Adaptive multigrid techniques for large-scale eigenvalue problems: Solutions of the Schrödinger problem in two and three dimensions

Sorin Costiner and Shlomo Ta'asan
Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, 76100, Israel and Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia 23681
(Received 23 June 1994)

Multigrid (MG) algorithms for large scale eigenvalue problems (EP), obtained from discretizations of partial differential EP, have often been shown to be more efficient than single level eigenvalue algorithms. This paper describes a robust and efficient, adaptive MG eigenvalue algorithm. The robustness of the present approach is a result of a combination of MG techniques introduced here, i.e., the completion of clusters; the adaptive treatment of clusters; the simultaneous treatment of solutions in each cluster; the multigrid projection (MGP) coupled with backrotations; and the robustness tests. Due to the MGP, the algorithm achieves a better computational complexity and better convergence rates than previous MG eigenvalue algorithms that use only fine level projections. These techniques overcome major computational difficulties related to equal and closely clustered eigenvalues. Some of these difficulties were not treated in previous MG algorithms. Computational examples for the Schrödinger eigenvalue problem in two and three dimensions are demonstrated for cases of special computational difficulties, which are due to equal and closely clustered eigenvalues. For these cases, the algorithm requires $O(qN)$ operations for the calculation of q eigenvectors of size N, using a second order approximation. The total computational cost is equivalent to only a few Gauss-Seidel relaxations per eigenvector.

PACS number(s): 02.70.Bf, 02.70.Rw, 02.60.Lj

I. INTRODUCTION

Large scale eigenvalue problems (EP) arising from physics, chemistry, and engineering often have special features which are not always exploited by eigenvalue solvers, such as the following: The EP can be approximated on several discretization levels, only a few eigenvalues and eigenvectors are sought, and the solutions are dominated by smooth components. In contrast with single level techniques, multigrid (MG) solvers can exploit naturally these features and provide powerful solving capabilities [1,2].

MG techniques involve, in general, the processing of the problem on a sequence of discretization levels. Usually, these levels are finite dimensional spaces of functions defined on increasingly finer grids. Such MG techniques can approximate the inverse power iteration, which is a very efficient eigenvalue technique, in order $O(N)$ operations.

Computer programs that solve eigenvalue problems are often faced with large computational difficulties, especially when close or equal eigenvalues are present, as usual in Schrödinger and in electromagnetism problems. These difficulties become more complicated when the eigenvalue problems are represented on several levels. Single level or MG procedures, such as relaxations or interlevel transfers, can amplify or introduce an eigenvector in the error component of a solution that is an approximated eigenvector. This we refer to as eigenvector mixing. When clusters of eigenvectors with close or equal eigenvalues are computed, the error of a computed eigenvector usually contains combinations of the other eigenvectors with close eigenvalues. A cluster of eigenvectors will be called complete relative to a procedure if it contains a whole set of eigenvectors which are mixed by that procedure. If a procedure acts on an incomplete cluster, the not-approximated eigenvectors of the completed cluster usually would enter into the computed solutions and prevent convergence. In addition to the computational difficulties related to close eigenvalues, to incomplete clusters, and to mixing of solutions, the structure of clusters may differ on different levels, and the coarse level eigenvectors may not approximate well the fine level ones. Other difficulties, not treated in MG algorithms before, result from the fact that the cluster structures, the multiplicity of eigenvalues, and the levels on which the solutions are poorly represented, are usually not known in advance.

The above mentioned difficulties are closely coupled and should be treated together to obtain robust and efficient algorithms. Several previous MG approaches relate to some of the mentioned difficulties. In no previous approach all of these difficulties were treated together. The above discussion suggests that (i) clusters should be completed, (ii) different clusters should be treated differently, i.e., adaptively, (iii) solutions should be treated in clusters simultaneously, and (iv) separation of eigenvectors should be done on different levels. MG algorithms which may fail in standard situations include nonadaptive algorithms, sequential algorithms, algorithms in which clusters are not completed, or algorithms in which solutions are not properly separated. Even when such algorithms work, the coarse level separation techniques

introduced here may improve their efficiency, and the presented adaptive techniques may improve their robustness.

This paper focuses on a robust and efficient algorithm for the calculation of a few eigenvalues and their corresponding eigenvectors. Its development is guided by the treatment of the above difficulties. The algorithm first solves the problem on coarse levels, then interpolates the solutions to finer levels where they serve as initial approximations to the finer level solutions. On each fine level, clusters are identified, tested for completeness, completed if necessary, and improved by MG cycles using coarser levels. The eigenvalue equations are relaxed on each level followed by full approximation scheme [27] (FAS) interlevel transfers. An MG projection (MGP) which generalizes the Rayleigh-Ritz projection, and backrotations are employed on coarse levels, in order to separate eigenvectors within clusters and to prevent mixing. Robustness tests control the algorithm's convergence and efficiency. These are done adaptively for different clusters on different levels.

The robustness of the present approach is determined by the introduced MG techniques, i.e., the adaptive completion and treatment of clusters, the simultaneous treatment of solutions in each cluster, the MGP coupled with backrotations, and the robustness tests. Moreover, due to MGP, the algorithm achieves a better computational complexity and better convergence rates than previous MG eigenvalue algorithms which use only fine level projections.

Computational examples are shown, for Schrödinger eigenvalue problems in two dimensions (2D) and 3D, with periodic boundary conditions. The problems have special difficulties such as very close and equal eigenvalues, the cluster structure differs on different levels, the number of eigenvalues in clusters and their multiplicity are not known, and coarse level solutions are poor representations of fine level solutions. A second order approximation is obtained in $O(qN)$ work, for q eigenvectors of size N on the finest level. The algorithm uses one to ten fine level cycles per cluster and in each cycle, two to four fine level relaxations per eigenvector are performed. The algorithm yields accurate results for very close eigenvalues, and accuracy of more than ten decimal places for equal eigenvalues. The MGP reduces the most time consuming part of previous algorithms, namely the fine level separation work with complexity $O(q^2N)$ to $O(qN)$.

We refer to the early works [3–5] for theory on MG eigenvalue solvers and first algorithms. A sequential MG algorithm for linear eigenvalue problems performing the projection on fine levels is presented in [1], this algorithm combined with a conjugate residual method is applied to the Hartree-Fock equation [6] for real problems. More theory and algorithms on MG eigenvalue problems are found in [2,7]; a related MG approach is presented in [8].

The present approach can be extended to nonlinear eigenvalue problems, an example being presented in [9], [10], and was efficiently used in sequences of EP and bifurcation problems [11]. In [12,13], an adaptive algebraic correction scheme cycle is used to compute the first eigenvector and its eigenvalue for the multigroup neutron diffusion equation. The elements of such an MG cycle, modified to the FAS form, can be used in the algorithm presented here. The combination of our technique with domain decomposition techniques for eigenvalue problems [14,15] is natural but was not analyzed yet. A review article on single level large-scale complex eigenvalue problems, containing many references, is [16]. For theory on Ritz projections and on algebraic eigenvalue problems we refer to [17–19]. The single level technique for obtaining the eigenvectors by relaxations and projections is referred in different places as subspace, simultaneous or Ritz iterations; see [20–22] for a single level algorithm and mathematical foundations.

The MG projection and backrotations were first introduced in [23] and in the reports [9,24,25]. Applications of the MG combined cycles to electromagnetism computations are presented in [11], and an outline of a related computational approach presented here is given in [26].

