Electronic Transactions on Numerical Analysis.
Volume 26, pp. 209-227, 2007.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

BLOCK TRIANGULAR PRECONDITIONERS FOR M-MATRICES AND MARKOV CHAINS*

MICHELE BENZI† AND BORA UÇAR‡

Abstract. We consider preconditioned Krylov subspace methods for solving large sparse linear systems under the assumption that the coefficient matrix is a (possibly singular) M-matrix. The matrices are partitioned into 2×2 block form using graph partitioning. Approximations to the Schur complement are used to produce various preconditioners of block triangular and block diagonal type. A few properties of the preconditioners are established, and extensive numerical experiments are used to illustrate the performance of the various preconditioners on singular linear systems arising from Markov modeling.

Key words. M-matrices, preconditioning, discrete Markov chains, iterative methods, graph partitioning

AMS subject classifications. 05C50, 60J10, 60J22, 65F10, 65F35, 65F50

1. Introduction.

We consider the solution of (consistent) linear systems $Ax = b$, where $A \in \mathbb{R}^{N\times N}$ is a large, sparse M-matrix. Our main interest is in singular, homogeneous ($b = 0$) systems arising from Markov chain modeling; our methods and results, however, are also applicable in the nonsingular case. The aim of this paper is to develop and investigate efficient preconditioners for Krylov subspace methods like GMRES [37] or BiCGStab [42]. Our main focus is on block preconditioners that are based on Schur complement approximations. Preconditioners of this type have proven very successful in the context of (generalized) saddle point problems arising in a variety of applications (see [5]); here we investigate such techniques for M-matrices and, in particular, for Markov chain problems.

We assume that A is partitioned into 2×2 block structure

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{1.1}$$

where A_{11} and A_{22} are square matrices of size, respectively, $n \times n$ and $m \times m$ with $N = n + m$. Building on the given structure, we consider block upper triangular preconditioners of the form

$$M_{BP} = \begin{bmatrix} M_{11} & A_{12} \\ O & M_{22} \end{bmatrix},$$

where M_{11} is equal to or an approximation of A_{11}, and M_{22} is an approximation of the Schur complement of A_{11} in A, i.e., $M_{22} \approx A_{22} - A_{21}A_{11}^{-1}A_{12}$. Obviously, block lower triangular preconditioners of the same type can also be constructed, as well as block diagonal ones. In any case, the main issue is the choice of the approximations $M_{11} \approx A_{11}$ and $M_{22} \approx A_{22} - A_{21}A_{11}^{-1}A_{12}$.

In the context of saddle-point problems, the 2×2 block structure is mandated by the application, and the approximation of the Schur complement is usually defined and interpreted in terms of the application. In contrast, the block preconditioners investigated in this paper

*Received October 17, 2006. Accepted for publication December 28, 2006. Recommended by D. Szyld.

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA (benzi@mathcs.emory.edu). The work of this author was supported in part by the National Science Foundation grant DMS-0511336.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA. Current Address: CERFACS, 42 Avenue G. Coriolis, 31057 Toulouse Cedex, France (ubora@cerfacs.fr). The work of this author was partially supported by The Scientific and Technological Research Council of Turkey (TÜBITAK) and by the University Research Committee of Emory University.

use well-known graph partitioning techniques to define the block structure, and make use of M-matrix properties to define approximations of the Schur complement.

For certain approximations of $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$, the block triangular preconditioners proposed in this work are akin to the product splitting preconditioners proposed in [10]. In fact, we experimentally show that in such cases these two types of preconditioners perform almost the same in terms of convergence rates for a large number of Markov chain problems from the MARCA collection [40]. However, the block triangular ones are cheaper to construct and apply than the product splitting ones.

The paper is organized as follows. We briefly review background material on M-matrices, discrete Markov chains, graph partitioning, matrix splittings and product splitting preconditioners in Section 2. We introduce the block triangular preconditioners in Section 3. Section 4 contains materials on partitioning the matrices into the 2×2 block structure (1.1) with an eye to future parallel implementations of the proposed preconditioners. In Section 5 we investigate the effect of the block triangular preconditioner under various partitionings, the properties of the 2×2 block structure imposed by the graph partitioning, and the performance of the proposed block triangular preconditioners relative to that of some other well-known preconditioners. We present our conclusions in Section 6.

2. Background.

Here we borrow some material mainly from [9, 10, 11, 43] to provide the reader with a short summary of the concepts and results that are used in building the proposed preconditioners. We also give a brief description of graph partitioning by vertex separator, which can be used to obtain the 2×2 block structure (1.1).

2.1. Nonnegative matrices and M-matrices.

A matrix $A_{N\times N}$ is nonnegative if all of its entries are nonnegative, i.e., $A \geq O$ if $a_{ij} \geq 0$ for all $1 \leq i, j \leq N$.

Any matrix A with nonnegative diagonal entries and nonpositive off-diagonal entries can be written in the form

$$A = sI - B, \quad s > 0, \quad B \geq O \, . \tag{2.1}$$

A matrix A of the form (2.1) with $s \geq \rho(B)$ is called an M-matrix. Here, $\rho(B)$ denotes the spectral radius of B. If $s = \rho(B)$ then A is singular, otherwise nonsingular. If A is a nonsingular M-matrix, then $A^{-1} \geq O$.

If A is a singular, irreducible M-matrix, then $\mathrm{rank}\,(A) = N - 1$ and each $k \times k$ principal square submatrix of A, where $1 \leq k < N$, is a nonsingular M-matrix. If, furthermore, A is the generator of an ergodic Markov chain (see below), then the Schur complement $S_{m\times m} = A_{22} - A_{21}A_{11}^{-1}A_{12}$ of A_{11} (cf. (1.1)) is a singular, irreducible M-matrix with rank $m - 1$ [9, 29].

2.2. Stationary distribution of ergodic Markov chains.

Discrete Markov chains with large state spaces arise in many applications, including for instance reliability modeling, queuing network analysis, web-based information retrieval, and computer system performance evaluation [41]. As is well known, the long-run behavior of an ergodic (irreducible) Markov chain is described by the stationary distribution vector of the corresponding matrix of transition probabilities. Recall that the stationary probability distribution vector of a finite, ergodic Markov chain with $N \times N$ transition probability matrix P is the unique $1 \times N$ vector π which satisfies

$$\pi = \pi P, \quad \pi_i > 0 \ \text{ for } \ i = 1, \ldots, N, \quad \sum_{i=1}^{N} \pi_i = 1 \, .$$

Here P is nonnegative ($p_{ij} \geq 0$ for $1 \leq i, j \leq N$), row-stochastic ($\sum_{j=1}^{N} p_{ij} = 1$ for $1 \leq i \leq N$), and due to the ergodicity assumption it is irreducible.

ETNA
Kent State University
etna@mcs.kent.edu

The matrix $A = I - P^T$, where I is the $N \times N$ identity matrix, is called the generator of the Markov process. The matrix A is a singular, irreducible M-matrix of rank $N-1$. Letting $x = \pi^T$ and hence $x^T = x^T P$, the computation of the stationary vector reduces to finding a nontrivial solution to the homogeneous linear system

$$Ax = 0 \,,$$

where $x \in \mathbb{R}^N$, $x_i > 0$ for $i = 1, \ldots, N$, and $\sum_{i=1}^N x_i = 1$. Perron–Frobenius theory [11] implies that such a vector exists and is unique.

Due to the very large number N of states typical of many real-world applications, there has been increasing interest in recent years in developing parallel algorithms for Markov chain computations; see [4, 6, 12, 24, 27, 30]. Most of the attention so far has focused on (linear) stationary iterative methods, including block versions of Jacobi and Gauss–Seidel [12, 27, 30], and on (nonlinear) iterative aggregation/disaggregation schemes specifically tailored to stochastic matrices [12, 24]. In contrast, comparatively little work has been done with parallel preconditioned Krylov subspace methods. The suitability of preconditioned Krylov subspace methods for solving Markov models has been demonstrated, e.g., in [32, 35], although no discussion of parallelization aspects was given there. Parallel computing aspects can be found in [6], where a symmetrizable stationary iteration (Cimmino's method) was accelerated using the Conjugate Gradient method on a Cray T3D, and in [27], where an out-of-core, parallel implementation of Conjugate Gradient Squared (with no preconditioning) was used to solve very large Markov models with up to 50 million states. In [9], parallel preconditioners based on sparse approximate pseudoinverses were used to speed-up the convergence of BiCGStab, and favorable parallelization results have been reported. In our recent work [10], we proposed product splitting preconditioners and discussed parallelization aspects of the proposed preconditioners.

