SIAM J. SCI. COMPUT.
Vol. 30, No. 3, pp. 1572–1595

MULTILEVEL PROJECTION-BASED NESTED KRYLOV ITERATION FOR BOUNDARY VALUE PROBLEMS*

YOGI A. ERLANGGA† AND REINHARD NABBEN†

Abstract. We propose a multilevel projection-based method for acceleration of Krylov subspace methods. The projection is constructed in a similar way as in deflation but shifts small eigenvalues to the largest one instead of to zero. In contrast with deflation, however, the convergence rate of a Krylov method combined with this new projection method is insensitive to the inaccurate solve of the Galerkin matrix, which with some particular choice of deflation subspaces is closely related to the coarse-grid solve in multigrid or domain decomposition methods. Such an insensitivity allows the use of inner iterations to solve the Galerkin problem. An application of a Krylov subspace method to the associated Galerkin system with the projection preconditioner leads to a multilevel, nested Krylov iteration. In this multilevel projection Krylov subspace method, information about small eigenvalues to be projected is contained implicitly in the Galerkin system associated with the matrix of the linear system to be solved. These small eigenvalues, from a Krylov method point of view, are responsible for slow convergence. In terms of projection methods, this is conceptually similar to multigrid but different in the sense that in multigrid the projection is done by the smoother. Furthermore, with the only condition being that the deflation matrices are full rank, we have in principle more freedom in choosing the deflation subspace. Intergrid transfer operators used in multigrid are some of the possible candidates. We present numerical results from solving the Poisson equation and the convection-diffusion equation, both in two dimensions. The latter represents the case where the related matrix of coefficients is nonsymmetric. By using a simple piecewise constant interpolation as the basis for constructing the deflation subspace, we obtain the following results: (i) h-independent convergence for the Poisson equation and (ii) almost independent of h and the Péclet number for the convection-diffusion equation.

Key words. Krylov subspace, GMRES, deflation, multilevel projection, Poisson equation, convection-diffusion equation

AMS subject classifications. 65F10, 65F50, 65N22, 65N55

DOI. 10.1137/070684550

1. Introduction.

It is well known that the convergence of Krylov subspace methods applied to the linear system

$$Au = b, \qquad A \in \mathbb{R}^{n\times n}, \tag{1.1}$$

depends to some extent on the spectrum of A. Suppose that A is symmetric positive definite (SPD). The convergence of conjugate gradient (CG) methods then depends in particular on the condition number of A, denoted by $\kappa(A)$ [19], which in this case is equal to the ratio of the largest eigenvalue to the smallest one. If A is obtained from a discretization of PDEs with elliptic, self-adjoint operators, the convergence is then typically characterized by the smallest eigenvalues, which can be of order $O(h^2)$. So, as the grid is refined, the convergence usually deteriorates.

For general matrices, no *practical* convergence bounds similar to the SPD case exist so far. Nevertheless, it is commonly accepted that a Krylov subspace method will converge faster if A has a more clustered spectrum. Similar to the SPD case, the convergence, however, may be hampered if some eigenvalues are very close to zero.

*Received by the editors March 7, 2007; accepted for publication (in revised form) September 20, 2007; published electronically April 18, 2008. The research was supported by the Deutsche Forschungsgemeinschaft (DFG) project NA248/2-2.

http://www.siam.org/journals/sisc/30-3/68455.html

†Institut für Mathematik, TU Berlin, MA 3-3, Strasse des 17. Juni 136, D-10623 Berlin, Germany (erlangga@math.tu-berlin.de, nabben@math.tu-berlin.de).

In order to improve the convergence, preconditioners are usually incorporated. A usual way of understanding the role of a preconditioner in Krylov subspace methods is again by looking at the condition number of the preconditioned system (in the case of SPD systems). An efficient preconditioning matrix M is usually chosen such that the condition number of $M^{-1}A$, denoted by $\kappa(M^{-1}A)$, is close to one. Hence, it is very natural to seek for M such that $M^{-1}A \approx I$, where I is the identity. Incomplete decompositions (e.g., incomplete LU and incomplete Cholesky) and the approximation inverse, among others, belong to this class of preconditioners. This approach, however, does not exploit detail information about the spectrum of A.

Since small eigenvalues are responsible for slow convergence, the convergence of a Krylov subspace method can be accelerated if by any means components in the residuals which correspond to the small eigenvalues can be removed during the iterations. One way to achieve this is by deflating a number of the smallest eigenvalues to zero. Nicolaides [14] shows that, by adding some vectors related to the smallest eigenvalues, the convergence of the CG may be improved. For GMRES, Morgan [10] shows that, by augmenting the Krylov subspace by some eigenvectors related to small eigenvalues, these eigenvectors no longer have components in the residuals, and the convergence bound of GMRES can be made smaller; thus, a faster convergence may be expected. See a somewhat unified discussion on this subject by Eiermann, Ernst, and Schneider in [1].

A similar approach is proposed in [6], where a projection matrix resembling deflation of some small eigenvalues is used as a preconditioner. Suppose that r smallest eigenvalues are to be deflated to zero. Define the projection

$$P_D = I - AZE^{-1}Y^T, \qquad E = Y^T AZ, \tag{1.2}$$

where $Z, Y \in \mathbb{R}^{n\times r}$ and have rank r. Here Z is called the *deflation subspace*. It can be proved [6, 12] that if A is SPD and $Z = Y$ any rectangular matrices with full rank, then the spectrum of $P_D A$ contains r zero eigenvalues. Furthermore, it has also been shown in [11] that, with larger r, the effective condition number reduces as well. Hence, if one uses a larger deflation subspace, the convergence can be improved considerably. A larger deflation subspace, however, raises a negative implication. Since in the preconditioning step one needs to compute E^{-1}, a larger deflation subspace can make this computation more difficult and expensive. Related to this, it has been shown in [12] that P_D is sensitive to inaccurate inversion of E. This means that E must be inverted exactly (by a direct method for sparse systems) or iteratively up to machine precision to obtain fast convergence.

We note here that the form $E = Y^T AZ$ in (1.2) is similar to the coarse-grid matrix used in multigrid and domain decomposition methods. If $Z \neq Y$, it is called *Petrov–Galerkin* coarse-grid approximation. In our discussion, we do not particularly call this term "the coarse-grid matrix." Instead, we generally refer the product $E = Y^T AZ$ to the "Galerkin matrix" associated with the matrix A and deflation subspaces Z and Y. The linear system related to this Galerkin matrix is then called the "Galerkin system."

Of importance is that, compared to some other forms of projection-like preconditioners, e.g., the abstract balancing preconditioner [9] or the additive coarse-grid correction preconditioner [15], under certain conditions, the deflation preconditioner leads to better conditioned systems and faster convergence [11, 12, 13, 3, 21]. Furthermore, with a straightforward implementation, a CG with deflation, for example, requires fewer matrix-vector multiplications than a CG with the abstract balancing preconditioner. The convergence rate of CG with the abstract balancing preconditioner is, however, insensitive to an inexact solve of the Galerkin system. It is actually

the sensitivity of deflation to an inexact solve of the Galerkin system which hinders the use of a large deflation subspace in order to enhance the convergence. In this case, work spent for this solve will undermine gains in the number of iterations.

In this paper, we propose another projection-like method which allows the exploitation of a large deflation subspace. The proposed method is found to be stable with respect to an inaccurate solve of the Galerkin system. With this advantage, an iterative procedure based on a Krylov method can be designed to accomplish this solve, which can then be considered as approximately solving a coarse-grid problem. The convergence rate for solving the Galerkin system is accelerated by employing the same projection-type preconditioner on the Galerkin matrix E, leading to a multilevel, nested iterations. We call the resultant method the "multilevel projection Krylov method."

Assuming that the inverse of the Galerkin matrix E is available, GMRES [17] combined with our stable projection preconditioner requires only one additional matrix-vector multiplication. This is the same as the unstable deflation preconditioner and less than the stable abstract balancing preconditioner.

This paper is organized as follows. In section 2, we discuss deflation from an eigenvalue computation point of view. We show that the deflation preconditioner (1.2) can be derived from the deflation process in eigenvalue computations. Based on this view, we propose our stable *abstract* projection-type preconditioner. In section 3, spectral analysis is given for the projected linear systems. We establish that, for arbitrary full rank deflation matrices Z and Y, the deflation and new projection preconditioner lead to projected linear systems of similar spectra. Sections 4 and 5 discuss implementation aspects of the *abstract* projection preconditioner. Particularly in section 5, we develop our multilevel projection algorithm and give a brief discussion on its relation with multigrid. One of the possible choices of the deflation subspace is discussed in section 6, which is based on agglomeration. Computational results are presented in section 7 for two model problems: the Poisson equation and the convection-diffusion equation. Section 8 draws some conclusions.

2. Wielandt deflation and deflation preconditioner. For the purpose of analysis, we consider more general notations for the linear system (1.1), namely,

$$\hat{A}\hat{u} = \hat{b}, \tag{2.1}$$

where $\hat{A}$ is nonsingular and in general nonsymmetric. $\hat{A}$ can be understood as the preconditioned matrix. For example, for left preconditioning, $\hat{A} = M^{-1}A$, and therefore $\hat{u} = u$, and $\hat{b} = M^{-1}b$. $\hat{A}$ can also represent a split preconditioned matrix, i.e., $\hat{A} = M_1^{-1}AM_2^{-1}$, with $\hat{u} = M_2u$ and $\hat{b} = M_1^{-1}b$.

We assume that the eigenvalues of $\hat{A}$, denoted by $\lambda(\hat{A})$ or λ if it is clear from the context, are all positive real numbers. Denote the spectrum of $\hat{A}$ by $\sigma(\hat{A})$. Then $\sigma(\hat{A}) = \{\lambda_1, \ldots, \lambda_n\}$ and $\lambda_i \leq \lambda_{i+1}$, $\forall i \in \mathbb{N}$. Hence, we order the eigenvalues of $\hat{A}$ increasingly. This spectral assumption is satisfied by symmetric positive definite matrices. For nonsymmetric matrices, this assumption does not generally hold. Unless the symmetric part is sufficiently dominating, the eigenvalues can be complex. For the convection-diffusion equation, this assumption is satisfied if the diffusion part is sufficiently dominating.

We start our discussion by looking at the action of the deflation preconditioner (1.2) on $\hat{A}$. But in doing so, we prefer not to follow the augmented Krylov subspace point of view. Instead, we will formulate the deflation preconditioner alternatively from the deflation process in computing a few largest or smallest eigenvalues in eigenvalue

computations. Note that, due to the use of new notations, (1.2) is now rewritten as $P_{\hat{D}} = I - \hat{A}Z\hat{E}^{-1}Y^T$, where $\hat{E} = Y^T\hat{A}Z$.

Consider the Power method [5, 27, 18] applied to $\hat{A}^{-1}$. The method iteratively computes an eigenvalue of $\hat{A}^{-1}$, which in this case is equal to the smallest eigenvalue (in magnitude) of $\hat{A}$, i.e., λ_1, and the corresponding eigenvector z_1. The second smallest eigenvalue can also be computed by using the Power method but after *deflating* λ_1 to zero. Define the deflation process as follows:

$$\hat{A}_1 = \hat{A} - \lambda_1 z_1 y^T, \qquad y^T z_1 = 1, \tag{2.2}$$

with y an arbitrary vector yet to be defined. This is called the *Wielandt* deflation [5, 27, 18]. Application of the Power method on $\hat{A}_1^{-1}$ results in the eigenpair (λ_2, z_2).