The paper is organized as follows. Section II presents the central MG separation techniques, i.e., the MG solver cycle, the MGP incorporating backrotations, and the MG combined cycles. Section III describes the adaptive techniques, i.e., the adaptive MG cycle, the cluster-completion, the robustness tests, and the adaptive full MG [27] (FMG). Section IV presents computational examples. It contains a final subsection with details and observations on the algorithms which were included there to keep the rest of the presentation simpler.

II. ON MULTIGRID SEPARATION TECHNIQUES

This section presents the two main separation techniques used in the introduced algorithms, the MG solver cycle and the MG projection (MGP). The main role of the MG solver cycle is to separate a cluster from the other clusters, while the main role of the MGP is to separate the eigenvectors inside a cluster. The MGP is combined with backrotations which prevent undesired rotation, sign flipping, and scaling of eigenvectors. Both separation techniques are used simultaneously in MG combined cycles.

In the rest of the paper, the problem (1)

$$AU = U\Lambda \tag{1}$$

is defined on a sequence of finite dimensional spaces, called levels. The U denotes an eigenvector associated to the eigenvalue Λ. The matrix corresponding to the level i problem is denoted by A_i and a transfer from level i to level j (e.g., interpolation) is denoted by I_i^j. For example, A_i may be the matrices obtained by discretizing a continuous eigenvalue problem, on a sequence of grids. In this case, the level i is the space of functions defined on grid i. In the presented computational applications it is considered that the levels correspond to discretizations (e.g., on grids), ordered from coarse to fine (level 1 corresponding to coarsest discretization) and the corresponding levels are called coarse or fine. Generally, this restriction is not necessary, the algorithms having an algebraic character.

The following FAS general formulation (full approxi-

mation scheme [27]) is used in the algorithms. Let

$$F_i U_i = T_i \tag{2}$$

be a level i problem, where F_i is a general operator and T_i is a right hand side. The level j problem

$$F_j U_j = T_j \tag{3}$$

is an FAS transfer of the level i problem (2) if

$$T_j = I_i^j (T_i - F_i U_i) + F_j I_i^j U_i \ . \tag{4}$$

See also [1]. The level j problem (3) is used in solving the level i problem (2). The level i solution U_i^{old} is corrected with the level j solution U_j by the FAS correction:

$$U_i^{\text{new}} = U_i^{\text{old}} + I_j^i (U_j - I_i^j U_i^{\text{old}}) \ . \tag{5}$$

A. Multigrid solver cycles

The MG solver cycle is a central tool for separating the desired eigenspaces and for separating eigenvectors when the eigenvalues are different and well enough approximated. It can be regarded as an approximation of the efficient inverse power iteration [19].

To motivate the MG solver cycle, consider the eigenvalue problem (1), where A is a square matrix. If Λ' approximates well enough the eigenvalue Λ (with multiplicity 1 for convenience), corresponding to an eigenvector U, then the inverse power iteration

$$U^{n+1} = (A - \Lambda' I)^{-1} U^n, \quad U^{n+1} = U^{n+1} / ||U^{n+1}|| \tag{6}$$

will converge fast (in a few iterations) to U [since the U component in U^n will be multiplied at each iteration by $1/(\Lambda - \Lambda') \approx \infty$, [19]]. For large A it is too expensive to compute $(A - \Lambda' I)^{-1}$, but one can approximate (6) by solving iteratively:

$$(A - \Lambda' I) U^{n+1} = U^n, \quad U^{n+1} = U^{n+1} / ||U^{n+1}|| \tag{7}$$

which is equivalent to (6). During the solution procedure, if U^n approximates well enough U, then Λ' can be improved, using a Rayleigh quotient equality

$$(U^n)^T A U^n = (U^n)^T U^n \Lambda'. \tag{8}$$

For large A, the iteration (7) is impractical for single level algorithms, but it can be approximated by MG cycles, which have often shown to be efficient [1,2].

Relation (1) can be considered in block form where U is a matrix whose columns are the eigenvectors corresponding to the eigenvalues of the diagonal matrix Λ. Relations (4) and (5) can be considered in block form in the same way. In a simultaneous MG solver cycle, the problem (1) is represented on the different levels in the FAS form:

$$F_i U_i := A_i U_i - U_i \Lambda_i = T_i \ , \tag{9}$$

where $T_m = 0$ on the initial level m (finest usually) and T_j are computed by (4) for $j < m$, with $j = i - 1$. Equation (9) is relaxed on each level and the solutions are corrected by (5).

An MG solver cycle from level m to level l ($l < m$ here), is defined by

$(U_m, \Lambda) \leftarrow$ **MG-Solver-Cycle** $(m, A_m, U_m, \Lambda, T_m, l)$
  For $k = m, \ldots, l$ **(step by** -1**) do:**
    $U_k \leftarrow$**Relax** $(m, A_k, U_k, \Lambda, T_k, k, l)$
    If $k > l$ **Transfer:**
      $U_{k-1} = I_k^{k-1} U_k$,
      $T_{k-1} = I_k^{k-1}(T_k - A_k U_k) + A_{k-1} U_{k-1}$
  End
  For $k = l, \ldots, m$ **(step by 1) do:**
    If $(k > l)$ **Correct** $U_k = U_k + I_{k-1}^k (U_{k-1} - I_k^{k-1} U_k)$
    $U_k \leftarrow$**Relax** $(m, A_k, U_k, \Lambda, T_k, k, l)$
  End

Such an MG cycle, where the algorithm goes from fine to a coarse level and comes back to the initial fine level is called V cycle [27]. In this MG solver cycle, the Λ is kept constant on all levels.

B. Generalized Rayleigh-Ritz projections

This subsection presents a generalization of the Rayleigh-Ritz projection [19], for eigenvalue problems with right hand side. The Rayleigh-Ritz projection is used to find the eigenvectors when only linear combinations of the eigenvectors are known (separation of eigenvectors).

Consider the eigenvalue relation:

$$AV = V\Lambda \ , \tag{10}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_q)$ contains on the diagonal the q sought eigenvalues corresponding to the sought eigen-

vectors which are the columns of V. Assume that U, which satisfies

$$V = UE \ , \tag{11}$$

is given instead of V, where E is a $q \times q$ invertible matrix to be found. Substituting (11) into (10) gives

$$AUE = UE\Lambda \ . \tag{12}$$

An FAS transfer (4) of (12) to another level yields an equation of the form

$$AUE = UE\Lambda + TE \ , \tag{13}$$

where the product TE is the FAS right hand side of (13) with known T. Solutions E and Λ for (13) can be computed by solving the $q \times q$ generalized eigenvalue problem

$$U^T(AU - T)E = (U^TU)E\Lambda \tag{14}$$

obtained by multiplying (13) by U^T. For $T = 0$, the usual Rayleigh-Ritz projection is obtained.

The process of obtaining (E, Λ) given (A, U, T) is denoted by

$$(E, \Lambda) \leftarrow \textbf{GRRP}(A, U, T) \tag{15}$$

and is referred later as the generalized Rayleigh-Ritz projection (GRRP).

