CWI

Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

On the construction of deflation-based preconditioners

J. Frank, C. Vuik

Modelling, Analysis and Simulation (MAS)

MAS-R0009 February, 2000

Report MAS-R0009
ISSN 1386-3703

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

On the construction of deflation-based preconditioners

J. Frank
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

C. Vuik
TU Delft, Fac. ITS
P.O. Box 356, 2600 AJ Delft, The Netherlands

ABSTRACT

In this article we introduce new bounds for the condition numbers of deflated symmetric positive definite systems, used with and without classical preconditioning. For the case of a subdomain deflation such as that of Nicolaides (1987), these lemmas can provide direction in choosing a proper decomposition into subdomains and a proper choice of classical preconditioner. If grid refinement is done keeping the subdomain resolutions fixed, the condition number can be shown to be independent of the number of subdomains.

1991 Mathematics Subject Classification: 65F10, 65F50, 65N22

Keywords and Phrases: deflation, incomplete factorizations, preconditioners, parallel computing, conjugate gradients

1. Background: preconditioning and deflation

It is well known that the convergence rate of the conjugate gradient method depends on 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 $b \in \mathbb{R}^n$ represents a discrete function on a grid Ω and that we are searching for the vector $x \in \mathbb{R}^n$ on Ω which solves the linear system

$$Ax = b.$$

Such systems are encountered, for example, when a finite volume/difference/element method is used to discretize an elliptic partial differential equation on a domain $\mathcal{D}$, the continuous analog of Ω.

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. [8], Thm. 10.2.6):

$$\|x - x_k\|_A \leq 2\,\|x - x_0\|_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \tag{1.1}$$

where $\kappa(A) = \lambda_n/\lambda_1$ is the condition number of A and the A-norm of x is given by $\|x\|_A = (x^TAx)^{1/2}$. 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 into groups [21].

When A is the discrete approximation to an elliptic PDE, the condition number can become very large as the grid is refined, slowing convergence. In this case it is advisable to solve, instead, a preconditioned system $K^{-1}Ax = K^{-1}b$, where the symmetric positive definite preconditioner K is chosen such that the spectrum of $K^{-1}A$ is either more clustered or has a smaller condition number than that of A. Furthermore, K must be relatively cheap to solve compared 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 factorization of A [16]. For discretizations of second order PDEs in two dimensions, defined on a grid with spacing h, we have with incomplete Cholesky factorization, $\kappa \sim h^{-2}$; with a modified IC factorization[9, 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* [18, 14, 15]. In this case we 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 I is the identity matrix of appropriate size. We assume that $m \ll n$ and that Z has rank m so that $A_c \equiv Z^TAZ$ may be easily computed and factored. Note that A_c is symmetric positive definite in this case. Since $x = (I - P^T)x + P^Tx$ and since

$$(I - P^T)x = Z(Z^TAZ)^{-1}Z^TAx = ZA_c^{-1}Z^Tb \tag{1.3}$$

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

$$PA\tilde{x} = Pb \tag{1.4}$$

for $\tilde{x}$ 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 $\tilde{x}$ 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^Tx is unique. Second, what consequences does the singularity of (1.4) imply for the conjugate gradient method?

Kaasschieter [12] notes that a positive semidefinite system can be solved as long as the right hand side is consistent (i.e. as long it contains no component in the null space span$\{Z\}$). This is certainly the case for (1.4), since the same projection is applied to both sides of the equation. Furthermore, he notes (with reference to [21]) that since the null space never enters the iteration, the corresponding zero-eigenvalues do not affect the convergence. To this end we define the *effective condition number* of a positive semidefinite matrix $A^* \in \mathbb{R}^{n\times n}$ with corank m to be the ratio of its largest to smallest *nonzero* eigenvalues:

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

Example. To see that the condition number of PA may be better than that of A, consider the case in which Z is an invariant subspace of A. 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. $Y^TZ = 0$ 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.

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

This idea has been used in various ways by several authors. For nonsymmetric systems, approximate eigenvectors can be extracted from the Krylov subspace produced by GMRES. Morgan [17] 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 space is removed from the spectrum. For more discussion on deflation methods for nonsymmetric systems, see [13, 7, 5, 19, 4, 2]. Other authors have attempted to choose *a priori* a subspace to effectively represent the slowest modes. In [24] deflation is used to remove a few stubborn but *known* modes from the spectrum. Mansfield [14] shows how Schur complement-type domain decomposition methods can be seen as a series of deflations. Nicolaides [18] 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 [15] uses the same "subdomain deflation" in combination with damped Jacobi smoothing, obtaining a preconditioner which is similar to the two-grid method.

