SIAM J. SCI. COMPUT.
Vol. 23, No. 2, pp. 442–462

ON THE CONSTRUCTION OF DEFLATION-BASED PRECONDITIONERS*

J. FRANK† AND C. VUIK‡

Abstract. In this article we introduce new bounds on the effective condition number of deflated and preconditioned-deflated symmetric positive definite linear systems. For the case of a subdomain deflation such as that of Nicolaides [*SIAM J. Numer. Anal.*, 24 (1987), pp. 355–365], these theorems can provide direction in choosing a proper decomposition into subdomains. If grid refinement is performed, keeping the subdomain grid resolution fixed, the condition number is insensitive to the grid size. Subdomain deflation is very easy to implement and has been parallelized on a distributed memory system with only a small amount of additional communication. Numerical experiments for a steady-state convection-diffusion problem are included.

Key words. deflation, preconditioners, optimal methods, parallel computing, conjugate gradients

AMS subject classifications. 65F10, 65F50, 65N22

PII. S1064827500373231

1. Background: Preconditioning and deflation.

It is well known that the convergence rate of the conjugate gradient method is bounded as a function of the condition number of the system matrix to which it is applied. Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite. We assume that the vector $f \in \mathbb{R}^n$ represents a discrete function on a grid Ω and that we are searching for the vector $u \in \mathbb{R}^n$ on Ω which solves the linear system

$$Au = f.$$

Such systems are encountered, for example, when a finite volume/difference/element method is used to discretize an elliptic partial differential equation (PDE) defined on the continuous analogue of Ω. In particular our goal is to develop efficient serial and parallel methods for applications in incompressible fluid dynamics; see [28, 27].

Let us denote the spectrum of A by $\sigma(A)$ and the ith eigenvalue in nondecreasing order by $\lambda_i(A)$ or simply by λ_i when it is clear to which matrix we are referring. After k iterations of the conjugate gradient method, the error is bounded by (cf. [10, Thm. 10.2.6]

$$\|u - u_k\|_A \leq 2\,\|u - u_0\|_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \tag{1.1}$$

where $\kappa = \kappa(A) = \lambda_n/\lambda_1$ is the spectral condition number of A and the A-norm of u is given by $\|u\|_A = (u^T Au)^{1/2}$. The error bound (1.1) does not tell the whole story, however, because the convergence may be significantly faster if the eigenvalues of A are clustered [23].

*Received by the editors June 6, 2000; accepted for publication September 27, 2000; published electronically July 10, 2001.
http://www.siam.org/journals/sisc/23-2/37323.html
†CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands (jason@cwi.nl). The research of this author was supported in part by Delft University of Technology.
‡Delft University of Technology, Faculty of Information Technology and Systems, Department of Applied Mathematical Analysis, P.O. Box 5031, 2600 GA Delft, The Netherlands (vuik@math.tudelft.nl).

When A is the discrete approximation of an elliptic PDE, the condition number can become very large as the grid is refined, thus slowing down convergence. In this case it is advisable to solve, instead, a preconditioned system $K^{-1}Au = K^{-1}f$, where the symmetric positive definite preconditioner K is chosen such that $K^{-1}A$ has a more clustered spectrum or a smaller condition number than that of A. Furthermore, K must be cheap to solve relative to the improvement it provides in convergence rate. A final desirable property in a preconditioner is that it should parallelize well, especially on distributed memory computers. Probably the most effective preconditioning strategy in common use is to take $K = LL^T$ to be an incomplete Cholesky (IC) factorization of A [18]. For discretizations of second order PDEs in two dimensions, defined on a grid with spacing h, one finds, with IC factorization, $\kappa \sim h^{-2}$; with a modified IC factorization [11, 1], $\kappa \sim h^{-1}$; and with a multigrid cycle, $\kappa \sim 1$. Preconditioners such as multigrid and some domain decomposition methods, for which the condition number of the preconditioned system is independent of the grid size, are termed *optimal*.

Another preconditioning strategy that has proven successful when there are a few isolated extremal eigenvalues is *deflation* [20, 16, 17]. Let us define the projection P by

$$P = I - AZ(Z^TAZ)^{-1}Z^T, \quad Z \in \mathbb{R}^{n\times m}, \tag{1.2}$$

where Z is the deflation subspace, i.e., the space to be projected out of the residual, and I is the identity matrix of appropriate size. We assume that $m \ll n$ and that Z has rank m. Under this assumption $A_c \equiv Z^TAZ$ may be easily computed and factored and is symmetric positive definite. Since $u = (I - P^T)u + P^Tu$ and because

$$(I - P^T)u = Z(Z^TAZ)^{-1}Z^TAu = ZA_c^{-1}Z^Tf \tag{1.3}$$

can be immediately computed, we need only compute P^Tu. In light of the identity $AP^T = PA$, we can solve the deflated system

$$PA\tilde{u} = Pf \tag{1.4}$$

for $\tilde{u}$ using the conjugate gradient method and premultiply this by P^T. Obviously (1.4) is singular, and this raises a few questions. First, the solution $\tilde{u}$ may contain an arbitrary component in the null space of PA, i.e., in span$\{Z\}$.[1] This is not a problem, however, because the projected solution $P^T\tilde{u}$ is unique. Second, what consequences does the singularity of (1.4) imply for the conjugate gradient method?

Kaasschieter [14] notes that a positive semidefinite system can be solved as long as the right-hand side is consistent (i.e., as long as $f = Au$ for some u). This is certainly true for (1.4), where the same projection is applied to both sides of the nonsingular system. Furthermore, he notes (with reference to [23]) that because the null space never enters the iteration, the corresponding zero-eigenvalues do not influence the convergence. Motivated by this fact, we define the *effective condition number* of a positive semidefinite matrix $C \in \mathbb{R}^{n\times n}$ with corank m to be the ratio of its largest to smallest *positive* eigenvalues:

$$\kappa_{\text{eff}}(C) = \frac{\lambda_n}{\lambda_{m+1}}.$$

[1] We will use the notation span$\{Z\}$ to denote the column space of Z.

Example. To see that the condition number of PA may be better than that of A, consider the case in which Z is the invariant subspace of A corresponding to the smallest eigenvalues. Note that $PAZ = 0$, so that PA has m zero-eigenvalues. Furthermore, since A is symmetric positive definite, we may choose the remaining eigenspace Y in the orthogonal complement of span$\{Z\}$, i.e., $Y^T Z = 0$ so that $PY = Y$. However, $AY = YB$ for some invertible B; therefore, $PAY = PYB = YB$, and span$\{Y\}$ is an invariant subspace of PA. Evidently, when Z is an invariant subspace of A,

$$\kappa_{\text{eff}}(PA) = \frac{\lambda_n(A)}{\lambda_{m+1}(A)}.$$

In summary, deflation of an invariant subspace cancels the corresponding eigenvalues, leaving the rest of the spectrum untouched.

This idea has been exploited by several authors. For nonsymmetric systems, approximate eigenvectors can be extracted from the Krylov subspace produced by GMRES. Morgan [19] uses this approach to improve the convergence after a restart. In this case, deflation is not applied as a preconditioner, but the deflation vectors are augmented with the Krylov subspace, and the minimization property of GMRES ensures that the deflation subspace is projected out of the residual. For more discussion on deflation methods for nonsymmetric systems, see [15, 8, 6, 21, 5, 2]. Other authors have attempted to choose a subspace a priori that effectively represents the slowest modes. In [29] deflation is used to remove a few stubborn but *known* modes from the spectrum. Mansfield [16] shows how Schur-complement-type domain decomposition methods can be seen as a series of deflations. Nicolaides [20] chooses Z to be a piecewise constant interpolation from a set of m subdomains and points out that deflation might be effectively used with a conventional preconditioner. Mansfield [17] uses the same "subdomain deflation" in combination with damped Jacobi smoothing, obtaining a preconditioner which is related to the two-grid method.

In this article we introduce new bounds on the effective condition number of deflated and preconditioned-deflated symmetric positive definite linear systems. For the case of a subdomain deflation such as that of Nicolaides [20], these theorems can provide direction in choosing a proper decomposition into subdomains. If grid refinement is done keeping the subdomain grid resolution fixed, the condition number is insensitive to the grid size. Subdomain deflation is very easy to implement and has been parallelized on a distributed memory system with only a small amount of additional communication. Numerical experiments for a steady-state convection-diffusion problem are included.