C. Multigrid projections

The solutions E and Λ of (13) can be obtained by an FAS MG procedure. Consider (13) written as a level i problem:

$$A_iU_iE - U_iE\Lambda = T_iE \ . \tag{16}$$

Then the FAS transfer of (16) to level j is

$$A_jU_jE - U_jE\Lambda = T_jE \ , \tag{17}$$

where $U_j = I_i^jU_i$. T_jE is computed by (4), and results in

$$T_j = I_i^j(T_i - A_iU_i) + A_jI_i^jU_i \ . \tag{18}$$

A solution (E, Λ) of (16) is a solution of (17). The solutions (E, Λ) of (17) can be obtained by GRRP.

Problems (16) and (17) have the same form. Hence problem (17) can be further transferred in the same FAS way to other levels and to perform the GRRP on the last level, e.g., on coarsest level. The process of obtaining (E, Λ) by transferring the eigenvalue problem to other levels will be called the MG projection (MGP). The FAS transfer (18) for the problem (17) is the same as the transfer used in the MG solver cycle for the problem $A_jU_j - U_j\Lambda = T_j$. This makes it possible to perform the MGP simultaneously with the MG solver cycle, in MG combined cycles, as presented in Sec. II E.

D. Backrotations

Backrotations are introduced to prevent rotations of solutions in subspaces of eigenvectors with equal or close eigenvalues, and to prevent permutations, rescalings, and sign changing of solutions during processing. For example, backrotations are used after the computation of (E, Λ) by an MGP, since E may permute or mix the eigenvectors in a degenerate eigenspace. Thus, if degenerate subspaces are present, the backrotation should bring E to a form close to block diagonal and having on diagonal blocks close to the identity matrix. Each such block associated to a degenerate subspace prevents mixing inside that subspace. These motivate the particular backrotation algorithms presented next. A backrotation step will be further denoted by

$$(E, \Lambda) \leftarrow \textbf{Backrotation}(E, \Lambda) \tag{19}$$

Backrotation
Input (E, Λ)
1) Sort the eigenvalues of Λ and
permute the columns of E accordingly
2) Determine the clusters of eigenvalues of Λ
to be considered degenerate, and
determine the clusters to be considered nondegenerate
3) For each diagonal block in E
associated with a nondegenerate cluster do:
bring to the diagonal the dominant elements of the block
permuting the columns of E,
and the diagonal of Λ correspondingly.
4) Let F be a block diagonal matrix
whose diagonal blocks are the diagonal blocks of E,
corresponding to the determined clusters.
replace each diagonal block which does not correspond
to a degenerate cluster by the corresponding identity matrix
5) Set $E = EF^{-1}$.
6) Change the signs of columns of E
to get positive elements on diagonal.
7) Normalize the columns of E.
Output (E, Λ)

E. Multigrid combined cycles

An MG simultaneous cycle combining an MG solver cycle with an MGP is described next. U_k is the matrix whose q columns are approximate solutions of the level k problem $A_k U_k = U_k \Lambda + T_k$, where T_k is obtained by an FAS transfer from the level $k+1$ problem. For level m, $T_m = 0$. In the applications, m is the finest level involved in the cycle, l_c is the coarsest level, and l_p is a level on which the GRRP and backrotations are performed.

$(U_m, \Lambda, T_m) \leftarrow$ **Solve-MGP** $(m, A_m, U_m, \Lambda, T_m, l_p, l_c, q)$
For $k = m, \ldots, l_c$ **do:**
Repeat ν_1^k **Times:**
If $k = l_p$ **then** $(U_k, \Lambda, T_k) \leftarrow$ **GRR-BR**$(m, A_k, U_k, \Lambda, T_k, k, l_p)$
$U_k \leftarrow$**Relax** $(m, A_k, U_k, \Lambda, T_k, k, l_c)$
If $k > l_c$ **Transfer:**
$U_{k-1} = I_k^{k-1} U_k$,
$T_{k-1} = I_k^{k-1}(T_k - A_k U_k) + A_{k-1} U_{k-1}$
End
For $k = l_c, \ldots, m$ **do:**
If $(k > l_c)$ **Correct** $U_k = U_k + I_{k-1}^k (U_{k-1} - I_k^{k-1} U_k)$
Repeat ν_2^k **Times**
$U_k \leftarrow$**Relax** $(m, A_k, U_k, \Lambda, T_k, k, l_c)$
If $k = l_p$ **then** $(U_k, \Lambda, T_k) \leftarrow$ **GRR-BR**$(m, A_k, U_k, \Lambda, T_k, k, l_p)$
End

The **GRR-BR** separation algorithm used above is the following:

$(U_k, \Lambda, T_k) \leftarrow$ **GRR-BR**$(m, A_k, U_k, \Lambda, T_k, k, l_p)$
$(E, \Lambda) \leftarrow$**GRR**(A_k, U_k, T_k)
$(E, \Lambda) \leftarrow$**Backrotation**(E, Λ)
$U_k = U_k E$
$T_k = T_k E$

The MG combined cycle, solve MGP, is the central building element of the adaptive algorithms presented in Sec. III.

III. ADAPTIVE MULTIGRID ALGORITHMS

The construction of adaptive MG techniques for eigenvalue problems is motivated by two types of difficulties. The first type is related to the problems while the second type is related to the algorithms involved. Difficulties related to the problems are as follows: existence of close and equal eigenvalues, unknown cluster structure, different cluster structures on different levels, interlevel cross correspondence of eigenvalues, and poor approximation of fine level eigenvalues and eigenvectors by coarse level eigenvalues and eigenvectors. Additionally, the eigenvectors may be highly sensitive with respect to some data, and the transfers may not conserve the dimensions of the eigenspaces.

Some of the central difficulties related to the algorithms are due to the following: incompleteness of clusters, mixing of solutions, and unknown parameters of the algorithms, such as iteration numbers, relaxation parameters, and levels on which to apply a given procedure.

These central difficulties can be further grouped in difficulties related to (a) clusters and mixing and (b) unknown parameters of subroutines. The techniques introduced for treating the difficulties related to clusters and mixing are the adaptive separation and completion of clusters on different levels, the simultaneous processing of clusters, and the MG projections and backrotations. The techniques introduced for treating the difficulties related to unknown parameters are the robustness tests. These techniques are incorporated in the following adaptive algorithms: the adaptive MG cycle, the cluster completion, the robustness tests, and the adaptive FMG.

A. Adaptive multigrid cycles

Efficiency and convergence considerations require that the GRRP should be done for different clusters on different levels in MG cycles. The coarsest level used to treat a given cluster may not coincide with the level on which the GRRP is done. Other parameters, such as the number of relaxations in an MG cycle, may vary too.

Following is a description of a basic adaptive MG cycle which invokes different projection levels for different clusters. Moreover, the coarsest levels used for different clusters are different.

Let q eigenvectors be approximated by j clusters on level k:

$$U_k = (U_k^1, \ldots, U_k^j) \ , \tag{20}$$

where, as before, each U_k^i approximates the solution of $A_k \bar{U}_k^i = \bar{U}_k^i \Lambda^i + T_k^i$, $i = 1, \ldots, j$. For each cluster U_k^i let l_p^i be the level on which the GRR-BR projection is done, and let l_c^i be the coarsest level used in the MG cycle for this cluster. Here it is assumed that $l_c^i \le l_p^i$. Denote $l_p = (l_p^1, \ldots, l_p^j)$, $l_c = (l_c^1, \ldots, l_c^j)$, and by $\Lambda = \mathrm{diag}(\Lambda^1, \ldots, \Lambda^j)$. Usually, on the finest level, $k = m$, $T_k = (T_k^1, \ldots, T_k^j) = (0, \ldots, 0)$. An MG cycle consisting of a sequence of cycles for each cluster in turn, for improving a given approximation (U_m, Λ, T_m), is

$(U_m, \Lambda, T_m) \leftarrow$ **Adaptive-MGP** $(m, A_m, U_m, \Lambda, T_m, l_p, l_c, q)$
For $i = 1, \ldots, j$ **do:**
$(U_m^i, \Lambda^i, T_m^i) \leftarrow$ **Solve-MGP** $(m, A_m, U_m^i, \Lambda^i, T_m^i, l_p^i, l_c^i, q^i)$
End

The choice of the different parameters of the algorithm is done by robustness tests discussed in Sec. III C.