2.3. Stationary iterations and matrix splittings. Consider again the solution of a linear system of the form $Ax = b$. The representation $A = B - C$ is called a splitting if B is nonsingular. A splitting gives rise to the stationary iterative method

$$x^{k+1} = Tx^k + c, \quad k = 0, 1, \ldots, \tag{2.2}$$

where $T = B^{-1}C$ is called the iteration matrix, $c = B^{-1}b$, and $x^0 \in \mathbb{R}^N$ is a given initial vector. The splitting $A = B - C$ is called (i) *regular* if $B^{-1} \geq O$ and $C \geq O$ [43], (ii) *weak regular* if $B^{-1} \geq O$ and $T \geq O$ [11], (iii) an *M-splitting* if B is an M-matrix and $C \geq O$ [38], and (iv) *weak nonnegative of the second kind* if $B^{-1} \geq O$ and $I - AB^{-1} \geq O$ [44]. If A is a nonsingular M-matrix, any of the conditions (i)–(iv) on the splitting $A = B - C$ is sufficient to ensure the convergence of the stationary iteration (2.2) to the unique solution of $Ax = b$, for any choice of the initial vector x^0.

A related approach is defined by the *alternating iterations*

$$\left\{ \begin{array}{lcl} x^{k+1/2} & = & M_1^{-1}N_1x^k + M_1^{-1}b \\ x^{k+1} & = & M_2^{-1}N_2x^{k+1/2} + M_2^{-1}b, \;\; k = 0, 1, \ldots, \end{array} \right. \tag{2.3}$$

where $A = M_1 - N_1 = M_2 - N_2$ are splittings of A, and x^0 is the initial vector. The convergence of alternating iterations was analyzed by Benzi and Szyld [7] under various assumptions on A, including the singular M-matrix case. They constructed a (possibly non-unique) splitting $A = B - C$ associated with the alternating iterations (cf. (10) in [7]) with

$$B^{-1} = M_2^{-1}(M_1 + M_2 - A)M_1^{-1} \,. \tag{2.4}$$

Clearly, the matrix $M_1 + M_2 - A$ must be nonsingular for (2.4) to be well-defined.

2.4. Product splitting preconditioners. The product splitting preconditioners [10] are based on alternating iterations (2.3) defined by two simple preconditioners. The first preconditioner can be taken to be the well-known block Jacobi preconditioner, i.e.,

$$M_{BJ} = \begin{bmatrix} A_{11} & O \\ O & A_{22} \end{bmatrix} .$$

Note that A_{11} and A_{22} are nonsingular M-matrices, and $A = M_{BJ} - (M_{BJ} - A)$ is a regular splitting (in fact, an M-splitting).

The second preconditioner is given by

$$M_{SC} = \begin{bmatrix} D_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} ,$$

where, $D_{11} \neq A_{11}$ stands for an approximation of A_{11}. In [10], we take D_{11} to be the diagonal matrix formed with the diagonal entries of A_{11}, e.g., $D_{11} = \text{diag}(A_{11})$. More generally, D_{11} is a matrix obtained from A_{11} by setting off-diagonal entries to zero. Thus, D_{11} is a nonsingular M-matrix [43, Theorem 3.12]. The Schur complement matrix $A_{22} - A_{21}D_{11}^{-1}A_{12}$ is therefore well-defined. It is easy to see that if A is a nonsingular M-matrix, so is $A_{22} - A_{21}D_{11}^{-1}A_{12}$; see, e.g., [1]. When A is a singular irreducible M-matrix, the Schur complement $A_{22} - A_{21}D_{11}^{-1}A_{12}$ is a nonsingular M-matrix under the (very mild) structural conditions given in [9, Theorem 3]. Therefore M_{SC} is a nonsingular M-matrix and $A = M_{SC} - (M_{SC} - A)$ is an M-splitting (hence, a regular splitting).

Since both M_{BJ} and M_{SC} define regular splittings, the product preconditioner M_{PS} given by

$$M_{PS}^{-1} = M_{SC}^{-1}(M_{BJ} + M_{SC} - A)M_{BJ}^{-1} ,$$

(see (2.4)), implicitly defines a weak regular splitting [7, Theorem 3.4]. Note that since the matrix

$$M_{BJ} + M_{SC} - A = \begin{bmatrix} D_{11} & O \\ O & A_{22} \end{bmatrix}$$

is invertible, M_{PS}^{-1} is well-defined, and so is the corresponding splitting of A.

2.5. Graph partitioning. Given an undirected graph $G = (V, E)$, the problem of K-way graph partitioning by vertex separator (GPVS) asks for a set of vertices V_S of minimum size whose removal decomposes the graph G into K disconnected subgraphs with balanced sizes. The problem is NP-hard [13]. Formally, $\Pi = \{V_1, \ldots, V_K; V_S\}$ is a K-way vertex partition by vertex separator V_S if the following conditions hold: $V_k \subset V$ and $V_k \neq \emptyset$ for $1 \le k \le K$; $V_k \cap V_\ell = \emptyset$ for $1 \le k < \ell \le K$ and $V_k \cap V_S = \emptyset$ for $1 \le k \le K$; $\bigcup_k V_k \cup V_S = V$; there is no edge between vertices lying in two different parts V_k and V_ℓ for $1 \le k < \ell \le K$; $W_{max}/W_{avg} \le \epsilon$, where W_{max} is the maximum part size (defined as $\max_k |V_k|$), W_{avg} is the average part size (defined as $(|V| - |V_S|)/K$), and ϵ is a given maximum allowable imbalance ratio. See the works [2, 14, 20, 21, 25] for applications of the GPVS and heuristics for GPVS.

In the weighted GPVS problem, the vertices of the given undirected graph have weights. The weight of the separator or a part is defined as the sum of the weights of the vertices that they contain. The objective of the weighted GPVS problem is to minimize the weight of the separator while maintaining a balance criterion on the part weights.

3. Block triangular preconditioners. Let $A \in \mathbb{R}^{N\times N}$ be an M-matrix, initially assumed to be nonsingular. Consider a 2×2 block structure as in (1.1). Recall from Section 2 that A_{11}, A_{22} and the Schur complement of A_{11} in A, i.e., $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ are all M-matrices. Note that if A_{11} is irreducible, then A_{11}^{-1} is dense and so is S; if A_{11} is reducible (e.g., block diagonal), then S will still be fairly dense unless the block sizes are very small.

Consider the ideal preconditioner [22, 31]

$$P_0 = \begin{bmatrix} A_{11} & A_{12} \\ O & S \end{bmatrix}.$$

It follows from the block LU factorization

$$A = \begin{bmatrix} I & O \\ A_{21}A_{11}^{-1} & I \end{bmatrix}\begin{bmatrix} A_{11} & A_{12} \\ O & S \end{bmatrix} = LP_0$$

that AP_0^{-1} (and therefore $P_0^{-1}A$) has the eigenvalue $\lambda = 1$ of multiplicity N as the only point in the spectrum. Also, $(AP_0^{-1} - I)^2 = (L-I)^2 = O$ which shows that the minimum polynomial of the preconditioned matrix has degree 2; hence, a minimal residual method (like GMRES) is guaranteed to converge in at most two steps. Unfortunately, preconditioning with P_0 is impractical. Here, similar to the situation for saddle point problems, we consider preconditioners obtained by different approximations of A_{11} and of S.