In this article we introduce new bounds for the condition numbers of deflated symmetric positive definite systems, used with and without classical preconditioning. For the case of a subdomain deflation such as that of Nicolaides (1987), these lemmas can provide direction in choosing a proper decomposition into subdomains and a proper choice of classical preconditioner. If grid refinement is done keeping the subdomain resolutions fixed, the condition number can be shown to be independent of the number of subdomains.

2. A condition number bound for deflation

Nicolaides [18] proves the following bound on the spectrum of PA:

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

where v is taken in $\text{span}\{Z\}^\perp$. 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 prove two supplemental lemmas. The first has to do with preservation of positive semidefiniteness under deflation.

Lemma 2.1 *Let C be positive semidefinite and P be a projection ($P^2 = P$), then if PC is symmetric, it is positive semidefinite.*

Proof. By hypothesis, $0 \leq x^T C x$ for all x. In particular, $0 \leq (P^T x)^T C (P^T x) = x^T P C P^T x$ so that $PCP^T = P^2 C = PC$ is positive semi-definite. □

The second lemma gives the norm of the projection matrix P.

Lemma 2.2 *For P defined as in (1.2), $\|P\|_2 = 1$.*

Proof. Define the spaces $V = \text{span}\{Z\}^\perp$ and $W = \text{span}\{AZ\}$ with dimensions $n - m$ and m, respectively. The direct sum of these spaces is $\mathbb{R}^n$. If this were not so, there would be a vector y in both spaces. Since $y \in W$, $y = Az$ for some nonzero $z \in \text{span}\{Z\}$. On the other hand, since $y \in V$, y must be perpendicular to z that is, $\langle z, Az \rangle = 0$, contradicting the positive definiteness of A.

Any vector in $\mathbb{R}^N$ can be written as the sum of components in W and V; i.e. $u = v + w$, $v \in V$, $w \in W$. Note that $Pw = w$ and $Pv = 0$. Now we have

$$\|P\|_2 = \max_{u \neq 0} \frac{\|Pu\|_2}{\|u\|_2} = \max_{u=v+w\neq 0} \frac{\|w\|_2}{\|v + w\|_2} = 1.$$

□

The following lemma provides a bound on the condition number of PA, and is our main result:

Lemma 2.3 *Let A be symmetric positive definite, P be defined by (1.2), and suppose there exists a splitting $A = A^* + C$ such that A^* and C are symmetric positive semidefinite with $\mathcal{N}(A^*) = \operatorname{span}\{Z\}$ the null space of A^*. Then*

$$\lambda_i(A^*) \leq \lambda_i(PA) \leq \lambda_i(A^*) + \lambda_{\max}(PC). \tag{2.1}$$

Moreover, the effective condition number of PA is bounded by

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

Proof. From (1.2) it is obvious that PA is symmetric. Since Z is in the null space of A^*, we have that $PA^* = A^*$ and is therefore also symmetric by hypothesis. Symmetry of $PC = PA - A^*$ follows immediately; and by assumption C is positive semidefinite, so we can apply Lemma 2.1 to arrive at $\lambda_{\min}(PC) \geq 0$, with equality holding in any case due to singularity of P. The bound (2.1) now follows from Theorem 8.1.5 of [8]:

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

Furthermore, $\lambda_n(PA) = \|PA\|_2 \leq \|P\|_2\|A\|_2 = \|A\|_2$, as a result of Lemma 2.2. This upper bound together with the lower bound in (2.1) proves (2.2). □

There is also a preconditioned version of the previous lemma.

Lemma 2.4 *Assume the conditions of Lemma 2.3 and let K be a symmetric positive definite preconditioner with Cholesky factorization $K = LL^T$. Then,*

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

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

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

Proof. Define $\hat{A} = L^{-1}AL^{-T}$, $\hat{A}^* = L^{-1}A^*L^{-T}$, $\hat{C} = L^{-1}CL^{-T}$ (all congruence transformations), $\hat{Z} = L^TZ$ 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, also that $\hat{Z}$ is in the null space of $\hat{A}^*$ so that $\hat{P}\hat{A}^* = \hat{A}^*$. Thus, Lemma 2.3 applies directly to the deflated system matrix $\hat{P}\hat{A}$. The conclusions follow immediately from the definitions of $\hat{A}$ and $\hat{A}^*$. □

Remark. Experience shows 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 greatest concern. It follows from Lemma 2.4 that one should choose the preconditioner $K = LL^T$ to be effective on the spectrum of A^* rather than on that of A. See Kaasschieter [12] for a discussion of the preconditioning of indefinite systems.