B. Cluster completion algorithm

When a procedure acts on an incomplete cluster, then the dominant error components of the solutions usually are formed of the nontreated eigenvectors of the completed cluster. It is hard to eliminate these error components. This suggests to complete the clusters and to treat simultaneously all solutions belonging to the complete cluster. Simultaneous techniques can be easier coupled with separation techniques at any stage of the algorithm. Since sequential techniques cannot invoke separation at an arbitrary stage and hardly avoid difficulties due to mixing, better efficiency and versatility are obtained for simultaneous techniques, as for sequential techniques.

The completion of a cluster is done by adding in turns a new vector u and improving it by MG cycles. The separation of u from the other eigenvectors is performed by a GRR-BR. An approximate eigenvalue is computed for this eigenvector, by a Rayleigh quotient. If the eigenvalue is close to the cluster, then the new vector is added to the cluster. If it does not belong to the cluster, then the cluster is considered complete. The convergence of the additional eigenvector is not sought. At the end, the complete cluster is improved by several adaptive MGP cycles.

Denote by d_j the current dimension of the cluster U_k^j. The cluster completion and cluster addition algorithms are given by

$(U_k^j, \Lambda^j, T_k^j, q) \leftarrow$**Cluster-Completion**$(j, A_k, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, q)$
Until (Cluster-Completion-Test = TRUE) Do
Choose random u
Until $\langle A_k u, u\rangle / \langle u, u\rangle$ **and residuals stabilize Do:**

$(u, \Lambda_{max}^j, T_k^j) \leftarrow$**Adaptive-MGP**$(k, A_k, u, \Lambda_{max}^j, T_k^j, 0, l_c^j, 1)$
Separate u **from** $(U_k^1, \ldots, U_k^j)$
Set $\lambda = \langle A_k u, u\rangle / \langle u, u\rangle$
$U_k^j \leftarrow (U_k^j, u)$
$\Lambda^j \leftarrow \mathrm{diag}(\Lambda^j, \lambda)$
$q = q + 1, \quad d_j = d_j + 1$
End
Perform
$(U_k^j, \Lambda^j, T_k^j) \leftarrow$**Adaptive-MGP**$(k, A_k, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, d_j)$

$(j, U_k, \Lambda, T_k, q) \leftarrow$**Add-Cluster**$(j, A_k, U_k, \Lambda, T_k, l_p, l_c, q)$
Set $j = j + 1$
$(U_k^j, \Lambda^j, T_k^j, q) \leftarrow$**Cluster-Completion**$(j, A_k, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, q)$
Set $U_k = (U_k^1, \ldots, U_k^j), \quad \Lambda = (\Lambda^1, \ldots, \Lambda^j)$

C. Robustness tests

Robustness tests are techniques which find the values of parameters to be used in a procedure, such that the procedure will be efficient for a given input. They are essential for robustness and efficiency. The values of the parameters are obtained by optimization which is usually performed on coarse levels, by a search, testing the procedure over a set of values of the parameters, and choosing the values for which the procedure performs best, e.g., has best convergence rate. Previous results are used to reduce the work involved in testing.

For a simple illustration, the robustness test which provides the values of the parameters (l_p, l_c) for the adaptive MGP cycle is presented. It is assumed that during the FMG for a given cluster these parameters will stabilize as the levels become finer.

A complete cluster on level L is called stabilized, if it corresponds to a complete cluster from level $L - 1$ or $L + 1$ in the sense of the number of eigenvectors in the cluster, the values of the eigenvalues, and the eigenvectors approximation. To reduce the work required by a fine level robustness test, it is assumed that correspond-

ing stabilized clusters will require the same parameters l_c, l_p. Thus, robustness tests are applied on coarse levels until clusters get stabilized. For nonstabilized clusters, which would usually exist on coarse levels only, a search is performed for obtaining best values for l_c, l_p. Such tests are inexpensive when performed on coarse enough levels, and often lead to significant fine level work savings.

Denote by $l_{p,m}, l_{c,m}$ the l_p and l_c parameters, for an MG cycle for a given cluster, (U_m, Λ), on level m, and by $\mu(l_{p,m}, l_{c,m}) := \mu$ [adaptive]-MGP$(m, A_m, U_m, \Lambda, T_m, l_{p,m}, l_{c,m}, q)$] the convergence rate (measured by the residual decrease) of the adaptive MGP cycle for the cluster (U_m, Λ), using the parameters $(l_{p,m}, l_{c,m})$. The following algorithm updates $(l_{p,m}, l_{c,m})$:

$(l_{p,m}, l_{c,m}) \leftarrow$ **Robustness-Test** $(m, A_m, U_m, \Lambda, T_m, l_p, l_c, q)$
If $(||\Lambda_{m-1} - \Lambda_{m-2}|| \leq \epsilon)$
then
$(l_{p,m}, l_{c,m}) = (l_{p,m-1}, l_{c,m-1})$
else
If $(||\Lambda_m - \Lambda_{m-1}|| \geq \epsilon)$ **or if** Λ_m **is not computed**
then
Solve for $(l_{p,m}, l_{c,m})$**:**
$\min_{l_P, l_C} \mu(l_P, l_C), \quad l_C \leq l_P \leq m,$
else
$(l_{p,m}, l_{c,m}) = (l_{p,m-1}, l_{c,m-1})$
endif
endif

Convergence of the adaptive MGP is always attained using the values found by the robustness test since at least the single level cycle converges, being a subspace iteration algorithm [19] (for $l_c = l_p = m$ when $\mu < 1$).

The minimization search is performed just for a few choices of parameters, since on coarse levels only a few combinations of coarse level values of parameters exist. Similar algorithms are used for determining the types, parameters, and numbers of relaxations in MG cycles.

D. The adaptive FMG algorithm

During the FMG, coarse levels approximate the desired subspaces and the clusters of eigenvalues. Coarse levels are also used to optimize the algorithm and to check the convergence of the sequence of discrete solutions obtained on the sequence of levels towards the differential solution. The full MG algorithm uses as building blocks the adaptive-MGP, add-cluster, cluster-completion, and robustness-test algorithms described before.

The full MG solver described below starts on the coarsest level. The solutions found there are used as initial approximation for finer level solutions where more eigenvectors are added if needed. The cluster completion is tested on all new finest levels and performed on several levels until the clusters are stabilized.