In eigenvalue computations, shifting the first, already-computed, small eigenvalues to zero, however, is not the only way to ensure computing the next smallest eigenvalue, if we are concerned only with a few, say, $r \ll n$, smallest eigenvalues. For example, we define the *generalized* deflation process as follows:

$$\hat{A}_{1,\gamma_1} = \hat{A} - \gamma_1 z_1 y^T, \qquad y^T z_1 = 1, \quad \gamma \in \mathbb{R}. \tag{2.3}$$

Regarding this generalized deflation, we have the following theorem.

THEOREM 2.1 (generalized Wielandt). *Let $\hat{A}_{1,\gamma_1}$ be defined as in* (2.3). *The spectrum of $\hat{A}_{1,\gamma}$ is then given by*

$$\sigma(\hat{A}_{1,\gamma_1}) = \{\lambda_1 - \gamma_1, \lambda_2, \ldots, \lambda_n\}. \tag{2.4}$$

Proof. The proof follows, e.g., [5, 27, 18]. For $i \neq 1$, the left eigenvector y_i of $\hat{A}_{1,\gamma_1}$ satisfies the relation $y_i^T\hat{A}_{1,\gamma_1} = y_i^T(\hat{A} - \gamma_1 z_1 y^T) = \tilde{\lambda}_i y_i^T$, or $\hat{A}_{1,\gamma_1}^T y_i = (\hat{A}^T - \gamma_1 y z_1^T)y_i = \hat{A}^T y_i - \gamma_1 y z_1^T y_i = \lambda_i y_i$, due to the orthogonality of z_1 and y_i. Since the eigenvalues of a matrix and its transpose are the same, the above statement is proved for $i \neq 1$. For $i = 1$, $\hat{A}_{1,\gamma_1} z_1 = (\hat{A} - \gamma_1 z_1 y^T)z_1 = \hat{A}z_1 - \gamma_1 z_1 y^T z_1 = (\lambda_1 - \gamma_1)z_1$. ∎

It is clear from the above theorem that by choosing $\gamma_1 = \lambda_1$ we obtain a spectrum, which is related to the Wielandt deflation (2.2). But, if we are concerned only with λ_2, this value can also be captured from $\hat{A}_{1,\gamma_1}^{-1}$ by the Power method if $\lambda_1 - \gamma_1 \geq \lambda_2$. The shift γ_1 cannot, however, be chosen arbitrarily, because it is possible that $0 < \lambda_1 - \gamma_1 < \lambda_2$. In this case, the Power method will return $\lambda_1 - \gamma_1$ as the smallest eigenvalue of $\hat{A}_{1,\gamma_1}$ and not λ_2. The problem, however, is that λ_2 is the value that we are going to compute from $\hat{A}_{1,\gamma_1}$; hence, λ_2 is not known yet. Since $\lambda_1, \lambda_2 \leq \lambda_n$, one alternative for the shift without having to know any information about λ_2 is by choosing $\gamma_1 = \lambda_1 - \lambda_n$. In this case, by Theorem 2.1 we actually shift λ_1 to λ_n, and the application of the Power method to $\hat{A}_{1,\gamma_1}^{-1}$ gives λ_2. We summarize these choices of shifts in the following corollary.

COROLLARY 2.2. *Let $\hat{A}_{1,\gamma_1}$ be defined as in* (2.3). *Then,*

(i) *for $\gamma_1 = \lambda_1$, $\sigma(\hat{A}_{1,\gamma_1}) = \{0, \lambda_2, \ldots, \lambda_n\}$ (Wielandt deflation), and*

(ii) *for $\gamma_1 = \lambda_1 - \lambda_n$, $\sigma(\hat{A}_{1,\gamma_1}) = \{\lambda_n, \lambda_2, \ldots, \lambda_n\}$.*

We now suppose that the r smallest eigenvalues are already known. The $(r+1)$th eigenvalue can next be computed by the Power method by first applying simultaneous deflation of the r smallest eigenvalues to $\hat{A}$. Denote by $Z = [z_1 \ldots z_r]$ the eigenvector matrix corresponding to the first r eigenvalues, and define

$$\hat{A}_r = \hat{A} - Z\Gamma_r Y^T, \qquad Y^T Z = I, \tag{2.5}$$

where $\Gamma_r = \mathrm{diag}(\gamma_1, \ldots, \gamma_r)$ and $Y = [y_1 \ldots y_r]$. Z is often called the deflation subspace. The following theorem is related to deflation (2.5).

THEOREM 2.3. *Let $\hat{A}_r$ be defined as in* (2.5)*, with $\Gamma_r = \mathrm{diag}(\gamma_1, \ldots, \gamma_r)$ and $Y^T Z = I$. Then, for $i = 1, \ldots, r$;*

(i) *$\gamma_i = \lambda_i$, the spectrum of $\hat{A}_r$ is $\sigma(\hat{A}_r) = \{0, \ldots, 0, \lambda_{r+1}, \ldots, \lambda_n\}$;*

(ii) *$\gamma_i = \lambda_i - \lambda_n$, the spectrum of $\hat{A}_r$ is $\sigma(\hat{A}_r) = \{\lambda_n, \ldots, \lambda_n, \lambda_{r+1}, \ldots, \lambda_n\}$.*

Proof. We consider the case with $\Gamma_r = \mathrm{diag}(\gamma_1, \ldots, \gamma_r)$. From (2.5), for $i = 1, \ldots, r$, we have

$$\begin{aligned}\hat{A}_r z_i &= (\hat{A} - Z\Gamma_r Y^T) z_i = \hat{A} z_i - [z_1 \ldots z_r]\mathrm{diag}(\gamma_1, \ldots, \gamma_r)[y_1 \ldots y_r]^T z_i \\ &= \lambda_i z_i - \gamma_i z_i = (\lambda_i - \gamma_i),\end{aligned}$$

because of the orthogonality condition in (2.5). For $i = r+1, \ldots, n$, the orthogonality condition leads to the relation $\hat{A}_r z_i = \lambda_i z_i$. The desired results follow immediately after substituting the particular choice of Γ_r. ∎

Thus, while the first r eigenvalues are simultaneously deflated (or shifted), the remaining $n - r$ eigenvalues are untouched. The rectangular matrix Y is, however, yet to be defined.

Let $Y = [y_1 \ldots y_r]$ be the eigenvector matrix of $\hat{A}^T$. In this case, the eigenvalue matrix $\hat{E} = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$ satisfies the relation $\hat{E} = Y^T \hat{A} Z$. We first consider the case where $\gamma_i = \lambda_i$, $i = 1, \ldots, r$. In this case, $\Gamma_r = \hat{E}$, and

$$\hat{A}_r = \hat{A} - Z\hat{E}Y^T = \hat{A} - Z\hat{E}\hat{E}^{-1}\hat{E}Y^T = \hat{A} - \hat{A}Z\hat{E}^{-1}Y^T\hat{A},$$

because $\hat{A}Z = Z\hat{E}$ and $\hat{A}^T Y = Y\hat{E}$. Thus,

$$\hat{A}_r = (I - \hat{A}Z\hat{E}^{-1}Y^T)\hat{A} =: P_{\hat{D}}\hat{A}, \tag{2.6}$$

or

$$\hat{A}_r = \hat{A}(I - Z\hat{E}^{-1}Y^T\hat{A}) =: \hat{A}Q_{\hat{D}}. \tag{2.7}$$

(In this paper, the notations P (e.g., $P_{\hat{D}}$) and Q (e.g., $Q_{\hat{D}}$) are reserved for denoting the left and the right *projection* preconditioner, respectively.) So, suppose that (2.1) is left preconditioned by $P_{\hat{D}} := I - \hat{A}Z\hat{E}^{-1}Y^T$, i.e.,

$$P_{\hat{D}}\hat{A}\hat{u} = P_{\hat{D}}\hat{b}. \tag{2.8}$$

The spectrum of $P_{\hat{D}}\hat{A}$ is then given by Theorem 2.3. With Z and Y consisting of the eigenvectors of $\hat{A}$ and $\hat{A}^T$, respectively, $\sigma(P_{\hat{D}}\hat{A}) = \sigma(\hat{A}Q_{\hat{D}})$.

If a Krylov subspace method is applied on (2.8), the method implicitly approximates eigenvalues of $P_{\hat{D}}\hat{A}$. Since the smallest eigenvalue of $P_{\hat{D}}\hat{A}$ is λ_{r+1}, the method will approximate λ_{r+1} and not λ_1. In the Krylov subspace terminology, this also means that the eigenspace spanned by Z, which is associated with eigenvalues λ_i, $i = 1, \ldots, r$, no longer has components in the residuals; these components are projected out of the residual. (This fact raises the notion of a "projection preconditioner," and $P_{\hat{D}}$ is one of the projection preconditioners.) It is then instructive to use the *effective* condition number κ_{eff} instead of the *usual* condition number κ to measure the convergence rate of a Krylov method. If $\hat{A}$ is symmetric positive definite, we then have that $\kappa_{eff} := \lambda_n/\lambda_{r+1} \leq \lambda_n/\lambda_1 =: \kappa$ because $\lambda_{r+1} \geq \lambda_1$. In this case, a Krylov subspace method applied to $P_{\hat{D}}\hat{A}$ will converge faster than if applied to $\hat{A}$.

We now consider the case where $\gamma_i = \lambda_i - \lambda_n$, $i = 1, \ldots, r$. In this case, $\Gamma_r = \hat{E} - \lambda_n I_r$, with I_r the identity matrix of appropriate size. For $Y = [y_1 \ldots y_r]$ the

eigenvector matrix of $\hat{A}^T$ and $\hat{E} = Y^T \hat{A} Z$, we have

$$\begin{aligned}\hat{A}_{r,\gamma} &= \hat{A} - Z\Gamma_r Y^T = \hat{A} - Z(\hat{E} - \lambda_n I_r)Y^T = \hat{A} - Z\hat{E}Y^T + \lambda_n ZY^T \\ &= \hat{A} - Z\hat{E}\hat{E}^{-1}\hat{E}Y^T + \lambda_n Z\hat{E}^{-1}\hat{E}Y^T = \hat{A} - \hat{A}Z\hat{E}^{-1}Y^T\hat{A} + \lambda_n Z\hat{E}^{-1}Y^T\hat{A}.\end{aligned}$$

Thus,

$$\hat{A}_{r,\gamma} = (I - \hat{A}Z\hat{E}^{-1}Y^T + \lambda_n Z\hat{E}^{-1}Y^T)\hat{A} =: P_{\hat{N}}\hat{A}. \tag{2.9}$$

Similar to $P_{\hat{D}}\hat{A}$, if a Krylov subspace method is applied to $P_{\hat{N}}\hat{A}$, λ_{r+1} will be the smallest eigenvalue to be approximated and not λ_1. Since $P_{\hat{D}}\hat{A}$ and $P_{\hat{N}}\hat{A}$ are spectrally similar (cf. Theorem 2.3), one may expect that the convergence of a Krylov subspace method applied to them should also be similar.

The form (2.9) is the *left* preconditioning version of this type of projection. A *right* preconditioning version of it can be derived as follows:

$$\begin{aligned}\hat{A}_{r,\gamma} &= \hat{A} - Z\Gamma_r Y^T = \hat{A} - Z\hat{E}Y^T + \lambda_n ZY^T \\ &= \hat{A} - Z\hat{E}\hat{E}^{-1}\hat{E}Y^T + \lambda_n Z\hat{E}\hat{E}^{-1}Y^T = \hat{A} - \hat{A}Z\hat{E}^{-1}Y^T\hat{A} + \lambda_n \hat{A}Z\hat{E}^{-1}Y^T.\end{aligned}$$

Thus,

$$\hat{A}_{r,\gamma} = \hat{A}(I - Z\hat{E}^{-1}Y^T\hat{A} + \lambda_n Z\hat{E}^{-1}Y^T) =: \hat{A}Q_{\hat{N}}. \tag{2.10}$$

Remark 2.4 (balancing preconditioner). Instead of shifting small eigenvalues to zero as in deflation, small eigenvalues can also be shifted to one as in the case of the balancing preconditioner [9, 8], widely used in domain decomposition methods. In abstract formulation, for $\hat{A} = A$, the balancing preconditioner can be written as

$$P_B = Q_D M^{-1} P_D + ZE^{-1}Y^T, \qquad E = Y^T AZ, \tag{2.11}$$

where M is the preconditioner associated with the coarse-grid solve. See also monographs [20, 22] for different expositions of the balancing preconditioner and [12, 3] for comparisons with deflation.