We start with the case where A_{11} is solved "exactly", while S is replaced by an approximation $\hat{S} = A_{22} - A_{21}M_{11}^{-1}A_{21}$ where M_{11}^{-1} is an approximate inverse of A_{11}. If we impose the condition that

$$O \le M_{11}^{-1} \le A_{11}^{-1} \quad \text{(entrywise)} \tag{3.1}$$

then $\hat{S}$ is guaranteed to be an M-matrix and invertible if A is invertible; see [1, pp. 263–265]. The condition (3.1) is satisfied (for example) when $A_{11} = M_{11} - (M_{11} - A_{11})$ is an M-splitting. In particular, if $M_{11} = \mathrm{diag}(A_{11})$ (or any other approximation obtained by setting any off-diagonal entries of A_{11} to zero), then the condition (3.1) is satisfied.

Assume that linear systems with $\hat{S}$ are solved exactly. Then, the preconditioner is

$$P_1 = \begin{bmatrix} A_{11} & A_{12} \\ O & \hat{S} \end{bmatrix}.$$

A simple calculation shows that

$$AP_1^{-1} = \begin{bmatrix} I & O \\ A_{21}A_{11}^{-1} & S\hat{S}^{-1} \end{bmatrix}.$$

Hence, AP_1^{-1} (or $P_1^{-1}A$) has the eigenvalue $\lambda = 1$ of multiplicity at least n, while the remaining m eigenvalues are those of $S\hat{S}^{-1}$. These m eigenvalues lie in the open disk centered at $(1,0)$ of radius 1. To show this, consider the following matrix splitting

$$A = P_1 - (P_1 - A) = \begin{bmatrix} A_{11} & A_{12} \\ O & \hat{S} \end{bmatrix} - \begin{bmatrix} O & O \\ -A_{21} & -A_{21}M_{11}^{-1}A_{12} \end{bmatrix}.$$

Note that this splitting is not a regular splitting, since $-A_{21}M_{11}^{-1}A_{12} \le O$. However, we have the following result.

THEOREM 3.1. *The splitting $A = P_1 - (P_1 - A)$ is a weak nonnegative splitting of the second kind.*

Proof. Due to the condition (3.1), P_1 is an M-matrix, and hence $P^{-1} \geq O$. We only need to check the nonnegativity of

$$I - AP_1^{-1} = \begin{bmatrix} O & O \\ -A_{21}A_{11}^{-1} & I - S\hat{S}^{-1} \end{bmatrix} .$$

Clearly $-A_{21}A_{11}^{-1} \geq O$ (since $A_{21} \leq O$ and $A_{11}^{-1} \geq O$). We need to show $I - S\hat{S}^{-1} \geq O$. We have

$$\begin{aligned} S &= A_{22} - A_{21}A_{11}^{-1}A_{12} \\ &= A_{22} - A_{21}(A_{11}^{-1} - M_{11}^{-1} + M_{11}^{-1})A_{12} \\ &= A_{22} - A_{21}M_{11}^{-1}A_{12} - A_{21}(A_{11}^{-1} - M_{11}^{-1})A_{12} \\ &= \hat{S} - R \,, \end{aligned}$$

where $R = A_{21}(A_{11}^{-1} - M_{11}^{-1})A_{12}$ is nonnegative, since $A_{11}^{-1} \geq M_{11}^{-1}$ (3.1). Recall that $\hat{S}$ is an M-matrix, therefore $I - S\hat{S}^{-1} = (\hat{S} - S)\hat{S}^{-1} = R\hat{S}^{-1} \geq O$. ☐

Since A is an invertible M-matrix and $A = P_1 - (P_1 - A)$ is a weak nonnegative splitting of the second kind, it is a convergent splitting: $\rho(I - AP_1^{-1}) < 1$; see [44]. In other words, all the eigenvalues of AP_1^{-1} or $(P_1^{-1}A)$ satisfy $|\lambda - 1| < 1$. For the special case $M_{11}^{-1} = A_{11}^{-1}$, we have $P_1 = P_0$ and $\sigma(AP_1^{-1}) = \{1\}$.

In practice, exact solves with A_{11} or $\hat{S}$ may be impractical or not advisable on the grounds of efficiency or numerical stability. This leads to the following three variants. The first one is

$$P_2 = \begin{bmatrix} \hat{A}_{11} & A_{12} \\ O & \hat{S} \end{bmatrix} , \quad \hat{A}_{11} \approx A_{11} \,,$$

where we have inexact solves with A_{11} and exact solves with $\hat{S}$. We can assume that $A_{11} = \hat{A}_{11} - (\hat{A}_{11} - A_{11})$ is a regular splitting, or even an M-splitting. Another variant is

$$P_3 = \begin{bmatrix} A_{11} & A_{12} \\ O & \tilde{S} \end{bmatrix} , \quad \tilde{S} \approx \hat{S} \,,$$

where we have exact solves with A_{11} and inexact solves with $\hat{S}$. Here we can assume that $\hat{S} = \tilde{S} - (\tilde{S} - \hat{S})$ is a regular splitting. P_3 can be analyzed exactly like P_1. Finally we consider

$$P_4 = \begin{bmatrix} \hat{A}_{11} & A_{12} \\ O & \tilde{S} \end{bmatrix} , \quad \hat{A}_{11} \approx A_{11} \,, \quad \tilde{S} \approx \hat{S} \,,$$

where we can assume that both $\hat{A}_{11}$ and $\tilde{S}$ induce regular splittings of A_{11} and $\hat{S}$, respectively. In practice, we use P_4 with $\hat{A}_{11}$ and $\tilde{S}$ coming from incomplete LU factorizations (ILU) of A_{11} and $\hat{S}$. Note that unlike position-based ILU's, threshold-based ILU's do not always lead to regular splittings (see Section 10.4.2 in [36]). In general, however, threshold-based ILUs often perform better than position-based ones, and the clustering of the eigenvalues of the preconditioned matrices around unity can be easily controlled by the drop tolerance: the smaller the drop tolerance, the tighter the cluster around (1,0) can be expected to be. We remark that although in the nonnormal case the eigenvalue distribution may not govern the rate of convergence of Krylov subspace methods, it is often the case in practice that a clustered spectrum (away from zero) results in rapid convergence. More precisely, for residual

ETNA
Kent State University
etna@mcs.kent.edu

minimizing methods (like GMRES), a sufficient condition for fast convergence is that the preconditioned matrix is diagonalizable with well-conditioned eigenvector matrix and with all of its eigenvalues clustered away from zero; see, e.g., the recent survey [39]. Unfortunately, it is generally very difficult to derive bounds on the condition number of the eigenvector matrix. Another sufficient condition is that the minimum polynomial of the preconditioned matrix be of low degree, since such degree is an upper bound on the number of GMRES steps needed to reach the exact solution. If the preconditioner is a good approximation of an "ideal" one that yields a preconditioned matrix with a minimum polynomial of low degree, convergence may be quite fast.

When A is a singular, irreducible M-matrix, then the splittings $A = P_i - (P_i - A)$ satisfy $\rho(I - AP_i^{-1}) = 1$, and the spectrum of AP_i^{-1} will also include the eigenvalue $\lambda = 0$ (for $i = 1, \ldots, 4$). We also note that $\lambda = 0$ is a simple eigenvalue, and the splitting $A = P_i - (P_i - A)$ is semiconvergent, if $\hat{S} - S$ contains no zero rows and $\hat{S}$ is irreducible; see [11]. The zero eigenvalue, in any event, does not negatively affect the convergence of Krylov methods like GMRES; in practice, the only effect is to introduce a set of (Lebesgue) measure zero of "bad" initial vectors. More precisely, it follows from the discussion in [17, 23] that the initial vector x^0 must not lie in the column space of the preconditioned matrix. This condition on x^0 can be easily fulfilled, for instance by using a random initial vector.

4. Building the block triangular preconditioner.

4.1. Defining the blocks. The first requirement to be met in permuting the matrix A into 2×2 block structure (1.1) is that the permutation should be symmetric. If A is nonsingular or obtained from the transition probability matrix of an irreducible Markov chain, then a symmetric permutation on the rows and columns of A guarantees that A_{11} and A_{22} are invertible M-matrices.

