Electronic Transactions on Numerical Analysis.
Volume 15, pp. 38-55, 2003.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

EFFICIENT SOLUTION OF SYMMETRIC EIGENVALUE PROBLEMS USING MULTIGRID PRECONDITIONERS IN THE LOCALLY OPTIMAL BLOCK CONJUGATE GRADIENT METHOD. *

ANDREW V. KNYAZEV† AND KLAUS NEYMEYR‡

Abstract. We present a short survey of multigrid–based solvers for symmetric eigenvalue problems. We concentrate our attention on "off the shelf" and "black box" methods, which should allow solving eigenvalue problems with minimal, or no, effort on the part of the developer, taking advantage of already existing algorithms and software. We consider a class of such methods, where the multigrid only appears as a black-box tool for constructing the preconditioner of the stiffness matrix, and the base iterative algorithm is one of well-known off-the-shelf preconditioned gradient methods such as the locally optimal block preconditioned conjugate gradient method. We review some known theoretical results for preconditioned gradient methods that guarantee the optimal, with respect to the grid size, convergence speed. Finally, we present results of numerical tests, which demonstrate practical effectiveness of our approach for the locally optimal block conjugate gradient method preconditioned by the standard V-cycle multigrid applied to the stiffness matrix.

Key words. symmetric eigenvalue problems, multigrid preconditioning, preconditioned conjugate gradient iterative method

AMS subject classifications. 65N25,65N55,65F15.

1. Introduction. At the end of the past Millennium, the multigrid technique has matured to the level of practical industrial applications. The progress in developing an algebraic multigrid has made possible implementing efficient algebraic multigrid preconditioners in commercial codes. Linear and nonlinear multigrid solvers have established a new high standard of effectiveness for the numerical solution of partial differential equations. Using multigrid for eigenvalue problems has also attracted significant attention.

Let us first present here a short and informal overview of different multigrid- based approaches for numerical solution of eigenvalue problems.

The most traditional multigrid approach to an eigenvalue problem is to treat it as a nonlinear equation and, thus, to apply a nonlinear multigrid solver; e.g., an FAS (full approximation scheme) [7], sometimes explicitly tuned for the eigenvalue computations, e.g., [4, 5, 24, 25, 26, 27, 28, 29, 30].

Such a solver can often be readily applied to nonlinear eigenvalue problems; see [10, 11] for the FAS applied to the nonlinear Schrödinger-Poisson equations and [5] for a homotopy continuation methods. For linear eigenvalue problems, this technique may not be always efficient. It may not be able to take advantage of specific properties of eigenvalue problems, if a general nonlinear solver is used. Tuning a nonlinear code for eigenvalue computations may, on the other hand, require elaborate programming. Nonlinear multigrid eigensolvers often involve special treatment of eigenvalue clusters, which further complicates their codes. Another known drawback is high approximation requirements on coarse grids, when a nonlinear multigrid solver needs accurate eigenfunctions on coarse grids, cf. [26].

The second popular approach is to use an outer eigenvalue solver, e.g., the Rayleigh quotient iteration (RQI) [43], which requires solving linear systems with a shifted stiffness matrix. The multigrid is used as an inner solver in such inner-outer iterations. The Newton

*Received May 18, 2001. Accepted for publication September 12, 2001. Recommended by Joe Pasciak.
†Department of Mathematics, University of Colorado at Denver, P.O. Box 173364, Campus Box 170, Denver, CO 80217- 3364. *WWW URL*: http://www-math.cudenver.edu/~aknyazev *e-mail*: andrew.knyazev@cudenver.edu
‡Mathematisches Institut der Universität Tübingen, Auf der Morgenstelle 10, 72076 Tübingen, Germany *e-mail*: neymeyr@na.uni-tuebingen.de

ETNA
Kent State University
etna@mcs.kent.edu

method for the Rayleigh quotient [1] or a homotopy continuation [40, 60] can be used here as an outer method instead of RQI, which allows one to consider such methods also in the first category, i.e. nonlinear solvers. The use of the shifted stiffness matrix typically leads to a very fast convergence of the outer iterative solver provided a high accuracy of the inner iterations, but requires a special attention as the linear systems, solved by multigrid, are nearly singular, e.g., [9]. In the well-known Jacobi–Davidson method [48, 52], this difficulty is somewhat circumvented by introducing a correction equation, which can be solved by multigrid [27]. The problem of choosing a proper stopping criteria for inner multigrid iterations is nontrivial as the choice may significantly affect the effectiveness of the inner-outer procedure.

More complex variants of the multigrid shift-and-invert methods, where the grids are changed in outer iterations as well, are known, e.g., [56]. In [57, 58], such a multigrid technique is adopted for so–called source iteration methods of solving spectral problems for systems of multigroup diffusion equations describing the steady state of nuclear reactors, and its implementation for the popular Russian WWER and BN reactors is described.

A very similar technique is to use the classical inverse iterations, often inverse subspace iterations, as an outer solver and the multigrid as an inner solver for systems with the stiffness matrix, e.g., [3, 42]. The linear convergence of the traditional inverse iterations is known to be slower compared to usually super-linear, e.g., cubic in some cases, convergence of the shift-and-invert iterations discussed in the previous paragraph, but one does not face now the troubles of nearly singular linear systems. Moreover, when the systems are solved inaccurately by inner iterations, which is desired to reduce the number of inner steps, the fast convergence of shift-and-invert outer iterations is typically lost. In this case, the convergence speed of outer iterations based on the inverse of the stiffness matrix and on the shift-and-invert techniques may be comparable; e.g., see [53] for comparison of inexact inverse and Rayleigh quotient iterations.

There are other methods, specifically designed multigrid eigensolvers, that do not fit our classification above, e.g., multilevel minimization of the Rayleigh quotient [9, 41], called the Rayleigh quotient multigrid (RQMG). This method has been integrated into an adaptive 2D Helmholtz eigensolver used for designing integrated optical chips [12, 13].

A class of methods we want to discuss in details in the present paper has two distinctive features: there are no inner-outer iterations and the multigrid can be used only as a preconditioner, not as a complete linear solver, nor as an eigensolver. These methods are usually called *preconditioned gradient methods* as historically the first method of this kind, suggested in [51], minimizes the Rayleigh quotient on every iteration in the direction of the preconditioned gradient. They have been studied mostly in the Russian literature; see, e.g., [17, 21, 23, 32, 34] as well as the monograph [18] and a recent survey [35], which include extensive bibliography. The most promising preconditioned gradient eigensolvers, the locally optimal preconditioned conjugate gradient (LOPCG) method and its block version (LOBPCG), suggested and analyzed in [33, 34, 35, 36], shall play the main role in the present paper.

Let us briefly mention here some relevant and even more recent results for symmetric eigenproblems. Paper [49] obtains asymptotic convergence rate estimate of the generalized Davidson method similar to that by [51] for the preconditioned steepest descent. In [44, 45], the second author of the present paper derives the first *sharp nonasymptotic* convergence rate estimates for the simplest method in the class, the preconditioned power method with a shift, called in [44, 45] the preconditioned inverse iteration (PINVIT) method. In [46], these estimates are generalized to similar subspace iterations. The most recent work, [37], written by the authors of the present paper, finds a short and elegant simplification of estimates of [44, 45, 46] and extends the results to generalized symmetric eigenproblems. We shall

ETNA
Kent State University
etna@mcs.kent.edu

describe one of the main estimates of [37] in Section 4.

Preconditioned gradient methods can be easily linked to the inexact inner-outer iterative solvers discussed earlier in the section. PINVIT is interpreted in [44, 45] as a perturbation of the classical inverse iterations. The same method can be also viewed as an inexact RQI with only one step of a preconditioned linear iterative solver performed on every outer iteration, e.g., [32, 34].