Remark 2.5 (additive coarse-grid correction). The idea of shifting the small eigenvalues towards the maximum eigenvalue is not new and has been discussed in [15]. The resultant projection operator in that paper is, however, different from ours. In fact, the projection operator in [15] belongs to the additive coarse-grid correction preconditioner.

Remark 2.6 (multistep fixed point iteration). Many existing multilevel methods can be viewed from multistep fixed point iterations. A multistep fixed point iteration consists of a sequence of *smoothing* and *coarse-grid correction*. A multigrid or domain decomposition method can always be represented by or decomposed into this sequence. From an abstract point of view, e.g., the balancing preconditioner is a fixed point iteration with the following steps: coarse-grid correction, smoothing, and then coarse-grid correction. The new error e^{new} can be related to the previous one by

$$e^{new} = Ke^{old}, \tag{2.12}$$

where K is the iteration matrix.

For $P_{\hat{N}}$, however, we could not find a decomposition which associates it with a multistep fixed point iteration. Hence, similar to deflation, it is somewhat necessary

to see $P_{\hat{N}}\hat{A}$ fully only from the Krylov subspace iteration context and not from a fixed point iteration.

It has been shown for SPD matrices in [12] and nonsymmetric matrices in [3] that for some starting vectors $\hat{u}_0$ a Krylov subspace method with deflation produces, respectively, the $\hat{A}$-norm of error and the 2-norm of residual, which are never larger than the abstract balancing preconditioner. Since $P_{\hat{N}}\hat{A}$ and $P_{\hat{D}}\hat{A}$ (as well as $\hat{A}Q_{\hat{N}}$ and $\hat{A}Q_{\hat{D}}$) have similar spectra, it is also worthwhile to consider $P_{\hat{N}}$ and $Q_{\hat{N}}$ as another type of projection-based preconditioner for $\hat{A}$.

We note here that our definition of $P_{\hat{D}}$ is similar to that used in [16]. By direct computation, we have that $P_{\hat{D}}^2 = P_{\hat{D}}$; hence, $P_{\hat{D}}$ is a projector. $P_{\hat{N}}$, however, is not a projector, because $P_{\hat{N}}^2 \neq P_{\hat{N}}$. We classify $P_{\hat{N}}$ as a projection preconditioner only because the action of $P_{\hat{N}}$ on $\hat{A}$ results in projection of some eigenvalues of $\hat{A}$ to a value suitable for convergence acceleration of a Krylov subspace method.

3. Further spectral properties. In this section, we focus mainly on the left projection preconditioner $P_{\hat{N}}\hat{A}$, with $\hat{A} = M^{-1}A$. We rewrite $P_{\hat{N}}\hat{A}$ below:

$$P_{\hat{N}} \equiv I - \hat{A}Z\hat{E}^{-1}Y^T + \lambda_n Z\hat{E}^{-1}Y^T = P_{\hat{D}} + \lambda_n Z\hat{E}^{-1}Y^T, \tag{3.1}$$

where $\hat{E} = Y^T\hat{A}Z = Y^TM^{-1}AZ$. We first note that, if $\hat{A}$ is symmetric, it is natural to set $Y = Z$ in order to preserve symmetry. This choice results in a symmetric $P_{\hat{D}}\hat{A}$ for a symmetric $\hat{A}$. But, even with $\hat{A}$ symmetric and $Z = Y$, $P_{\hat{N}}\hat{A}$ is nonsymmetric. In relation with $P_{\hat{D}}$, we have the following properties, which are valid for any possible preconditioning formulation for $\hat{A}$: $M^{-1}A$ (left), AM^{-1} (right), and $M_1^{-1}AM_2^{-1}$ (split preconditioning).

LEMMA 3.1. *Let $Z, Y \in \mathbb{R}^{n\times r}$ be any matrices of rank r. $\hat{A}$ is nonsingular. Then the following relations hold:*

(i) $P_{\hat{D}}\hat{A}Z = Z^T\hat{A}Q_{\hat{D}} = 0$;

(ii) $Z^TP_{\hat{D}} = Q_{\hat{D}}Z = 0$;

(iii) $P_{\hat{D}}\hat{A} = \hat{A}Q_{\hat{D}}$.

Proof. As all equalities can be established by direct computations, we show only the proof for the first part of (i); the rest can be proved similarly. In this case, we have

$$P_{\hat{D}}\hat{A}Z = \hat{A}Z - \hat{A}Z\hat{E}^{-1}Y^T\hat{A}Z = 0, \tag{3.2}$$

due to $\hat{E} = Y^T\hat{A}Z$. ∎

In section 2 we considered a projection which is based on invariant vectors associated with $\hat{A}$ (and $\hat{A}^T$). In practice, such invariant vectors are not known and have yet to be computed. This computation itself is already expensive. In order to make the projection preconditioner viable, we need to use vectors which are different from eigenvectors to construct Z and Y.

In the discussions to follow, we neither specify the vectors for Z and Y nor determine how to construct them. We just assume that such vectors are available. We now need to see how the spectrum of $P_{\hat{N}}\hat{A}$ (and $\hat{A}Q_{\hat{N}}$) behaves under "arbitrary" Z and Y. We have an intermediate result, which establishes spectral equivalence of $P_{\hat{N}}\hat{A}$ with another but similar preconditioner, as follows.

THEOREM 3.2. *Let $Z, Y \in \mathbb{R}^{n\times r}$ be any rectangular matrices of rank r, and let $P_{\hat{N}}$ be defined as in* (3.1). *Define $P_{\hat{B}} = Q_{\hat{D}}P_{\hat{D}} + \lambda_n Z\hat{E}^{-1}Y^T$. Then*

$$\sigma(P_{\hat{N}}\hat{A}) = \sigma(P_{\hat{B}}\hat{A}). \tag{3.3}$$

Proof. We have that

$$\begin{aligned} P_{\hat{N}}\hat{A} &\equiv P_{\hat{D}}\hat{A} + \lambda_n Z\hat{E}^{-1}Y^T\hat{A} = P_{\hat{D}}^2\hat{A} + \lambda_n Z\hat{E}^{-1}Y^T\hat{A} \\ &= P_{\hat{D}}\hat{A}Q_{\hat{D}} + \lambda_n Z\hat{E}^{-1}Y^T\hat{A} = P_{\hat{D}}\hat{A}Q_{\hat{D}} + \lambda_n I - \lambda_n Q_{\hat{D}} \\ &= \lambda_n(I + (\lambda_n^{-1}P_{\hat{D}}\hat{A} - I)Q_{\hat{D}}). \end{aligned}$$

Thus,

$$\lambda \in \sigma(P_{\hat{N}}\hat{A}) = \sigma(\lambda_n(I + (\lambda_n^{-1}P_{\hat{D}}\hat{A} - I)Q_{\hat{D}})) = \lambda_n + \mu,$$

with $\mu \in \sigma(\lambda_n(\lambda_n^{-1}P_{\hat{D}}\hat{A} - I)Q_{\hat{D}})$. But

$$\sigma(\lambda_n(\lambda_n^{-1}P_{\hat{D}}\hat{A} - I)Q_{\hat{D}}) = \sigma(\lambda_n Q_{\hat{D}}(\lambda_n^{-1}P_{\hat{D}}\hat{A} - I)).$$

Hence,

$$\begin{aligned} \lambda \in \sigma(P_{\hat{N}}\hat{A}) &= \sigma(\lambda_n I + (P_{\hat{D}}\hat{A} - \lambda_n I)Q_{\hat{D}}) = \sigma(\lambda_n I + Q_{\hat{D}}(P_{\hat{D}}\hat{A} - \lambda_n I)) \\ &= \sigma(\lambda_n I + Q_{\hat{D}}P_{\hat{D}}\hat{A} - \lambda_n Q_{\hat{D}}) = \sigma((Q_{\hat{D}}P_{\hat{D}} + \lambda_n Z\hat{E}^{-1}Y^T)\hat{A}), \end{aligned}$$

which completes the proof. ∎

Notice that the preconditioner $P_{\hat{B}} := Q_{\hat{D}}P_{\hat{D}} + \lambda_n Z\hat{E}Y^T$ is very similar to the abstract balancing preconditioner and is the same as the abstract balancing preconditioner if $M = I$ and $\lambda_n = 1$ (cf. (2.11) in Remark 2.4).

We now consider $P_{\hat{D}}\hat{A}$. For arbitrary full ranked Z and Y, the first r eigenvalues of $P_{\hat{D}}\hat{A}$ are zero. (This is because $P_{\hat{D}}\hat{A}Z = 0$; cf. Lemma 3.1.) Denote by μ_i the eigenvalues of $P_{\hat{D}}\hat{A}$ for $r+1 \le i \le n$. The corresponding eigenvectors are $\widetilde{z}_i$; thus $P_{\hat{D}}\hat{A}\widetilde{z}_i = \mu_i\widetilde{z}_i$. The next theorem relates the spectrum of $P_{\hat{D}}\hat{A}$ with $P_{\hat{B}}\hat{A}$.

THEOREM 3.3. *Let $Z, Y \in \mathbb{R}^{n\times r}$ be any matrices of rank r. Let $\hat{A}$ be nonsingular. If*

$$\sigma(P_{\hat{D}}\hat{A}) = \{0, \dots, 0, \mu_{r+1}, \dots, \mu_n\}, \tag{3.4}$$

then

$$\sigma(P_{\hat{B}}\hat{A}) = \{\lambda_n, \dots, \lambda_n, \mu_{r+1}, \dots, \mu_n\}. \tag{3.5}$$

Proof. For $i = 1, \dots, r$, we know that $P_{\hat{D}}\hat{A}Z = 0$. Thus,

$$P_{\hat{B}}\hat{A}Z = Q_{\hat{D}}P_{\hat{D}}\hat{A}Z + \lambda_n Z\hat{E}^{-1}Y^T\hat{A}Z = \lambda_n Z. \tag{3.6}$$

In this case $Z = [z_1 \dots z_r]$ are the eigenvectors of $P_{\hat{D}}\hat{A}$ and $P_{\hat{B}}\hat{A}$. For $r+1 \le i \le n$, we have

$$\begin{aligned} P_{\hat{B}}\hat{A}Q_{\hat{D}}\widetilde{z}_i &= Q_{\hat{D}}P_{\hat{D}}\hat{A}Q_{\hat{D}}\widetilde{z}_i + \lambda_n Z\hat{E}^{-1}Y^T\hat{A}Q_{\hat{D}}\widetilde{z}_i \\ &= Q_{\hat{D}}P_{\hat{D}}\hat{A}\widetilde{z}_i = \mu_i Q_{\hat{D}}\widetilde{z}_i, \end{aligned}$$

because of Lemma 3.1. Thus, $\mu_i \in \sigma(P_{\hat{B}}\hat{A})$, and the corresponding eigenvector is $Q_{\hat{D}}\widetilde{z}_i$. ∎

Combining Theorems 3.2 and 3.3 results in the following spectral comparison.