In the next section we consider applications of Lemmas 2.3 and 2.4 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 selection of a 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 $\mathcal{I} = \{i | x_i \in \Omega\}$ into m nonoverlapping subdomains Ω_j, $j = 1, \ldots, m$ with respective index sets $\mathcal{I}_j = \{i \in \mathcal{I} | x_i \in \Omega_j\}$. We assume that the Ω_j are simply connected regions covering Ω. Define Z by:

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

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 [18] and Mansfield [14, 15].

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 [11]. As a multigrid method, P can be seen as a coarse grid correction using a piecewise constant interpolation operator with very extreme coarsening.

Note that the matrix $A_c = Z^T A Z$, the projection of A onto the deflation subspace Z, has the same sparsity pattern as 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 become to solve A_c recursively. In a parallel implementation this would lead to additional idle time, as it does with multigrid.

3.1 Application to Stieltjes matrices

Using subdomain deflation, we can identify matrices A^* and C needed for application of the deflation Lemmas 2.3 and 2.4 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 $i \neq j$.
- $a_{ii} - \sum_{i \neq j} a_{ij} \geq 0$ 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 with A by

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

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 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.[2] It follows from Corollary 6.4.11 of [10] that the B_{jj} are again Stieltjes matrices.

[2] 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.

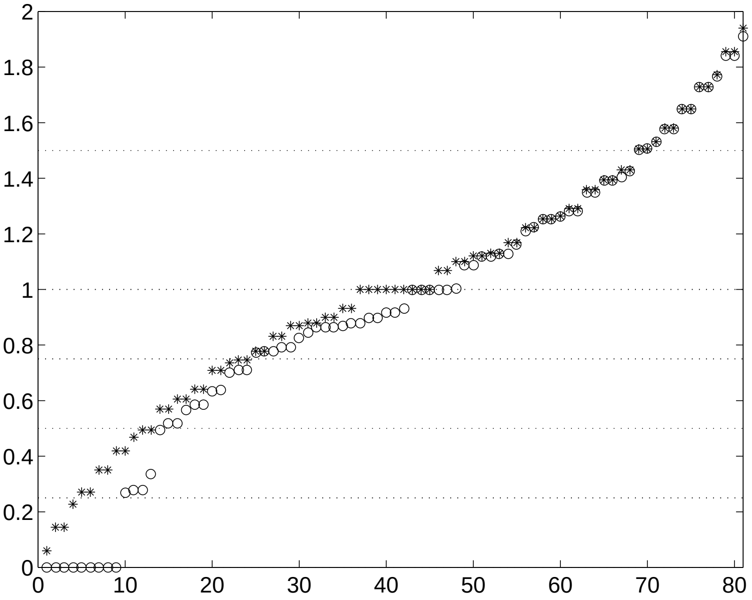

Figure 1: The eigenvalues of $A(\circ)$, $PA(*)$ and $A^*(\cdots)$.

Additionally, define $\Sigma(A) \equiv A\mathbf{1}$ to be the vector containing the row sums of A, where $\mathbf{1}$ denotes the vector of appropriate length containing all ones. Let the matrix A^* be defined by

$$A^* = B - \operatorname{diag}(\Sigma(B)). \tag{3.3}$$

Each block A^*_{jj} of A^* 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 one. It follows that the matrix Z defined by (3.1) is a basis for the null space of A^*.

Putting these ideas together we formulate:

Theorem 3.1 *If A is an irreducibly diagonally dominant M-matrix and A^* defined by (3.3) has only irreducible blocks, then the hypotheses of Lemma 2.3 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 3 × 3 blocks. The system matrix A is pre- and post-multiplied by the square root of its diagonal. Figure 1 shows the eigenvalues of A, PA and A^*. The extreme positive eigenvalues of these three matrices are:

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

Both the table and the figure support the conclusions of Lemma 2.3; namely, that the largest eigenvalue of A and the smallest nonzero eigenvalue of A^* bound the spectrum of PA. Note that the bounds are reasonably sharp.