Preconditioned gradient methods for computing several extreme eigenpairs of symmetric eigenvalue problems have several advantages compared to other preconditioned eigensolvers:

- algorithmic simplicity;
- low costs per iteration and minimal memory requirements;
- practical robustness with respect to initial guesses;
- robust computation of eigenvalue clusters as a result of using block methods as that by the classical inverse subspace iterations;
- fast convergence, comparable to that of the block Lanczos applied to $A^{-1}B$, when a good preconditioner and an advanced preconditioned gradient method, e.g., the LOBPCG method, are used;
- designed to operate in a matrix-free environment and equally efficient for regular, $Ax = \lambda x$, and generalized, $Ax = \lambda Bx$, symmetric eigenvalue problems;
- relatively well developed theory, which, in particular, guarantees optimal, with respect to the mesh size, convergence with multigrid preconditioning;
- trivial multigrid implementation, e.g., simply by using black-box multigrid preconditioning of the stiffness matrix.

The main drawback of present preconditioned gradient methods is their inability to compute efficiently eigenvalues in a given interval in the middle of the spectrum. For this and nonsymmetric cases, we refer the reader to preconditioned shift-and-invert methods reviewed in [2].

Multigrid preconditioned gradient methods are popular for applications, e.g., in structural dynamics [8] and in electronic structure calculations for carbon nanotubes [22], where two slightly different versions of the block preconditioned steepest descent are used.

Preconditioning for gradient methods is not limited to multigrid, e.g., band structure calculations in two- and three-dimensional photonic crystals in [15, 16] are performed by using the fast Fourier transform for the preconditioning.

Some eigenvalue problems in mechanics, e.g., vibration of a beam supported by springs, lead to equations with nonlinear dependence on the spectral parameter. Preconditioned eigensolvers for such equations are analyzed in [54, 55], where, in particular, a generalization of the theory of a preconditioned subspace iteration method of [19, 20] is presented.

Here, we use multigrid only as a preconditioner so that all iterations are performed on the finest grid. This implementation separates completely iterative methods from the grid structure, which significantly simplifies the code and, we repeat, allows the use of black-box multigrid, e.g., one of algebraic multigrid preconditioners. Our recommended choice to precondition the stiffness matrix *without any shifts* further simplifies the algorithm as it removes a possibility of having singularities typical for preconditioning of shifted stiffness matrices.

In research codes, well-known multigrid tricks could also be used, such as nested iterations, where first iterations are performed on coarser grids and the grid is refined with the number of iterations, e.g., [18]. We note, however, that doing so would violate the off-the-shelf and black-box concepts, thus, significantly increasing the code complexity. Nevertheless, adaptive mesh refinement strategies are available for the eigenproblem [47] where the adaptive eigensolver is based on the finite element code KASKADE [14]. The efficiency of

ETNA
Kent State University
etna@mcs.kent.edu

such schemes has been demonstrated in [12, 39].

Let us finally highlight that in the multigrid preconditioned gradient eigensolvers we recommend, no eigenvalue problems are solved on coarse grids; thus, there are no limitations on the coarse grid size associated with approximation of eigenfunctions.

The rest of the paper is organized as follows. Section 2 introduces the notation and presents main assumptions. In Section 3, we describe the LOBPCG method of [33, 34, 35, 36], and a simpler, but slower, method, the block preconditioned steepest descent (BPSD), analyzed in [6]. Section 4 reproduces the most recent convergence rate estimate for the two methods, derived in [37].

Numerical results for a model problem, the Laplacian on the unit square, using a few typical choices of the multigrid preconditioning are given in Section 5. We compare several preconditioned eigensolvers: PSD, LOPCG and their block versions. These numerical experiments provide clear evidence for regarding LOBPCG as practically the optimal scheme (within that class of preconditioned eigensolvers we consider). Moreover, we compare the LOBPCG method with the preconditioned *linear solvers*: the standard preconditioned conjugate gradient (PCG) method for $Ax = b$ and the PCG applied to computing a null–space of $A-\lambda_{\min}B$ (see our notation in the next section), called PCGNULL in [36]. The latter experiments suggest that LOBPCG is a *genuine conjugate gradient* method.

2. Mesh symmetric eigenvalue problems. We consider a generalized *symmetric positive definite* eigenvalue problem of the form $(A-\lambda B)x = 0$ with real symmetric positive definite n-by-n matrices A and B. That describes a regular matrix pencil $A-\lambda B$ with a discrete spectrum (set of eigenvalues λ). It is well known that the problem has n real positive eigenvalues

$$0 < \lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = \lambda_{\max},$$

and corresponding (right) eigenvectors x_i, satisfying $(A-\lambda_i B)x_i = 0$, which can be chosen orthogonal in the following sense: $(x_i, Ax_j) = (x_i, Bx_j) = 0,\ i \neq j$.

In our notation, A is the stiffness matrix and B is the mass matrix. In some applications, the matrix B is simply the identity, $B = I$, and then we have the standard symmetric eigenvalue problem with matrix A. For preconditioned gradient eigensolvers, it is not important whether the matrix B is diagonal; thus, a mass condensation in the finite element method (FEM) is not necessary.

We consider the problem of computing m smallest eigenvalues λ_i and the corresponding eigenvectors x_i.

Let T be a real symmetric positive definite n-by-n matrix. T will play the role of the preconditioner; more precisely, an application of a preconditioner to a given vector x must be equivalent to the matrix vector product Tx. In many practical applications, T, as well as A and B, is not available as a matrix, but only as a function performing Tx, i.e., we operate in a matrix–free environment. The assumption that T is positive definite is crucial for the theory, but T does not have to be fixed, i.e. it may change from iteration to iteration. This flexibility may allow us to use a wider range of smoothers in multigrid preconditioning.

When solving a linear system with the stiffness matrix, $Ax = b$, it is evident that the preconditioner T should approximate A^{-1}. For eigenvalue problems, it is not really clear what the optimal target for the preconditioning should be. If one only needs to compute the smallest eigenvalue λ_1, one can argue [34] that the optimal preconditioner would be the pseudoinverse of $A-\lambda_1 B$.

Choosing the stiffness matrix A as the target for preconditioning in mesh eigenvalue problems seems to be a reasonably practical compromise, though it may not be the optimal choice from a purely mathematical point of view. On the one hand, it does provide, as we

ETNA
Kent State University
etna@mcs.kent.edu

shall see in Section 4, the optimal convergence in a sense that the rate of convergence does not deteriorate when the mesh gets finer. On the other hand, it simplifies the theory of block preconditioned eigensolvers and streamlines the corresponding software development.

To this end, let us assume that

$$\delta_0(x,Tx) \le (x,A^{-1}x) \le \delta_1(x,Tx), \forall x \in R^n, 0<\delta_0 \le \delta_1. \tag{2.1}$$

The ratio δ_1/δ_0 can be viewed as the spectral condition number $\kappa(TA)$ of the preconditioned matrix TA and measures how well the preconditioner T approximates, up to a scaling, the matrix A^{-1}. A smaller ratio δ_1/δ_0 typically ensures faster convergence. For mesh problems, matrices A^{-1} and T are called *spectrally equivalent* if the ratio is bounded from above uniformly in the mesh size parameter; see [18]. For variable preconditioners, we still require that all of them satisfy (2.1) with the same constants.

3. Algorithms: the block preconditioned steepest descent and the locally optimal block preconditioned conjugate gradient method. In the present paper, we shall consider two methods: the preconditioned steepest descent (PSD) and the locally optimal preconditioned conjugate gradient method (LOPCG) as well as their block analogs: the block preconditioned steepest descent (BSPD) and the locally optimal block preconditioned conjugate gradient method (LOBPCG), in the form they appear in [34, 36].