2. A condition number bound for deflation. Nicolaides [20] proves the following bound on the spectrum of PA:

$$\lambda_{m+1}(PA) = \min \frac{v^T v}{v^T A^{-1} v}, \quad \lambda_n(PA) = \max \frac{v^T v}{v^T A^{-1} v},$$

where v is taken in span$\{Z\}^\perp$. In this section we give a bound of a different flavor which will be used in the subsequent sections to construct a preconditioning strategy with an optimal convergence property.

First we need the following result on the preservation of positive semidefiniteness under deflation.

LEMMA 2.1. *Let R be positive semidefinite and P be a projection ($P^2 = P$); then if PR is symmetric, it is positive semidefinite.*

Proof. By hypothesis, $0 \leq u^T R u$ for all u. In particular, $0 \leq (P^T u)^T R (P^T u) = u^T P R P^T u$ so that $PRP^T = P^2 R = PR$ is positive semidefinite. ∎

The next theorem provides a bound on the condition number of PA and is our main result.

THEOREM 2.2. *Let A be symmetric positive definite, let P be defined by* (1.2)*, and suppose there exists a splitting $A = C + R$ such that C and R are symmetric positive semidefinite with $\mathcal{N}(C) = \text{span}\{Z\}$ the null space of C. Then*

$$\lambda_i(C) \leq \lambda_i(PA) \leq \lambda_i(C) + \lambda_{\max}(PR). \tag{2.1}$$

Moreover, the effective condition number of PA is bounded by

$$\kappa_{\text{eff}}(PA) \leq \frac{\lambda_n(A)}{\lambda_{m+1}(C)}. \tag{2.2}$$

Proof. From (1.2) it is obvious that PA is symmetric. Since Z is in the null space of C, we have that $PC = C$ and is therefore also symmetric by hypothesis. Symmetry of $PR = PA - C$ follows immediately; and by assumption R is positive semidefinite, so we can apply Lemma 2.1 to arrive at $\lambda_{\min}(PR) \geq 0$, with equality holding in any case due to singularity of P. The bound (2.1) now follows from Theorem 8.1.5 of [10]:

$$\lambda_i(PC) + \lambda_{\min}(PR) \leq \lambda_i(PA) \leq \lambda_i(PC) + \lambda_{\max}(PR).$$

Furthermore, because $PA = A - AZ(Z^T AZ)^{-1}(AZ)^T$ is the difference of positive (semi-)definite matrices, the same theorem (Theorem 8.1.5 of [10]) gives $\lambda_{\max}(PA) \leq \lambda_{\max}(A)$. This upper bound together with the lower bound in (2.1) proves (2.2). ∎

There is also a preconditioned version of the previous theorem.

THEOREM 2.3. *Assume the conditions of Theorem* 2.2 *and let K be a symmetric positive definite preconditioner with Cholesky factorization $K = LL^T$. Then*

$$\lambda_i(L^{-1}CL^{-T}) \leq \lambda_i(L^{-1}PAL^{-T}) \leq \lambda_i(L^{-1}CL^{-T}) + \lambda_{\max}(L^{-1}PRL^{-T}), \tag{2.3}$$

and the effective condition number of $L^{-1}PAL^{-T}$ is bounded by

$$\kappa_{\text{eff}}(L^{-1}PAL^{-T}) \leq \frac{\lambda_n(L^{-1}AL^{-T})}{\lambda_{m+1}(L^{-1}CL^{-T})}. \tag{2.4}$$

Proof. Define $\hat{A} = L^{-1}AL^{-T}$, $\hat{C} = L^{-1}CL^{-T}$, $\hat{R} = L^{-1}RL^{-T}$ (all congruence transformations), $\hat{Z} = L^T Z$, and

$$\hat{P} = I - \hat{A}\hat{Z}(\hat{Z}^T \hat{A}\hat{Z})^{-1}\hat{Z}^T = L^{-1}PL.$$

Note that $\hat{P}$ is a projection and $\hat{P}\hat{A}$ is symmetric, and also that $\hat{Z}$ is in the null space of $\hat{C}$ so that $\hat{P}\hat{C} = \hat{C}$. Thus, Theorem 2.2 applies directly to the deflated system matrix $\hat{P}\hat{A}$. The conclusions follow immediately from the definitions of $\hat{A}$ and $\hat{C}$. ∎

Remark. Experience with discretized PDEs indicates that the greatest improvement in convergence is obtained by removing the smallest eigenvalues from the spectrum. It is therefore the lower bounds of (2.1) and (2.3) which are of most concern. Theorem 2.3 suggests that it might be better to construct a preconditioner for C rather than for A in this case. However, care should be taken that a good preconditioner for C does not increase the upper bound in (2.3) when applied to A. See Kaasschieter [14] for a discussion about preconditioning indefinite systems.

In the next section we consider applications of Theorems 2.2 and 2.3 in lieu of a specific choice of the subspace of deflation Z.

3. Subdomain deflation. The results of the previous section are independent of the choice of deflation subspace Z in (1.2). As mentioned in section 1, deflation of an eigenspace cancels the corresponding eigenvalues without affecting the rest of the spectrum. This has led some authors to try to deflate with "nearly invariant" subspaces obtained during the iteration, and led others to try to choose in advance subspaces which represent the extremal modes.

For the remainder of this article we make a specific choice for the subspace Z in (1.2), based on a decomposition of the domain Ω with index set $\mathcal{I} = \{i \,|\, u_i \in \Omega\}$ into m nonoverlapping subdomains Ω_j, $j = 1, \ldots, m$, with respective index sets $\mathcal{I}_j = \{i \in \mathcal{I} \,|\, u_i \in \Omega_j\}$. We assume that the Ω_j are simply connected graphs covering Ω. Define Z by

$$z_{ij} = \begin{cases} 1, & i \in \mathcal{I}_j, \\ 0, & i \notin \mathcal{I}_j. \end{cases} \tag{3.1}$$

With this choice of Z, the projection (1.2) will be referred to as *subdomain deflation*. Such a deflation subspace has been used by Nicolaides [20] and Mansfield [16, 17].

This choice of deflation subspace is related to domain decomposition and multigrid methods. The projection P can be seen as a subspace correction in which each subdomain is agglomerated into a single cell; see, for example, [13]. Within the multigrid framework, P can be seen as a coarse grid correction using a piecewise constant interpolation operator with very extreme coarsening.

Note that the matrix $A_c = Z^T AZ$, the projection of A onto the deflation subspace Z, has sparsity pattern similar to that of A. We will see that the effective condition number of PA improves as the number of subdomains is increased (for a fixed problem size). However, this implies that the dimension of A_c also increases, making direct solution expensive. By analogy with multigrid, it might be advantageous in some applications to solve A_c recursively.[2] In a parallel implementation this would lead to additional idle processor time, as it does with multigrid.

3.1. Application to Stieltjes matrices. Using subdomain deflation, we can identify matrices C and R needed for application of Theorems 2.2 and 2.3 to the class of irreducibly diagonally dominant Stieltjes matrices (i.e., symmetric M-matrices). Such matrices commonly arise as a result of discretization of symmetric elliptic and parabolic PDEs. For our purposes the following characteristics are important:

- A is symmetric positive definite and irreducible.
- $a_{ii} > 0$, $a_{ij} \leq 0$ for $i \neq j$.
- $a_{ii} + \sum_{j \neq i} a_{ij} \geq 0$ with strict inequality holding for some i.

For a matrix A, define the subdomain block-Jacobi matrix $B(A) \in \mathbb{R}^{n \times n}$ associated to A by

$$b_{ij} = \begin{cases} a_{ij} & \text{if } i, j \in \mathcal{I}_k, \text{ for some } k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Notice that since each block B_{jj} is a principle submatrix of A, it is symmetric positive definite. Also, since B is obtained from A by deleting off-diagonal blocks containing only negative elements, the B_{jj} are at least as diagonally dominant as the corresponding rows of A. Furthermore, the irreducibility of A implies that A itself cannot be

[2] A referee pointed out to us that the two-level method with direct solution of A_c has suboptimal complexity. On the other hand, for the examples considered in this article, A_c is too small for a second coarsening.

written in block diagonal form, so to construct B it is necessary to delete at least one nonzero block from each block-row. As a result, at least one row of each B_{jj} is strictly diagonally dominant. We will further assume that the so-constructed B_{jj} are irreducible.[3] It follows from Corollary 6.4.11 of [12] that the B_{jj} are again Stieltjes matrices.