Adaptive-FMG(m, q, A)
Set $k = 1,\ q' = 0,\ j = 0,\ \ l_p = k,\ l_c = k$
Until $(q' \geq q$ or $q' \geq \alpha \dim_k)$ **Perform**
$(j, U_k, \Lambda, T_k, q') \leftarrow$**Add-Cluster**$(j, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$(U_k, \Lambda, T_k) \leftarrow$**Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
Until $k \geq m$ **Do:**
If $k < m$ **then:**
Set $k = k + 1,\ U_k = I_{k-1}^k U_{k-1},\ T_k = 0$
endif
$(l_p, l_c) \leftarrow$ **Robustness-Test** $(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
If $(q' \geq q)$ **then:**
If (Cluster-Completion-Test=TRUE) then:
$(U_k, \Lambda, T_k) \leftarrow$**Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
Else
$(U_k^j, \Lambda^j, T_k^j, q') \leftarrow$**ClusterCompletion**$(j, A_k, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, q')$
$(U_k, \Lambda, T_k) \leftarrow$**Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
endif
Else
Until $(q' \geq q$ **or** $q' \geq \alpha \dim_k)$ **Perform**
$(j, U_k, \Lambda, T_k, q') \leftarrow$**Add-Cluster**$(j, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$(U_k, \Lambda, T_k) \leftarrow$**Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
endif
Endo

The notation **k-FMG-V**(ν_i, ν_j) denotes an FMG algorithm in which k cycles, type V, $V(\nu_i, \nu_j)$, are performed per level, besides the adaptive computations (cluster completion, add cluster, and robustness tests). In each $V(\nu_i, \nu_j)$ cycle, ν_i (ν_j) relaxations are performed per level on the path from fine to coarse (coarse to fine).

Our MG approach differs from previous MG approaches [1–8,12] mainly by the emphasis on robustness, the adaptive and simultaneous cluster processing, the MG projection and backrotations, the treatment of eigenvector mixing, and the treatment of close and equal eigenvalues.

E. Storage and complexity

Assume that the eigenvectors have size N on the finest level. For the adaptive-FMG algorithm, storage is required for the eigenvectors and the corresponding right hand sides, on all levels, giving an overall memory estimate of order $O(3(N+1))$ per eigenvector, for problems in 2D and 3D. If q eigenvectors are processed simultaneously, the memory is of order $3q(N+1)$, while for a sequential processing, the memory would be only $qN+2$, since only one right hand side and a single coarse storage is required for all eigenvectors. It is not necessary to process all eigenvectors simultaneously but only the eigenvectors in each cluster. This can lead to significant memory savings. The FMG work requires $O(N)$ operations per eigenvector. The work performed on the coarsest levels should be added to these estimates. In the performed tests, where several (up to tens) eigenvectors were sought, the coarse level work was usually a fraction of the finest level work. If exact orthogonality of eigenvectors is needed on the finest levels, then orthonormalizations or projections may be required within the finest level degenerate or closely clustered eigenspaces. If separation is employed on fine levels inside clusters, the work becomes $O(q^2N)$ operations per cluster of q eigenvectors. However, as can be seen in the computational examples, accurate orthogonality inside degenerate clusters may be obtained by coarse level separation also.

In the computational examples presented here, a complexity of $O(qN)$ for q eigenvectors is obtained.

IV. COMPUTATIONAL SOLUTIONS OF THE SCHRÖDINGER EIGENVALUE PROBLEM IN 2D AND 3D

Computational examples for the Schrödinger eigenvalue problem in 2D and 3D are presented to show the central difficulties related to clusters and mixing, and to illustrate the efficiency of the presented techniques in overcoming these difficulties. In the first example the following difficulties are present: existence of clusters with very close and equal eigenvalues, the cluster structure is not the same on the different levels, and the coarse level representation of the solutions is poor. The adaptive FMG algorithm is described in detail for this case. The second example shows that it is enough to treat the clusters in a sequential manner, and that separation has to be done within each cluster only in order to obtain good convergence and accurate separation. The third example shows that the new MGP may be performed on the coarsest levels, even in cases with closely clustered or equal eigenvalues, thus reducing the computational work significantly. The last example shows that the same efficiency as for problems in 2D is obtained for problems in 3D, even for cases with close and equal eigenvalues.

In all examples the Schrödinger eigenvalue problem

$$(\Delta - V)u = \lambda u \tag{21}$$

with periodic boundary conditions, defined on $\Omega = [0, a]^d$ (d=2 or 3), where $a = 2\pi/10$, is considered. The ith eigenvalue and eigenvector will be denoted next by λ_i and v_i. The potentials V are chosen such that distributions of eigenvalues with special difficulties are obtained. The usual second order finite difference discretization of the Laplacian on rectangular grids is used, although higher order discretizations can be used as well. Richardson type extrapolations based on the sequence of solutions obtained on the different levels can be used to obtain higher order accuracy. During the MG cycles, linear interpolation is used, while in the FMG, when passing to the next new finest level, local cubic interpolation is used. Gauss-Seidel type relaxations in red-black ordering are used during the cycles, and Kaczmarz and Richardson relaxations are used on coarsest levels. The symmetry of the chosen potentials is used to provide the clusters and the mentioned difficulties, but the symmetry is not used to simplify the problems or the algorithm.

Example I: Adaptive algorithm

A problem in 2D with the potential $V(x, y) = 5 + 3\sin(10x)$ is presented. The first $q = 12$ eigenvalues were required, and have been approximated using an adaptive 1-FMG-V(1,1) algorithm where the coarsest level was a 4×4 grid. The results are presented in Tables I and II.

The boxes in Table I show the clusters of close or equal eigenvalues (with minus sign) found by the algorithm (the formats are chosen to outline the equal digits in clusters). The cluster structure on the different levels is not the same, i.e., the level 2 cluster structure differs from the level 1 cluster structure. The cluster of six eigenvalues on level 1 $\{\lambda_6 - \lambda_{11}\}$, with multiplicities 1 - 4 - 1, has no correspondence on level 2. The first level eigenvalues are poor approximations of the second level eigenvalues. The eigenvalue λ_{16} on first level is very close to the eigenvalues $\{\lambda_{10} - \lambda_{13}\}$ on the second level. Such cross correspondences give rise to serious convergence difficulties for algorithms which do not treat them. The coarse level eigenvectors are poor approximations of the fine level eigenvectors.

The algorithm described in Sec. III D is used. To clarify the adaptive flow of the algorithm, a full history of the computation is given.

The algorithm started on level 1 adding eigenvectors until the cluster containing λ_{12} was completed. The last

TABLE I. The first 16 eigenvalues (E) of the discretized Schrödinger eigenvalue problem in 2D, on five levels, computed by a 1-FMG-$V(1,1)$ adaptive algorithm. The boxes represent the clusters of eigenvalues obtained on each level at the end of the last MG cycle. The different formats show the common digits of eigenvalues in each cluster. The notation $[n]$ stands for multiplication by 10^n.