The second requirement, as already discussed in Section 3, is to keep the order n of A_{11} as large as possible to maximize the number of (near) unit eigenvalues of the preconditioned matrix (AP_i^{-1} for $i = 1, 2, 3,$ and 4). The requirement is important in order to have fast convergence of the Krylov subspace method, since m, the size of A_{22}, is an upper bound on the number of non-unit eigenvalues. Strictly speaking, this is true only if we assume exact solves in the application of the preconditioner, e.g., for AP_1^{-1}. In practice we will use inexact solves, and rather than having n eigenvalues (or more) exactly equal to 1, there will be a cluster of at least n eigenvalues near the point $(1, 0)$. Still, we want this cluster to contain as many eigenvalues as possible.

The second requirement is also desirable from the point of view of parallel implementation. A possible parallelization approach would be constructing and solving the linear systems with approximate Schur complements on a single processor, and then solving the linear systems with A_{11} in parallel. This approach has been taken previously in parallelizing applications of approximate inverse preconditioners in Markov chain analysis [9]. Another possible approach would be parallelizing the solution of the approximate Schur complement systems either by allowing redundancies in the computations (each processor can form the whole system or a part of it) or by running a parallel solver on those systems. In both cases, the solution with the approximate Schur complement system constitutes a serial bottleneck and requires additional storage space.

The third requirement, not necessary for the convergence analysis but crucial for an efficient implementation, is that A_{11} should be block diagonal with subblocks of approximately equal size and density. Given K subblocks in the (1,1) block A_{11}, the K linear systems stemming from A_{11} can be solved independently. Meeting this requirement for a serial implementation will enable solution of very large systems, since the subblocks can be handled one at a time. In any admissible parallelization, each of these subblocks would more likely be

ETNA
Kent State University
etna@mcs.kent.edu

assigned to a single processor. Therefore, maintaining balance on the sizes and the densities of the subblocks will relate to maintaining balance on computational loads of the processors. Furthermore, it is desirable that the sizes of these subblocks be larger than the order m of A_{22}, if possible, for the reasons given for the second requirement.

Meeting all of the above three requirements is a very challenging task. Therefore, as a pragmatic approach we apply well established heuristics for addressing the remaining three requirements. As it is common, we adopt the standard undirected graph model to represent a square matrix $A_{N\times N}$. The vertices of the graph $G(A) = (V, E)$ correspond to the rows and columns of A and the edges correspond to the nonzeros of A. The vertex $v_i \in V$ represents the ith row and the ith column of A, and there exists an edge $(v_i, v_j) \in E$ if a_{ij} and a_{ji} are nonzero.

Consider a partitioning $\Pi = \{V_1, \ldots, V_K; V_S\}$ of $G(A)$ with vertex separator V_S. The matrix A can be permuted into the 2×2 block structure (1.1) by permuting the rows and columns associated with the vertices in $\bigcup_k V_k$ before the rows and columns associated with the vertices in V_S. That is, V_S defines the rows and columns of the (2,2) block A_{22}. Notice that the resulting permutation is symmetric, and hence the first requirement is met. Furthermore, since GPVS tries to minimize the size of the separator set V_S, it tries to minimize the order of the block A_{22}. Therefore, the permutation induced by Π meets the second requirement as well.

Consider the A_{11} block defined by the vertices in $\bigcup_k V_k$. The rows and columns that are associated with the vertices in V_k can be permuted before the rows and columns associated with the vertices in V_ℓ for $1 \le k < \ell \le K$. Such a permutation of A_{11} gives rise to diagonal subblocks. Since we have already constructed A_{22} using V_S, we end up with the following structure:

$$A = \begin{bmatrix} A_1 & & & & B_1 \\ & A_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & A_K & B_K \\ C_1 & C_2 & \cdots & C_K & A_S \end{bmatrix} .$$

The diagonal blocks $A_1, \ldots, A_K$ correspond to the vertex parts $V_1, \ldots, V_K$, and therefore have approximately the same order. The off-diagonal blocks B_i, C_i represent the connections between the subgraphs, and the diagonal block A_S represents the connections between nodes in the separator set. Note that if A is nonsingular or obtained from the transition probability matrix of an irreducible Markov chain, then each block A_i is a nonsingular M-matrix. Thus, graph partitioning induces a reordering and block partitioning of the matrix A in the form (1.1) where

$$A_{11} = \mathrm{diag}(A_1,\ A_2, \ldots, A_K),\quad A_{22} = A_S$$

and

$$A_{12} = [B_1^T\ B_2^T\ \cdots\ B_K^T]^T,\quad A_{21} = [C_1\ C_2\ \cdots\ C_K]\ .$$

Therefore, the permutation induced by the GPVS partially addresses the third requirement. Note that the GPVS formulation ignores the requirement of balancing the densities of the diagonal subblocks of A_{11}. In fact, obtaining balance on the densities of the diagonal blocks is a complex partitioning requirement that cannot be met before a partitioning takes place (see [33] for a possible solution) even with a weighted GPVS formulation.

If the matrix is structurally nonsymmetric, which is common for matrices arising from Markov chains, then A cannot be modeled with undirected graphs. In this case, a 2×2 block structure can be obtained by partitioning the graph of $A + A^T$.

ETNA
Kent State University
etna@mcs.kent.edu

4.2. Approximating the Schur complement. Recall from Section 3 that we are interested in approximations of the Schur complement of the form $\hat{S} = A_{22} - A_{21}M_{11}^{-1}A_{21}$ where M_{11}^{-1} is an approximate inverse of A_{11} satisfying (3.1). The approximate Schur complement $\hat{S}$ is required to be nonsingular. As mentioned in Section 2.4, this requirement is satisfied under the structural conditions given in [9, Theorem 3]. We found the rather crude approximation $M_{11} = \mathrm{diag}(A_{11})$ to be quite satisfying. It is easy to invert and apply, and also it maintains a great deal of sparsity in $\hat{S}$. Apart from this approximation, we also tried $M_{11} = \hat{A}_{11}$ with $\hat{A}_{11} = \bar{L}_{11}\bar{U}_{11}$ (an incomplete factorization of A_{11}), the same approximation used in the (1,1) blocks of P_2 and P_4. This works well in terms of reducing the number of iterations and in all experiments it delivered the smallest number of iterations; however, this good rate of convergence came at the price of a prohibitive preconditioner construction overhead, and in a few cases it failed due to lack of space for $\hat{S}$. In actual computation with both choices of M_{11}, the Schur complement matrix $\hat{S}$ was always observed to be nonsingular.

In an attempt to find a midway, we tried to construct a block diagonal (up to a symmetric permutation) M_{11} with 1×1 and 2×2 blocks. We create a list of all pairs $\langle i,j\rangle$ with $i<j$ and either a_{ij} or a_{ji} or both are nonzero. Each of these pairs is a candidate for a 2×2 block of the form $\begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix}$. Then we visit the candidates in the descending order of the determinants of the corresponding 2×2 blocks, where the determinant of the pair $\langle i,j\rangle$ is computed using $a_{ii}a_{jj} - a_{ij}a_{ji}$. At candidate $\langle i,j\rangle$, if both i and j are not included in any 2×2 block, we form the corresponding 2×2 block. Otherwise, we proceed to the next candidate. Any row i (and hence column) that is not included in a 2×2 block defines a 1×1 block a_{ii} in M_{11}. Note that this greedy algorithm tries to maximize the minimum of the determinants of the 2×2 blocks in M_{11} and hence tries to yield a well-conditioned matrix M_{11} without any attempt to maximize the number of those blocks. Similar algorithms were used in [18, 19] in preconditioning indefinite systems and were found to be useful. In our case, however, this choice of M_{11} did not improve upon the simple diagonal one.

5. Numerical experiments. In this section, we report on experimental results obtained with a Matlab 7.1.0 implementation on a 2.2 GHz dual core AMD Opteron Processor 875 with 4GB main memory. The main goal was to test the proposed block preconditioners and to compare them with a few other techniques, including the product splitting preconditioner [10]. The Krylov method used was GMRES [37]. For completeness we performed experiments with the stationary iterations corresponding to the various splittings (without GMRES acceleration), but they were found to converge too slowly to be competitive with preconditioned GMRES. Therefore, we do not show these results.