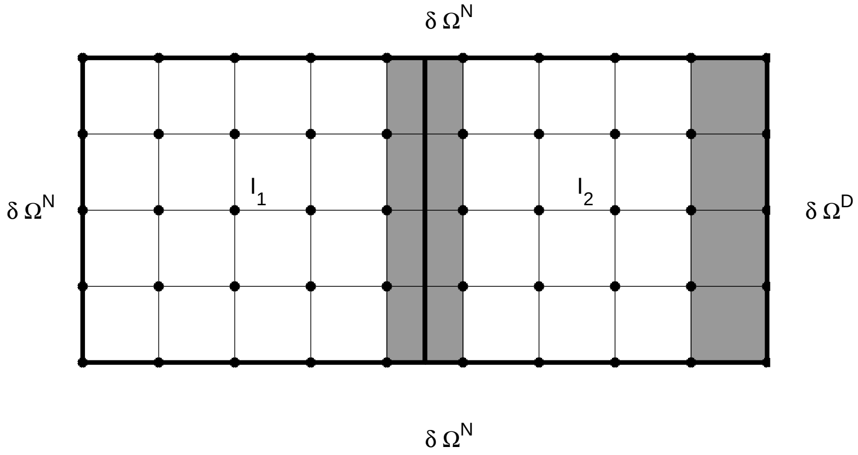

Figure 2: The domain Ω is decomposed into two subdomains (the shaded region is I_r)

Note that the diagonal blocks A^*_{jj} of the matrix A^* as defined by (3.3) can be interpreted as the discretizations of a set of m related Neumann problems on the corresponding subdomain grids. By Lemma 2.3, the effective condition number of the deflated matrix PA is determined by the smallest nonzero eigenvalue of A^*—in this case, the smallest nonzero eigenvalue over the set of related Neumann problems on the subdomain grids, i.e.

$$\lambda_{m+1}(PA) = \min_j \lambda_2(A^*_{jj}).$$

Lemma 2.3 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

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

and $\mathcal{I}_r = \mathcal{I} \backslash \cup_j \mathcal{I}_j$. Figure 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 is prescribed somewhere on the boundary. The matrix A^* needed for Lemma 2.3 is defined by

$$A^* = \sum_{i \in \mathcal{I} \setminus \mathcal{I}_r} A_{e_i}.$$

Note that A^* is block diagonal and the blocks A^*_{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 $\lambda_1(A^*_{jj}) = 0$ and that Z is in the null space of A^*. Clearly A^* is positive semidefinite, as is

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

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

$$\Delta u = f, \quad \partial u/\partial n = 0, \; u \in \partial\Omega.$$

This equation is discretized with standard central finite differences 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

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

The eigenvalues of the discretization matrix[3] 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 get

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

[3] We are grateful to Jos van Kan for supplying this formula for the eigenvalues of the discrete Laplacian.

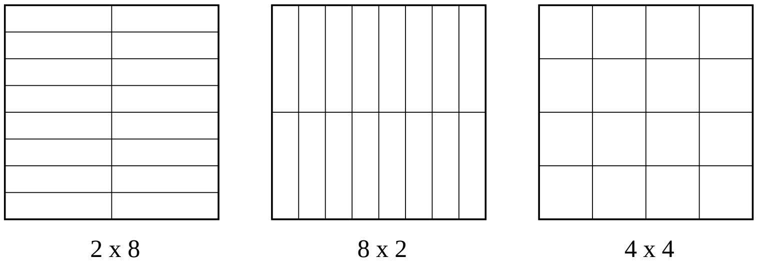

Figure 3: Three decompositions of the unit square into 16 subdomains.

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

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

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:

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

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 3:

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

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

The decomposition problem discussed above assumes that the number of domains is given. 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 are 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 A^* 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 equal smaller subdomains, the condition number of A^* is improved by a factor 4, roughly speaking. On the other hand, the dimension of the coarse grid matrix A_c will be increased by a factor 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 Strongly varying coefficients

When a problem has a large jump in coefficients at some location, it may be helpful to apply 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.

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

$$\frac{\partial}{\partial x}\alpha(x)\frac{\partial u}{\partial x} = f(x), \quad x \in [0, 8.5], \quad \frac{dy}{dx}(0) = 0,\ y(8.5) = 1,$$

and a jump discontinuity in the coefficient

$$\alpha(x) = \begin{cases} 1, & x \le 3.5, \\ \epsilon, & x > 3.5, \end{cases}$$

for some $\epsilon > 0$. The interval is discretized on a grid $x_i = i + 0.5$, $i = 0, \dots, 8$, where the Neumann boundary condition is discretized using a fictitious grid point $x_{-1} = -0.5$, and the prescribed value at $x = 8.5$ is eliminated from the system. The resulting matrix A is symmetric and positive definite, so that its diagonal is positive and we may define the preconditioner $D^{1/2} = \text{diag}(A)^{1/2}$.