THEOREM 3.4. *Let $Z, Y \in \mathbb{R}^{n\times r}$ be any matrices of rank r. Let $\hat{A}$ be nonsingular. If*

$$\sigma(P_{\hat{D}}\hat{A}) = \{0, \ldots, 0, \mu_{r+1}, \ldots, \mu_n\}, \tag{3.7}$$

then

$$\sigma(P_{\hat{N}}\hat{A}) = \{\lambda_n, \ldots, \lambda_n, \mu_{r+1}, \ldots, \mu_n\}. \tag{3.8}$$

Thus, for any full ranked Z and Y, the spectra of $P_{\hat{N}}\hat{A}$ and $P_{\hat{D}}\hat{A}$ are still similar.

Suppose that $\hat{A}$ is SPD, and set $Y = Z$. In this case, $\hat{E}$ is symmetric, and $P_{\hat{D}}\hat{A} = \hat{A} - \hat{A}ZE^{-1}Z^T\hat{A}$ is also symmetric and amounts to the difference between an SPD matrix $\hat{A}$ and a positive semidefinite matrix $\hat{A}Z\hat{E}^{-1}Z^T\hat{A}$. If λ_i and μ_i are the eigenvalues of $\hat{A}$ and $P_{\hat{D}}\hat{A}$, respectively, then $\mu_n \leq \lambda_n$ (cf. [7]). The condition number of $P_{\hat{D}}\hat{A}$ is $\kappa(P_{\hat{D}}\hat{A}) = \mu_n/\mu_{r+1}$. $P_{\hat{N}}\hat{A}$ is, however, not symmetric, even if $\hat{A}$ is SPD and $Y = Z$. The condition number of $P_{\hat{N}}\hat{A}$ becomes very complicated to determine.

We define for any nonsingular matrix with positive real eigenvalues *the quality of spectral clustering* as the ratio between the largest and the smallest eigenvalue, i.e., $\lambda_{\max}(A)/\lambda_{\min}(A) =: \hat{\kappa}(A)$. For $P_{\hat{N}}\hat{A}$ we have $\hat{\kappa}(P_{\hat{N}}\hat{A}) := \lambda_n/\mu_{r+1} \geq \mu_n/\mu_{r+1} =: \hat{\kappa}(P_{\hat{D}}\hat{A})$; i.e., the spectrum of $P_{\hat{D}}\hat{A}$ is better clustered than the spectrum of $P_{\hat{N}}\hat{A}$.

Since Krylov subspace methods converge rapidly for a linear system whose matrix has a much-clustered spectrum, this conclusion again shows the effectiveness of deflation for Krylov subspace convergence acceleration. But there exists an $\omega_1 \in \mathbb{R}$ such that $\mu_n = \omega_1\lambda_n$. Thus, if λ_n in $P_{\hat{N}}$ is replaced by $\omega_1\lambda_n$, then $\hat{\kappa}(P_{\hat{D}}\hat{A}) = \hat{\kappa}(P_{\hat{N}}\hat{A})$, and $P_{\hat{D}}\hat{A}$ and $P_{\hat{N}}\hat{A}$ have the same quality of clustering. In this situation, we may expect a similar convergence for $P_{\hat{D}}\hat{A}$ and $P_{\hat{N}}\hat{A}$.

Before closing this section, we establish the spectral equivalence between $P_{\hat{A}}\hat{A}$ and $\hat{A}Q_{\hat{N}}$ for any full ranked $Z, Y \in \mathbb{R}^{n\times r}$ in the following theorem.

THEOREM 3.5. *Let $Z, Y \in \mathbb{R}^{n\times r}$ be any matrices of rank r. Then*

$$\sigma(P_{\hat{N}}\hat{A}) = \sigma(\hat{A}Q_{\hat{N}}). \tag{3.9}$$

Proof. By using the definition of $Q_{\hat{N}}$, we have

$$\begin{aligned}\hat{A}Q_{\hat{N}} &= \hat{A}Q_{\hat{D}} + \lambda_n\hat{A}Z\hat{E}^{-1}Y^T = \hat{A}Q_{\hat{D}}^2 + \lambda_n\hat{A}Z\hat{E}^{-1}Y^T \\ &= P_{\hat{D}}\hat{A}Q_{\hat{D}} + \lambda_n(I - P_{\hat{D}}) = \lambda_n(I + P_{\hat{D}}(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I)).\end{aligned}$$

Thus,

$$\lambda \in \sigma(\hat{A}Q_{\hat{N}}) = \sigma(\lambda_n(I + P_{\hat{D}}(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I))) = \lambda_n + \mu,$$

with $\mu \in \sigma(\lambda_n P_{\hat{D}}(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I))$. Since

$$\sigma(\lambda_n P_{\hat{D}}(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I)) = \sigma(\lambda_n(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I)P_{\hat{D}}),$$

we then have

$$\begin{aligned}\lambda \in \sigma(\hat{A}Q_{\hat{N}}) &= \sigma(\lambda_n I + \lambda_n P_{\hat{D}}(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I)) = \sigma(\lambda_n I + \lambda_n(\lambda_n^{-1}\hat{A}Q_{\hat{D}} - I)P_{\hat{D}}) \\ &= \sigma(\lambda_n I + \hat{A}Q_{\hat{D}}P_{\hat{D}} - \lambda_n P_{\hat{D}}) = \sigma(\hat{A}(Q_{\hat{D}}P_{\hat{D}} + \lambda_n Z\hat{E}^{-1}Y^T)) \\ &= \sigma((Q_{\hat{D}}P_{\hat{D}} + \lambda_n Z\hat{E}^{-1}Y^T)\hat{A}).\end{aligned}$$

Because of Theorem 3.2, we finally get $\sigma(P_{\hat{N}}\hat{A}) = \sigma(\hat{A}Q_{\hat{N}})$. ∎

Therefore, the spectral analysis given previously for $P_{\hat{N}}\hat{A}$ also holds for $\hat{A}Q_{\hat{N}}$. Furthermore, the convergence behavior of GMRES applied to $\hat{A}Q_{\hat{N}}$ can also be well explained by the spectrum of $P_{\hat{N}}\hat{A}$.

4. Implementation aspects. If we use a small deflation subspace Z, the results from section 3 appear to be in favor of $P_{\hat{D}}$. In this case, the matrix $\hat{E}$ is still small, and its inverse can therefore be easily computed and stored. In this section, we discuss an important aspect related to the inversion of $\hat{E}$ needed in our projection-based preconditioner.

So far, we have not particularly specified Z and Y. Since $E \equiv Y^T AZ$ is the Galerkin product associated with the matrix A, we will just call E "the Galerkin matrix" throughout this section. The linear system related to $\hat{E}$ will be then called "the Galerkin system."

4.1. Inexact coarse-grid solve. For deflation, it has been shown theoretically as well as numerically that larger deflation subspace leads to better convergence [12, 6]. Since the inverse of the Galerkin matrix $\hat{E}$ must be computed, there is, however, a limit with respect to the size of the deflation subspace in order to make the overall performance efficient. For a large Galerkin matrix $\hat{E}$, one may think of using an iterative method to approximately invert it. It has been shown in [12, 21], however, that the convergence rate of a Krylov method with deflation is sensitive to the inaccuracy in solving the Galerkin system. In this case, the smallest eigenvalues are not shifted (or deflated) exactly to zero but to very small values $0 < |\epsilon| \ll \lambda_1$. The presence of very small eigenvalues makes the convergence even worse. If we choose to use an iterative method to solve the Galerkin system, then a very tight termination criterion must be employed.

Such a problematic behavior of $P_{\hat{D}}$ does not appear in $P_{\hat{N}}$. This is shown in the following proposition.

PROPOSITION 4.1. *Let $Z = [z_1 \dots z_r]$ be a matrix whose columns are the eigenvectors of $\hat{A}$, and assume that, in $P_{\widetilde{N}} = I - \hat{A}Z\widetilde{E}^{-1}Y^T + \lambda_n Z\widetilde{E}^{-1}Y^T$,*

$$\widetilde{E}^{-1} = \operatorname{diag}\left(\frac{1-\epsilon_1}{\lambda_1} \dots \frac{1-\epsilon_r}{\lambda_r}\right),$$

where $|\epsilon_i|_{i=1,r} \ll 1$. Then the spectrum of $P_{\widetilde{N}}\hat{A}$ is

$$\sigma(P_{\widetilde{N}}\hat{A}) = \{(1-\epsilon_1)\lambda_n + \lambda_1\epsilon_1, \dots, (1-\epsilon_r)\lambda_n + \lambda_r\epsilon_r, \lambda_{r+1}, \dots, \lambda_n\}. \tag{4.1}$$

Proof. For $i = 1, \dots, r$, we have

$$\begin{aligned}
P_{\widetilde{N}}\hat{A}Z &= \hat{A}Z - \hat{A}Z\widetilde{E}^{-1}Y^T\hat{A}Z + \lambda_n Z\widetilde{E}^{-1}Y^T\hat{A}Z \\
&= \hat{A}Z - \hat{A}Z\operatorname{diag}\left(\frac{1-\epsilon_1}{\lambda_1}, \dots, \frac{1-\epsilon_r}{\lambda_r}\right)\operatorname{diag}(\lambda_1, \dots, \lambda_r) \\
&\quad + \lambda_n Z\operatorname{diag}\left(\frac{1-\epsilon_1}{\lambda_1} \dots \frac{1-\epsilon_r}{\lambda_r}\right)\operatorname{diag}(\lambda_1, \dots, \lambda_r) \\
&= \hat{A}Z - \hat{A}Z\operatorname{diag}(1-\epsilon_1, \dots, 1-\epsilon_r) + Z\operatorname{diag}(\lambda_n(1-\epsilon_1), \dots, \lambda_n(1-\epsilon_r)) \\
&= Z\operatorname{diag}((1-\epsilon_1)\lambda_n + \lambda_1\epsilon_1, \dots, (1-\epsilon_r)\lambda_n + \lambda_r\epsilon_r).
\end{aligned}$$

For $r+1 \le i \le n$,

$$\begin{aligned}
P_{\widetilde{N}}\hat{A}z_i &= \hat{A}z_i - \hat{A}Z\widetilde{E}^{-1}Z^T\hat{A}z_i + \lambda_n Z\widetilde{E}^{-1}Z^T\hat{A}z_i \\
&= \lambda_i z_i - \hat{A}Z\widetilde{E}^{-1}Z^T(\lambda_i z_i) + \lambda_n Z\widetilde{E}^{-1}Z^T(\lambda_i z_i) = \lambda_i z_i,
\end{aligned}$$

because of orthogonality. ∎

With the Galerkin system solved only approximately, we have that $\hat{\kappa}(P_{\widetilde{N}}\hat{A}) := \max\{\max\{(1-\epsilon_i)\lambda_n/\lambda_{r+1} : i = 1,\ldots,r\}, \lambda_n/\lambda_{r+1}\}$, with $\hat{\kappa}$ indicating the quality of spectral clustering. For $|\epsilon_i| \ll 1$, $\hat{\kappa}(P_{\widetilde{N}}\hat{A}) \cong \lambda_n/\lambda_{r+1} = \hat{\kappa}(P_{\hat{N}}\hat{A})$. Therefore, with $P_{\hat{N}}$, the convergence rate of a Krylov method will not be dramatically deteriorated if $\hat{E}^{-1}$ is computed without a sufficient accuracy.

For $P_{\hat{N}}$, having such a *stability* with respect to inexact solves of the Galerkin system allows us to enlarge the deflation subspace and solve the Galerkin system with an iterative method. This will shift many of the small eigenvalues and cluster the spectrum as much as possible around λ_n. Such a spectral clustering will in general lead to a faster convergence in terms of number of iterations.