For brevity, we reproduce here the algorithms in the block form only. The single-vector form is simply the particular case $m=1$, where m is the block size, which equals to the number of sought eigenpairs.

First, we present Algorithm 3.1 of the BPSD method, which is somewhat simpler.

ALGORITHM 3.1 : **BPSD.**
Input: *m starting vectors $x_1^{(0)},\dots x_m^{(0)}$, functions to compute: matrix-vector products Ax, Bx and Tx for a given vector x and the vector inner product (x,y).*

1. *Start: select $x_j^{(0)}$, and set $p_j^{(0)}=0$, $j=1,\dots,m$.*
2. *Iterate: For $i=0,\dots,$ Until Convergence Do:*
3. $\lambda_j^{(i)} := (x_j^{(i)},Ax_j^{(i)})/(x_j^{(i)},Bx_j^{(i)})$, $j=1,\dots,m$;
4. $r_j := Ax_j^{(i)} - \lambda_j^{(i)}Bx_j^{(i)}$, $j=1,\dots,m$;
5. $w_j^{(i)} := Tr_j$, $j=1,\dots,m$;
6. *Use the Rayleigh–Ritz method for the pencil $A-\lambda B$ on the trial subspace $\mathrm{span}\{w_1^{(i)},\dots,w_m^{(i)},x_1^{(i)},\dots,x_m^{(i)}\}$ to compute the next iterate $x_j^{(i+1)}$ as the j-th Ritz vector corresponding to the j-th smallest Ritz value, $j=1,\dots,m$;*
7. *EndDo*

Output: *the approximations $\lambda_j^{(i+1)}$ and $x_j^{(i+1)}$ to the smallest eigenvalues λ_j and corresponding eigenvectors, $j=1,\dots,m$.*

A different version of the preconditioned block steepest descent is described in [8], where the Rayleigh–Ritz method on step 6 is split into two parts, similar to that of the LOBPCG II method of [36] which we discuss later. Other different versions are known, e.g., the successive eigenvalue relaxation method of [50]. Our theory replicated in the next section covers only the version of the preconditioned block steepest descent of Algorithm 3.1.

Our second Algorithm 3.2 of the LOBPCG method is similar to Algorithm 3.1, but utilizes an extra set of vectors, analogous to conjugate directions used in the standard preconditioned conjugate gradient linear solver.

ALGORITHM 3.2 : **LOBPCG.**

Input: *m starting vectors $x_1^{(0)}, \ldots x_m^{(0)}$, functions to compute: matrix-vector products Ax, Bx and Tx for a given vector x and the vector inner product (x,y).*

1. *Start: select $x_j^{(0)}$, and set $p_j^{(0)} = 0,\ j = 1, \ldots, m$.*
2. *Iterate: For $i = 0, \ldots,$ Until Convergence Do:*
3. $\lambda_j^{(i)} := (x_j^{(i)}, Ax_j^{(i)})/(x_j^{(i)}, Bx_j^{(i)}),\ j = 1, \ldots, m;$
4. $r_j := Ax_j^{(i)} - \lambda_j^{(i)} Bx_j^{(i)},\ j = 1, \ldots, m;$
5. $w_j^{(i)} := Tr_j,\ j = 1, \ldots, m;$
6. *Use the Rayleigh–Ritz method for the pencil $A - \lambda B$ on the trial subspace $\mathrm{span}\{w_1^{(i)}, \ldots, w_m^{(i)}, x_1^{(i)}, \ldots, x_m^{(i)}, p_1^{(i)}, \ldots, p_m^{(i)}\}$ to compute the next iterate $x_j^{(i+1)} := \sum_{k=1,\ldots,m} \alpha_k^{(i)} w_k^{(i)} + \tau_k^{(i)} x_k^{(i)} + \gamma_k^{(i)} p_k^{(i)}$ as the j-th Ritz vector corresponding to the j-th smallest Ritz value, $j = 1, \ldots, m$;*
7. $p_j^{(i+1)} := \sum_{k=1,\ldots,m} \alpha_k^{(i)} w_k^{(i)} + \gamma_k^{(i)} p_k^{(i)};$
8. *EndDo*

Output: *the approximations $\lambda_j^{(i+1)}$ and $x_j^{(i+1)}$ to the smallest eigenvalues λ_j and corresponding eigenvectors, $j = 1, \ldots, m$.*

Here, on Step 6 the scalars $\alpha_k^{(i)}$, $\tau_k^{(i)}$, and $\gamma_k^{(i)}$ are computed implicitly as components of the corresponding eigenvectors of the Rayleigh–Ritz procedure.

We want to highlight that the main loop of Algorithm 3.2 of LOBPCG can be implemented with only one application of the preconditioner T, one matrix-vector product Bx and one matrix-vector product Ax, per iteration; see [36] for details.

Storage requirements are small in both Algorithms 3.1 and 3.2: only several n-vectors, and no n-by-n matrices at all. Such methods are sometimes called matrix-free.

For the stopping criterion, we compute norms of the preconditioned residual $w^{(i)}$ on every iteration. Residual norms provide accurate two-sided bounds for eigenvalues and *a posteriori* error bounds for eigenvectors; see [32].

A different version of the LOBPCG, called LOBPCG II, is described in [36], where the Rayleigh–Ritz method on step 6 of Algorithm 3.2 is split into two parts. There, on the first stage, the Rayleigh–Ritz method is performed m times on three-dimensional trial subspaces, spanned by $w_j^{(i)}$, $x_j^{(i)}$, and $p_j^{(i)}$, $j = 1, \ldots, m$, thus, computing m approximations to the eigenvectors. In the second stage, the Rayleigh–Ritz method is applied to the m-dimensional subspace, spanned by these approximations. In this way, the Rayleigh–Ritz method is somewhat less expensive and can be more stable, as the dimension of the large trial subspace is reduced from $3m$ to m. We do not yet have enough numerical evidence to suggest using LOBPCG II widely, even though a similar preconditioned block steepest descent version of [8] is successfully implemented in an industrial code Finite Element Aggregation Solver (FEAGS).

Other versions of LOBPCG are possible, e.g., the successive eigenvalue relaxation technique of [50] can be trivially applied to the LOBPCG. However, only the original version of the LOBPCG, shown in Algorithm 3.2, is supported by our theory of the next section.

Comparing the BPSD and LOBPCG algorithms, one realizes that the only difference is that the LOBPCG uses an extra set of directions $p_j^{(i)}$ in the trial subspace of the Rayleigh–Ritz method. In single-vector versions PSD and LOPCG, when $m = 1$, this leads to a difference between using two and three vectors in the iterative recursion. Such a small change results in considerable acceleration, particularly significant when the preconditioner is not of a high

ETNA
Kent State University
etna@mcs.kent.edu

quality, as demonstrated numerically in [34, 35]. In Section 5, we observe this effect again in our numerical tests with multigrid preconditioning.

A seemingly natural idea is to try to accelerate the LOPCG and LOBPCG by adding more vectors to the trial subspace. Let us explore this possibility for the single vector ($m = 1$) method, LOPCG, by introducing a group of methods we call LOPCG+k, $k \geq 1$ in Section 5. It is explained in [36] that in our Algorithm 3.2 with $m = 1$ the trial subspace can be written in two alternative forms:

$$\operatorname{span}\left\{w^{(i)}, x^{(i)}, p^{(i)}\right\} = \operatorname{span}\left\{w^{(i)}, x^{(i)}, x^{(i-1)}\right\}.$$

Here and until the end of the section we drop the lower index $j = 1$ for simplicity of notation. The former formula for the subspace, used in Algorithm 3.2, is more stable in the presence of round-off errors. The latter formula is simpler and offers an insight for a possible generalization: for a fixed $k \geq 1$ we define the method LOPCG+k by using the following extended trial subspace

$$\operatorname{span}\left\{w^{(i)}, x^{(i)}, x^{(i-1)}, \ldots, x^{(i-1-k)}\right\}$$

in the Rayleigh-Ritz method. When k is increased, the trial subspace gets larger, thus providing a potential for an improved accuracy.