To define A^* we decompose the problem into two subdomain problems

$$\frac{d^2y}{dx^2} = f(x), \quad x \in [0, 4], \quad \frac{dy}{dx}(0) = \frac{dy}{dx}(4) = 0$$

and

$$\epsilon\frac{d^2y}{dx^2} = f(x), \quad x \in [4, 8], \quad \frac{dy}{dx}(4) = \frac{dy}{dx}(8) = 0,$$

each discretized on 4 grid points, again using fictive grid points to discretize the boundary conditions. The subdomain deflation space Z is defined by (3.1) with $\Omega_1 = [0, 4]$ and $\Omega_2 = [4, 8]$.

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.) are shown in Figure 4 for $\epsilon = 1$ and $\epsilon = 0.01$ with the eigenvalues of $D^{-1}A^*$ appearing as dotted lines. Note that the smallest positive eigenvalue of $D^{-1}A^*$ bounds from below the smallest positive eigenvalue of $D^{-1}PA$, as predicted by Lemma 2.4.

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	$1.9 \cdot 10^{-2}$	$1 \cdot 10^{2}$	$2.9 \cdot 10^{-1}$	6.5
10^{-2}	$3.3 \cdot 10^{-4}$	$6 \cdot 10^{-3}$	$2.9 \cdot 10^{-1}$	6.8
10^{-4}	$3.3 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	$2.9 \cdot 10^{-1}$	6.8

Due to diagonal preconditioning, the smallest eigenvalue of $D^{-1}A^*$ is independent of ϵ. As predicted by Lemma 2.4, 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 $D^{-1}Ax = D^{-1}b$.

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.

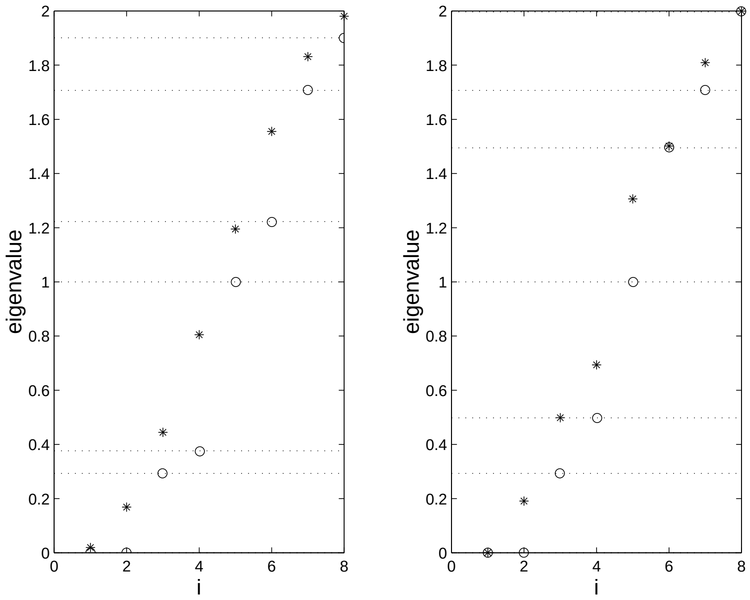

Figure 4: Eigenvalues of $D^{-1}A$ ($*$) and $D^{-1}PA$ ($\circ$) for $\epsilon = 1$ (left) and $\epsilon = 0.01$ (right). The spectrum of $D^{-1}A^*$ is indicated by the dotted lines.

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 [24]. 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 $Ax = b$ using deflation, note that x can be written as

$$x = (I - Q)x + Qx$$

and that $(I - Q)x = Z(Y^TAZ)^{-1}Y^TAx = Z(Y^TAZ)^{-1}Y^Tb$ can be computed immediately (cf. (1.3)). Furthermore Qx can be obtained by solving the deflated system

$$PA\tilde{x} = Pb \tag{5.1}$$

for $\tilde{x}$ (cf. (1.4)) and pre-multiplying 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: solve $\tilde{x}$ from

$$K^{-1}PA\tilde{x} = K^{-1}Pb, \tag{5.2}$$

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

and form $Q\tilde{x}$, or solve $\tilde{y}$ from

$$PAK^{-1}\tilde{y} = Pb, \tag{5.3}$$

and form $QK^{-1}\tilde{y}$. Both systems can be solved by one's favorite Krylov subspace solver, such as: GMRES [20], GCR [6, 23], Bi-CGSTAB [22] etc.

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

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 $X^{-1}AX = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$. For the first choice, which is related to Hotelling deflation (see [25] p. 585), we have the following result.