The various methods were tested on the generator matrices of some Markov chain models provided in the MARCA (MARkov Chain Analyzer) collection [40]. The models are discussed in [16, 32, 34] and have been used to compare different solution methods in [9, 10, 15] and elsewhere. These matrices are infinitesimal generators of time-continuous Markov chains, but can be easily converted (as we did) to the form $A = I - P^T$, with P row-stochastic, so that A corresponds to a discrete-time Markov chain, known as the *embedded Markov chain*; see [41, Chapter 1.4.3]. The preconditioning techniques described in this paper can be applied to either form of the generator matrix.

We performed a large number of tests on numerous matrices; here we present a selection of results for a few test matrices, chosen to be representative of our overall findings. Table 5.1 displays the properties of the chosen test matrices. Each matrix is named by its family followed by its index in the family. For example, `mutex09` refers to the 9th matrix in the `mutex` family. The matrices from the `mutex` and `ncd` families are structurally symmetric, the matrices from the `qnatm` and `twod` families are structurally nonsymmetric, and the ma-

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 5.1
Properties of the generator matrices.

Matrix	number of rows/cols N	number of nonzeros total	average row/col	row min	row max	col min	col max
mutex09	65535	1114079	17.0	16	17	16	17
mutex12	263950	4031310	15.3	9	21	9	21
ncd07	62196	420036	6.8	2	7	2	7
ncd10	176851	1207051	6.8	2	7	2	7
qnatm06	79220	533120	6.7	3	9	4	7
qnatm07	130068	875896	6.7	3	9	4	7
tcomm16	13671	67381	4.9	2	5	2	5
tcomm20	17081	84211	4.9	2	5	2	5
twod08	66177	263425	4.0	2	4	2	4
twod10	263169	1050625	4.0	2	4	2	4

trices from the `tcomm` family are very close to being structurally symmetric—the nonzero patterns of `tcomm20` and `tcomm16` differ from the nonzero patterns of their transposes in only 60 locations.

We compared the block preconditioner (BT) with the block Jacobi (BJ), block Gauss-Seidel (BGS), block successive overrelaxation (BSOR), and product splitting (PS) preconditioners. The block Gauss-Seidel (BGS) and the block successive overrelaxation (BSOR) preconditioners are given, respectively, by

$$M_{BGS} = \begin{bmatrix} A_{11} & A_{12} \\ O & A_{22} \end{bmatrix} \text{ or } M_{BGS} = \begin{bmatrix} A_{11} & O \\ A_{21} & A_{22} \end{bmatrix},$$
$$M_{BSOR} = \begin{bmatrix} \frac{1}{\omega}A_{11} & A_{12} \\ O & \frac{1}{\omega}A_{22} \end{bmatrix} \text{ or } M_{BSOR} = \begin{bmatrix} \frac{1}{\omega}A_{11} & O \\ A_{21} & \frac{1}{\omega}A_{22} \end{bmatrix}.$$

In agreement with previously reported results [15] on the MARCA collection, we observed that $\omega = 1.0$ (which reduces the BSOR to BGS) or very close to 1.0 is nearly always the best choice of the relaxation parameter for BSOR. We also observed that for most MARCA problems, the block lower triangular versions of the BGS and BSOR preconditioners are indistinguishable from the block upper triangular versions under either the storage or performance criteria. Therefore, we report only the experiments with the upper triangular BGS preconditioner.

As discussed for instance in [22, 31], the ideal block triangular preconditioner P_0 has a natural block diagonal (BD) counterpart, namely

$$M_{BD} = \begin{bmatrix} A_{11} & O \\ O & S \end{bmatrix}.$$

As in P_0, exact solutions with M_{BD} is not feasible. Therefore, we replace A_{11} and the Schur complement S with approximations, as in P_4, and compare the BT preconditioners with the corresponding inexact BD as well.

5.1. Observations on the block preconditioners. We partitioned the matrix into the 2×2 block structure (1.1) using Metis [26] library. We recursively applied the Metis function `MlevelNodeBisectionMultiple` using the default options prescribed for `METIS_NodeND` [26]. In all cases, the partitioning time is negligible compared to the solve time. For the structurally symmetric `mutex` and `ncd` matrices, we used the graph of A, and for the other matrices we used the graph of $A + A^T$ as mentioned in Section 4. As discussed

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 5.2
Properties of the partitions and the induced block structures averaged over all matrices excluding `mutex` *matrices. The column "sep" refers to the number of rows in the 2nd row block of A normalized by the number of rows in A, i.e.,* m/N*; the column "part" refers to the average part size normalized by the number of rows in A, i.e.,* $(n/K)/N$*; the columns* A_{ij} *for* $i, j = 1, 2$ *refer to the number of nonzeros in the* (i, j) *block normalized by the number of nonzeros in A, i.e.,* $nnz(A_{ij})/nnz(A)$*.*

K	Partition		Blocks			
	sep	part	A_{11}	A_{12}	A_{21}	A_{22}
2	0.008	0.496	0.986	0.006	0.006	0.002
4	0.016	0.246	0.971	0.012	0.012	0.004
8	0.028	0.121	0.951	0.021	0.021	0.007
16	0.045	0.060	0.921	0.034	0.034	0.012
32	0.068	0.029	0.881	0.051	0.051	0.017

in Section 4, we maintain balance on the size, rather than the densities, of the subblocks of A_{11}. We have conducted experiments with $K = 2, 4, 8, 16$, and 32 subblocks in the (1,1) block. For each K value, K-way partitioning of a test matrix constitutes a partitioning instance. Since Metis incorporates randomized algorithms, it was run 10 times starting from different random seeds for each partitioning instance with a maximum allowable imbalance ratio of 25%. In all partitioning instances except the `mutex` matrices, the imbalance ratios among the parts were within the specified limit. The following tables give the average of these 10 different runs for each partitioning instance.

Only the `mutex` matrices have a large number of rows in the second row block, i.e., a large separator among all partitioning instances. For these matrices, the average part size is larger than the size of the separator set only in $K = 2$-way partitioning. For the other matrices, the average part size is larger than the size of the separator in all partitioning instances with $K = 2, 4, 8$, and 16 except in $K = 16$-way partitioning of `ncd07`, `ncd10`, and `qnatm06`, giving the average figures in Table 5.2.

We have conducted experiments with block triangular preconditioners P_1 through P_4. A somewhat surprising find is that those variants requiring exact solves with A_{11}, e.g., P_1 and P_3, besides being rather expensive (as expected), are prone to numerical instabilities. By this we mean that at least one block A_k in A_{11} was found to have an upper triangular factor U with a huge condition number, causing the convergence of GMRES to deteriorate. (The unit lower triangular factor is always well-conditioned for the problems considered here.) We encountered this difficulty with all test matrices and for all values of K, except for the `qnatm` matrices. This phenomenon can be explained by noting that the diagonal blocks A_k, while guaranteed to be nonsingular, are often close to singular, in particular when A is nearly reducible; see [29]. Hence, the corresponding upper triangular factor must have an exceedingly small pivot, and consequently its inverse must have a huge norm. This problem disappears when the complete factorizations of the diagonal blocks A_k are replaced by incomplete ones. This is not surprising: it has been shown in [28] that in an incomplete factorization of an M-matrix, the pivots (i.e., the diagonal entries of the upper triangular factor) cannot become smaller, and in practice they always increase. As a result, the condition number of the incomplete upper triangular factor is several orders of magnitude smaller than that of the complete factor, and no instabilities arise. Therefore, we have the somewhat unexpected conclusion that in practice inexact solvers result in greater robustness and faster convergence than exact ones.