4.2. Estimation of the maximum eigenvalue. Another point related to $P_{\hat{N}}$ is the determination of λ_n. In practice, λ_n is not known and has yet to be computed. This computation is expensive and must be avoided. At first sight, this seems to be a serious drawback of $P_{\hat{N}}$ as compared to $P_{\hat{D}}$. In many cases, however, λ_n can be approximated, for example, by using Gerschgorin's theorem, discussed below.

First, the next proposition shows the spectrum of $P_{\hat{N}}\hat{A}$ if λ_n is obtained from an approximation. Its proof is similar to, e.g., the proof of Proposition 4.1.

PROPOSITION 4.2. *Let λ_n and $|\delta| \ll \lambda_n$ be the largest eigenvalue of $\hat{A}$ and a constant, respectively. Z and Y consist of the right and left eigenvectors of $\hat{A}$, respectively, associated with the first r eigenvalues. In the case where $P_{\widetilde{N}} = I - \hat{A}Z\hat{E}^{-1}Y + \lambda_{n,est}Z\hat{E}^{-1}Y^T$, where $\lambda_{n,est} = \lambda_n + \delta$, the spectrum of $P_{\widetilde{N}}\hat{A}$ is*

$$\sigma(P_{\widetilde{N}}\hat{A}) = \{\lambda_{n,est},\ldots,\lambda_{n,est},\lambda_{r+1},\ldots,\lambda_n\}.$$

Thus, as long as $\lambda_{n,est}$ is not too far from λ_n (i.e., the error δ is not of the same order as λ_n), the convergence rate of a Krylov subspace method with $P_{\hat{N}}$ is insensitive to $\lambda_{n,est}$. But, in this case, there again exists an $\omega_2 \in \mathbb{R}$ such that $\lambda_n = \omega_2\lambda_{n,est}$, and therefore $\hat{\kappa}(P_{\widetilde{N}}\hat{A}) = \hat{\kappa}(P_{\hat{N}}\hat{A})$.

A simple way to estimate λ_n is based on the Gerschgorin theorem. We skip this theorem and its proof, as they are well known, and refer the reader to, e.g., [24]. One consequence of the Gerschgorin theorem, which is relevant to our estimate on the maximum eigenvalue, is stated below.

THEOREM 4.3. *Let $\rho(\hat{A})$ be the maximum eigenvalue of $\hat{A}$ in magnitude. I.e., $\rho(\hat{A}) = \max\{|\lambda| : \lambda \in \sigma(\hat{A})\}$. $\rho(\hat{A})$ is the spectral radius of $\hat{A}$. For any $\hat{A} \in \mathbb{C}^{n\times n}$, then*

$$\rho(\hat{A}) \leq \max_{i\in N} \sum_{j\in N} |\hat{a}_{i,j}|. \tag{4.2}$$

Proof. See [24, p. 4]. ∎

Example 4.4. To illustrate how sharp this estimate is, we first consider a symmetric case obtained from a finite difference discretization of the one-dimensional (1D) Poisson equation:

$$-\frac{d^2u}{dx^2} = f, \qquad x = (0,1),$$

with Dirichlet boundary conditions on $x = 0, 1$. Set $M = I$. Hence, $\hat{A} = A$. The matrix A is SPD, $\rho(A) = \lambda_n(A)$, and the bound (4.2) reduces to the bound for $\lambda_n(A)$, i.e.,

$$\lambda_n \leq \max_{i\in N} \sum_{j\in N} |a_{i,j}| = \lambda_{n,est}. \tag{4.3}$$

TABLE 4.1
The computed maximum eigenvalue and its estimate for the 1D Poisson equation.

n	λ_n	$\lambda_{n,est}$
10	3.92E+2	4.E+2
50	9.99E+3	1.E+4
100	3.99E+4	4.E+4

TABLE 4.2
The computed maximum eigenvalue and its estimate for the 1D convection-diffusion equation.

Pe	$\lvert\lambda_n\rvert$	$\lvert\lambda_{n,est}\rvert$
20	3.15E+2	3.20E+2
40	6.29E+2	6.40E+2
100	1.58E+3	1.60E+3

The computed eigenvalue λ_n and the estimate $\lambda_{n,est}$ are presented in Table 4.1 for different numbers of grid points. As clearly seen, $\lambda_{n,est}$ is very close to λ_n.

Example 4.5. Now we consider a finite difference discretization of the 1D convection-diffusion equation:

$$\frac{du}{dx} - \frac{1}{Pe}\frac{d^2u}{dx^2} = q, \qquad x = (0,1),$$

with Dirichlet boundary conditions on $x = 0, 1$. Pe is the Péclet number and q the source term. The resultant matrix of coefficients A is nonsymmetric. The nonsymmetric part becomes dominant if the Péclet number is large (convection-dominated problem). We set $M = I$; therefore, $\hat{A} = A$. The computed largest eigenvalue and its estimate are shown in Table 4.2.

As compared to the computed maximum eigenvalue, the estimate again is quite good, even when Pe is large, and in some eigenvalues may be complex-valued. In this case the spectral radius $\rho(A) = \max\{|\lambda| : \lambda \in \sigma(A)\}$, *but* $\rho(A)$ *is not necessarily the same as the actual value of it.*

All convergence results presented in section 7 *are based on this type of estimation of the maximum eigenvalue* $\lambda_n(\hat{A})$.

5. Multilevel projection, nested iteration. As discussed early in section 3, a very much-clustered spectrum of $P_{\hat{N}}\hat{A}$ can be obtained by taking the deflation subspace $Z \in \mathbb{R}^{n\times r}$ to be sufficiently large. A large matrix Z will, however, lead to a large Galerkin matrix $\hat{E} \in \mathbb{R}^{r\times r}$ (but still $r < n$). In such a case, solving the Galerkin system will be very costly. Proposition 4.1 suggests, however, that using only an approximation to $\hat{E}^{-1}$ would not hamper the convergence rate of a Krylov subspace method dramatically. One may, for example, use an incomplete LU factorization to $\hat{E}$ and use these factors to approximately solve the Galerkin system. An ILU factorization, however, is not well parallelizable and requires significant memory during factorization and to store the factors.

In this section, we discuss the use of (inner) iterations to perform the inversion of $\hat{E}$. In the end, we show that such inner iterations can be constructed in a fashion that leads to another type of multilevel methods.

Before proceeding, we recall that $P_{\hat{N}}\hat{A}$ is nonsymmetric, even if $\hat{A}$ is symmetric. Therefore, a Krylov subspace method for a nonsymmetric system should be employed, e.g., GMRES or Bi-CGSTAB [23]. Particularly in our case, we base the construction of the iterative algorithm on GMRES.

We note that the left and right preconditioned GMRES produces different residuals. The left preconditioned GMRES, for example, produces the preconditioned residual during the iteration. In cases where the preconditioner does not accurately approximate A, the actual error associated with the preconditioned residual may still not be small when the termination criterion based on the preconditioned residual is reached. Since the actual residual is not a by-product of the algorithm, evaluation of the actual residual must be done at the expense of unnecessary computation of the approximate solution and one matrix-vector multiplication to compute the residual. This is not the case if the right preconditioned GMRES is used. In fact, the right preconditioned GMRES produces the actual residual during the iteration. Since we are going to also use GMRES for the *inner* iteration, it is important that the inner iteration terminates when the actual residual is already small. In this case, the matrix E should have already been inverted sufficiently accurately. This motivates the use of the right preconditioned GMRES throughout the rest of our discussion.

The right preconditioned GMRES solves the system

$$\hat{A}Q_{\hat{N}}\hat{u} = \hat{b} \tag{5.1}$$

or, with $\hat{A} = AM^{-1}$,

$$AM^{-1}Q_{\hat{N}}\hat{u} = b, \text{ where } u = M^{-1}Q_N\hat{u}, \tag{5.2}$$

for any nonsingular M. Since we assume any full ranked Z and Y, and since we are going to use an estimate for $\lambda_n(\hat{A})$, we need the scaling ω_1 and ω_2 to be able to shift r small eigenvalues as close as possible to $\mu_n = \lambda_n(P_{\hat{N}}\hat{A})$. Recall from sections 3 and 4.2 that $\mu_n = \omega_1\lambda_n$ and $\lambda_n = \omega_2\lambda_{n,est}$, and we have $\mu_n = \omega_1\omega_2\lambda_{n,est}$. Below, we fully write $Q_{\hat{N}}$ after accommodating this scaling:

$$Q_{\hat{N}} = I - Z\hat{E}^{-1}Y^TAM^{-1} + \omega\lambda_{n,est}Z\hat{E}^{-1}Y^T, \qquad \hat{E} = Y^T(AM^{-1})Z, \tag{5.3}$$

where $\omega := \omega_1\omega_2$ is called the *shift scaling factor*. Since $P_{\hat{N}}\hat{A}$ and $\hat{A}Q_{\hat{N}}$ are spectrally the same (cf. Theorem 3.5), the convergence behavior of both implementations can be expected to be the same.

5.1. Two-level projection. To make the presentation self-contained, we write in Algorithm 1 below the right preconditioned GMRES for solving (5.2).

ALGORITHM 1. FGMRES PRECONDITIONED BY M AND $Q_{\hat{N}}$.

1. Choose u_0, ω, and $\lambda_{n,est}$. Compute Q_N.
2. Compute $r_0 = b - Au_0$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$.
3. For $j = 1, 2, \ldots, k$
4. $\quad x_j := Q_{\hat{N}}v_j$;
5. $\quad w := AM^{-1}x_j$.
6. $\quad$ For $i = 1, 2, \ldots, j$
7. $\quad\quad h_{i,j} = (w, v_j)$;
8. $\quad\quad w := w - h_{i,j}v_i$.
9. $\quad$ Endfor
10. $\quad h_{j+1,j} := \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.
11. Endfor
12. Set $X_k := [x_1 \ldots x_k]$ and $\hat{H}_k = \{h_{i,j}\}_{1\le i\le j+1; 1\le j\le k}$.
13. Compute $y_k = \arg\min_y \|\beta e_1 - \hat{H}_k y\|_2$ and $u_k = u_0 + M^{-1}X_k y_k$.

First notice that, in Algorithm 1, we store X_k instead of $V_k := [v_1 \ldots v_k]$ in order to accommodate variable preconditioners. If $Q_{\hat{N}}$ and M are constant during the

course of iterations, they can be taken out from X_k, and the approximate solution is extracted from V_k. In this case, X_k is no longer needed, and only V_k is to be stored in memory. With constant M and $Q_{\hat{N}}$, the second part of line 13 becomes $u_k = u_0 + M^{-1}Q_{\hat{N}}V_k y_k$. Algorithm 1 is actually the *flexible* version of GMRES (or FGMRES).

Second, in line 4 of Algorithm 1, we need to compute $x_j = Q_{\hat{N}}v_j$. In general, the matrix $Q_{\hat{N}}$ is not sparse. Therefore, computing $Q_{\hat{N}}$ only once and storing it in the memory are not advisable, especially if the size of $\hat{A}$ is large. In the actual implementation, the action of $Q_{\hat{N}}$ on $\hat{A}$ is done implicitly as follows. By using the definition of $Q_{\hat{N}}$ (5.3), line 4 of Algorithm 1 is rewritten as follows:

$$\begin{aligned} x_j &= v_j - Z\hat{E}^{-1}Y^T\hat{A}v_j + \omega\lambda_{n,est}Z\hat{E}^{-1}Y^T v_j \\ &= v_j - Z\hat{E}^{-1}Y^T(AM^{-1} - \omega\lambda_{n,est}I)v_j. \end{aligned}$$

By setting $s := AM^{-1}v_j$ and $\widehat{v} := Y^T(s - \omega\lambda_{n,est}v_j)$, we then have

$$x_j = v_j - Z\hat{E}^{-1}\widehat{v}. \tag{5.4}$$

In (5.4), $\hat{E}^{-1}\widehat{v} =: \widetilde{v}$ can in principle be approximately computed by solving the Galerkin system

$$\widehat{v} = \hat{E}\widetilde{v}, \qquad \widehat{v}, \widetilde{v} \in \mathbb{R}^r, \tag{5.5}$$

for $\widetilde{v}$ iteratively. Once $\widetilde{v}$ is determined, we can continue computing $t := Z\widetilde{v}$ and finally $x_j := s - t$.