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

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

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

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

□

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

Lemma 5.2 *For $Y = Z$ one has:*

(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 $Z = [x_1 \dots x_m]$, the eigenvalues of PA are $\{0, \lambda_{m+1}, ..., \lambda_n\}$, where m is the multiplicity of eigenvalue 0.*

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 $i = 1, \dots, m$. For the other eigenvalues we define the vectors

$$v_i = x_i - AZA_c^{-1}Z^Tx_i, \quad i = m+1, \dots, n.$$

These vectors are nonzero, because $x_1, ..., x_n$ form an independent set. Multiplication of v_i by PA yields:

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

which proves the lemma. □

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 $z_1, \dots, z_m$ 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).

Lemmas 2.3 and 2.4 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 $Ax = b$ 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 (i, j) of Z^TAZ can be done locally on processor i by summing the coefficients corresponding to block A_{ij} of (5.4).

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

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

Assuming $(Z^TAZ)^{-1}$ 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 compute and store on processor i the vectors

$$\{c_j = A_{ij}\mathbf{1};\ j|A_{ij} \neq 0\}. \tag{5.5}$$

Then, local computation of $AZ\tilde{e}$ for a given (m-dimensional) vector $\tilde{e}$ consists of scaling the nonzero c_j by the corresponding $\tilde{e}_j$ and summing them up. The vectors c_j, $j \neq i$ have nonzero elements only for local unknowns with connections to the virtual cells. Furthermore, for many applications the elements of c_i corresponding to grid points interior to a subdomain will be zero.

In parallel, we first compute and store the (nonzero parts of the) c_j and $(Z^TAZ)^{-1}$ (factored) on each processor. Then to compute PAv we first perform the matrix-vector multiplication $w = Av$, requiring nearest neighbor communications. Next we compute the local contribution to the restriction $\tilde{w} = Z^Tw$ and distribute this to all processes. With this done, we can solve $\tilde{e} = (Z^TAZ)^{-1}\tilde{w}$ and compute $AZ\tilde{e}$ locally.

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

The computational and communication costs plus storage requirements of subdomain deflation are summarized in the following table, assuming a five-point discretization stencil on an $N_x \times N_y$ grid with $M_x \times M_y$ decomposition into $n_x \times n_y$ blocks ($n_x = N_x/M_x$, $n_y = N_y/M_y$). The abbreviation *GaBr (m)* refers to a gather-broadcast operation in which a set of m distributed floating point numbers are gathered from the participating processors and then whole set 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, block-wise) incomplete factorization preconditioner with zero fill-in, ILU(0).

	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: $s = Z^TAv$	N_xN_y		n_xn_y		0
Backsolve: $A_c\tilde{e} = s$	$4M_x^2M_y$		$4M_x^2M_y$		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

In conducting numerical experiments, we are interested in the following issues:

- verification of the theoretical results of this article
- the properties of subdomain deflation for nonsymmetric systems
- the parallel performance of the method

To this end we consider the Poisson equation with nonsymmetric preconditioning. The model problem[5] reads:

$$u_{xx} + u_{yy} = 1.$$

We discretize this using a cell-centered finite volume method on a rectangular grid.

The test cases are:

I. a uniform grid on $\Omega = [0, 1] \times [0, 1]$, $u = 0$ on $\partial\Omega$; and

II. a stretched grid on $\Omega = [0, 3] \times [0, 1]$, $\Delta x = 3/N_x$ and $\Delta y = 1/N_y$, $u = 0$ on $\partial\Omega$.

We solve the resulting discrete (symmetric) system using GMRES with a restart of 20 and subdomain deflation:

$$PAx = Pb.$$

Due to boundary conditions, a nonsymmetric matrix may be obtained by multiplying $Ax = b$ on the left by the inverse of $D = \text{diag}(A)$. We assume that this has been done prior to constructing the deflation operators and preconditioner, and do not write D explicitly. The preconditioned nonsymmetric system to be solved is therefore:

$$K^{-1}PAx = K^{-1}Pb.$$

[5] We checked that replacing the right hand side by a random vector gives the same qualitative behavior reported here.

Convergence is declared when, in the Jth iteration, $\|r_J\| \leq tol \cdot \|r_0\|$, for $tol = 10^{-6}$.

The preconditioner used on the blocks is the relaxed incomplete LU (RILU) 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 only occur on the diagonal) and is reasonably effective. Certainly, more advanced preconditioners could be designed based on the blocks of A^*.

6.1 Convergence results

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

6.1.1 Near grid independence In this section we illustrate the extent to which subdomain deflation can provide nearly grid-independent convergence. The symmetric discretization matrix for Problem I is considered, without diagonal scaling and preconditioning. Keeping the resolution of each subdomain fixed, the number of subdomains is increased. In so doing, the blocks of A^* 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 the domain is decomposed into $M_x \times M_y$ subdomains of equal size, each containing a $n_x \times n_y$ grid.