We test numerically LOPCG+k methods for a few values of k, see Section 5, and observe no noticeable accuracy improvement at all. Based on these numerical results, we come to the same conclusion as in [36], namely, that the LOPCG method is apparently the optimal preconditioned eigensolvers; in particular, it cannot be significantly accelerated by adding more vectors to the trial subspace.

This conclusion, however, is not yet supported by a rigorous theory. In the next section, we discuss known theoretical results for BPSD and LOBPCG methods.

4. Available theory. The following result of [37] provides us with a short and elegant convergence rate estimate for Algorithms 3.1 and 3.2.

THEOREM 4.1. *The preconditioner is assumed to satisfy (2.1) on every iteration step. For a fixed index $j \in [1, m]$, if $\lambda_j^{(i)} \in [\lambda_{k_j}, \lambda_{k_j+1}[$ then it holds for the Ritz value $\lambda_j^{(i+1)}$ computed by Algorithm 3.1, or 3.2, that either $\lambda_j^{(i+1)} < \lambda_{k_j}$ (unless $k_j = j$), or $\lambda_j^{(i+1)} \in [\lambda_{k_j}, \lambda_j^{(i)}[$.*

In the latter case,

$$\frac{\lambda_j^{(i+1)} - \lambda_{k_j}}{\lambda_{k_j+1} - \lambda_j^{(i+1)}} \leq \left(q\left(\kappa(TA), \lambda_{k_j}, \lambda_{k_j+1}\right)\right)^2 \frac{\lambda_j^{(i)} - \lambda_{k_j}}{\lambda_{k_j+1} - \lambda_j^{(i)}}, \tag{4.1}$$

where

$$q\left(\kappa(TA), \lambda_{k_j}, \lambda_{k_j+1}\right) = 1 - \left(1 - \frac{\kappa(TA) - 1}{\kappa(TA) + 1}\right)\left(1 - \frac{\lambda_{k_j}}{\lambda_{k_j+1}}\right). \tag{4.2}$$

In the context of multigrid preconditioning, the most important feature of this estimate is that is guarantees the optimal, with respect to the mesh size, convergence rate, provided that $\kappa(TA)$ is uniformly bounded from above in the mesh size parameter h and that the eigenvalues of interest do not contain clusters, so that $\lambda_{k_j} - \lambda_{k_j+1}$ is large compared to h. Estimate (4.1) can be trivially modified to cover the case of multiple eigenvalues.

For the block case, $m > 1$, there is only one other comparable nonasymptotic convergence rate result, but it is for a simpler method, see [6]. This estimate is not applicable to BPSD and LOBPCG as it cannot be used recursively.

For the single vector case, $m = 1$, which is, of course, much simpler, more convergence results are known, e.g., see [32, 34, 35, 36, 37] and references there. In particular, an estimate of [31, 32] for the PSD is in some cases, e.g., for high quality preconditioners, sharper than that of Theorem 4.1. Asymptotically this estimate is similar to the estimate of [51]. We present it in the next section for comparison.

Advantages of the Theorem 4.1 are that:

- it is applicable to any initial subspaces,
- the convergence rate estimate can be used recursively,
- the estimates for the Ritz values are individually sharp for the most basic PINVIT scheme; see [37] for details,
- the convergence rate estimate for a fixed index j is exactly the same as for the single–vector scheme, cf. [37].

One serious disadvantage of the estimate (4.2) is that it deteriorates when eigenvalues of interest $\lambda_1, \ldots, \lambda_m$ include a cluster. The actual convergence of Algorithms 3.1 and 3.2 in numerical tests is known not to be sensitive to clustering of eigenvalues, and the estimate of [6] does capture this property, essential for subspace iterations.

Theorem 4.1 provides us with the only presently known nonasymptotic theoretical convergence rate estimate of the LOBPCG with $m > 1$. Numerical comparison of PINVIT, PSD and LOBPCG according to [34, 35, 36] demonstrates, however, that the LOBPCG method is much faster in practice. Therefore, our theoretical convergence estimate of Theorem 4.1 is not sharp enough yet to explain excellent convergence properties of the LOBPCG in numerical simulations, which we illustrate next.

5. A numerical example. We consider an eigenproblem for the Laplacian on $[0,\pi]^2$ with the homogeneous Dirichlet boundary conditions. The problem is discretized by using linear finite elements on a uniform triangle mesh with the grid parameter $h = \pi/64$ and 3969 inner nodes. The discretized eigenproblem is a generalized matrix eigenvalue problem for the pencil $A - \lambda B$, where A is the stiffness matrix for the Laplacian and B is the mass matrix.

Our goal is to test PSD, BPSD, LOPCG and LOBPCG methods using multigrid preconditioners for the stiffness matrix A:

- $V(i,i)$–cycle preconditioners performing each i steps of Gauss–Seidel symmetric pre- and post-smoothing (alternatively Jacobi–smoothing) on a hierarchy of grids $h_l = \pi/2^l$, $l = 2, \ldots, 6$, with the exact solution on the coarsest grid,
- a hierarchical basis (HB) preconditioner [59] for h_l, $l = 1, \ldots, 6$, so that the coarsest finite element space consists only of a single basis function.

5.1. Comparison of convergence factors. To begin with, we compare the computed convergence factors for the schemes PSD, LOPCG, PCGNULL and PCG, which are each started with the same initial vectors out of 200 randomly chosen. By PCGNULL we denote the standard PCG method applied to the singular system of linear equations

$$(A - \lambda_1 B)x_1 = 0,$$

where we suppose λ_1 to be given; i.e., we compute the eigenvector x_1 as an element of the null–space of $A - \lambda_1 B$ by PCG. PCGNULL was suggested by the first author of the present paper in [36] as a benchmark. For the PCG runs, when solving $Ax = b$, we choose random right–hand sides b.

In the following we apply the *same* preconditioners to the eigensolvers and to linear solvers PCGNULL and PCG; and we consider only preconditioners for the stiffness matrix

	(4.2)	(5.3)	PSD	(5.4)	PCGNULL	LOPCG	PCG for $Ax = b$
$V(2,2)$	0.43	0.30	0.26	0.15	0.13	0.13	0.03
$V(1,1)$	0.48	0.36	0.29	0.19	0.17	0.16	0.06
HB	0.96	0.96	0.9	0.76	0.7	0.7	0.7

TABLE 5.1
Actual and theoretical convergence factors.

A. We test the $V(i,i)$–cycle preconditioners using Gauss–Seidel smoothing for $i = 1,2$ and the HB preconditioner.

The iterations of PSD and LOBPCG are stopped if $\lambda_1^{(i)} - \lambda_1$ is less than 10^{-8}. In all our tests they converge for each random initial vector tried to the extreme eigenpair (u_1,λ_1). In other words, iterations do not get stuck in eigenspaces corresponding to higher eigenvalues. The schemes PCGNULL and PCG are stopped if $(r,Tr) < 10^{-10}$, where r denotes the actual residual vector: $r = (A-\lambda_1 B)x$ in PCGNULL and $r = Ax - b$ in PCG.

The computed convergence factors and corresponding theoretical estimates are listed in Table 5.1.