Additionally, define $\mathbf{1} = (1, \ldots, 1)^T$ with the dimension following from the context, such that $A\mathbf{1}$ is the vector of row sums of A. Let the matrix C be defined by

$$C = B - \operatorname{diag}(B\mathbf{1}). \tag{3.3}$$

Each block C_{jj} of C has zero row sums—so $\mathbf{1}$ is in the null space of each block—but is further irreducible and weakly diagonally dominant and has the M-matrix property. According to Theorem 4.16 of [3], a singular M-matrix has a null space of rank exactly 1. It follows that the matrix Z defined by (3.1) is a basis for the null space of C.

Putting these ideas together we formulate the following.

THEOREM 3.1. *If A is an irreducibly diagonally dominant Stieltjes matrix and C defined by* (3.3) *has only irreducible blocks, then the hypotheses of Theorem* 2.2 *are met.*

Example. Consider a Poisson equation on the unit square with homogeneous Dirichlet boundary conditions

$$-\Delta u = f, \quad u = 0, u \in \partial\Omega, \quad \Omega = [0,1] \times [0,1]. \tag{3.4}$$

The problem is discretized using central finite differences on a 9×9 grid, and subdomain deflation is applied with a 3×3 decomposition into blocks of resolution 3×3. The system matrix A is pre- and postmultiplied by the inverse square root of its diagonal. Figure 3.1 shows the eigenvalues of A, PA, and C. The extreme positive eigenvalues of these three matrices are

	$\lambda_{\min}$	$\lambda_{\max}$
A	0.06	1.94
PA	0.27	1.91
C	0.25	1.50

Both the table and the figure support the conclusions of Theorem 2.2; namely, that the largest eigenvalue of A and the smallest nonzero eigenvalue of C bound the spectrum of PA. (Note that each eigenvalue of C has multiplicity equal to the number of blocks—9 in this case.) We observe also that the bounds are reasonably sharp.

Each diagonal block C_{jj} of the matrix C as defined by (3.3) can be interpreted as the discretization of a related Neumann problem on the jth subdomain. By Theorem 2.2, the effective condition number of the deflated matrix PA is determined by the smallest nonzero eigenvalue of C—in this case, the smallest nonzero eigenvalue over the set of related Neumann problems on the subdomain grids, i.e.,

$$\lambda_{m+1}(PA) = \min_j \lambda_2(C_{jj}).$$

[3] This is generally the case with matrices arising from discretization of PDEs on simply connected domains. If a block B_{ii} *is* reducible, then it may be possible to decompose B_{ii} into additional subdomains which *are* irreducible.

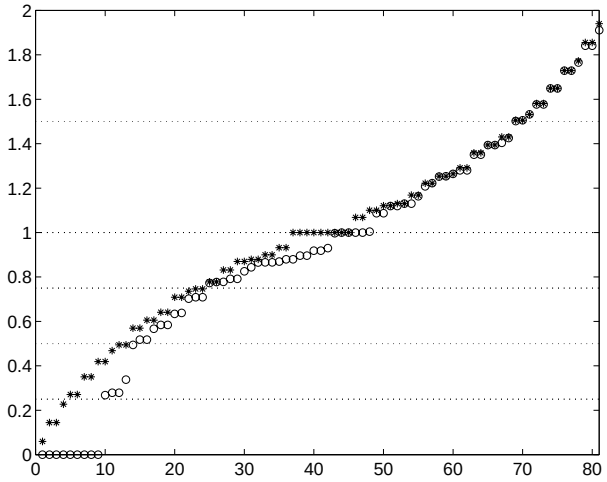

FIG. 3.1. *The eigenvalues of A (*), PA (∘), and C (· · ·).*

Theorem 2.2 thus says that *subdomain deflation effectively decouples the original system into a set of independent Neumann problems on the subdomains,* with convergence governed by the "worst-conditioned" Neumann problem. This implies an optimality result, since—if we can somehow refine the grid without affecting the worst-conditioned Neumann problem—the condition number will also remain unchanged. For an isotropic problem on a uniform grid, for example, this can be achieved by simply fixing the subgrid resolutions and performing refinement by adding more subdomains. The numerical experiments of section 6 support this observation.

3.2. Application to finite element stiffness matrices. A result similar to the above discussion on M-matrices holds for finite element stiffness matrices. We briefly describe it here. Suppose we have a domain Ω whose boundary is given by $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$, with Dirichlet boundary conditions on $\partial\Omega^D$ and Neumann boundary conditions on $\partial\Omega^N$. Let Ω be decomposed into m nonoverlapping subdomains Ω_j, $j = 1, \ldots, m$, and define the finite element decomposition of Ω by

$$\bar{\Omega} = \cup_{i\in\mathcal{I}} \bar{e}_i.$$

Let the index set $\mathcal{I}$ be divided into $m+1$ disjoint subsets $\mathcal{I}_1, \ldots, \mathcal{I}_m$ and $\mathcal{I}_r$, defined by

$$\mathcal{I}_j = \left\{ i \in \mathcal{I} \,|\, e_i \subset \Omega_j \text{ and } \bar{e}_i \cap \partial\Omega^D = \varnothing \right\},$$

and $\mathcal{I}_r = \mathcal{I} \backslash \cup_j \mathcal{I}_j$. Figure 3.2 shows an example of a domain with quadrilateral elements and two subdomains.

The stiffness matrix A is defined as the sum of elemental stiffness matrices A_{e_i}:

$$A = \sum_{i\in\mathcal{I}} A_{e_i},$$

where the elemental matrices are assumed to be positive semidefinite. This is always the case when the integrals in the element matrices are computed analytically. We assume that A is symmetric positive definite. This is normally true if the solution

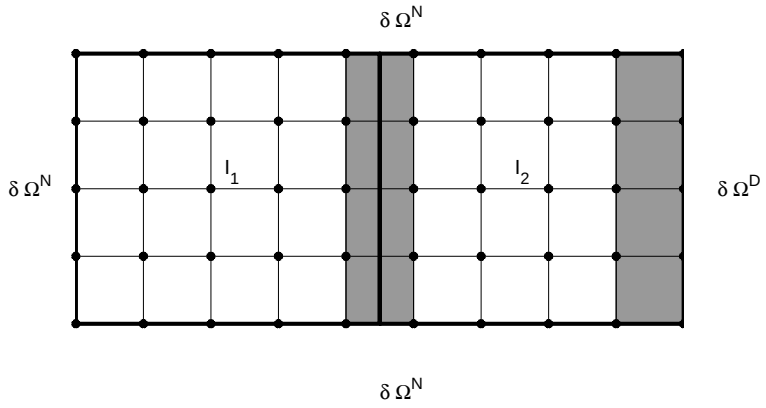

FIG. 3.2. *The domain Ω is decomposed into two subdomains (the shaded region is $\mathcal{I}_r$).*

is prescribed somewhere on the boundary. The matrix C needed for Theorem 2.2 is defined by

$$C = \sum_{i \in \mathcal{I} \setminus \mathcal{I}_r} A_{e_i}.$$

Note that C is block diagonal and the blocks C_{jj} can be interpreted as a finite element discretization of the original system on the subdomain Ω_j with homogeneous Neumann boundary conditions. This implies that $\lambda_1(C_{jj}) = 0$ and that Z is in the null space of C. Clearly C is positive semidefinite, as is

$$R = \sum_{i \in \mathcal{I}_r} A_{e_i}.$$

To ensure that $\lambda_{m+1}(C) \neq 0$, it is necessary that every grid point $x_k \in \bar{\Omega} \setminus \partial\Omega^D$ be contained in a finite element e_i with $i \in \cup_{j=1}^{m} \mathcal{I}_j$; otherwise the ith row of C contains only zero elements.

4. Guidelines for selecting subdomains. We can use the results of the previous section to give guidance in choosing a good decomposition of the domain Ω such that the "worst-conditioned related Neumann problem" is as well conditioned as possible. We consider two cases: a Poisson equation on a stretched uniform grid, and a diffusion equation with a discontinuity in the diffusion coefficient.