We note that $Z : \mathbb{R}^r \mapsto \mathbb{R}^n$ and $Y^T : \mathbb{R}^n \mapsto \mathbb{R}^r$. Without being specific to any particular choice of Z and Y, we call the linear map of $x \in \mathbb{R}^n$ into $\mathbb{R}^r$ a *restriction* and the linear map of $w \in \mathbb{R}^r$ into $\mathbb{R}^n$ a *prolongation*. In this case, Z is the prolongation operator, and Y^T is the restriction operator.

In summary, we write Algorithm 2 below. This algorithm is equivalent to Algorithm 1 but with line 4 expanded according to the above discussion.

ALGORITHM 2. FGMRES PRECONDITIONED BY M AND $Q_{\hat{N}}$.

1. Choose u_0, ω, and $\lambda_{n,est}$.
2. Compute $r_0 = b - Au_0$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$.
3. For $j = 1, 2, \ldots, k$
4. $\quad s := AM^{-1}v_j$.
5. $\quad$ Restriction: $\widehat{v} := Y^T(s - \omega\lambda_{n,est}v_j)$.
6. $\quad$ Solve for $\widetilde{v}$: $\hat{E}\widetilde{v} = \widehat{v}$.
7. $\quad$ Prolongation: $t := Z\widetilde{v}$.
8. $\quad x_j := v_j - t$.
9. $\quad w := AM^{-1}x_j$.
10. $\quad$ For $i = 1, 2, \ldots, j$
11. $\quad\quad h_{i,j} = (w, v_j)$;
12. $\quad\quad w := w - h_{i,j}v_i$.
13. $\quad$ Endfor
14. $\quad$ Compute $h_{j+1,j} := \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.
15. Endfor
16. Set $X_k := [x_1 \ldots x_k]$ and $\hat{H}_k = \{h_{i,j}\}_{1 \le i \le j+1; 1 \le j \le k}$.
17. Compute $y_k = \arg\min_y \|\beta e_1 - \hat{H}_k y\|_2$ and $u_k = u_0 + M^{-1}X_k y_k$.

Notice from Algorithm 2 that at each FGMRES iteration one requires two matrix-vector multiplications with A, two preconditioner solves, and one solve of small matrix

$\hat{E}$. Considering the case with $M = I$, the work needed by FGMRES with $Q_{\hat{N}}$ is the same as the work with deflation (in this case, $\lambda_{n,est} = 0$) and less than the work with the abstract balancing preconditioner P_B. In terms of computational work, this shows a clear advantage of $Q_{\hat{N}}$ over the abstract balancing preconditioner.

5.2. Multilevel, nested iterations. We now focus on solving the Galerkin system (5.5). We will base our solution method for the Galerkin system on Krylov subspace iterations.

To proceed, we need to change our notations. Without loss of generality, we set $M = I$; hence, $\hat{A} = A$. We denote now the matrix A by $A^{(1)}$, i.e., $A^{(1)} := A$, and the Galerkin matrix E by $A^{(2)}$, i.e., $A^{(2)} := E = Y^T A^{(1)} Z$. The projection preconditioner for $A^{(1)}$ is denoted by $Q_N^{(1)}$, where

$$Q_N^{(1)} = I^{(1)} - A^{(1)} Z^{(1,2)} {A^{(2)}}^{-1} {Y^{(1,2)}}^T + \omega^{(1)} \lambda_{n,est}^{(1)} Z^{(1,2)} {A^{(2)}}^{-1} {Y^{(1,2)}}^T, \tag{5.6}$$

where $\omega^{(1)}$ is the shift scaling factor related to $\lambda_n^{(1)}(A^{(1)})$, and so on. Equation (5.6) is the *two-level* projection, discussed in section 5.1.

If, for a special case, $A^{(1)}$ is SPD, and $Z^{(1,2)} = Y^{(1,2)}$, then $A^{(2)}$ is also SPD because

$$q^T A^{(2)} q = q^T {Z^{(1,2)}}^T A^{(1)} Z^{(1,2)} q = (Z^{(1,2)} q)^T A^{(1)} (Z^{(1,2)} q) > 0 \tag{5.7}$$

for $q \neq 0$. We cannot in general derive an interlacing property between $\sigma(A^{(1)})$ and $\sigma(A^{(2)})$ due to arbitrariness in $Z^{(1,2)}$. But suppose that, in the worst case, $\sigma(A^{(2)})$ contains $0 < \lambda_{\min}(A^{(2)}) = \lambda_1(A^{(2)}) \ll \lambda_1(A^{(1)})$ and $\lambda_{\max}(A^{(2)}) = \lambda_n(A^{(2)}) \gg \lambda_n(A^{(1)})$. In this case, we may expect a worse convergence rate of a Krylov method for solving the Galerkin system than the original one. This means that a substantial amount of work has to be spent on the Galerkin system (5.5). We can improve the convergence rate by applying a projection method similar to (5.6) to the Galerkin system (5.5). In the case that the Galerkin system is better conditioned than the original system, incorporating a projection preconditioner into the Galerkin system may improve the convergence further.

Consider the Galerkin system (5.5) with $M^{(1)} = I$. With right projection preconditioning, we write the Galerkin problem (5.5) as, after changing notations:

$$A^{(2)} Q_N^{(2)} p^{(2)} = \widehat{v}^{(2)}, \qquad \widehat{v}^{(2)} = Q_N^{(2)} p^{(2)}, \qquad p^{(2)} \in \mathbb{R}^r, \tag{5.8}$$

where

$$Q_N^{(2)} = I^{(2)} - Z^{(2,3)} {A^{(3)}}^{-1} {Y^{(2,3)}}^T A^{(2)} + \omega^{(2)} \lambda_{n,est}^{(2)} Z^{(2,3)} {A^{(3)}}^{-1} {Y^{(2,3)}}^T. \tag{5.9}$$

Here $A^{(3)} := {Y^{(2,3)}}^T A^{(2)} Z^{(2,3)}$, and $Z^{(2,3)}$ and $Y^{(2,3)}$ are again any rectangular matrices with full rank. A (*inner*) Krylov subspace method is then employed to solve the preconditioned Galerkin system (5.8) up to a specified accuracy.

In $Q_N^{(2)}$ one needs to solve a Galerkin system problem associated with the Galerkin matrix $A^{(3)}$. If $A^{(3)}$ is small, this can be done exactly by using a direct method. If $A^{(3)}$ is larger, a similar process to (5.8) has to be performed.

Given a set of matrices $A^{(\ell)}$, $\ell = 1, \ldots, m$, a multilevel nested Krylov subspace iteration results if, at every level $\ell < m$, the right preconditioned system

$$A^{(\ell)} Q_N^{(\ell)} p^{(\ell)} = \widehat{v}^{(\ell)}, \qquad \widehat{v}^{(\ell)} = Q_N^{(\ell)} p^{(\ell)}, \tag{5.10}$$

is solved, with

$$Q_N^{(\ell)} = I^{(\ell)} - Z^{(\ell,\ell+1)} A^{(\ell+1)^{-1}} Y^{(\ell,\ell+1)^T} A^{(\ell)} + \omega^{(\ell)} \lambda_{n,est}^{(\ell)} Z^{(\ell,\ell+1)} A^{(\ell+1)^{-1}} Y^{(\ell,\ell+1)^T} \tag{5.11}$$

and

$$A^{(\ell+1)} = Y^{(\ell,\ell+1)^T} A^{(\ell)} Z^{(\ell,\ell+1)}. \tag{5.12}$$

At $\ell = m$, the Galerkin matrix $A^{(m)}$ is already small, and the associated Galerkin system can be solved exactly by a sparse direct method. At level $\ell = 1, \ldots, m-1$, the problem (5.10) is solved iteratively by FGMRES.

Remark 5.1. Different from the usual nested Krylov subspace iteration, where the *inner* iteration is used to solve a preconditioner of the same dimension as $A^{(1)}$, in this multilevel projection method, since $\dim A^{(2)} \ll \dim A^{(1)}$, the inner iteration solves a much smaller system.

The final algorithm of the multilevel projection with nested Krylov methods is written in Algorithm 3, with $M^{(1)} \neq I$ and $\hat{A}^{(1)} = A^{(1)} M^{(1)^{-1}}$. In this algorithm, the FGMRES parts are simplified, by removing the solution computation steps.

ALGORITHM 3. MULTILEVEL PROJECTION WITH FGMRES.
Given $A^{(1)}$, $M^{(1)}$, $Z^{(\ell,\ell+1)}$, $Y^{(\ell,\ell+1)}$ for $\ell = 1, \ldots, m$, with $A^{(1)} = A$ and $M^{(1)} = M$. Compute $\hat{A}^{(\ell+1)} = Y^{(\ell,\ell+1)^T} \hat{A}^{(\ell)} Z^{(\ell,\ell+1)}$, where $\hat{A}^{(1)} = A^{(1)} M^{(1)^{-1}}$, and $\lambda_{n,est}^{(\ell)}$ for $\ell = 1, \ldots, m-1$, and choose $\omega^{(\ell)}$.

1. $\ell = 1$.
   With an initial guess $u_0^{(1)} = 0$, solve $A^{(1)} u^{(1)} = b^{(1)}$ with FGMRES by computing
2. $s^{(1)} := A^{(1)} M^{(1)^{-1}} v^{(1)}$;
3. restriction: $\widehat{v}^{(2)} := Y^{(1,2)^T} (s^{(1)} - \omega^{(1)} \lambda_{n,est}^{(1)} v^{(1)})$;
4. if $\ell + 1 = m$
5. solve exactly $\hat{A}^{(2)^{-1}} \widehat{v}^{(2)} = \widetilde{v}^{(2)}$;
6. else
7. $\ell = \ell + 1$.
   With $\widetilde{v}_0^{(2)} = 0$, solve $\hat{A}^{(2)} \widetilde{v}^{(2)} = \widehat{v}^{(2)}$ with FGMRES by computing
8. $s^{(2)} := \hat{A}^{(2)} v^{(2)}$;
9. restriction: $\widehat{v}^{(3)} := Y^{(2,3)^T} (s^{(2)} - \omega^{(2)} \lambda_{n,est}^{(2)} v^{(2)})$;
10. if $\ell + 1 = m$
11. solve exactly $\hat{A}^{(3)^{-1}} \widehat{v}^{(3)} = \widetilde{v}^{(3)}$;
12. else
13. $\ell = \ell + 1$;
14. ...
15. endif
16. prolongation: $t^{(2)} = Z^{(2,3)} \widetilde{v}^{(3)}$;
17. $x^{(2)} = v^{(2)} - t^{(2)}$;
18. $w^{(2)} = \hat{A}^{(2)} x^{(2)}$;
19. endif
20. prolongation: $t^{(1)} = Z^{(1,2)} \widetilde{v}^{(2)}$;
21. $x^{(1)} = v^{(1)} - t^{(1)}$;
22. $w^{(2)} = A^{(1)} M^{(1)^{-1}} x^{(2)}$.