The convergence factors for PSD and LOPCG in Table 5.1 are the mean values of all convergence factors

$$\sqrt{\frac{\lambda_1^{(i+1)} - \lambda_1}{\lambda_2 - \lambda_1^{(i+1)}} \frac{\lambda_2 - \lambda_1^{(i)}}{\lambda_1^{(i)} - \lambda_1}}$$

(for all 200 initial vectors) computed from the numerical data recorded when $\lambda < \lambda_2$. The convergence factors for PCGNULL and PCG in Table 5.1 are computed by calculating the ratio of Euclidean norms of the initial and the final residuals and then taking the average ratio per iteration.

The theoretical convergence factor q given by formula (4.2) of Theorem 4.1 is computed as follows. We first roughly estimate the spectral condition number of TA using the actual convergence factors of PCG. Let q_{PCG} be the convergence factor of PCG as given in Table 5.1. From the standard PCG convergence rate estimate we get

$$q_{PCG} \approx \frac{\sqrt{\kappa(TA)} - 1}{\sqrt{\kappa(TA)} + 1}; \tag{5.1}$$

therefore, we take

$$\kappa(TA) := \left(\frac{1 + q_{PCG}}{1 - q_{PCG}}\right)^2. \tag{5.2}$$

Then, inserting this $\kappa(TA)$ and $\lambda_1 = 2$ and $\lambda_2 = 5$ in (4.2) gives the corresponding column of Table 5.1.

We also provide two other theoretical convergence rate factors, for the PSD and the PCGNULL, correspondingly. They are based on the spectral condition number $\kappa(T(A-\lambda_1 B))$, which is defined as a ratio of the largest and the smallest nonzero eigenvalues of the matrix $T(A - \lambda_1 B)$. Namely, the asymptotic convergence rate factor of the PSD is

$$\frac{\kappa(T(A-\lambda_1 B)) - 1}{\kappa(T(A-\lambda_1 B)) + 1}, \tag{5.3}$$

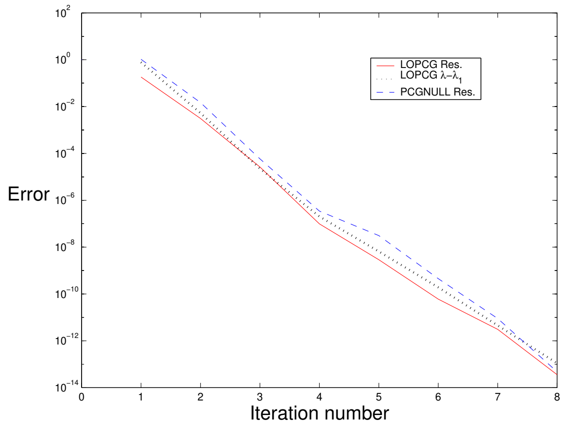

FIG. 5.1. *LOPCG vs. PCGNULL.*

see [31, 32, 51], while the convergence rate factor of the PCGNULL per iteration is approximately equal [36] to

$$\frac{\sqrt{\kappa(T(A-\lambda_1 B))}-1}{\sqrt{\kappa(T(A-\lambda_1 B))}+1}. \tag{5.4}$$

According to Theorem 3.1 of [32],

$$\kappa(T(A-\lambda_1 B)) \le \kappa(TA)\left(1-\frac{\lambda_1}{\lambda_2}\right)^{-1}. \tag{5.5}$$

Thus, knowing the values $\kappa(TA)$ from (5.2) and $\lambda_1 = 2$, $\lambda_2 = 5$ we compute convergence factors (5.3) and (5.4) and present them in Table 5.1.

Let us now discuss the numerical data of Table 5.1. We first observe that our theoretical convergence factor q given by (4.2) of Theorem 4.1 does provide an upper bound for the actual convergence factors, thus supporting the statement of Theorem 4.1. However, this upper bound is clearly pessimistic even for the PSD. The asymptotic convergence rate factor (5.3) fits better the actual convergence rate of the PSD. Let us notice, however, comparing the first two lines with the data that an improvement of the quality of the preconditioner from $V(1,1)$ to $V(2,2)$ reduces the value of q of (4.2) and accelerates the actual convergence of the PSD 1.1 times, while the value of (5.3) gets smaller 1.2 times. This observation suggests that neither (4.2), nor (5.3) captures accurately the actual convergence behavior of the PSD. The problem of obtaining a sharp nonasymptotic convergence rate estimate of the PSD remains open, except for the case $T = A^{-1}$, which we discuss at the end of the subsection.

Equation (5.4) provides an accurate estimate of the convergence rate of the PCGNULL, as expected.

The convergence factors of LOPCG and those of PCGNULL are nearly identical in Table 5.1. This supports the supposition of [33] that the convergence rate factor of LOPCG depends on $\kappa(TA)$ the same way as the convergence rate factor (5.4) of PCGNULL; see [36] for extensive numerical comparison of LOPCG and PCGNULL.

A direct comparison, as presented in Table 5.1, of the convergence factors of LOPCG with those of PCGNULL must be done with care as we tabulate different quantities: the square root of ratios of differences of eigenvalue approximations for the former, but the ratios of the residuals for the latter. We scrutinize this potential discrepancy by a direct comparison

ETNA
Kent State University
etna@mcs.kent.edu

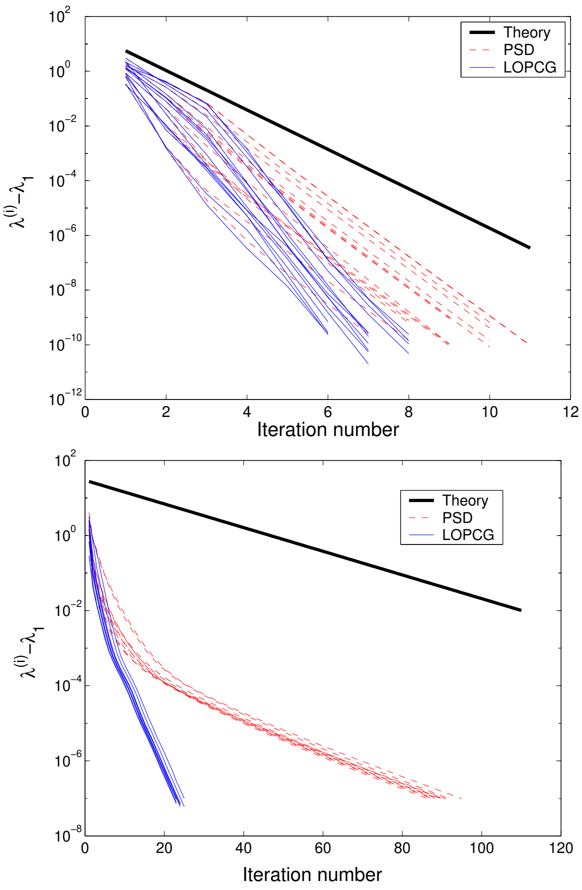

FIG. 5.2. PSD and LOPCG with $V(2,2)$–Gauss–Seidel (left) and HB (right) preconditioning.

of the convergence history lines for LOPCG and PCGNULL on Figure 5.1. Both schemes with the same V(2,2) Gauss–Seidel preconditioner are applied to the same initial guess, in this case simply a vector with all components equal to one. For LOPCG we draw the error $\lambda_1^{(i)} - \lambda_1$ as well as the square of the Euclidean norm of the residual of the actual eigenvector approximation and for PCGNULL only the square of the Euclidean norm of the residual. We observe not only a similar convergence speed but a striking correspondence of the error history lines. This confirms a conclusion of [36], which can also be drawn from Table 5.1, that the LOPCG appears as a *genuine conjugate gradient* method.