4.1. Large domain/grid aspect ratios. Consider the Poisson equation with homogeneous Neumann boundary conditions on a rectangular domain Ω:

$$-\Delta u = f, \quad \partial u / \partial \hat{n} = 0, \ u \in \partial\Omega,$$

where $\hat{n}$ denotes the unit normal vector to the boundary. This equation is discretized using cell-centered, central finite volumes on a uniform $N_x \times N_y$ grid having cell dimensions $h_x \times h_y$:

$$\frac{1}{h_x^2}(-u_{j-1,k} + 2u_{j,k} - u_{j+1,k}) + \frac{1}{h_y^2}(-u_{j,k-1} + 2u_{j,k} - u_{j,k+1}) = f_{j,k}$$

for $j = 0, \ldots, N_x$ and $k = 0, \ldots, N_y$. Assume central discretization of the boundary conditions

$$u_{-1,k} = u_{0,k}, \text{ etc.;}$$

then, the eigenvalues of the discretization matrix are given by

$$\lambda_{j,k} = \frac{4}{h_x^2} \sin^2 \left(\frac{j\pi}{2(N_x+1)} \right) + \frac{4}{h_y^2} \sin^2 \left(\frac{k\pi}{2(N_y+1)} \right). \tag{4.1}$$

The largest eigenvalue is λ_{N_x,N_y} and the smallest nonzero eigenvalue is the minimum of $\lambda_{0,1}$ and $\lambda_{1,0}$. Substituting into (4.1), and assuming $N_x, N_y \gg 1$, we find

$$\begin{aligned} \lambda_{N_x,N_y} &\approx \frac{4}{h_x^2} + \frac{4}{h_y^2}, \\ \lambda_{0,1} &\approx \frac{4}{h_y^2} \left(\frac{\pi}{2(N_y+1)} \right)^2 = \frac{\pi^2}{h_y^2 (N_y+1)^2}, \\ \lambda_{1,0} &\approx \frac{4}{h_x^2} \left(\frac{\pi}{2(N_x+1)} \right)^2 = \frac{\pi^2}{h_x^2 (N_x+1)^2}. \end{aligned} \tag{4.2}$$

The decomposition problem can be stated as follows: For a fixed cell aspect ratio $\mathcal{Q}_c \equiv h_x/h_y$ and a fixed total number of cells $\gamma \equiv N_x N_y = \text{const}$, find the grid aspect ratio $\mathcal{Q}_g \equiv N_x/N_y$ minimizing the effective condition number

$$\begin{aligned} \kappa_{\text{eff}} &= \max \left\{ \frac{\lambda_{N_x,N_y}}{\lambda_{0,1}}, \frac{\lambda_{N_x,N_y}}{\lambda_{1,0}} \right\} \\ &= 4/\pi^2 \max \left\{ (1+\mathcal{Q}_c^{-2})(\gamma/N_x+1)^2, (1+\mathcal{Q}_c^2)(N_x+1)^2 \right\}. \end{aligned}$$

Since both arguments of the maximum are monotone functions of positive N_x, one increasing and the other decreasing, the condition number is minimized when these arguments are equal:

$$\begin{aligned} (1+\mathcal{Q}_c^{-2})(\gamma/N_x+1)^2 &= (1+\mathcal{Q}_c^2)(N_x+1)^2, \\ \frac{1}{\mathcal{Q}_c^2} = \frac{1+\mathcal{Q}_c^{-2}}{1+\mathcal{Q}_c^2} &= \frac{(N_x+1)^2}{(N_y+1)^2} \approx \mathcal{Q}_g^2. \end{aligned}$$

Thus, for constant coefficients and a uniform grid, one should choose a decomposition such that the subdomain grid aspect ratio is the reciprocal of the cell aspect ratio; that is, one should strive for a subdomain aspect ratio $\mathcal{Q}_d \equiv (N_x h_x)/(N_y h_y)$ of 1:

$$\mathcal{Q}_d = \mathcal{Q}_g \mathcal{Q}_c = 1.$$

Example. Again take the Poisson equation on the unit square (3.4), with a grid resolution $N_x = 16$, $N_y = 32$. We compare the condition number of PA for three decompositions into 16 subdomains as shown in Figure 4.1:

	$\lambda_{\min}(C)$	$\lambda_{\min}(PA)$	$\kappa(PA)$
2×8	0.013	0.024	83.0
4×4	0.053	0.062	32.2
8×2	0.014	0.024	81.8

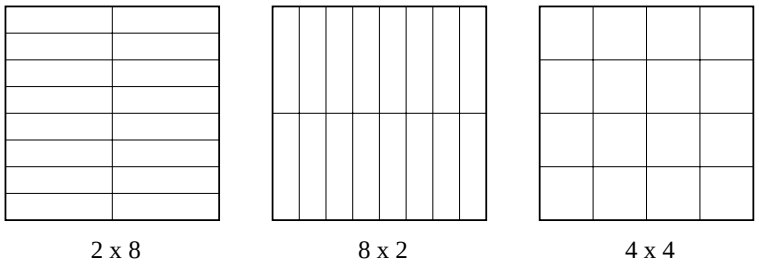

FIG. 4.1. *Three decompositions of the unit square into* 16 *subdomains.*

The 4×4 decomposition yields a subdomain aspect ratio of $\mathcal{Q}_d = 1$, and this is the best-conditioned case, as predicted.

The decomposition problem described above assumes that the grid size and the number of domains is given, and that one would like to choose the decomposition for optimal convergence rate. This would be the case, for example, if a parallel decomposition is desired on a prescribed number of processors. For a serial computation, or if there is an unlimited number of available processors, a better approach would be to ask what number of domains gives the fastest solution. Suppose we decompose into subdomains of unit aspect ratio, as described above. By comparison with (4.2), the smallest positive eigenvalue of C scales as $1/N_x^2$, with N_x the number of grid cells in the x direction for the worst-conditioned Neumann problem. Thus if we split each subdomain horizontally and vertically into four equivalent smaller subdomains, the condition number of C is improved by a factor of 4, roughly speaking. On the other hand, the dimension of the coarse grid matrix A_c will be increased by a factor of 4, causing the direct (or recursive) solution of this system to be relatively more expensive. In the extreme case of one unknown per subdomain, $A_c = A$, so that solving A_c is as expensive as solving A. Clearly, there must be an optimal value for the number of subdomains; however, this will depend on the convergence of the conjugate gradients process, and therefore also on the distribution of eigenvalues.

4.2. Discontinuous coefficients. When a problem has a large jump in coefficients at some location, poor scaling may result in slow convergence. It may be possible to improve the convergence by applying subdomain deflation, choosing the subdomain interface at the discontinuity. Since the related Neumann problems are decoupled, a diagonal scaling preconditioner is sufficient to make the condition number independent of the jump in coefficients. This is best illustrated with an example.

Consider a one-dimensional diffusion problem with Neumann and Dirichlet boundary conditions

$$-\frac{d}{dx}\alpha(x)\frac{du}{dx} = f(x), \quad x \in (0,1), \quad \frac{du}{dx}(0) = 0,\ u(1) = 1,$$

and a jump discontinuity in the coefficient

$$\alpha(x) = \begin{cases} 1, & x \leq 0.5, \\ \epsilon, & x > 0.5 \end{cases}$$

for some $\epsilon > 0$. Choose an even number n and define $h = 1/n$. The grid points are given by $x_i = ih, i = 0, \ldots, n$ and u_i is the numerical approximation for $u(x_i)$. For all $i \in \{0, 1, \ldots, n-1\} \setminus \{n/2\}$ we use the standard central difference scheme.