For these reasons we do not present results with P_1 and P_3. In addition, P_2 has a large construction overhead due to exact factorization of $\hat{S}$ and did not perform better than P_4 in reducing the number of iterations. Therefore, in the following we present results only with P_4. We tried all three approximations of the Schur complement discussed in Section 4.2 with

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 5.3

Data pertaining to the P_4 preconditioner with $\tilde{S} \approx \hat{S} = A_{22} - A_{21}(\bar{L}_{11}\bar{U}_{11})^{-1}A_{12}$. The column "its" refers to the average number of iterations, "dens" refers to the total number of nonzeros in the matrices appearing in preconditioner solve phase of P_4 divided by the number of nonzeros in the generator matrices, "prec" refers to the time spent in constructing the preconditioners, and "solve" refers to the time spent during GMRES iterations.

Matrix	K	its	dens	Total time	
				prec	solve
ncd07	2	11	1.11	29.19	1.11
	4	12	1.23	14.86	1.16
	8	12	1.20	16.87	1.25
	16	13	1.18	20.26	1.35
	32	13	1.18	20.50	1.46
ncd10	2	11	1.10	187.40	3.68
	4	11	1.13	120.63	4.29
	8	12	1.12	99.12	4.14
	16	13	1.14	81.39	4.31
	32	13	1.12	95.72	4.51
qnatm06	2	35	2.97	45.86	6.07
	4	35	2.65	43.09	6.21
	8	37	2.61	37.19	6.57
	16	38	2.49	31.51	6.85
	32	40	2.48	31.73	7.66
qnatm07	2	39	2.99	86.45	11.66
	4	40	2.58	96.43	12.50
	8	42	2.59	84.09	12.88
	16	42	2.52	69.27	13.05
	32	45	2.49	73.82	14.32
tcomm16	2	14	3.03	0.38	0.24
	4	16	2.47	0.60	0.24
	8	19	2.13	0.84	0.29
	16	26	2.01	1.27	0.40
	32	25	1.93	1.64	0.40
tcomm20	2	16	2.93	0.48	0.34
	4	19	2.24	0.73	0.35
	8	22	2.17	1.02	0.42
	16	30	2.09	1.51	0.61
	32	28	1.99	2.23	0.59
twod08	2	10	2.24	7.20	0.91
	4	12	2.30	12.18	1.03
	8	17	3.05	17.17	1.56
	16	18	2.77	18.68	1.63
	32	18	2.87	26.38	1.68
twod10	2	17	4.64	156.29	8.71
	4	18	4.63	163.76	8.60
	8	20	5.11	190.06	9.51
	16	21	5.01	199.50	10.30
	32	23	4.37	229.49	10.50

the block triangular preconditioner P_4. The one with block diagonal M_{11} with 1×1 and 2×2 blocks did not improve upon the simple diagonal one. Therefore, we omit the results with this choice of M_{11}. The one with $M_{11} = \hat{A}_{11}$ has very large memory requirements; for example, it was not possible to run it with the `mutex` matrices. Therefore, it is not recommended as a general purpose solution. However, it merits presenting because it gives the smallest number of iterations and has fairly robust behavior with respect to increasing K (see Table 5.3). In

ETNA
Kent State University
etna@mcs.kent.edu

the following discussion, BT thus refers to the block triangular preconditioner P_4:

$$P_4 = \begin{bmatrix} \hat{A}_{11} & A_{12} \\ O & \tilde{S} \end{bmatrix}, \quad \hat{A}_{11} \approx A_{11}, \quad \tilde{S} \approx \hat{S} = A_{22} - A_{21} M_{11}^{-1} A_{12} \quad ,$$

where $M_{11} = \text{diag}(A_{11})$.

Each subblock A_k, for $k = 1, \ldots, K$, of A_{11} and the (2,2) block A_{22} in the BJ and BGS preconditioners were ordered using symmetric reverse Cuthill-McKee (for better numerical properties; see [8]) and factored using the incomplete LU factorization (ILUTH) with threshold parameter $\tau = 0.01$ for the `qnatm` matrices and $\tau = 0.001$ for the other matrices. The threshold of 0.001 was too small for the `qnatm` matrices: the resulting preconditioners had 8 times more nonzeros than the generator matrices. We observed that reordering the approximate Schur complement matrix causes ILUTH to take too much time (presumably due to the need for pivoting), but reduces the number of nonzeros in the factors only by a small amount. Therefore, the approximate Schur complement matrices were factored using ILUTH without any prior ordering.

The densities of the preconditioners, i.e., the number of nonzeros in the matrices appearing in the preconditioner solve phase divided by the number of nonzeros in the corresponding generator matrices, are given in Table 5.4. As seen from the table, memory requirements of the BT preconditioners are less than those of the PS preconditioners and comparable to those of the three, relatively simple, "classical" preconditioners.

5.2. Performance comparisons. The underlying Krylov subspace method was GMRES, restarted every 50 iterations (if needed). Right preconditioning was used in all the tests. The stopping criterion was set as

$$\|r_k\|_2 / \|r_0\|_2 < 10^{-10} ,$$

where r_k is the residual at the kth iteration and r_0 is the initial residual. In all cases, the initial solution is set to the one-vector normalized to have an ℓ_1-norm of 1.0. We allowed at most 250 iterations, i.e., 5 restarts, for the GMRES iteration. Therefore, the number 250 in the following tables marks the cases in which GMRES failed to deliver solutions with the prescribed accuracy within 250 iterations. Iteration counts for GMRES(50) on the various test matrices with no permutation or preconditioning (GMRES) and averages of the iteration counts with the preconditioners BJ, BD, BGS, PS and BT are given in Table 5.5. Note that without preconditioning, GMRES(50) converges only for the `mutex` matrices. In all instances, preconditioned GMRES(50) with the PS and the proposed BT preconditioners converged under the stopping criterion given above. Preconditioned GMRES(50) with BJ, BD, and BGS did not converge in, respectively, 103, 141, and 8 of the 500 instances. The largest of the ℓ_1-norms of the residuals corresponding to the approximate solutions returned by the preconditioned GMRES(50) were less than 1.1e-11 for the converged instances with the BGS, PS, and BT preconditioners. With the BJ and BD preconditioners, the largest of the ℓ_1-norms of the residuals corresponding to the approximate solutions returned by the preconditioned GMRES(50) was 4.6e-10.

As seen from Table 5.5, the PS and BT preconditioners perform consistently better than the BJ, BD, and BGS preconditioners. The proposed BT preconditioner performs almost as well as the PS preconditioner, and it outperforms the BGS one by a factor of two, on the average, in terms of iteration counts, at the expense of a slight increase in memory requirements (see Table 5.4).

We close this section by discussing running times for GMRES. The running times are measured using Matlab's `cputime` command, and these measurements are given in Ta-

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 5.4
The densities of the preconditioners, i.e., the total number of nonzeros in the matrices appearing in the preconditioner solve phase divided by the number of nonzeros in the corresponding generator matrices.