5.3. Multilevel projection and multigrid. At a first view, multigrid methods and our new multilevel projection Krylov method appear to be similar. But in principal they are different. To illustrate this, we take as an example a multigrid iteration with one pre- and postsmoothing. The error relation between two consecutive multigrid iterations can be written as

$$e^{j+1} = (I - M^{-1}A_h)(I - A_h I_H^h A_H^{-1} I_h^H)(I - M^{-1}A_h)e^j, \tag{5.13}$$

where $A_h = A$ is the fine-grid matrix, A_H the coarse-grid matrix, M a matrix related to the smoother, and I_h^H and I_H^h the restriction and interpolation matrix, respectively. In multigrid, $A_H = I_h^H A_h I_H^h$ is called "the Galerkin coarse-grid approximation matrix." Thus, compared to $P_{\hat{N}}$ (or $Q_{\hat{N}}$) in our multilevel projection Krylov method, they have similar ingredients.

Indeed both methods try to project certain quantities, which are responsible for slow convergence of an iterative method. In the classical multigrid method, the projection is done by smoothing steps, which can be done with a Richardson-like iteration as Jacobi, Gauss–Seidel, or other more complicated methods. Some parts of errors which cannot be projected efficiently by smoothers then are projected out by the smoothers in the coarse-grid-correction step, which is the multilevel part in a multigrid method.

In our multilevel projection Krylov method, some eigenvalues of the original operator are projected into one point. Information about the projected eigenvalues is contained in the Galerkin matrix. Hence, the accuracy of solving the Galerkin problem determines the effectiveness of our projection method. This projection, however, is done in a recursive multilevel way. Since we use, e.g., FGMRES for solving the Galerkin system, FGMRES then can be considered as a sort of smoother in our multilevel projection Krylov method. Its role, however, is not necessarily to smooth certain quantities as *the* smoother in multigrid. FGMRES (or any Krylov method) does not actually distinguish smooth or rough quantities (such as errors). Hence, such a distinction may not be relevant in our multilevel projection method.

We also note that, in the classical geometric multigrid, the interpolation matrix is usually chosen such that errors on the coarse grid can accurately be interpolated into the fine grid. This can be achieved only if the errors on the coarse grid satisfies certain smoothness conditions.

In the multilevel projection method proposed here, however, there exists more freedom in choosing the deflation subspace Z and Y. The only condition that must be satisfied is that Z and Y are full rank. This condition is naturally satisfied by the interpolation and restriction matrices in multigrid. But such matrices are only one of many possible options for constructing Z and Y. In the multilevel projection Krylov method, Z and Y can be built without any *geometrical* relation between the fine grid and the coarse grid. If one uses, e.g., eigenvectors of A and A^T for Z and Y, respectively, for a class of matrices A satisfying the assumptions in section 2, $E = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. In this case, we can hardly interpret the matrix E as a sort of coarse-grid discretization of PDEs. Hence, in our multilevel projection method, to call E "the coarse-grid matrix" appears to be too restrictive.

In the next section, we make a specific choice of prolongation and restriction operators for our multilevel projection Krylov method, which is based on geometrical information about the underlying problem.

6. The choice of deflation subspaces Z and Y. In this section, we present one possible choice of constructing the deflation subspaces.

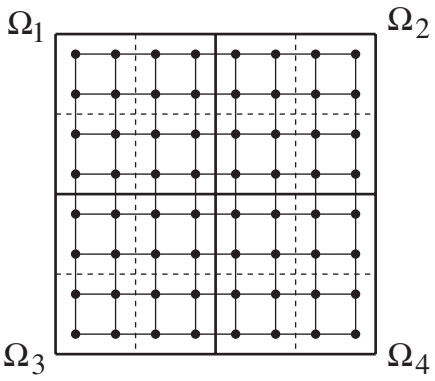

FIG. 6.1. *2D mesh with partitions.*

6.1. Agglomeration and redistribution. For ease of presentation, we assume in this section that, in $\hat{A}$, $M = I$. Hence, in the two-level projection notation, $E = Y^T AZ$.

For some classes of problems, the bilinear interpolation is a powerful technique to construct the prolongation operator in geometric multigrid methods. The restriction operator can be taken as the transpose of the prolongation operator. We can in principle directly bring these operators into our multilevel projection setting and use them as the deflation subspaces. We are, however, not going to do this. Instead, we will use a simpler technique than the bilinear interpolation, which is discussed below.

Suppose that the domain Ω with index set $\mathcal{I} = \{i | u_i \in \Omega\}$ is partitioned into l nonoverlapping subdomains Ω_j, $j = 1, \ldots, l$, with respective index $\mathcal{I}_j = \{i \in \mathcal{I} | u_i \in \Omega_j\}$. Then $Z = [z_{ij}]$ is defined as

$$z_{ij} = \begin{cases} 1, & i \in \mathcal{I}_j, \\ 0, & i \notin \mathcal{I}_j. \end{cases} \tag{6.1}$$

We then set $Y = Z$. (This construction can also be considered as interpolation, called the piecewise constant interpolation [14].)

To illustrate this construction, consider a square domain Ω discretized into an 8×8 equidistant mesh. The domain Ω is partitioned into four nonoverlapping subdomains: $\Omega_1, \Omega_2, \Omega_3, \Omega_4$, as shown in Figure 6.1. In this case, Z is a rectangular matrix with 4 columns. The first column has entries equal to one for points in Ω_1 and zero for $\Omega \setminus \Omega_1$, etc. As the result, the matrix $Z \in \mathbb{R}^{64 \times 4}$ has only 64 nonzeros, which is considered sparse.

We can refine the partition by increasing the number of subdomains. Consider the case where every subdomain Ω_j occupies only four points, shown in Figure 6.1 by the dashed lines. This mimics the standard coarsening in the *geometric* multigrid. However, in this case there exists no connectivity between points in Ω_j and points in $\Omega \setminus \Omega_j$. With (6.1), the prolongation and restriction processes are simpler than the prolongation and restriction matrices in the geometric multigrid. The resultant prolongation and restriction matrices are also sparser. Furthermore, this construction can easily be extended to arbitrary triangulation in finite elements, with complex local refinements.

Consider again Figure 6.1 partitioned with four subdomains $\Omega_1, \ldots, \Omega_4$. We number the unknowns as $u = [u_1 \ldots u_4]^T$. Thus,

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}, \tag{6.2}$$

with A_{jj} a block matrix of coefficients related to $i \in \mathcal{I}_j$, $j = 1, \ldots, 4$. Denote by $\mathbf{1}_j = \{1, \ldots, 1\}^T$ a vector of ones with length equal to the number of nodes in Ω_j, and $E = [e_{ij}]$. It is easy to show that

$$e_{ij} = \mathbf{1}_i^T A_{ij} \mathbf{1}_j. \tag{6.3}$$

Thus, the coefficients e_{ij} are nothing but the sum of all coefficients of A_{ij}. If there is no connectivity between two subdomains, then $A_{ij} = 0$, and as the consequence $e_{ij} = 0$. Thus, the Galerkin matrices E and A have a similar structure. (6.3) also suggests that e_{ij} need not be computed explicitly from $E = Z^T AZ$ and can be determined by simply summing up all coefficients of A_{ij}.

For comparison, in multigrid the restriction and prolongation matrices are usually stored in memory. In a few cases, one can avoid storing these matrices by computing the matrices once they are needed. This, however, should be done with an extra expense in computational work.

We can further save some work and memory with respect to the action of Z and Y^T on a vector. For example, because $Y = Z$, line 5 of Algorithm 2 can be written as, after dropping the iteration counter,

$$\widehat{v} = Z^T(s - \omega\lambda_n v) = \left(\sum_{i \in \mathcal{I}_1} (s_i - \omega\lambda_n v_i) \cdots \sum_{i \in \mathcal{I}_k} (s_i - \omega\lambda_n v_i) \right)^T. \tag{6.4}$$

Hence, the action of Z^T on the vector $s - \omega\lambda_n v$ can be regarded as *agglomerating* values on the fine grid into the coarse grid. Again dropping the iteration counter, for $t := vZ\widetilde{v}$ (line 7, Algorithm 2), notice that

$$t_i = \widetilde{v}_i, \qquad i \in \mathcal{I}_j, \quad j = 1, \ldots, k. \tag{6.5}$$

This is nothing but *redistributing* values on the coarse grid into the fine grid.

With this choice of Z we observe that the fine-coarse grid transfer reduces to only *agglomerating* values of the fine grid and *redistributing* values of the coarse grid. They are simple processes. Our numerical computations reveal that, even with these simple processes and without any additional preconditioner M, the convergence is quiet satisfactory for some classes of problems. This construction of prolongation and restriction operators will not, however, lead to a good multigrid method, as will be seen in our numerical results in section 7.

6.2. Additional preconditioner M. A simple preconditioner which can be incorporated in the algorithm and at the same time preserves the structure of A at every coarse-grid level is the diagonal scaling. In this case, $M = \mathrm{diag}(A)$. In the multigrid language, the diagonal scaling is associated with the point Jacobi smoother.

Consider again the partition of domain Ω into 4 subdomains as depicted in Figure 6.1. By using the same numbering as for (6.2), we have

$$AM^{-1} = \begin{bmatrix} D_{11}^{-1}A_{11} & D_{22}^{-1}A_{12} & D_{33}^{-1}A_{13} & D_{44}^{-1}A_{14} \\ D_{11}^{-1}A_{21} & D_{22}^{-1}A_{22} & D_{33}^{-1}A_{23} & D_{44}^{-1}A_{24} \\ D_{11}^{-1}A_{31} & D_{22}^{-1}A_{32} & D_{33}^{-1}A_{33} & D_{44}^{-1}A_{34} \\ D_{11}^{-1}A_{41} & D_{22}^{-1}A_{42} & D_{33}^{-1}A_{43} & D_{44}^{-1}A_{44} \end{bmatrix}, \tag{6.6}$$

where $D_{jj} = \mathrm{diag}(A_{jj})$. An eigenvalue estimate for AM^{-1} can be obtained by using Theorem 4.3. In this case, the estimate is given by

$$\lambda_{n,est}(AM^{-1}) = \max_{i \in N} \sum_{j \in N} |a_{i,j}/a_{i,i}|. \tag{6.7}$$

7. Numerical experiments. In this section we present the application of the multilevel projection method for solving linear systems of equations arising from some PDEs. We consider two problems: the Poisson equation and the convection-diffusion equation.

The Poisson equation represents the class of PDEs where the discretized form is symmetric (and positive definite). In this case standard methods, e.g., multigrid and domain decomposition methods, or deflated incomplete Cholesky CG (ICCG) already work sufficiently well. The second problem is related to the nonsymmetric linear system.

7.1. The Poisson equation. The first equation to solve is the Poisson equation:

$$-\nabla \cdot \nabla u = g \qquad \text{in } \Omega = (0,1)^2. \tag{7.1}$$

For the boundary condition we set the homogeneous Dirichlet conditions on $\Gamma = \partial\Omega$. The source term is the unit source placed in the middle of the domain. The computational domain is discretized by the central difference. The resultant matrix of coefficients is symmetric positive definite.

We run the multilevel FGMRES to solve the linear system. The multilevel FGMRES is terminated if the residual of the iteration at level $\ell = 1$ has been reduced by six orders of magnitude. No restart is used for the FGMRES. All coarse-grid problems are solved with a few FGMRES iterations (mostly only two). At the last level ($\ell = m - 1$), the related mth Galerkin problem is solved exactly. We set the shift scaling factor $\omega^{(\ell)} = 1$. The deflation subspace $Z^{(\ell,\ell+1)}$ is constructed by using (6.1) based on agglomeration of four neighboring points; see Figure 6.1.