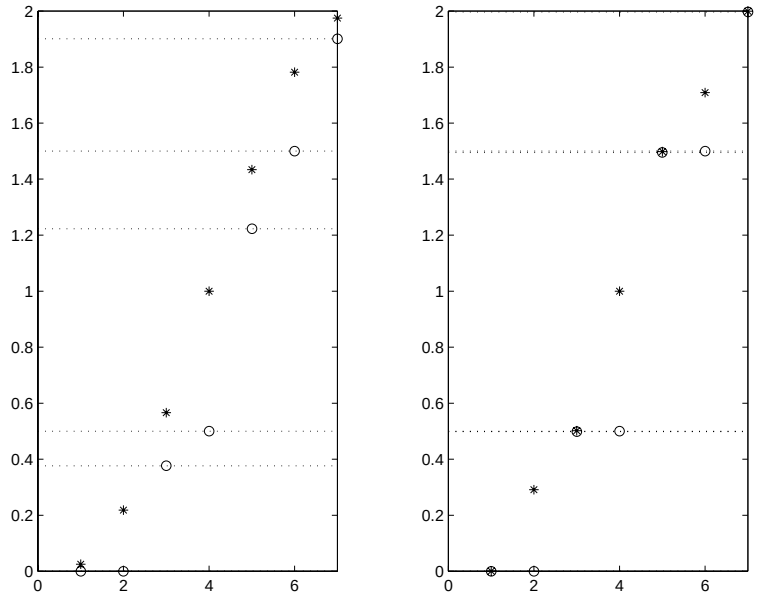

FIG. 4.2. *Eigenvalues of $D^{-1}A$ ($*$) and $D^{-1}PA$ ($\circ$) for $\epsilon = 1$ (left) and $\epsilon = 0.01$ (right). The spectrum of $D^{-1}C$ is indicated by the dotted lines.*

The unknown u_n is eliminated from the system of equations by using the Dirichlet boundary condition. For $i = 0$ the value u_{-1} is eliminated by a central discretization of the Neumann boundary condition. The resulting equation is multiplied by $1/2$ to make the coefficient matrix symmetric. Finally for $i = n/2$ the discrete equation is

$$\frac{\frac{u_{n/2}-u_{n/2-1}}{h} - \epsilon\frac{u_{n/2+1}-u_{n/2}}{h}}{h} = f(x_{n/2}).$$

The domain $\Omega = [0, 1]$ is subdivided into two subdomains $\Omega_1 = [0, 0.5]$ and $\Omega_2 = (0.5, 1]$. Note that grid point $x_{n/2} = 0.5$ belongs to Ω_1. The subdomain deflation space Z is defined by (3.1).

To construct C from A we decouple the matrix A according to the subdomains, so

$$c_{n/2+1,n/2} = c_{n/2,n/2+1} = 0.$$

The other off-diagonal elements of A and C are identical. Finally the diagonal elements of C are made equal to minus the sum of the off-diagonal elements, so

$$\sum_{j=1}^{n} c_{ij} = 0.$$

Let D be the diagonal of A. The eigenvalues of $D^{-1}A$ and $D^{-1}PA$ (equivalent to the eigenvalues of the symmetrically preconditioned case $D^{-1/2}AD^{-1/2}$, etc.) with $n = 8$ are shown in Figure 4.2 for $\epsilon = 1$ and $\epsilon = 0.01$ with the eigenvalues of $D^{-1}C$ appearing as dotted lines. Note that the smallest positive eigenvalue of $D^{-1}C$ bounds from below the smallest positive eigenvalue of $D^{-1}PA$, as predicted by Theorem 2.3.

In the following table we give the effective condition numbers relevant for convergence of the preconditioned conjugate gradient method.

ϵ	$\lambda_1(D^{-1}A)$	$\kappa(D^{-1}A)$	$\lambda_3(D^{-1}PA)$	$\kappa_{\text{eff}}(D^{-1}PA)$
1	$2.5 \cdot 10^{-2}$	$7.9 \cdot 10^{1}$	$3.8 \cdot 10^{-1}$	5.0
10^{-2}	$4.1 \cdot 10^{-4}$	$4.8 \cdot 10^{3}$	$5.0 \cdot 10^{-1}$	4.0
10^{-4}	$4.2 \cdot 10^{-6}$	$4.8 \cdot 10^{5}$	$5.0 \cdot 10^{-1}$	4.0

Due to diagonal preconditioning, the smallest nonzero eigenvalue of $D^{-1}C$ is independent of ϵ. As predicted by Theorem 2.3, the same holds for $D^{-1}PA$. The smallest eigenvalue of $D^{-1}A$, however, decreases proportionally to ϵ, leading to a large condition number and slow convergence of the conjugate gradient method applied to $D^{-1}Au = D^{-1}f$.

5. Additional considerations. In this section we discuss extension of deflation methods to the nonsymmetric case and describe an efficient parallel implementation of the subdomain deflation method.

5.1. The nonsymmetric case. A generalization of the projection P for a nonsymmetric matrix $A \in \mathbb{R}^{n\times n}$ is used in [29]. In this case there is somewhat more freedom in selecting the projection subspaces. Let P and Q be given by

$$P = I - AZ(Y^TAZ)^{-1}Y^T, \quad Q = I - Z(Y^TAZ)^{-1}Y^TA,$$

where Z and Y are suitable subspaces of dimension $n \times m$. The operator A_c on the projection subspace is given by $A_c = Y^TAZ$.[4] We have the following properties for P and Q:

- $P^2 = P$, $Q^2 = Q$.
- $PAZ = Y^TP = 0$, $Y^TAQ = QZ = 0$.
- $PA = AQ$.

To solve the system $Au = f$ using deflation, note that u can be written as

$$u = (I - Q)u + Qu$$

and that $(I-Q)u = Z(Y^TAZ)^{-1}Y^TAu = Z(Y^TAZ)^{-1}Y^Tf$ can be computed immediately (cf. (1.3)). Furthermore Qu can be obtained by solving the deflated system

$$PA\tilde{u} = Pf \tag{5.1}$$

for $\tilde{u}$ (cf. (1.4)) and premultiplying the result with Q.

Also in the nonsymmetric case, deflation can be combined with preconditioning. Suppose K is a suitable preconditioner of A, then (5.1) can be replaced by the following: solve $\tilde{u}$ from

$$K^{-1}PA\tilde{u} = K^{-1}Pf \tag{5.2}$$

and form $Q\tilde{u}$, or solve $\tilde{v}$ from

$$PAK^{-1}\tilde{v} = Pf \tag{5.3}$$

and form $QK^{-1}\tilde{v}$. Both systems can be solved by one's favorite Krylov subspace solver, such as GMRES [22], GCR [7, 25], Bi-CGSTAB [24], etc.

The question remains how to choose Y. We consider two possibilities:

[4] In multigrid terminology, Z is the projection or interpolation operator, and Y^T is the restriction operator.

1. Suppose Z consists of eigenvectors of A. Choose Y as the corresponding eigenvectors of A^T.
2. Choose $Y = Z$.

For both choices we can prove some results about the spectrum of PA.

ASSUMPTION 5.1. *We assume that A has real eigenvalues and is nondefective.*

Whenever A satisfies Assumption 5.1 there exists a matrix $X \in \mathbb{R}^{n\times n}$ such that $X^{-1}AX = \text{diag}(\lambda_1, \ldots, \lambda_n)$. For the first choice, which is related to Hotelling deflation (see [30, p. 585]), we have the following result.

LEMMA 5.1. *If A satisfies Assumption* 5.1, *$Z = [x_1 \cdots x_m]$, and Y is the matrix composed of the first m columns of X^{-T}, then*

$$X^{-1}PAX = \text{diag}(0, \ldots, 0, \lambda_{m+1}, \ldots, \lambda_n).$$

Proof. From the definition of P we obtain $PAZ = 0$, so $PAx_i = 0,\ i = 1, \ldots, m$. For the other vectors $x_i,\ i = m+1, \ldots, n$, we note that

$$PAx_i = Ax_i - AZ(Y^TAZ)^{-1}Y^TAx_i = \lambda_i x_i - AZ(Y^TAZ)^{-1}\lambda_i Y^T x_i = \lambda_i x_i. \qquad \square$$

The second choice $Y = Z$ has the following properties.

LEMMA 5.2. *For $Y = Z$ one has the following:*

(i) *If A is positive definite and Z has full rank, $A_c = Z^TAZ$ is nonsingular.*

(ii) *If A satisfies Assumption* 5.1 *and $Z = [x_1 \cdots x_m]$, the eigenvalues of PA are $\{0, \lambda_{m+1}, \ldots, \lambda_n\}$, where the zero-eigenvalue has multiplicity m.*

Proof. (i) For $Y = Z$ the matrix $A_c = Z^TAZ$ is nonsingular since $s^TA_cs > 0$ for all $s \in \mathbb{R}^m$ and $s \neq 0$.