E	level 1	level 2	level 3	level 4	level 5
1	0.496347395806[+1]	0.495721389176[+1]	0.495552134150[+1]	0.495509317773[+1]	0.495498173425[+1]
2	0.860204208719[+2]	0.999213342469[+2]	0.103677004418[+3]	0.104634633842[+3]	0.104874695012[+3]
3	0.860204208719[+2]	0.999213342469[+2]	0.103677004418[+3]	0.104634633842[+3]	0.104874695012[+3]
4	0.860569469139[+2]	0.9995[+2]	0.10371[+3]	0.1046[+3]	0.10491[+3]
5	0.860569469139[+2]	0.99998[+2]	0.10375[+3]	0.1047[+3]	0.10495[+3]
6	0.1670[+3]	0.194919376181[+3]	0.202435153808[+3]	0.204351758395[+3]	0.204831900326[+3]
7	0.167113893828[+3]	0.194919376181[+3]	0.202435153808[+3]	0.204351758395[+3]	0.204831900326[+3]
8	0.167113893828[+3]	0.194962161804[+3]	0.202479632146[+3]	0.204395[+3]	0.204876918643[+3]
9	0.167113893828[+3]	0.194962161804[+3]	0.202479632146[+3]	0.204396[+3]	0.204876918643[+3]
10	0.167113893828[+3]	0.329185001547[+3]	0.384812002762[+3]	0.399841022256[+3]	0.403673103803[+3]
11	0.16715[+3]	0.329185001547[+3]	0.384812002762[+3]	0.399841022256[+3]	0.403673103808[+3]
12	0.248170840742[+3]	0.329227787655[+3]	0.3848590736[+3]	0.399888846[+3]	0.403720980600[+3]
13	0.248170840742[+3]	0.329227787656[+3]	0.3848590739[+3]	0.399888846[+3]	0.403720980600[+3]
14	0.248207366784[+3]	0.424191908011[+3]	0.483580557031[+3]	0.499567983067[+3]	
15	0.248207366784[+3]	0.424295844705[+3]			
16	0.329264313697[+3]				

eigenvalue found, λ_{16}, belongs to the next cluster, confirming the completeness of the last sought cluster. On level 1, λ_{12} belongs to a cluster consisting of two degenerate subspaces, each of dimension 2, and the eigenvalues corresponding to these degenerate subspaces are close to within $\sim 10^{-4}$ relative difference.

The relevant eigenvectors $\{v_1, \ldots, v_{15}\}$ were interpolated to level 2 where they provided initial guesses for the level 2 problem. Here the completion of clusters restarted but this time working with the cluster structure from level 1 and using two level cycles. A test was done for the efficiency of a simultaneous cycle with fine level projection. The cycle was performed to provide first approximations of the level 2 eigenvalues. The cluster structure and eigenvalues obtained were compared with the ones of level 1. Since the agreement was not satisfactory, except for v_1, a cluster completion algorithm started with v_2. The completion continued until the complete cluster containing the last sought eigenvector was obtained (e.g., for level 2, the desired v_{12} belongs to the cluster $\{v_{10} - v_{13}\}$, the completion was ensured by the far value of λ_{14}). Then the relevant eigenvectors were updated by a few cycles.

The solution obtained on level 2 was interpolated to level 3, where a cluster completion test was satisfied only by the first cluster, v_1. The cluster completion algorithm was applied to the remaining eigenvectors (using robustness tests and the cluster completion tests). These resulted in a few cycles per eigenvector. The parameters l_c and l_p were found in the following way: (1) for the first cluster $\{v_1\}$, the values were obtained from the previous level since this cluster was stabilized from level 2;

TABLE II. The residuals of the 16 eigenvectors (E) of the discrete Schrödinger eigenvalue problem in 2D, on five levels, computed by a 1 -FMG-$V(1,1)$ adaptive algorithm. The residuals in the left column are computed after the interpolation to the new fine level, and the residuals in the right column are computed at the end of work on each level, during the FMG. The decrease of the residuals by a factor of 4 from one level to the next (on fine levels, left column) indicates a second order convergence towards the differential solution. The left columns show the convergence factor of order 10^{-2} for the first fine level $V(1,1)$ cycle. The notation $[\pm n]$ stands for multiplication by $10^{\pm n}$.

E	level 1		level 2		level 3		level 4		level 5	
1	0.48[+2]	0.37[-13]	0.69[+0]	0.97[-13]	0.22[+0]	0.30[-12]	0.60[-1]	0.64[-4]	0.15[-1]	0.14[-4]
2	0.53[+2]	0.44[-13]	0.30[+2]	0.14[-12]	0.11[+2]	0.35[-12]	0.30[+1]	0.86[-3]	0.76[+0]	0.73[-4]
3	0.61[+2]	0.38[-13]	0.30[+2]	0.80[-13]	0.11[+2]	0.29[-12]	0.30[+1]	0.86[-3]	0.76[+0]	0.73[-4]
4	0.66[+2]	0.68[-13]	0.30[+2]	0.17[-12]	0.11[+2]	0.30[-12]	0.30[+1]	0.54[-2]	0.76[+0]	0.11[-2]
5	0.55[+2]	0.39[-13]	0.30[+2]	0.24[-12]	0.11[+2]	0.45[-12]	0.30[+1]	0.54[-2]	0.76[+0]	0.11[-2]
6	0.52[+2]	0.12[-12]	0.11[+3]	0.32[-12]	0.16[+2]	0.35[-12]	0.44[+1]	0.42[-2]	0.11[+1]	0.82[-3]
7	0.59[+2]	0.31[-12]	0.45[+2]	0.54[-11]	0.16[+2]	0.32[-1]	0.44[+1]	0.42[-1]	0.11[+1]	0.82[-3]
8	0.61[+2]	0.20[-12]	0.45[+2]	0.57[-11]	0.16[+2]	0.41[-12]	0.44[+1]	0.39[-2]	0.11[+1]	0.83[-3]
9	0.62[+2]	0.17[-12]	0.45[+2]	0.71[-11]	0.16[+2]	0.33[-12]	0.44[+1]	0.1[-2]	0.11[+1]	0.93[-3]
10	0.73[+2]	0.13[-12]	0.45[+2]	0.15[-9]	0.12[+3]	0.30[-9]	0.43[+2]	0.72[-8]	0.12[+2]	0.33[-2]
11	0.58[+2]	0.34[-12]	0.11[+3]	0.41[-9]	0.12[+3]	0.16[-9]	0.43[+2]	0.20[-8]	0.12[+2]	0.33[-2]
12	0.54[+2]	0.39[-12]	0.12[+3]	0.81[-11]	0.12[+3]	0.26[-9]	0.43[+2]	0.21[-5]	0.12[+2]	0.29[-1]
13	0.51[+2]	0.70[-12]	0.12[+3]	0.50[-4]	0.12[+3]	0.50[-9]	0.43[+2]	0.16[-5]	0.12[+2]	0.29[-1]
14	0.44[+2]	0.96[-12]	0.12[+3]	0.19[-5]	0.12[+3]	0.17[-6]	0.44[+2]	0.34[-2]		
15	0.53[+2]	0.16[-12]	0.12[+3]	0.55[+1]						
16	0.69[+2]	0.19[-6]								

(2) for clusters 2 and 3 $\{v_6 - v_9\}$ and $\{v_{10} - v_{13}\}$, l_c and l_p were taken from level 2 since these clusters became stabilized after the cluster completion on level 3; (3) robustness tests were used for cluster 4 since the eigenvalues $\{\lambda_{10} - \lambda_{13}\}$ on level 3 and the corresponding ones from level 2 were not close enough. Then one cycle $V(1,1)$ was performed for each cluster.

On level 4, the first three clusters, eigenvectors $\{v_1 - v_9\}$ became stabilized, and their parameters were taken from level 3. The cluster completion algorithm was applied to cluster 4, $\{v_{10} - v_{13}\}$, where a few cycles were sufficient, and the parameters were taken from level 3 since the cluster became stabilized after the cycles. Then a $V(1,1)$ cycle was performed for each cluster.