Not surprisingly, knowing results of numerical tests of [34, 35, 36], LOPCG converges significantly faster than PSD, according to Table 5.1. We additionally illustrate this on Figure 5.2.

Figure 5.2 displays the convergence history of PSD and LOPCG using the $V(2,2)$ Gauss–Seidel and HB preconditioners. Therein, $\lambda_1^{(i)} - \lambda_1$ is plotted versus the iteration index for 15 different randomly chosen initial vectors. The slope of the bold line in Figure 5.2 is determined by (4.1) for q as given in the (4.2) column of Table 5.1, i.e., we have drawn $q^{2i}(\lambda_2 - \lambda_1)$ against the iteration index i.

Looking at the last column of Table 5.1, we notice that the PCG for the linear system $Ax = b$ converges much faster than the PCGNULL for $V(2,2)$ and $V(1,1)$ preconditioners, but about with the same speed when the HB preconditioner is used. The reason for this is that the preconditioner T here is chosen to approximate the stiffness matrix A without any shifts, for the reasons already discussed in the introduction. Therefore, according to (5.5), PCG should always converge faster than the PCGNULL, but it would be mostly noticeable for preconditioner of a high quality. This is exactly the behavior of actual convergence factors of PCGNULL and PCG in Table 5.1.

Our final comments on Table 5.1 concern the data in the raw corresponding to the $V(2,2)$ preconditioner. For our problem, this preconditioner provides an excellent approximation to the stiffness matrix with $\kappa(TA) \approx 1.1$; thus, practically speaking, in this case we have $T \approx A^{-1}$ up to scaling. If $T = A^{-1}$ we have additional theoretical convergence rate estimates for eigensolvers we can compare with.

The PSD method with $T = A^{-1}$ is studied in details in [38], where a sharp convergence factor is obtained. In our notation that is

$$\frac{1-\xi}{1+\xi} = .25, \text{ where } \xi = 1 - \frac{\lambda_1}{\lambda_2} = 0.6. \tag{5.6}$$

Let us note that the earlier presented asymptotic PSD convergence factor (5.3) turns into (5.6), when $\kappa(TA) = 1$. As $\kappa(TA) \approx 1.1$ for the $V(2,2)$ preconditioner, we get the value .25 from (5.6) consistent with the actual PSD convergence factor, which is in this case .26, and with the value .3 of (5.3).

The LOPCG method is not yet theoretically investigated even with $T = A^{-1}$. Instead, let us show here the convergence factor of the classical Lanczos method, applied to find the smallest eigenvalue of $A^{-1}B$. The standard estimate, based on Chebyshev polynomials, gives the following convergence factor per iteration:

$$\frac{1-\sqrt{\xi}}{1+\sqrt{\xi}} = 0.13. \tag{5.7}$$

This is a perfect fit with the actual convergence factor of the LOPCG, which suggests that LOPCG is a natural extension of the Lanczos method in the class of preconditioned eigensolvers.

5.2. The optimal convergence of LOPCG. Next, we compare results of PSD, LOPCG and LOPCG+k, where we use the $V(2,2)$–cycle preconditioner with two steps of Jacobi pre- and post-smoothing each.

Figure 5.3 displays the error $\lambda^{(i)} - \lambda_1$ of the computed eigenvalue approximations $\lambda^{(i)}$ versus the iteration number i. Each curve represents the case of the poorest convergence toward λ_1 for 100 random initial vectors, the same for each scheme. The relatively poor convergence in the first steps accounts for attraction to eigenvalues different from λ_1, when $\lambda^{(i)} > \lambda_2$. The bold straight line is drawn based on the theoretical convergence factor q given by (4.2) of Theorem 4.1 in an analogous way as that described above.

The outcome of this experiment exemplifies that:

- The convergence factor q of (4.2) is a pessimistic upper bound for PSD and LOPCG.
- PSD is slower than LOPCG.
- Most importantly, LOPCG appears as the optimal scheme of those tested, since the slope of the convergence curves for LOPCG+k, $k = 1,2,3$, is approximately the same as the one of LOPCG, but LOPCG+k, $k > 0$ methods are more expensive as they involve optimization over larger trial subspaces.

The optimality of LOPCG is described by the first author of the present paper in [33, 36] with more details.

ETNA
Kent State University
etna@mcs.kent.edu

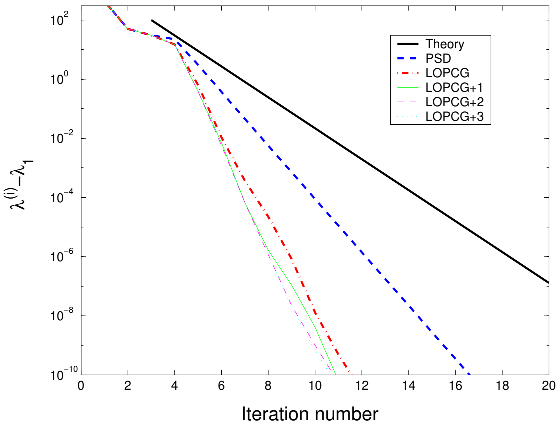

FIG. 5.3. Convergence of PSD, LOPCG and LOPCG+k, $k = 1,2,3$ for $V(2,2)$–Jacobi.

5.3. Convergence of subspace schemes. Here, we report on the results of preconditioned subspace iteration for $V(2,2)$ Gauss–Seidel preconditioning. Therefore, we construct a 7–dimensional initial subspace $U^{(0)} \in \mathbf{R}^{n\times 7}$ whose kth column is given as the grid restriction of the function $(x/\pi)^{k/2} + (y/\pi)^{k/3}$. Block versions, BPSD and LOBPCG are each started on $U^{(0)}$.

On Figure 5.4 we plot the differences $\lambda_j^{(i)} - \lambda_j$, for $j = 1,\ldots,4$, versus the iteration number i. The iteration is stopped if $\lambda_4^{(i)} - \lambda_4 \leq 10^{-8}$. This is the case after 13 BPSD iterations but only 8 LOBPCG steps, which again shows the superiority of LOBPCG.

Figure 5.4 demonstrates several properties of LOBPCG that we also observe in other similar tests:

- The convergence rate is better for the eigenpairs with smaller indexes.
- For the first eigenpair, the convergence of the block version is faster than the convergence of the single-vector version, i.e. the increase of the block size of the LOBPCG accelerates convergence of extreme eigenpairs.

The LOBPCG in this test behaves similarly to the block Lanczos method applied to $A^{-1}B$.

5.4. Optimality with respect to the mesh size. In our final set of numerical simulations we test scalability with respect to the mesh size. According to the theoretical convergence rate estimates we already discussed, the convergence should not slow down when the mesh gets finer. Combined with well-known efficiency of multigrid preconditioning, this should lead to overall costs depending linearly on the number of unknowns.

To check these statements numerically for our model problem we run the LOPCG method preconditioned using $V(2,2)$ Jacobi for the initial vector with all components equal to one on a sequence of uniform grids with $N = (2^k - 1)^2$, $k = 3,\ldots,10$ nodes. After ten iterations the residuals drop below 10^{-6} for all k, which supports the claim of a uniform in N convergence rate. The number of flops, measured by MATLAB, grows proportionally to $N^{1.1}$ in these tests, which is in a good correspondence with the theoretical prediction of the linear dependence.

Conclusion.