Consider Problem I with $M_x = M_y$ and $n_x = n_y$. Table 1 gives the number of GMRES iterations required to solve Problem I as the grid is refined keeping the subdomain resolution n_x fixed.

Table 1: Iterations Required for Problem I with Grid Refinement

$m = M_x^2$	$n_x = 5$	$n_x = 10$	$n_x = 20$	$n_x = 50$	$n_x = 100$
1	4	11	44	175	596
4	11	44	138	596	2119
9	18	51	152	642	2336
16	27	56	155	644	2408
25	26	54	152	637	2442
36	27	54	147	627	2458
64	26	52	139	602	2474

It is apparent 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.

6.1.2 Stretched grid We consider Problem II with $N_x = 36$ 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 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 2 gives the number of iterations required for convergence for 5 different decompositions into 12 equally sized subdomains. The 6 × 2 decomposition with $\mathcal{Q}_d = 1$ gives the minimum number of iterations, in keeping with the discussion.

6.2 Parallel performance

For the results in this section, the RILU preconditioner is constructed on the blocks of $B(A)$ rather than A^*. To measure the parallel performance we consider test Problem I. The domain Ω is decomposed in to $M_x \times M_y$ subdomains of size $1/M_x \times 1/M_y$ upon each of which an $n_x \times n_y$ uniform grid is constructed (for the experiments presented here, $M_x = M_y$ and $n_x = n_y$). An initial guess $u^{(0)} = 0$ is used.

Table 2: Iterations required for Problem II for different decompositions

$M_x \times M_y$	$n_x \times n_y$	$\mathcal{Q}_d$	J
2×6	18×12	9	369
3×4	12×18	4	245
4×3	9×24	9/4	247
6×2	6×36	1	189
12×1	3×72	1/4	191

Our implementation does not take advantage of symmetry or the fact that some of the row sums may 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 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 $M_x = 1, 2, 3, 4, 5, 6, 8$ and $n_x = N_x/M_x$. The results are given in Tables 3 and 4 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 3 the parallel efficiency decreases from 44 percent on four processors to only 16 percent on 64 processors, whereas in Table 4 greater than 100 percent efficiency (compared to the single subdomain, nondeflated case) was attained for all parallel runs. This behavior is not yet understood, but the following factors may contribute:

- 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.
- There may be cache effects in play.

Table 3: Speedup for Problem I on a 120 × 120 grid.

p	J	t_{const}	t_{iter}	s	eff
1	40	$8.5 \cdot 10^{-3}$	2.72	–	–
4	95	$1.2 \cdot 10^{-2}$	1.56	1.8	0.44
9	124	$6.4 \cdot 10^{-3}$	1.04	2.6	0.29
16	120	$4.3 \cdot 10^{-3}$	0.60	4.6	0.29
25	107	$6.7 \cdot 10^{-3}$	0.50	5.5	0.22
36	96	$7.9 \cdot 10^{-3}$	0.41	6.6	0.18
64	76	$1.1 \cdot 10^{-2}$	0.27	9.9	0.16

Table 4: Speedup for Problem I on a 480 × 480 grid.

p	J	t_{const}	t_{iter}	s	*eff*
1	484	$1.4 \cdot 10^{-1}$	638.3	–	–
4	321	$1.3 \cdot 10^{-1}$	95.3	6.7	1.68
9	351	$6.4 \cdot 10^{-2}$	54.6	11.7	1.30
16	378	$3.4 \cdot 10^{-2}$	27.4	23.3	1.45
25	316	$2.5 \cdot 10^{-2}$	17.6	36.3	1.45
36	409	$2.2 \cdot 10^{-2}$	14.1	45.4	1.26
64	317	$1.4 \cdot 10^{-2}$	5.9	108.3	1.69

6.2.2 Scaled performance for fixed subdomain size. Table 5 gives the computation times obtained for fixed subdomain sizes $n_x = 5, 10, 20, 50, 100, 200$ as the number of processors is increased, with and without subdomain deflation. It is clear that the effect of deflation is to make the parallel computation time less sensitive to the number of processors. The $*$ indicates the method did not converge in 3000 iterations for the case in question.

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. At the time of printing, the 64-processor cases with $n_x = 100$ and $n_x = 200$ (indicated by †) were not available.

Table 5: Scaled performance for Problem I with fixed subdomain size n.