On level 5, the cluster completion test was satisfied by all relevant eigenvectors $\{v_1 - v_{13}\}$, all clusters being stabilized from previous levels. A $V(1,1)$ cycle was performed for each cluster. The l_c and l_p for the separate clusters, in the final cycles, on levels 3, 4, 5, were found as follows: for $\{v_1\}$: $l_c = l_p = 1$, for the other clusters, containing $\{v_2, \ldots, v_{13}\}$, $l_c = l_p = 2$ were obtained (a test for the asymptotic convergence rate, for cluster $\{v_{10} - v_{13}\}$, may lead to $l_c = l_p = 3$, but such a test was not used in this run).

The additional last eigenvector obtained in the cluster completion test, used just to ensure that the previous cluster was complete, was not needed and not used in further steps. Usually its convergence was poor since the algorithm did not separate it from the next eigenvectors in its cluster, e.g., on level 2, λ_{14} was not separated from the next seven eigenvectors with close eigenvalues.

The left columns, in Table II, show the residuals after the cubic interpolation in the FMG. These residuals decrease with a factor of 4 (for fine levels) from one level to the next, indicating a second order convergence towards the differential solution. The right columns, for each level in Table II, show the residuals at the end of the cycle in the 1-FMG, on each level, demonstrating a convergence factor of order 10^{-2} for the first cycle on fine levels 4 and 5.

A simultaneous cycle for all clusters with separation on the coarsest common level for all clusters (here level 2) would improve the efficiency of the first cycle but this was not needed. (This also would improve the scalar products which resulted of order 10^{-4} after first cycle in the FMG, in this case. Accurate orthogonality is obtained by the algorithm described in the next example.)

This algorithm is of order $O(qN)$ if one does not use fine level separation inside the clusters. The adaptive coarse level work on levels 1,2, took approximately 1/6 of the total computer time and on levels 1,2,3 approximately 1/4 of the total computer time. This is a fixed time and it would be equivalent to 1/16 of the total computer time if level 6 would be employed too.

Example II: Fine level separation

In this example, the potential $V(x,y) = 5+3\sin(10x)+2\cos(10y)$ causes a further splitting of the eigenvalues. The clusters were treated sequentially and the projection for each cluster was performed on the finest level to provide accurate finest level separation inside clusters. The results, for nine eigenvectors, are presented in Tables III and IV. A 10-FMG-$V(1,1)$ algorithm was used to show the constant convergence of order 0.1 rate per cycle. The coarsest relaxation level for clusters 2 and 3 was level $l_c = 2$ and for the first eigenvector was level $l_c = 1$. On levels 1 and 2 the adaptive algorithm and few cycles were used. On finer levels, l_c and l_p were taken from level 2. All eigenvectors came out accurately orthogonal (10^{-13} scalar products on level 4). This shows that it is enough to perform separation only within clusters. On levels 3 and 4, 10 $V(1,1)$ cycles reduced the residuals for all eigenvectors by 10 orders of magnitude.

Example III: Coarse level separation, 2D and 3D

In the next two examples (Tables V and VI) we show that separation on the coarsest level ($l_c = l_l = 1$) may be

TABLE III. The residuals of the first nine eigenvectors (E) of the discretized Schrödinger eigenvalue problem in 2D, on four levels, computed by a 10-FMG-$V(1,1)$ adaptive algorithm. The residuals in the left column are computed after the interpolation to the new fine level, and the residuals in the right column are computed at the end of work on each level, during the FMG. The residuals decrease in 10 MG cycles with 10 orders of magnitude (to values of 10^{-10}) on the fine levels. The notation $[\pm n]$ stands for multiplication by $10^{\pm n}$.

E	level 1		level 2		level 3		level 4	
1	0.48[+2]	0.14[-13]	0.83[+0]	0.12[-12]	0.27[+0]	0.11[-11]	0.72[-1]	0.41[-11]
2	0.46[+2]	0.83[-9]	0.30[+2]	0.48[-9]	0.11[+2]	0.42[-12]	0.30[+1]	0.21[-11]
3	0.52[+2]	0.29[-9]	0.30[+2]	0.12[-8]	0.11[+2]	0.75[-12]	0.30[+1]	0.52[-11]
4	0.56[+2]	0.56[-10]	0.30[+2]	0.73[-9]	0.11[+2]	0.93[-12]	0.30[+1]	0.56[-11]
5	0.54[+2]	0.85[-9]	0.30[+2]	0.55[-8]	0.11[+2]	0.17[-11]	0.30[+1]	0.12[-10]
6	0.53[+2]	0.57[-2]	0.11[+3]	0.40[-5]	0.16[+2]	0.61[-12]	0.44[+1]	0.16[-10]
7	0.53[+2]	0.68[-11]	0.45[+2]	0.57[-5]	0.16[+2]	0.10[-11]	0.44[+1]	0.39[-10]
8	0.41[+2]	0.13[-10]	0.45[+2]	0.29[-5]	0.16[+2]	0.82[-12]	0.44[+1]	0.33[-10]
9	0.43[+2]	0.80[-2]	0.11[+3]	0.14[-5]	0.16[+2]	0.83[-12]	0.44[+1]	0.48[-10]

TABLE IV. The first nine eigenvalues (E) of the discretized Schrödinger eigenvalue problem in 2D from the Table III example, computed by a 10-FMG-$V(1,1)$ adaptive algorithm. The notation $[+n]$ stands for multiplication by 10^n.

E	level 1	level 2	level 3	level 4
1	-0.494698319454[+1]	-0.493789518604[+1]	-0.493543833853[+1]	-0.493481214576[+1]
2	-0.860202443918[+2]	-0.9991[+2]	-0.10367[+3]	-0.10463[+3]
3	-0.860202443918[+2]	-0.999361[+2]	-0.1036931[+3]	-0.104650[+3]
4	-0.860406326305[+2]	-0.999362[+2]	-0.1036937[+3]	-0.104651[+3]
5	-0.860406326305[+2]	-0.9997[+2]	-0.1037[+3]	-0.10469[+3]
6	-0.1670[+3]	-0.19491[+3]	-0.20243[+3]	-0.20434[+3]
7	-0.167113893828[+3]	-0.19493[+3]	-0.20245[+3]	-0.20436[+3]
8	-0.167113893828[+3]	-0.19495[+3]	-0.20247[+3]	-0.20439[+3]
9	-0.16713[+3]	-0.19497[+3]	-0.20249[+3]	-0.2044[+3]

sufficient for providing accurate finest level separation, even for clusters containing very close and degenerate eigenvalues. In the degenerate clusters the eigenvectors were not orthogonalized on the finest levels but resulted so from the FMG. This implies an $O(qN)$ algorithm even for close clustered cases.

Table V shows results for a problem in 2D with a potential $V(x,y) = 2 + 0.1\sin(10x + 10y)$, which produces a splitting of the first cluster of four eigenvalues into two degenerate clusters, whose eigenvalues are close to within 10^{-4}. A second order approximation was obtained by a 1-FMG-$V(1,1)$ algorithm with an asymptotic convergence rate of 0.1 per cycle on fine levels. The 13 equal digits of the degenerate eigenvalues, on all levels, are seen. On level 5, eight cycles were performed to show the constant convergence rate per cycle (see cycles 3 and 8, where the convergence rate is accurately 0.1). The eigenvectors came out accurately orthogonal, even in the degenerate eigenspaces, although the projection was performed only on the coarsest level (the eigenvectors scalar products being of order 10^{-13} on level 5).