- A short survey of multigrid–based solvers for symmetric eigenvalue problems is presented with particular attention to off-the-shelf and black-box methods, which

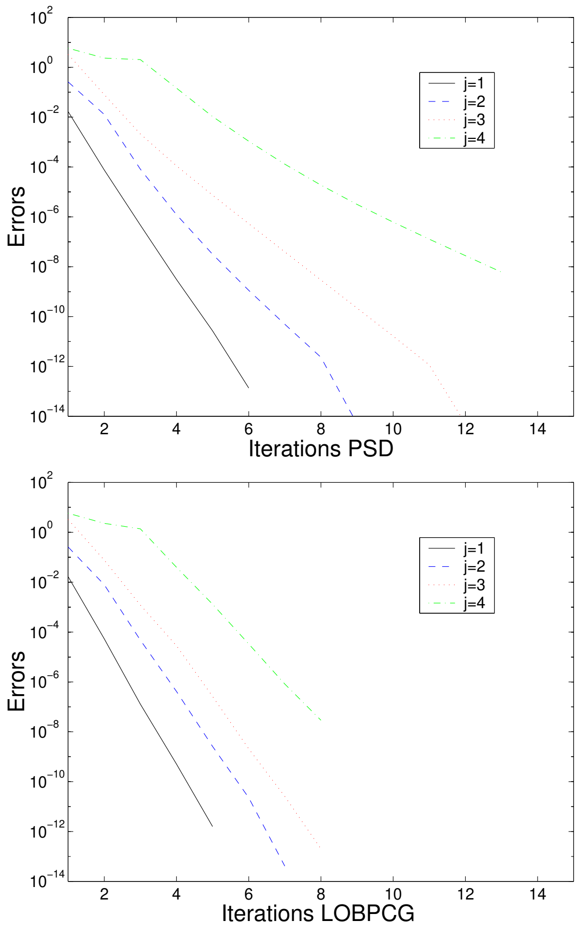

FIG. 5.4. *Preconditioned subspace iterations BPSD and LOBPCG with $V(2,2)$ preconditioning.*

should allow solving eigenvalue problems with minimal, or no, effort on the part of the developer, taking advantage of already existing algorithms and software.

- A class of such methods, where the multigrid only appears as a black-box tool of constructing the preconditioner of the stiffness matrix, and the base iterative algorithm is one of well-known off-the-shelf preconditioned gradient methods, such as the LOBPCG method, is argued to be a reasonable choice for large scale engineering computations.
- The LOBPCG method can be recommended as practically the optimal method on the whole class of preconditioned eigensolvers for symmetric eigenproblems.
- The multigrid preconditioning of the stiffness matrix is robust and practically effective for eigenproblems.
- Results of numerical tests, which demonstrate practical effectiveness and optimality of the LOBPCG method preconditioned by the standard V-cycle multigrid applied to the stiffness matrix are demonstrated.
- An efficient multigrid preconditioning of the stiffness matrix used in the LOBPCG method leads to a “textbook multigrid effectiveness” for computing extreme eigenpairs of symmetric eigenvalue problems.

References.

[1] G. ASTRAKHANTSEV, *The iterative improvement of eigenvalues*, USSR Comput. Math. and Math. Physics, 16,1 (1976), pp. 123–132.

[2] Z. BAI, J. DEMMEL, J. DONGARRA, A. RUHE, AND H. VAN DER VORST, eds., *Templates for the solution of algebraic eigenvalue problems*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[3] R. E. BANK, *Analysis of a multilevel inverse iteration procedure for eigenvalue problems*, SIAM J. Numer. Anal., 19 (1982), pp. 886–898.

[4] ———, *PLTMG: a software package for solving elliptic partial differential equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Users' guide 8.0.

[5] R. E. BANK AND T. F. CHAN, *PLTMGC: a multigrid continuation program for parameterized nonlinear elliptic systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 540–559.

[6] J. H. BRAMBLE, J. E. PASCIAK, AND A. V. KNYAZEV, *A subspace preconditioning algorithm for eigenvector/eigenvalue computation*, Adv. Comput. Math., 6 (1996), pp. 159–189.

[7] A. BRANDT, S. MCCORMICK, AND J. RUGE, *Multigrid methods for differential eigenproblems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 244–260.

[8] V. E. BULGAKOV, M. V. BELYI, AND K. M. MATHISEN, *Multilevel aggregation method for solving large-scale generalized eigenvalue problems in structural dynamics*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 453–471.

[9] Z. CAI, J. MANDEL, AND S. MCCORMICK, *Multigrid methods for nearly singular linear equations and eigenvalue problems*, SIAM J. Numer. Anal., 34 (1997), pp. 178–200.

[10] S. COSTINER AND S. TA'ASAN, *Adaptive multigrid techniques for large-scale eigenvalue problems: solutions of the Schrödinger problem in two and three dimensions*, Phys. Rev. E (3), 51 (1995), pp. 3704–3717.

[11] ———, *Simultaneous multigrid techniques for nonlinear eigenvalue problems: solutions of the nonlinear Schrödinger-Poisson eigenvalue problem in two and three dimensions*, Phys. Rev. E (3), 52 (1995), pp. 1181–1192.

[12] P. DEUFLHARD, T. FRIESE, AND F. SCHMIDT, *A nonlinear multigrid eigenproblem solver for the complex Helmholtz equation*, Scientific report 97–55, Konrad–Zuse–Zentrum für Informationstechnik Berlin, 1997.

[13] P. DEUFLHARD, T. FRIESE, F. SCHMIDT, R. MÄRZ, AND H.-P. NOLTING, *Effiziente Eigenmodenberechnung für den Entwurf integriert-optischer Chips. (Efficient eigenmodes computation for designing integrated optical chips).*, in Hoffmann, Karl-Heinz (ed.) et al., Mathematik: Schlüsseltechnologie für die Zukunft. Verbundprojekte zwischen Universität und Industrie. Berlin: Springer. 267-279 , 1997.

[14] P. DEUFLHARD, P. LEINEN, AND H. YSERENTANT, *Concepts of an adaptive hierarchical finite element code*, Impact Comput. Sci. Engrg., 1 (1989), pp. 3–35.

[15] D. C. DOBSON, *An efficient method for band structure calculations in 2D photonic crystals*, J. Comput. Phys., 149 (1999), pp. 363–376.

[16] D. C. DOBSON, J. GOPALAKRISHNAN, AND J. E. PASCIAK, *An efficient method for band structure calculations in 3D photonic crystals*, J. Comput. Phys., 161 (2000), pp. 668–679.

[17] E. G. D'YAKONOV, *Iteration methods in eigenvalue problems*, Math. Notes, 34 (1983), pp. 945–953.

[18] E. G. D'YAKONOV, *Optimization in solving elliptic problems*, CRC Press, Boca Raton,

FL, 1996. Translated from the 1989 Russian original, Translation edited and with a preface by Steve McCormick.

[19] E. G. D'YAKONOV AND A. V. KNYAZEV, *Group iterative method for finding lower-order eigenvalues*, Moscow University, Ser. XV, Computational Math. and Cybernetics, (1982), pp. 32–40.

[20] ———, *On an iterative method for finding lower eigenvalues*, Russian J. of Numerical Analysis and Math. Modelling, 7 (1992), pp. 473–486.

[21] E. G. D'YAKONOV AND M. Y. OREKHOV, *Minimization of the computational labor in determining the first eigenvalues of differential operators*, Mathematical Notes, 27 (1980), pp. 382–391.

[22] J. FATTEBERT AND J. BERNHOLC, *Towards grid-based O(N) density-functional theory methods: Optimized nonorthogonal orbitals and multigrid acceleration*, PHYSICAL REVIEW B (Condensed Matter and Materials Physics), 62 (2000), pp. 1713–1722.

[23] S. K. GODUNOV, V. V. OGNEVA, AND G. K. PROKOPOV, *On the convergence of the modified steepest descent method in application to eigenvalue problems*, in American Mathematical Society Translations. Ser. 2. Vol. 105. (English) Partial differential equations (Proceedings of a symposium dedicated to Academician S. L. Sobolev)., Providence, R. I, 1976, American Mathematical Society, pp. 77–80.