(ii) Again $PAx_i = 0$ for $i = 1, \ldots, m$. For the other eigenvalues we define the vectors

$$v_i = x_i - AZA_c^{-1}Z^Tx_i = x_i - ZD_mA_c^{-1}Z^Tx_i,\ \ i = m+1, \ldots, n,$$

where $D_m = \text{diag}(\lambda_1, \ldots, \lambda_m)$. These vectors are nonzero, because $x_1, \ldots, x_n$ form an independent set. Multiplication of v_i by PA yields

$$PAv_i = PA(x_i - ZD_mA_c^{-1}Z^Tx_i) = PAx_i = Ax_i - AZA_c^{-1}Z^TAx_i = \lambda_i v_i,$$

which proves the lemma. $\square$

From these lemmas we conclude that both choices of Y lead to the same spectrum of PA. The second choice has the following advantages: when A is positive definite we have proven that A_c is nonsingular; it is not necessary to determine (or approximate) the eigenvectors of A^T; and finally only one set of vectors $z_1, \ldots, z_m$ has to be stored in memory. This motivates us to use the choice $Y = Z$. In our applications Z is not an approximation of an invariant subspace of A but is defined as in (3.1).

Theorems 2.2 and 2.3 do not apply to the nonsymmetric case. However, our experience has shown that the convergence of (5.1) is similar to that of (1.4) as long as the asymmetric part of A is not too dominant.

5.2. Parallel implementation. In this section we describe an efficient parallel implementation of the subdomain deflation method with Z defined by (3.1). We distribute the unknowns according to subdomain across available processors. For the discussion we will assume one subdomain per processor. The coupling with neighboring domains is realized through the use of virtual cells added to the local grids. In

this way, a block-row of $Au = f$ corresponding to the subdomain ordering

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \vdots & \vdots \\ A_{m1} & \cdots & A_{mm} \end{bmatrix} \tag{5.4}$$

can be represented locally on one processor: the diagonal block A_{ii} represents coupling between local unknowns of subdomain i, and the off-diagonal blocks of block-row i represent coupling between local unknowns and the virtual cells.

Computation of element $A_{c_{ij}}$ of $A_c = Z^T AZ$ can be done locally on processor i by summing the coefficients corresponding to block A_{ij} of (5.4): $A_{c_{ij}} = \mathbf{1}^T A_{ij} \mathbf{1}$.

Use of the deflation P within a Krylov subspace method involves premultiplying a vector v by PA:

$$PAv = (I - AZ(Z^T AZ)^{-1} Z^T) Av.$$

Assuming A_c^{-1} has been stored in factored form, this operation requires two multiplications with A. However, the special form of Z given by (3.1) allows some simplification. Since Z is piecewise constant, we can efficiently compute and store the vectors

$$w_j = Az_j = \begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix} \mathbf{1} \tag{5.5}$$

corresponding to row sums of the jth block-column of A. Note that for the ith block system the local block of w_j is nonzero only if there is coupling between subdomains i and j, and only the nonzero blocks of w_j need be stored. Thus, for a five-point stencil the number of nonzero vectors w_j which have to be stored per block is five. Furthermore, for many applications, the row sums are zero, and w_j is only nonzero on subdomain boundaries.

With the w_j stored, local computation of $AZ\tilde{e}$ for a given (m-dimensional) vector $\tilde{e}$ consists of scaling the nonzero w_j by the corresponding $\tilde{e}_j$ and summing them up: $AZ\tilde{e} = \sum_j \tilde{e}_j w_j$. The number of vector updates is five for a five-point stencil.

In parallel, we first compute and store the (nonzero parts of the) w_j and $(Z^T AZ)^{-1}$ (factored) on each processor. In particular, on processor i we store the local part $w_j = A_{ij}\mathbf{1}$ for all nonzero A_{ij}. Then to compute PAv we first perform the matrix-vector multiplication $\tilde{q} = Av$, requiring nearest neighbor communications. Next we compute the local contribution to the restriction $q = Z^T \tilde{q}$ (local summation over all grid points) and distribute the result to all processes. With this done, we solve for $\tilde{e}$ from $A_c \tilde{e} = q$ and finally compute $AZ\tilde{e} = \sum_j \tilde{e}_j w_j$ locally.

The total parallel communication involved in the matrix-vector multiplication and deflation are a nearest neighbor communication of the length of the interfaces and a global gather-broadcast of dimension m.

The computational and communication costs plus storage requirements of subdomain deflation are summarized in Table 5.1, assuming a five-point discretization stencil on an $N_x \times N_y$ grid with $M_x \times M_y$ decomposition into blocks of revolution $n_x \times n_y$ ($N_x = n_x M_x$, $N_y = n_y M_y$). The abbreviation *GaBr* (m) refers to a gather-broadcast operation in which a set of m distributed floating point numbers is gathered from the participating processors and then the whole set is returned to each processor. The construction costs are incurred only once, whereas the iteration costs are in

each conjugate gradient iteration. Also included in the table are the costs of an (in the parallel case, blockwise) incomplete factorization preconditioner with zero fill-in, ILU(0).

TABLE 5.1
Work, storage, and communication costs for deflation-based preconditioning.

	Sequential		Parallel		
	Work	Storage	Work	Storage	Comms.
Construction:					
ILU(0)	$6N_xN_y$	N_xN_y	$6n_xn_y$	n_xn_y	0
A_c	$5N_xN_y$	$5M_xM_y$	$5n_xn_y$	$5M_xM_y$	GaBr ($5M_xM_y$)
Band-factor A_c	$2M_x^3My$	$2M_x^2M_y$	$2M_x^3M_y$	$2M_x^2M_y$	0
AZ	$9N_xN_y$	$5N_xN_y$	$9n_xn_y$	$9n_xn_y$	0
Iteration:					
Backsolve IC(0):	$10N_xN_y$		$10n_xn_y$		0
Restriction: $q = Z^TAv$	N_xN_y		n_xn_y		0
Backsolve: $A_c\tilde{e} = q$	$4M_x^2M_y$		$4M_x^2M_y$		GaBr (M_xM_y)
Prolongation: $AZ\tilde{e}$	$5N_xN_y$		$5n_xn_y$		0
Vector update: $Av - AZ\tilde{e}$	N_xN_y		n_xn_y		0

Besides the items tabulated above, there are computation and communication costs associated with the matrix-vector multiplication and inner products as well as computational costs of vector updates, associated with the CG method. Based on this table, we expect the added iteration expense of deflation to be less expensive than an ILU(0) factorization, and that the method will parallelize very efficiently on a distributed memory computer.

6. Numerical experiments. All experiments in this section are conducted with PDEs discretized using cell-centered, central finite volumes on Cartesian grids in rectangular regions. The theory discussed until now makes no such assumptions, however, and should hold in a more general, unstructured setting.

In conducting numerical experiments, we are interested in the following issues: (i) verification of the theoretical results of this article, (ii) the properties of subdomain deflation for nonsymmetric systems, and (iii) the parallel performance of the method. To this end we consider three test cases:

I. Poisson equation: $-\Delta u(x, y) = f$.

II. Diffusion equation: $-\nabla \cdot \nu(x, y)\nabla u(x, y) = f$.

III. Steady-state convection-diffusion equation: $\nabla\cdot(\mathbf{a}(x, y)u(x, y))-\Delta u(x, y) = f$.

In most examples we take $f \equiv 1$, having checked that similar results are observed for a random right-hand side function. We use a global grid resolution $N_x \times N_y$, with decomposition into $M_x \times M_y$ subdomains, each of resolution $n_x \times n_y$ (thus, $N_x = n_xM_x$ and $N_y = n_yM_y$).

We solve the resulting discrete (symmetric) system using the CG method and subdomain deflation. The initial iterate is chosen to be $u^{(0)} = 0$, and convergence is declared when, in the Jth iteration, $\|r_J\| \leq tol \cdot \|r_0\|$ for $tol = 10^{-6}$.

When classical preconditioning is included, we solve $K^{-1}PAu = K^{-1}Pf$, where the preconditioner K used on the blocks is the relaxed incomplete Cholesky (RIC) factorization of [1], with relaxation parameter $\omega = 0.975$. We choose this preconditioner because it is simple to implement (for a five-point stencil, modifications occur only on the diagonal) and is reasonably effective. Certainly, more advanced preconditioners could be employed on the blocks of C.