Matrix	K	Preconditioners				
		BJ	BD	BGS	PS	BT
mutex09	2	1.30	1.52	1.48	2.00	1.70
	4	0.78	1.24	1.06	1.99	1.52
	8	0.51	1.18	0.86	2.07	1.52
	16	0.36	1.14	0.73	2.11	1.51
	32	0.30	1.16	0.69	2.17	1.55
mutex12	2	1.87	2.06	2.06	2.53	2.25
	4	0.83	1.21	1.14	1.92	1.51
	8	0.45	0.96	0.82	1.82	1.33
	16	0.29	0.90	0.70	1.83	1.31
	32	0.23	0.89	0.66	1.85	1.32
ncd07	2	1.09	1.10	1.11	1.28	1.11
	4	1.18	1.19	1.21	1.40	1.21
	8	1.11	1.13	1.16	1.38	1.17
	16	1.05	1.07	1.11	1.37	1.13
	32	0.99	1.03	1.08	1.38	1.12
ncd10	2	1.08	1.08	1.09	1.26	1.09
	4	1.09	1.10	1.11	1.30	1.12
	8	1.05	1.06	1.09	1.29	1.10
	16	1.04	1.06	1.09	1.33	1.11
	32	0.99	1.01	1.06	1.32	1.08
qnatm06	2	2.94	2.95	2.95	3.13	2.96
	4	2.58	2.61	2.60	2.81	2.63
	8	2.46	2.52	2.50	2.76	2.55
	16	2.25	2.33	2.30	2.63	2.39
	32	2.10	2.23	2.18	2.59	2.31
qnatm07	2	2.97	2.98	2.98	3.15	2.98
	4	2.52	2.54	2.53	2.73	2.56
	8	2.48	2.52	2.51	2.75	2.55
	16	2.34	2.40	2.38	2.67	2.44
	32	2.20	2.30	2.27	2.62	2.36
tcomm16	2	3.03	3.03	3.03	3.24	3.03
	4	2.45	2.45	2.46	2.67	2.46
	8	2.09	2.09	2.10	2.33	2.10
	16	1.91	1.93	1.94	2.20	1.95
	32	1.76	1.78	1.80	2.10	1.82
tcomm20	2	2.93	2.93	2.93	3.14	2.93
	4	2.22	2.23	2.23	2.44	2.23
	8	2.13	2.14	2.14	2.37	2.15
	16	2.01	2.02	2.03	2.28	2.04
	32	1.83	1.84	1.86	2.16	1.88
twod08	2	2.24	2.24	2.24	2.49	2.24
	4	2.26	2.26	2.27	2.53	2.27
	8	2.98	2.98	2.99	3.26	2.99
	16	2.65	2.66	2.66	2.95	2.67
	32	2.70	2.71	2.72	3.03	2.73
twod10	2	4.64	4.64	4.64	4.89	4.64
	4	4.60	4.60	4.61	4.86	4.61
	8	5.06	5.06	5.06	5.33	5.07
	16	4.92	4.92	4.92	5.20	4.93
	32	4.24	4.25	4.25	4.53	4.26

bles 5.6 and 5.7 in units of seconds. In Table 5.6, the total time for GMRES without preconditioning is the total time spent in performing the GMRES iterations. In both of the tables, the total time for the preconditioned GMRES is dissected into the preconditioner construction and the solve phases. We first discuss the case of `mutex` matrices, since all the precondi-

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 5.5
Average number of iterations to reduce the ℓ_2-norm of the initial residual by ten orders of magnitude using GMRES(50) with at most 5 restarts. The number 250 means that the method did not converge in 250 iterations.

Matrix	GMRES	Preconditioned GMRES					
		K	Preconditioners				
			BJ	BD	BGS	PS	BT
mutex09	97	2	24	24	13	9	9
		4	27	27	14	9	9
		8	28	28	14	8	9
		16	29	30	15	9	9
		32	29	30	15	9	9
mutex12	91	2	26	27	14	8	10
		4	28	29	15	8	9
		8	28	29	15	8	9
		16	29	31	16	7	9
		32	30	31	16	7	8
ncd07	250	2	40	68	23	12	12
		4	212	250	87	13	13
		8	222	250	101	15	15
		16	243	250	129	16	16
		32	250	250	192	18	18
ncd10	250	2	122	168	57	14	15
		4	160	250	47	15	15
		8	244	250	105	15	15
		16	250	250	145	18	17
		32	250	250	215	19	19
qnatm06	250	2	60	60	38	36	36
		4	78	83	41	36	36
		8	104	119	46	39	39
		16	177	181	55	43	43
		32	223	250	83	46	47
qnatm07	250	2	51	56	40	39	39
		4	83	91	44	41	41
		8	110	138	51	44	44
		16	163	186	74	47	47
		32	232	246	92	55	56
tcomm16	250	2	30	30	18	16	16
		4	48	51	27	21	21
		8	112	131	39	29	30
		16	250	250	85	42	42
		32	250	250	105	46	46
tcomm20	250	2	33	34	20	18	18
		4	55	61	29	23	23
		8	132	157	43	32	32
		16	247	250	94	44	44
		32	250	250	221	96	98
twod08	250	2	28	27	15	10	10
		4	41	40	22	13	13
		8	49	48	25	18	18
		16	59	61	29	24	24
		32	92	93	36	29	30
twod10	250	2	38	38	21	18	18
		4	47	47	26	21	21
		8	58	58	30	24	25
		16	77	78	35	28	29
		32	94	94	39	32	32

tioners lead to convergence for these matrices. In all partitioning instances of the `mutex` matrices, the solve phase time with the proposed BT preconditioner is less than those with the other preconditioners. On the other hand, the total running time of BGS preconditioner

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 5.6
Running times (in seconds) for GMRES(50) without preconditioning (GMRES column) and with BJ, BD, BGS, PS, and BT preconditioning for the `mutex` *matrices.*

Matrix	GMRES Total time Solve	K	Preconditioner const BJ	BD	BGS	PS	BT	Solve BJ	BD	BGS	PS	BT
mutex09	7.0	2	3.55	4.28	3.60	4.53	4.30	3.55	3.94	1.82	1.96	1.42
		4	1.39	2.84	1.52	3.33	2.95	3.45	4.23	1.82	2.37	1.51
		8	0.78	2.92	1.04	3.67	3.16	3.19	4.45	1.99	2.86	1.72
		16	0.51	3.15	1.06	4.44	3.62	3.22	4.80	2.60	3.94	2.07
		32	0.48	3.28	1.45	5.47	4.16	2.95	4.83	3.63	5.54	2.82
mutex12	27.2	2	17.75	20.61	17.95	21.75	20.72	17.69	19.42	9.18	8.07	7.23
		4	5.06	10.84	5.49	12.63	11.10	14.76	18.59	8.05	8.29	6.21
		8	2.44	11.49	3.37	14.58	12.36	12.97	17.87	7.79	9.98	6.46
		16	1.74	11.05	3.42	16.11	12.69	11.70	18.89	9.47	12.13	7.47
		32	1.59	11.11	4.62	19.81	13.97	11.42	18.83	13.35	17.92	9.13

is always the minimum except in $K = 32$-way partitioning of these two `mutex` matrices, in which case BJ gives the minimum total running time. We observe that the `mutex` matrices are the worst case for the construction of the BD, PS, and BT preconditioners, since the size of the separator set is very large already for $K = 2$, thus forming the Schur complement is very time-consuming. Note that PS and BT also suffer from the cost of copying the large A_{12} blocks (and A_{21} in PS).

Table 5.7 contains the running times of the preconditioned GMRES with the BGS, PS, and BT preconditioners for the larger matrices in each matrix family. As seen from the table, for these matrices (whose partitions have small separators) the preconditioner construction phase times are always smaller than the solve phase times with all preconditioners. Although the construction phase times with the BGS preconditioners are always smaller than those with the BT preconditioner, BT preconditioner is faster than BGS in all instances. Note that the ith iteration of GMRES after a restart requires i inner product computations with vectors of length N [3]. Therefore, the performance gains in the solve phase with the BT preconditioners are not only due to the savings in preconditioner solves and matrix-vector multiplies, but also due to the savings in the inner product computations. Recall from Table 5.5 that the number of iterations with the BT and and PS preconditioners are almost the same. However, the BT preconditioner is always faster than the PS preconditioner.

6. Conclusions. We have investigated block triangular preconditioning strategies for M-matrices, with an emphasis on the singular systems arising from Markov chain modeling. These preconditioners require a 2×2 block partitioning and suitable Schur complement approximations. The resulting preconditioner can be applied exactly or inexactly, leading to several possible variants, including block diagonal ones.

Numerical experiments with preconditioned GMRES using test matrices from MARCA allowed us to identify the most efficient variant and to rule out some of the others. Block triangular preconditioning with inexact solves obtained by means of (local) ILUTH factorizations was found to be superior to several other approaches, including block diagonal, block Gauss–Seidel, and product splitting preconditioning. Furthermore, the numerical experiments indicate that, for most problems, the number of iterations grows slowly with the number of parts (subdomains). This suggests that the inexact block triangular preconditioner should perform very well in a parallel implementation.

Acknowledgment. We thank Billy Stewart for making the MARCA software available.

TABLE 5.7
Running times (in seconds) for GMRES(50) with BGS, PS, and BT preconditioners for the larger matrices in each family.