[24] W. HACKBUSCH, *On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method*, SIAM J. Numer. Anal., 16 (1979), pp. 201–215.

[25] ———, *Multigrid eigenvalue computation*, in Advances in multigrid methods (Oberwolfach, 1984), Vieweg, Braunschweig, 1985, pp. 24–32.

[26] W. HACKBUSCH, *Multigrid methods and applications*, Springer-Verlag, Berlin, 1985.

[27] V. HEUVELINE AND C. BERTSCH, *On multigrid methods for the eigenvalue computation of nonselfadjoint elliptic operators*, East-West J. Numer. Math., 8 (2000), pp. 275–297.

[28] T. HWANG AND I. D. PARSONS, *A multigrid method for the generalized symmetric eigenvalue problem: Part I algorithm and implementation*, International Journal for Numerical Methods in Engineering, 35 (1992), p. 1663.

[29] ———, *A multigrid method for the generalized symmetric eigenvalue problem: Part II performance evaluation*, International Journal for Numerical Methods in Engineering, 35 (1992), p. 1677.

[30] ———, *Multigrid solution procedures for structural dynamics eigenvalue problems*, Computational Mechanics, 10 (1992), p. 247.

[31] A. V. KNYAZEV, *Computation of eigenvalues and eigenvectors for mesh problems: algorithms and error estimates*, Dept. Numerical Math. USSR Academy of Sciences, Moscow, 1986. (In Russian).

[32] ———, *Convergence rate estimates for iterative methods for mesh symmetric eigenvalue problem*, Soviet J. Numerical Analysis and Math. Modelling, 2 (1987), pp. 371–396.

[33] ———, *A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace*, in International Ser. Numerical Mathematics, v. 96, Eigenwertaufgaben in Natur- und Ingenieurwissenschaften und ihre numerische Behandlung, Oberwolfach, 1990., Basel, 1991, Birkhauser, pp. 143–154.

[34] ———, *Preconditioned eigensolvers—an oxymoron?*, Electron. Trans. Numer. Anal., 7 (1998), pp. 104–123 (electronic). Large scale eigenvalue problems (Argonne, IL, 1997).

[35] ———, *Preconditioned eigensolvers: practical algorithms*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 352–368. Section

11.3. An extended version published as a technical report UCD-CCM 143, 1999, at the Center for Computational Mathematics, University of Colorado at Denver, http://www-math.cudenver.edu/ccmreports/rep143.ps.gz.

[36] A. V. KNYAZEV, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541 (electronic). Copper Mountain Conference (2000).

[37] A. V. KNYAZEV AND K. NEYMEYR, *A geometric theory for preconditioned inverse iteration. III: A short and sharp convergence estimate for generalized eigenvalue problems*, Linear Algebra Appl., 358 (2003), pp. 95–114.

[38] A. V. KNYAZEV AND A. L. SKOROKHODOV, *On exact estimates of the convergence rate of the steepest ascent method in the symmetric eigenvalue problem*, Linear Algebra and Applications, 154–156 (1991), pp. 245–257.

[39] P. LEINEN, W. LEMBACH, AND K. NEYMEYR, *An adaptive subspace method for elliptic eigenproblems with hierarchical basis preconditioning*, scientific report, Sonderforschungsbereich 382, Universitäten Tübingen und Stuttgart, 1997.

[40] S. H. LUI, H. B. KELLER, AND T. W. C. KWOK, *Homotopy method for the large, sparse, real nonsymmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 312–333.

[41] J. MANDEL AND S. MCCORMICK, *A multilevel variational method for* $A\mathbf{u} = \lambda B\mathbf{u}$ *on composite grids*, J. Comput. Phys., 80 (1989), pp. 442–452.

[42] J. MARTIKAINEN, T. ROSSI, AND J. TOIVANEN, *Computation of a few smallest eigenvalues of elliptic operators using fast elliptic solvers*, Communications in Numerical Methods in Engineering, 17 (2001), pp. 521–527.

[43] S. F. MCCORMICK, *A mesh refinement method for* $Ax = \lambda Bx$, Math. Comput., 36 (1981), pp. 485–498.

[44] K. NEYMEYR, *A geometric theory for preconditioned inverse iteration. I: Extrema of the Rayleigh quotient*, Linear Algebra Appl., 322 (2001), pp. 61–85.

[45] ———, *A geometric theory for preconditioned inverse iteration, II: Sharp convergence estimates*, Linear Algebra Appl., 322 (2001), pp. 87–104.

[46] K. NEYMEYR, *A geometric theory for preconditioned inverse iteration applied to a subspace*, Math. Comp., 71 (2002), pp. 197–216.

[47] ———, *A posteriori error estimation for elliptic eigenproblems*, Numer. Linear Algebra Appl., 9 (2002), pp. 263–279.

[48] Y. NOTAY, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Lin. Alg. Appl., 9 (2002), pp. 21–44.

[49] S. OLIVEIRA, *On the convergence rate of a preconditioned subspace eigensolver*, Computing, 63 (1999), pp. 219–231.

[50] E. E. OVTCHINNIKOV AND L. S. XANTHIS, *Successive eigenvalue relaxation: a new method for generalized eigenvalue problems and convergence estimates*, Proc. R. Soc. Lond. A, 457 (2001), pp. 441–451.

[51] B. SAMOKISH, *The steepest descent method for an eigenvalue problem with semi-bounded operators*, Izvestiya Vuzov, Math., (1958), pp. 105–114. (In Russian).

[52] G. L. G. SLEIJPEN AND H. A. VAN DER VORST, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[53] P. SMIT AND M. H. C. PAARDEKOOPER, *The effects of inexact solvers in algorithms for symmetric eigenvalue problems*, Linear Algebra Appl., 287 (1999), pp. 337–357. Special issue celebrating the 60th birthday of Ludwig Elsner.

[54] S. I. SOLOV'EV, *Convergence of the modified subspace iteration method for nonlinear eigenvalue problems*, Preprint SFB393/99-35, Sonderforschungsbereich 393 an der

Technischen Universität Chemnitzs, Technische Universität D-09107 Chemnitz, Germany, 1999. Available at http://www.tu-chemnitz.de/sfb393/Files/PS/sfb99-35.ps.gz.

[55] ———, *Preconditioned gradient iterative methods for nonlinear eigenvalue problems*, Preprint SFB393/00-28, Sonderforschungsbereich 393 an der Technischen Universität Chemnitz, Technische Universität D-09107 Chemnitz, Germany, 2000. 17 pp. Available at http://www.tu-chemnitz.de/sfb393/Files/PS/sfb00-28.ps.gz.

[56] L. STRAKHOVSKAYA, *An iterative method for evaluating the first eigenvalue of an elliptic operator*, USSR Comput. Math. and Math. Physics, 17,3 (1977), pp. 88–101.

[57] L. STRAKHOVSKAYA AND R. P. FEDORENKO, *Solution of the principal spectral problem and mathematical modeling of nuclear reactors*, Comput. Math. Math. Phys., 40 (2000), pp. 880–888.

[58] ———, *Solution of the principal spectral problem for a system of multigroup diffusion equations*, Comput. Math. Math. Phys., 40 (2000), pp. 1312–1321.

[59] H. YSERENTANT, *Hierarchical bases*, in ICIAM 91, O'Malley and R.E., eds., Washington, DC, 1992, SIAM, Philadelphia, pp. 256–276.

[60] T. ZHANG, K. H. LAW, AND G. H. GOLUB, *On the homotopy method for perturbed symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1625–1645 (electronic).