The convergence results are shown in Table 7.1. To obtain the results, five grid levels are employed. Hence, for example, the coarsest grid for a 256^2 mesh is 16^2. In general, one can still go further until only one point remains in the domain.

On top of the convergence results in Table 7.1, the integers between parentheses indicate the numbers of FGMRES iterations used at level $\ell > 1$. So, for instance, in the first group of convergence results, 4 FGMRES iterations are used at the second level, 2 at the third level, and so on, denoted compactly by $(4, 2, 2, 2)$, etc.

From Table 7.1 we see that the method mostly converges after 14 iterations. The error at convergence, i.e., $e_{conv} = u - u_{conv}$, where $u = A^{-1}b$ is obtained from a direct method, is also quite satisfactory (which is of order 10^{-6}). Only when the method is run with intermediate iterations $(2, 2, 2, 2)$ does the convergence slightly deteriorate. Since, compared to others, only two iterations are done at the second level ($\ell = 2$), it gives an indication that the Galerkin system at this level is not solved sufficiently

TABLE 7.1
FGMRES iteration counts for the 2D Poisson equation.

	(4,2,2,2)		(4,3,3,3)	
N	Iter	$\|u-u_{conv}\|_2$	Iter	$\|u-u_{conv}\|_2$
32^2	14	4.53E–07	14	6.38E–07
64^2	14	1.08E–06	14	1.55E–06
128^2	14	4.50E–06	14	3.11E–06
256^2	14	5.55E–06	14	6.01E–06

	(2,2,2,2)		(6,2,2,2)	
N	Iter	$\|u-u_{conv}\|_2$	Iter	$\|u-u_{conv}\|_2$
32^2	15	1.05E–06	14	2.04E–07
64^2	16	8.60E–07	14	4.70E–07
128^2	16	2.45E–06	14	9.71E–07
256^2	16	3.40E–06	14	1.82E–06

accurately. The convergence, however, seems to be insensitive to the accuracy of the solutions at the remaining levels. Therefore, it is the accuracy at the second level which is important.

Using only two-level projection with the Galerkin system at level $\ell = 2$ solved exactly, FGMRES converges after 14 iterations. Hence, for $(4, 2, 2, 2)$, e.g., the method already converges with the best it can achieve if only one level preconditioner is used. Observe that the method converges independently of the grid size h. For comparison, multigrid with V-cycle, one pre- and post- red-black Gauss–Seidel smoothing, and bilinear interpolation achieves the same order of residual reduction after 11 iterations. If bilinear interpolation is replaced by the piecewise constant interpolation, multigrid does not converge after 40 iterations. In this case, for a 64^2 mesh the averaged residual reduction factor is only about 0.98.

7.2. The convection-diffusion equation. In this section we consider a nonsymmetric linear system arising from a discretization of the 2D convection-diffusion equation:

$$\frac{\partial u}{\partial y} - \frac{1}{Pe}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \quad \text{in } \Omega = (-1,1)^2, \tag{7.2}$$

where Pe is the Péclet number. The boundary conditions are determined as follows: $u(-1, y) \approx -1$, $u(1, y) \approx 1$, $u(x, -1) = x$, and $u(x, 1) = 0$. This problem resembles vertical flows with a boundary layer at the upper wall ($y = 1$). This problem is described in [2] and solved under finite element settings.

For our second example, we use the vertex-centered finite volume discretization described in [26]. The flux term is approximated by using an upwind scheme. Figure 7.1 shows solutions for $Pe = 20$ and 200; the boundary layer near the upper wall becomes thinner by an increase in Pe.

As discussed in [2], a multigrid method based on Galerkin coarse-grid discretization and bilinear interpolation intergrid transfer does not lead to an efficient method for $Pe > 1$. In fact, it is a divergent method. Convergence can still be obtained if, e.g., interpolation based on [28] is used, even though the convergent is not of the "textbook multigrid" one. By using this interpolation one needs to incorporate properties of the flows into the grid transfers [28, 25].

First we consider problems with $Pe = 20, 50, 100$, and 200 and construct the related linear systems based on equidistant meshes with 128^2, 256^2, and 512^2 grid

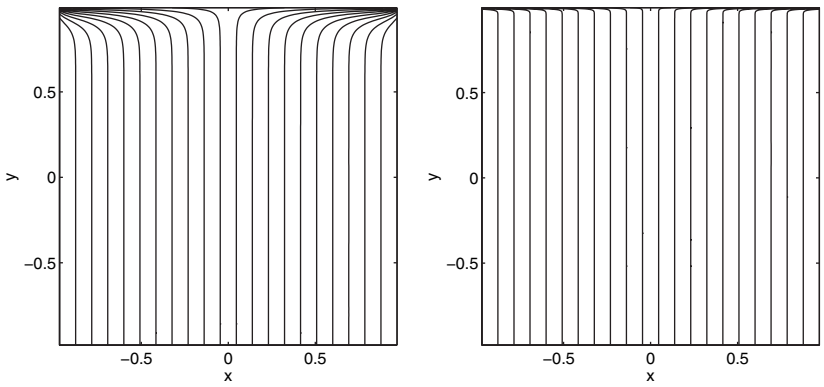

FIG. 7.1. *Contour of solutions of the 2D convection-diffusion equation with vertical wind:* $Pe = 20$ *(left) and* 200 *(right).*

TABLE 7.2
FGMRES iteration counts for the 2D convection-diffusion equation solved on equidistant grids.

	Pe:			
Grid	20	50	100	200
128^2	16	16	18	24
256^2	16	16	16	17
512^2	15	16	16	15

TABLE 7.3
FGMRES iteration counts for the 2D convection-diffusion equation preconditioned by diagonal scaling with local grid refinements near the upper wall.

	Pe:			
Grid	20	50	100	200
128^2	17	16	18	25
256^2	16	16	16	23

points. The convergence results from FGMRES are shown in Table 7.2. The (4,2,2,2) multilevel projection is performed. Furthermore, we set $\omega^{(\ell)} = 0.8$. The deflation subspace $Z^{(\ell,\ell+1)}$ is constructed by using (6.1) based on agglomeration of four neighboring points, and $Y^{(\ell,\ell+1)} = Z^{(\ell,\ell+1)}$. The norm of error between the solutions obtained by FGMRES and a direct method is of order 10^{-5}. From Table 7.2, we observe an almost h- and Pe-independent convergence. For $Pe = 200$ solved on the 128^2 mesh, convergence is reached after 24 iterations. In this case, however, nonphysical spurious wiggles appear in the solution at the vicinity of the upper wall ($y = +1$). Even though the solution is not correct due to these wiggles, the method is still convergent.

In order to get rid of wiggles, a grid refinement is employed near the wall $y = +1$. Table 7.3 shows the results after including the grid refinement. We still employ the same construction of Z as in the case without grid refinements, so we do not take into account the influence of grid stretching in the refinement zone. We have also used diagonal scaling as the additional preconditioner M. For all cases wiggles do not appear in the solutions. The norm of error between the solutions obtained by FGMRES and a direct method is of order 10^{-5}. The convergence is observed to be almost independent of h and Pe.

8. Conclusion. In this paper, a multilevel projection-based Krylov subspace method has been discussed. From an abstract point of view, the projection is done similarly to the way the deflation preconditioner is used for Krylov subspace methods. With this projection, some small eigenvalues are shifted to the largest one. We showed that the spectrum of the projected system is similar to the spectrum of the original system preconditioned by the deflation preconditioner. This projection, however, has been shown to be insensitive with respect to an inexact solve of the Galerkin (or coarse-grid) system. This allows the construction of multilevel projected Krylov subspace iterations, working on a sequence of Galerkin (coarse-grid) problems.

We claimed that the associated projection operator could not be decomposed into a sequence of smoothing and coarse-grid correction, typically used in multigrid and domain decomposition methods. Hence, the method presented here is not of the fixed point iteration type and must be considered solely from Krylov subspace iteration methods.

While we in general have freedom in choosing the deflation subspace (we need only the full rank condition to determine the deflation subspace), in this paper we used a simple way of constructing the deflation subspace, which was based on the piecewise constant interpolation. With this particular choice, the process could be seen as simply agglomerating values to the coarse grid, solving coarse-grid problems, and redistributing values from the coarse grid to the fine grid. We could obtain convergence, which is almost independent of mesh and physical parameters, for a class of problems considered in this paper. With the algebraic approach we have used, at this moment we could not explain this nice convergence property.

An application of our multilevel projection Krylov method with similar convergence behavior has also been done for the indefinite Helmholtz equation and is submitted for publication as well [4].

Finally, we state that this new multilevel Krylov method consists of several ingredients, such as a preconditioner for Krylov iterations, restriction and prolongation operators, approximation of the maximum eigenvalue, and approximation of the Galerkin matrix. In this paper, we have chosen a specific choice of all of these ingredients, some of which are integral parts of successful methods, such as multigrid (or domain decomposition) methods. Nevertheless, other choices or new developments in those methods can be easily implemented as well in our new multilevel Krylov framework (to obtain an even faster convergence).

REFERENCES

[1] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, J. Comput. Appl. Math., 123 (2000), pp. 261–292.

[2] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, 2005.

[3] Y. A. Erlangga and R. Nabben, *Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices*, SIAM J. Matrix Anal. Appl., to appear.

[4] Y. A. Erlangga and R. Nabben, *On a Multilevel Krylov Method for the Helmholtz Equation Preconditioned by the Shifted Laplacian*, submitted.

[5] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*, Freeman, San Francisco, 1963.

[6] J. Frank and C. Vuik, *On the construction of deflation-based preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp. 442–462.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[8] J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp., 216 (1996), pp. 1387–1401.

[9] J. Mandel, *Balancing domain decomposition*, Comm. Numer. Methods Engrg., 9 (1993), pp. 233–241.
[10] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[11] R. Nabben and C. Vuik, *A comparison of deflation and coarse grid correction applied to porous media flow*, SIAM J. Numer. Anal., 42 (2004), pp. 1631–1647.
[12] R. Nabben and C. Vuik, *A comparison of deflation and the balancing preconditioner*, SIAM J. Sci. Comput., 27 (2006), pp. 1742–1759.
[13] R. Nabben and C. Vuik, *A Comparison of Abstract Versions of Deflation, Balancing and Additive Coarse Grid Correction Preconditioners*, report 07-01, Delft University of Technology, Department of Applied Mathematical Analysis, Delft, 2007.
[14] R. A. Nicolaides, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.
[15] A. Padiy, O. Axelsson, and B. Polman, *Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 793–818.
[16] S. Röllin and W. Fichtner, *Accuracy and performance issues of spectral preconditioners in semiconductor device simulation*, in Proceedings of the European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006, P. Wesseling, E. Onate, and J. Periaux, eds., TU Delft, 2006.
[17] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 7 (1986), pp. 856–869.
[18] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halstead Press, New York, 1992.
[19] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[20] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, 1996.
[21] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga, *Theoretical and Numerical Comparison of Various Projection Methods Derived from Deflation, Domain Decomposition and Multigrid Methods*, report 07-04, Delft University of Technology, Delft Institute of Applied Mathematics, Delft, 2007.
[22] A. Toselli and O. Widlund, *Domain Decomposition Methods*, Springer-Verlag, Berlin, 2005.
[23] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Comput., 13 (1992), pp. 631–644.
[24] R. S. Varga, *Gerschgorin and His Circles*, Springer-Verlag, Berlin, 2004.
[25] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley and Sons, Chichester, 1991.
[26] P. Wesseling, *Principles of Computational Fluids Dynamics*, Springer-Verlag, Berlin, 2001.
[27] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1995.
[28] P. M. de Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 3 (1990), pp. 1–27.