Matrix	K	Total time					
		Precond const			Solve		
		BGS	PS	BT	BGS	PS	BT
ncd10	2	1.18	1.93	1.68	19.40	5.39	4.13
	4	1.04	1.99	1.59	15.85	6.16	4.01
	8	1.01	2.31	1.58	33.95	7.74	3.85
	16	1.01	3.04	1.67	45.84	12.34	4.30
	32	1.09	4.55	2.00	71.01	20.82	5.03
qnatm07	2	3.68	4.18	4.04	11.43	13.62	11.20
	4	2.34	2.98	2.71	11.69	15.12	11.02
	8	1.66	2.53	2.04	13.23	18.73	11.23
	16	1.34	2.75	1.84	18.07	26.62	12.13
	32	1.30	3.72	1.96	23.85	46.35	14.73
tcomm20	2	0.10	0.13	0.11	0.43	0.50	0.40
	4	0.10	0.10	0.10	0.63	0.74	0.50
	8	0.10	0.18	0.10	0.93	1.28	0.65
	16	0.10	0.20	0.10	1.98	2.47	0.97
	32	0.10	0.30	0.10	5.01	8.30	2.21
twod10	2	7.04	7.92	7.64	10.46	10.90	8.89
	4	6.65	7.55	7.15	12.85	14.32	10.43
	8	4.95	6.47	5.64	14.41	18.83	11.17
	16	4.09	6.43	4.89	16.78	30.60	13.00
	32	3.02	7.17	4.18	18.27	53.05	13.82

REFERENCES

[1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, New York, NY, USA, 1994.

[2] C. AYKANAT, A. PINAR, AND Ü. V. ÇATALYÜREK, *Permuting sparse rectangular matrices into block-diagonal form*, SIAM J. Sci. Comput., 25 (2004), pp. 1860–1879.

[3] R. BARRETT, M. BERRY, T. F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE, AND H. A. VAN DER VORST, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994.

[4] M. BENZI AND T. DAYAR, *The arithmetic mean method for finding the stationary vector of Markov chains*, Parallel Algorithms Appl., 6 (1995), pp. 25–37.

[5] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[6] M. BENZI, F. SGALLARI, AND G. SPALETTA, *A parallel block projection method of the Cimmino type for finite Markov chains*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer Academic Publishers, Boston/London/Dordrecht, 1995, pp. 65–80.

[7] M. BENZI AND D. B. SZYLD, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–321.

[8] M. BENZI, D. B. SZYLD, AND A. VAN DUIN, *Orderings for incomplete factorization preconditioning of nonsymmetric problems*, SIAM J. Sci. Comput., 20 (1999), pp. 1652–1670.

[9] M. BENZI AND M. TŮMA, *A parallel solver for large-scale Markov chains*, Appl. Numer. Math., 41 (2002), pp. 135–153.

[10] M. BENZI AND B. UÇAR, *Product preconditioning for Markov chain problems*, in Proceedings of the 2006 Markov Anniversary Meeting, A. N. Langville and W. J. Stewart, eds., Raleigh, NC, 2006, (Charleston, SC), Boson Books, pp. 239–256.

[11] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, 1979. Reprinted by SIAM, Philadelphia, PA, 1994.

[12] P. BUCHHOLZ, M. FISCHER, AND P. KEMPER, *Distributed steady state analysis using Kronecker algebra*, in Numerical Solutions of Markov Chains (NSMC'99), B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 76–95.

[13] T. N. BUI AND C. JONES, *Finding good approximate vertex and edge partitions is NP hard*, Inform. Process. Lett., 42 (1992), pp. 153–159.

[14] T. N. BUI AND C. JONES, *A heuristic for reducing fill in sparse matrix factorization*, in Proc. 6th SIAM

Conf. Parallel Processing for Scientific Computing, SIAM, Philadelphia, PA, 1993, pp. 445–452.

[15] T. DAYAR AND W. J. STEWART, *Comparison of partitioning techniques for two-level iterative solvers on large, sparse Markov chains*, SIAM J. Sci. Comput., 21 (2000), pp. 1691–1705.

[16] P. FERNANDES, B. PLATEAU, AND W. J. STEWART, *Efficient descriptor-vector multiplications in stochastic automata networks*, J. ACM, 45 (1998), pp. 381–414.

[17] R. W. FREUND AND M. HOCHBRUCK, *On the use of two QMR algorithms for solving singular systems and applications in Markov chain modeling*, Numer. Linear Algebra Appl., 1 (1994), pp. 403–420.

[18] R. W. FREUND AND F. JARRE, *A QMR-based interior-point algorithm for solving linear programs*, Math. Program., 76 (1996), pp. 183–210.

[19] M. HAGEMANN AND O. SCHENK, *Weighted matchings for preconditioning symmetric indefinite linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 403–420.

[20] B. HENDRICKSON AND R. LELAND, *A multilevel algorithm for partitioning graphs*, in Supercomputing'95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM), New York, NY, USA, 1995, ACM Press, p. 28.

[21] B. HENDRICKSON AND E. ROTHBERG, *Improving the run time and quality of nested dissection ordering*, SIAM J. Sci. Comput., 20 (1998), pp. 468–489.

[22] I. C. F. IPSEN, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[23] I. C. F. IPSEN AND C. D. MEYER, *The idea behind Krylov subspace methods*, Amer. Math. Monthly, 105 (1998), pp. 889–899.

[24] M. JARRAYA AND D. EL BAZ, *Asynchronous iterations for the solution of Markov systems*, in Numerical Solutions of Markov Chains (NSMC'99), B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 335–338.

[25] G. KARYPIS AND V. KUMAR, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1998), pp. 359–392.

[26] G. KARYPIS AND V. KUMAR, *MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices version 4.0*, University of Minnesota, Department of Computer Science/Army HPC Research Center, Minneapolis, MN 55455, September 1998.

[27] W. J. KNOTTENBELT AND P. G. HARRISON, *Distributed disk-based solution techniques for large Markov models*, in Numerical Solutions of Markov Chains (NSMC'99), B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 58–75.

[28] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[29] C. D. MEYER, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.

[30] V. MIGALLÓN, J. PENADÉS, AND D. B. SZYLD, *Experimental studies of parallel iterative solutions of Markov chains with block partitions*, in Numerical Solutions of Markov Chains (NSMC'99), B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 96–110.

[31] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.

[32] B. PHILIPPE, Y. SAAD, AND W. J. STEWART, *Numerical methods in Markov chain modeling*, Oper. Res., 40 (1992), pp. 1156–1179.

[33] A. PINAR AND B. HENDRICKSON, *Partitioning for complex objectives*, in Proceedings of the 15th International Parallel and Distributed Processing Symposium (CDROM), Washington, DC, USA,, 2001, IEEE Computer Society, p. 121.

[34] P. K. POLLET AND S. E. STEWART, *An efficient procedure for computing quasi-stationary distributions of Markov chains with sparse transition structure*, Adv. in Appl. Probab., 26 (1994), pp. 68–79.

[35] Y. SAAD, *Preconditioned Krylov subspace methods for the numerical solution of Markov chains*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer Academic Publishers, Boston/London/Dordrecht, 1995, pp. 65–80.

[36] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.

[37] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[38] H. SCHNEIDER, *Theorems on M-splittings of a singular M-matrix which depend on graph structure*, Linear Algebra Appl., 58 (1984), pp. 407–424.

[39] V. SIMONCINI AND D. SZYLD, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[40] W. J. STEWART, *MARCA Models: A collection of Markov chain models*. URL http://www.csc.ncsu.edu/faculty/stewart/MARCA_Models/MARCA_Models.html.

[41] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

ETNA
Kent State University
etna@mcs.kent.edu

[42] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems*, SIAM J. Sci. Comput., 12 (1992), pp. 631–644.

[43] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall,Englewood Cliffs, New Jersey, 1962. Second Edition, revised and expanded, Springer, Berlin, Heidelberg, New York, 2000.

[44] Z. I. WOŹNICKI, *Nonnegative splitting theory*, Japan. J. Indust. Appl. Math., 11 (1994), pp. 289–342.