6.1. Convergence results. In this section we give convergence results with problems I, II, and III to illustrate the insensitivity of the convergence to the number of subdomains, the optimal decomposition on stretched grids, the effectiveness of the method for problems with discontinuous coefficients, and the convergence behavior for nonsymmetric problems.

6.1.1. Near grid independence. First we illustrate the sense in which subdomain deflation can lead to nearly grid-independent convergence. The symmetric discretization matrix of problem I on $(0,1)\times(0,1)$ with homogeneous Dirichlet boundary conditions is used without preconditioning. Keeping the resolution of each subdomain fixed, the number of subdomains is increased. In so doing, the blocks of C remain roughly the same as the grid is refined, and the bound in (2.1) becomes insensitive to the number of blocks m for large enough m.

Assume $M_x = M_y$ and $n_x = n_y$. Figure 6.1 shows the scaled number of CG iterations J/n_x (note that n_x is constant along each line in the figure) for problem I as the grid is refined keeping the subdomain resolution n_x fixed at values of 10, 50, and 200. The lines are almost indistinguishable from one another. It is apparent from the figure that—using only subdomain deflation—the number of iterations required for convergence is bounded independent of the number of subdomains. The same qualitative behavior is observed with preconditioning.

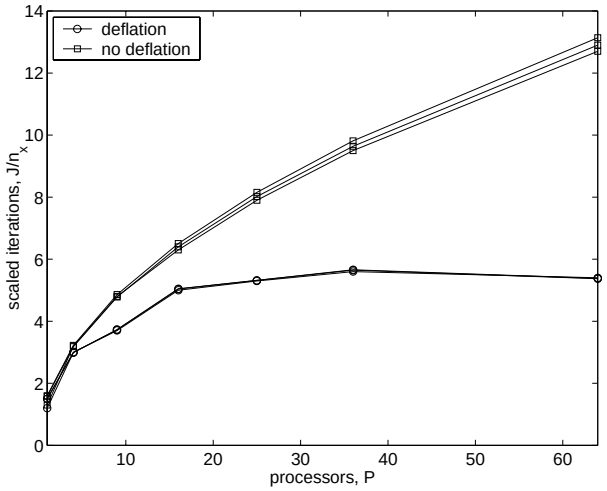

FIG. 6.1. *Number of iterations J divided by the subdomain resolution $n_x \equiv n_y \in \{10, 50, 200\}$ with and without deflation.*

6.1.2. Stretched grid. We consider problem I on $(0,3)\times(0,1)$ with homogeneous Dirichlet boundary conditions, and $N_x = 36$ and $N_y = 72$. The cell aspect ratio is $\mathcal{Q}_c = h_x/h_y = (3/36)/(1/72) = 6$. Based on the discussion of section 4.1, the best condition number is expected for a subdomain aspect ratio $\mathcal{Q}_d = 1$, associated with a subdomain grid aspect ratio of $\mathcal{Q}_g = \mathcal{Q}_d/\mathcal{Q}_c = 1/6$. Table 6.1 gives the number of iterations required for convergence for 5 different decompositions into 12 equally sized subdomains. The solution tolerance of the nonpreconditioned CG algorithm was set to $tol = 10^{-2}$, prior to the onset of superlinear convergence, to obtain these results. The 6×2 decomposition with $\mathcal{Q}_d = 1$ gives the minimum number of iterations, in keeping with the discussion. We note that if iteration is continued to

high tolerance, the superlinear convergence effect may give quite different results than shown here. This domain decomposition selection strategy is most useful when the condition number governs the convergence rate.

TABLE 6.1
Iterations required for problem I *for different decompositions.*

$M_x \times M_y$	$n_x \times n_y$	$\mathcal{Q}_d$	J
2×6	18×12	9	73
3×4	12×18	4	63
4×3	9×24	9/4	56
6×2	6×36	1	48
12×1	3×72	1/4	50

6.1.3. Discontinuous coefficients. To further illustrate the discussion of section 4.2 we give results for problem II on $(0,1) \times (0,1)$ with boundary conditions $u_x(0,y) \equiv u_y(x,0) \equiv u_y(x,1) \equiv 0$, $u(1,y) \equiv 0$. We define the diffusion coefficient to have value $\nu(x,y) = 1$ on the lower left subdomain, including its interfaces, and $\nu(x,y) = \epsilon$ elsewhere. Table 6.2 lists the iterations for the CG method with diagonal preconditioning for $M_x = M_y = 3$ and $n_x = n_y = 30$, as ϵ is decreased.

One observes that this is a very effective strategy for eliminating the effect of the jump in coefficients.

TABLE 6.2
Iterations for problem II *with discontinuous coefficients.*

ϵ	No deflation	Deflation
1	295	151
10^{-2}	460	183
10^{-4}	521	189
10^{-6}	628	189

6.1.4. A nonsymmetric example. We also illustrate the convergence of the deflation method for a convection dominated problem III on $(0,1) \times (0,1)$ with recirculating wind field $a_1(x,y) = -80xy(1-x)$, $a_2(x,y) = 80xy(1-y)$ and boundary conditions $u(x,0) \equiv u(y,0) \equiv u(x,1) \equiv 0$, $u_x(1,y) = 0$. The grid parameters are $N_x = N_y$, $M_x = M_y$, $n_x = n_y$ with grid spacing given by

$$x_i = (i/N_x)^2(3 - 2(i/N_x)).$$

The resulting system is solved using GCR truncated to a subspace of 20 vectors by dropping the vector most nearly orthogonal to the current search direction [26]. Classical preconditioning in the form of RILU(0.975) is incorporated. The restriction matrix for deflation is chosen to be $Y = Z$.

Table 6.3 compares the required number of GCR iterations as the number of subdomains is increased keeping the subdomain resolution fixed at $n_x = 50$. Although the number of iterations is not bounded in the deflated case, it grows much slower than the nondeflated case.

6.2. Parallel performance. For the results in this section, problem I will be solved on $(0,1) \times (0,1)$ with homogeneous Dirichlet boundary conditions everywhere.

The resulting equations are solved with CG preconditioned with RIC(0.975). Our implementation does not take advantage of the fact that some of the row sums may

TABLE 6.3
Scalability for a nonsymmetric problem, subdomain grid 50×50.

M_x	No deflation	Deflation
1	42	42
2	122	122
3	224	191
4	314	235
5	369	250
6	518	283
7	1007	377

TABLE 6.4
Speedup for problem I *on a* 120×120 *grid.*

p	J	t_{const}	t_{iter}	s	*eff*
1	38	$8.7 \cdot 10^{-3}$	1.3	–	–
4	58	$1.2 \cdot 10^{-2}$	0.57	2.3	0.58
9	68	$5.0 \cdot 10^{-3}$	0.33	4.0	0.44
16	64	$5.3 \cdot 10^{-3}$	0.18	7.2	0.45
25	57	$4.3 \cdot 10^{-3}$	0.15	9.0	0.36
36	50	$7.6 \cdot 10^{-3}$	0.11	11.7	0.33
64	41	$1.1 \cdot 10^{-2}$	0.11	12.3	0.19

TABLE 6.5
Speedup for problem I *on a* 480×480 *grid.*

p	J	t_{const}	t_{iter}	s	*eff*
1	120	$1.4 \cdot 10^{-1}$	67.3	–	–
4	137	$1.3 \cdot 10^{-1}$	21.8	3.1	0.77
9	138	$6.3 \cdot 10^{-2}$	9.65	7.0	0.78
16	139	$3.6 \cdot 10^{-2}$	5.60	12.0	0.75
25	121	$2.5 \cdot 10^{-2}$	3.21	21.0	0.84
36	118	$2.2 \cdot 10^{-2}$	2.27	29.7	0.82
64	100	$1.3 \cdot 10^{-2}$	1.19	56.6	0.88

be zero in (5.5). Each processor is responsible for exactly one subdomain. Parallel communications were performed with MPI, using simple point-to-point and collective communications. No exploitation of the network topology was used. Parallel results were obtained from a Cray T3E. Wall-clock times in seconds were measured using the MPI timing routine.