		$n_x = 5$	$n_x = 10$	$n_x = 20$	$n_x = 50$	$n_x = 100$	$n_x = 200$
$p = 1$	no deflation	$6 \cdot 10^{-4}$	$2 \cdot 10^{-3}$	$9 \cdot 10^{-3}$	0.17	1.19	12.16
$p = 4$	no deflation	$8 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	$7 \cdot 10^{-2}$	0.87	9.38	74.49
	deflation	$8 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	$7 \cdot 10^{-2}$	0.90	8.15	56.08
$p = 9$	no deflation	$3 \cdot 10^{-2}$	$9 \cdot 10^{-2}$	0.24	2.99	21.71	174.85
	deflation	$3 \cdot 10^{-2}$	$7 \cdot 10^{-2}$	0.21	2.10	14.44	110.35
$p = 16$	no deflation	$4 \cdot 10^{-2}$	0.11	0.42	4.66	40.90	373.80
	deflation	$3 \cdot 10^{-2}$	$8 \cdot 10^{-2}$	0.24	2.46	14.86	125.13
$p = 25$	no deflation	$9 \cdot 10^{-2}$	0.21	0.91	7.99	$*$	$*$
	deflation	$6 \cdot 10^{-2}$	0.13	0.34	3.04	15.78	129.31
$p = 36$	no deflation	0.12	0.34	1.19	14.13	$*$	$*$
	deflation	$7 \cdot 10^{-2}$	0.15	0.41	3.20	19.32	158.20
$p = 64$	no deflation	0.12	0.48	1.43	$*$	$*$	$*$
	deflation	$7 \cdot 10^{-2}$	0.16	0.44	3.17	†	†

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 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 number of subdomains, and that the method can be very efficiently implemented on distributed memory parallel computer.

Acknowledgements

We thank HPαC for the use of the Cray T3E.

References

1. O. Axelsson and G. Linskog. On the eigenvalue distribution of a class of preconditioning methods. *Numerische Mathematik*, 48:479–498, 1986.
2. J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel. Adaptively preconditioned GMRES algorithms. *SIAM J. Sci. Comput.*, 20(1):243–269, 1999.
3. A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences.* Classics in applied mathematics. SIAM, Philadelphia, 1994.
4. K. Burrage, J. Erhel, B. Pohl, and A. Williams. A deflation technique for linear systems of equations. *SIAM J. Sci. Comput.*, 19:1245–1260, 1998.
5. A. Chapman and Y. Saad. Deflated and augmented Krylov subspace techniques. *Numer. Linear Algebra Appl.*, 4(1):43–66, 1997.
6. S. C. Eisenstat, H. C. Elman, and M. H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Num. Anal.*, 20:345–357, 1983.
7. J. Erhel, K. Burrage, and B. Pohl. Restarted GMRES preconditioned by deflation. *J. Comput. Appl. Math.*, 69(2):303–318, 1996.
8. G. H. Golub and C. F. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.
9. I. Gustafsson. A class of first order factorization methods. *BIT*, 18:142–156, 1978.
10. W. Hackbusch. *Iterative solution of large sparse systems of equations.* Springer-Verlag, New York, 1993.
11. C. B. Jenssen and P. Å. Weinerfelt. Coarse grid correction scheme for implicit multiblock Euler calculations. *AIAA Journal*, 33(10):1816–1821, 1995.
12. E .F. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *J. of Comput. Appl. Math.*, 24:265–275, 1988.
13. S. A. Kharchenko and A. Yu. Yeremin. Eigenvalue translation based preconditioners for the GMRES(k) method. *Numer. Linear Algebra Appl.*, 2(1):51–77, 1995.
14. L. Mansfield. On the conjugate gradient solution of the Schur complement system obtained from domain decomposition. *SIAM J. Numer. Anal.*, 27(6):1612–1620, 1990.
15. L. Mansfield. Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers. *SIAM J. Sci. Stat. Comput.*, 12(6):1314–1323, 1991.
16. 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. *Mathematics of Computation*, 31:148–162, 1977.
17. R. B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, 16(4):1154–1171, 1995.
18. R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2):355–365, 1987.
19. Y. Saad. Analysis of augmented Krylov subspace methods. *SIAM J. Math. Anal. Appl.*, 18(2):435–449, 1997.
20. Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.
21. A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numerische Mathematik*, 48:543–560, 1986.

22. H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13:631–644, 1992.

23. H. A. van der Vorst and C. Vuik. GMRESR: a family of nested GMRES methods. *Num. Lin. Alg. Appl.*, 1:369–386, 1994.

24. 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.*, 152:1–19, 1999.

25. J. H. Wilkinson. *The algebraic eigenvalue problem.* Clarendon Press, Oxford, 1992.