6.2.1. Speedup for fixed problem size. To measure the speedup, we choose $p = M_x^2$ processors for $M_x \in \{1, 2, 3, 4, 5, 6, 8\}$. The results are given in Tables 6.4 and 6.5 for $N_x = 120$ and $N_x = 480$, respectively. The total number of iterations is denoted by J; the time to construct the incomplete factorization and deflation operator is denoted by t_{const}; and the time spent in iterations is denoted by t_{iter}. The speedup is determined from $s = (t_{\text{iter}}|_{p=1}) / (t_{\text{iter}}|_{p=M_x^2})$ and parallel efficiency by $\mathit{eff} = s/p$.

In Table 6.4 the parallel efficiency decreases from 58% on 4 processors to only 19% on 64 processors, whereas in Table 6.5 efficiency increases slightly from 77% to 88%. We expect that the poorer performance in the first table is due to both a relatively large cost of solving the coarse operator A_c and a large communication-to-computation ratio for small subdomains. The following factors contribute to the parallel performance:

- As more subdomains are added, the relative size of the deflation system A_c

increases, making it more expensive to solve, but at the same time, its solution becomes a better approximation of the global solution.
- As the size of the subdomain grids decreases, the *RILU* preconditioner becomes a better approximation of the exact solution of the subdomain problems.
- Global communications become more expensive for many subdomains.
- Additionally there may be architecture-dependent effects in play.

6.2.2. Scaled performance for fixed subdomain size. Table 6.6 gives the computation times in seconds obtained with and without deflation, keeping the subdomain size fixed at $n_x \in \{5, 10, 20, 50, 100, 200\}$ as the number of processors is increased. It is clear that the effect of deflation is to make the parallel computation time less sensitive to the number of processors.

We have already seen that the number of iterations levels off as a function of the number of subdomains. The results of this table show that also the parallel iteration time becomes relatively insensitive to an increase in the number of blocks. Some overhead is incurred in the form of global communications and in solving the deflation subsystem. As a result, the computation times are not bounded independent of the number of subdomains.

Comparing the iteration counts for this problem, we note that the ratio of iterations with and without deflation is very similar to that of Figure 6.1 without preconditioning. Furthermore, the cost per iteration scales with n_x^2 for $n_x \geq 20$ (for smaller n_x, the cost of deflation offsets the advantage gained). The effect of preconditioning is to reduce the necessary number of iterations in both the deflated and undeflated cases such that the ratio of iterations remains fixed. We therefore expect that the ratio of computation times with and without deflation should reflect the ratios of Figure 6.1 as well.

TABLE 6.6
Scaled performance for problem I *with fixed subdomain size* n_x.

n_x		$p=1$	$p=4$	$p=9$	$p=16$	$p=25$	$p=36$	$p=64$
5	no defl.	$4\cdot10^{-4}$	$4\cdot10^{-3}$	$1\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-2}$	$4\cdot10^{-2}$	$4\cdot10^{-2}$
	defl.	—	$5\cdot10^{-3}$	$1\cdot10^{-2}$	$1\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-2}$	$4\cdot10^{-2}$
10	no defl.	$1\cdot10^{-3}$	$9\cdot10^{-3}$	$3\cdot10^{-2}$	$3\cdot10^{-2}$	$5\cdot10^{-2}$	$6\cdot10^{-2}$	$7\cdot10^{-2}$
	defl.	—	$1\cdot10^{-2}$	$3\cdot10^{-2}$	$4\cdot10^{-2}$	$5\cdot10^{-2}$	$6\cdot10^{-2}$	$6\cdot10^{-2}$
20	no defl.	$6\cdot10^{-3}$	$3\cdot10^{-2}$	$6\cdot10^{-2}$	$8\cdot10^{-2}$	0.12	0.15	0.18
	defl.	—	$3\cdot10^{-2}$	$7\cdot10^{-2}$	$8\cdot10^{-2}$	0.10	0.11	0.13
50	no defl.	0.11	0.34	0.51	0.69	0.94	1.10	1.37
	defl.	—	0.35	0.57	0.64	0.71	0.75	0.77
100	no defl.	0.78	2.11	2.98	4.10	5.29	6.23	8.00
	defl.	—	2.10	3.27	3.46	3.58	3.89	3.97
200	no defl.	4.96	13.3	18.6	25.3	32.8	38.6	49.7
	defl.	—	12.9	17.6	20.4	20.8	22.5	23.3

7. Conclusions. In this paper we have given new effective condition number bounds for deflated systems, both with and without conventional preconditioning. Specifically, we show that choosing the deflation subspace to be piecewise constant on subdomains effectively decouples the problem into a set of related Neumann problems, with the convergence governed by the "worst-conditioned" Neumann problem. This knowledge can help to choose an effective decomposition of the domain and is especially useful for problems with large discontinuities in the coefficients. Numerical experiments illustrate that the convergence rate is nearly independent of the num-

ber of subdomains for some problems, and that the method can be very efficiently implemented on distributed memory parallel computers.

We see the deflation approach presented here as offering improved convergence rate at a small additional cost for parallel computations using blockwise application of conventional preconditioners. The reader is referred to [9] for a comparison of blockwise incomplete factorization in the framework of nonoverlapping domain decomposition. In that reference is also a comparison of blockwise incomplete factorization with single-block incomplete factorization. In turn, to put these results in perspective, Botta et al. [4] compare a number of modern strategies including ICCG and multigrid methods.

Acknowledgments. We thank HPαC for the use of the Cray T3E, and Pieter Wesseling, Guus Segal, Jos van Kan, and a referee for helpful discussions and suggestions.

REFERENCES

[1] O. Axelsson and G. Lindskog, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math., 48 (1986), pp. 479–498.
[2] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1999), pp. 243–269.
[3] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.
[4] E. F. F. Botta, K. Dekker, Y. Notay, A. van der Ploeg, C. Vuik, F. W. Wubs, and P. M. de Zeeuw, *How fast the Laplace equation was solved in* 1995, Appl. Numer. Math., 24 (1997), pp. 439–455.
[5] K. Burrage, J. Erhel, B. Pohl, and A. Williams, *A deflation technique for linear systems of equations*, SIAM J. Sci. Comput., 19 (1998), pp. 1245–1260.
[6] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.
[7] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.
[8] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 303–318.
[9] J. Frank and C. Vuik, *Parallel implementation of a multiblock method with approximate subdomain solution*, Appl. Numer. Math., 30 (1999), pp. 403–423.
[10] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[11] I. Gustafsson, *A class of first order factorization methods*, BIT, 18 (1978), pp. 142–156.
[12] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, New York, 1993.
[13] C. B. Jenssen and P. A. Weinerfelt, *Coarse grid correction scheme for implicit multiblock Euler calculations*, AIAA J., 33 (1995), pp. 1816–1821.
[14] E. F. Kaasschieter, *Preconditioned conjugate gradients for solving singular systems*, J. Comput. Appl. Math., 24 (1988), pp. 265–275.
[15] S. A. Kharchenko and A. Y. Yeremin, *Eigenvalue translation based preconditioners for the GMRES(k) method*, Numer. Linear Algebra Appl., 2 (1995), pp. 51–77.
[16] L. Mansfield, *On the conjugate gradient solution of the Schur complement system obtained from domain decomposition*, SIAM J. Numer. Anal., 27 (1990), pp. 1612–1620.
[17] L. Mansfield, *Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1314–1323.
[18] 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. Comput., 31 (1977), pp. 148–162.
[19] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[20] R. A. Nicolaides, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.

[21] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.
[22] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[23] A. van der Sluis and H. A. van der Vorst, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.
[24] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
[25] H. A. van der Vorst and C. Vuik, *GMRESR: A family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
[26] C. Vuik, *Further experiences with GMRESR*, Supercomputer, 55 (1993), pp. 13–27.
[27] C. Vuik and J. Frank, *A parallel block preconditioner accelerated by coarse grid correction*, in High-Performance Computing and Networking, Proceedings of the 8th International Conference, HPCN Europe 2000, M. Bubak, H. Afsarmanesh, R. Williams, and B. Hertzberger, eds., Lecture Notes in Comput. Sci. 1823, Springer-Verlag, Berlin, 2000, pp. 99–108.
[28] C. Vuik, J. Frank, and A. Segal, *A parallel block-preconditioned GCR method for incompressible flow problems*, Future Generation Computer Systems, to appear.
[29] C. Vuik, A. Segal, and J. A. Meijerink, *An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients*, J. Comput. Phys., 152 (1999), pp. 1–19.
[30] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1992.