The same efficiency is obtained for problems in 3D, as can be seen in Table VI. The potential $V(x,y,z) = 2 + \sin(20x + 10y - 10z)$ determines a cluster of six close

TABLE V. The eigenvalues and residuals of the first five eigenvectors of the discretized Schrödinger eigenvalue problem in 2D, on five levels, computed by a 1-FMG-$V(1,1)$ algorithm. On the finest level, eight cycles were performed to show the asymptotic convergence rate of 0.1 for the cluster presenting degenerate and very close eigenvalues. The separation was performed on the coarsest level, each cycle performed two Gauss-Seidel relaxations per level [$V(1,1)$ cycles]. The notation $[\pm n]$ stands for multiplication by $10^{\pm n}$.

cycle	vector	first residual	last residual	eigenvalue
		Level 4		
1	1	0.18[-2]	0.13[-3]	-0.19999752449715[+1]
	2	0.30[+1]	0.43[-2]	-0.10162979203934[+3]
	3	0.30[+1]	0.43[-2]	-0.10162979203934[+3]
	4	0.30[+1]	0.43[-2]	-0.10172931140738[+3]
	5	0.30[+1]	0.43[-2]	-0.10172931140738[+3]
		Level 5		
1	1	0.46[-3]	0.36[-4]	-0.19999750026388[+1]
	2	0.76[+0]	0.40[-3]	-0.10186970728937[+3]
	3	0.76[+0]	0.40[-3]	-0.10186970728937[+3]
	4	0.76[+0]	0.40[-3]	-0.10196970729590[+3]
	5	0.76[+0]	0.40[-3]	-0.10196970729590E+3]
3	1	0.35[-5]	0.33[-6]	-0.19999749801202[+1]
	2	0.27[-4]	0.26[-5]	-0.10186970049930[+3]
	3	0.27[-4]	0.26[-5]	-0.10186970049930[+3]
	4	0.27[-4]	0.26[-5]	-0.10196970049780[+3]
	5	0.27[-4]	0.26[-5]	-0.10196970049780[+3]
8	1	0.97[-11]	0.26[-10]	-0.19999749799142[+1]
	2	0.29[-9]	0.31[-10]	-0.10186970048459[+3]
	3	0.29[-9]	0.31[-10]	-0.10186970048459[+3]
	4	0.29[-9]	0.31[-10]	-0.10196970048302[+3]
	5	0.29[-9]	0.31[-10]	-0.10196970048302[+3]

TABLE VI. The eigenvalues and residuals of the first seven eigenvectors of the discretized Schrödinger eigenvalue problem in 3D, on three levels, computed by a 1-FMG-$V(1,1)$ algorithm. The boxes show the clusters of very close and equal eigenvalues. The separation was performed on the coarsest level, each cycle performed two Gauss-Seidel relaxations per level [$V(1,1)$ cycles]. The notation [$\pm n$] stands for multiplication by $10^{\pm n}$.

cycle	vector	first residual	last residual	eigenvalue
		Level 3		
3	1	0.15[-2]	0.10[-3]	-0.19991341655960[+1]
	2	0.25[-1]	0.25[-2]	-0.10072012662990[+3]
	3	0.25[-1]	0.25[-2]	-0.10072012662990[+3]
	4	0.23[-1]	0.21[-2]	-0.10072068269198[+3]
	5	0.23[-1]	0.21[-2]	-0.10072068269198[+3]
	6	0.23[-1]	0.21[-2]	-0.10072068269198[+3]
	7	0.23[-1]	0.21[-2]	-0.10072068269198[+3]

eigenvalues. The convergence rate of 0.1 in cycle 3 and the first six common digits of the eigenvalues in the cluster are observed. It is a close similarity between the 2D and 3D results.

These examples indicate that the reduction of complexity of the new algorithms from $O(q^2N)$ to $O(qN)$ holds for a class of problems which include difficult cases as shown.

A. Observations on the algorithms

Observations and details of the algorithms, not introduced before in order to keep the expositoin simpler, are mentioned in this section.

(1) When the operators A_i are obtained by discretizing differential problems, it is not needed to compute and store A_i.

(2) In the shown examples, only local operations are needed in relaxations, transfers, and corrections, operations which involve only the unknown at each point and its neighbors.

(3) Different relaxations can be used in the algorithms, like damped Jacobi, Gauss-Seidel, Richardson, Kaczmarz, block relaxations; see for example [27].

On the GRRP

(1) For the GRRP, the matrix A is not needed, but it is enough to provide a procedure that calculates AU. No operations are performed on the matrix A, e.g., to precondition or bring A to a special form.

(2) The vectors U^T in (14) can be replaced by a more general set of vectors Y^T.

(3) Solutions (E, Λ) of (13) may not exist. However, as in the usual Rayleigh-Ritz projection, an E and a Λ can be found such that the projection of the residual of (13) on the columns of U is minimized, i.e., performing GRRP.

(4) The complexity of solving the generalized eigenvalue problem in GRRP on the coarsest level, for q vectors, is of order $O(q^3)$, which is often much smaller than $O(q^2N)$, the cost of computing E and Λ on a fine level. By this procedure the fine level eigenvalues are computed on coarse levels. The coarse level updated eigenvalues enhance the efficiency of MG cycles.

On MG solver cycles

(1) In the presented form, the MG solver cycles update the solutions simultaneously but MG solver cycles can be performed sequentially, in turns for each eigenvector or for each cluster.

(2) Other types of solver cycles can be defined in the same way, incorporating different sequences of visiting the levels, e.g., W type cycles [27]. The usage of W cycles was generally not needed in algorithms, although in some cases the convergence rate for W cycles was better, but also the work increased by W cycles. Sometimes W cycles increase the mixing of solutions.

(3) Additional procedures can be performed during the MG cycles, like updating the eigenvalues by Rayleigh quotients.

On MG combined cycles

(1) At different stages of the MG combined cycle, for example on the coarsest level only, the solutions can be normalized using an FAS normalization, i.e., setting $||U|| = T$, where T is a scalar computed like in (4) where FU is replaced by $||U||$. This can be done after the backrotations but normalization of solutions can be performed also on the finest level. Accurate normalization, if needed, can be performed as the last step on the finest level, e.g., in the last cycle of the FMG. This does not change the complexity of the algorithm.

(2) The MGP is also in agreement with the general principle of performing global steps on coarse levels.

On adaptive FMG algorithms

(1) On coarse levels, only a part of the sought eigenvectors may be approximated, e.g., if the coarse levels cannot approximate more eigenvectors. More eigenvec-

tors can be added and processed on finer levels.

(2) Transfers from fine to coarse levels may not conserve the dimensions of the transferred subspaces. This difficulty is handled by robustness tests (which do not detect the loss of dimension but the inefficiency of the MG cycles in such situations).

(3) The separation of solutions $U_j = U_j E$ cannot be combined for any E with the usual FAS correction of U_i, (5), since this would usually destroy an exact solution U_i, e.g., if E is not the identity but a permutation matrix. To overcome this difficulty we propose a *backrotation FAS correction:*

$$U_i = U_i E + I_j^i(U_j - I_i^j U_i E), \qquad T_i = T_i E \ . \tag{22}$$

In this correction the right hand side T is updated also. In (22) the multiplication $U_i E$ is of the same order of work as needed for a Rayleigh-Ritz separation for U_i. Still, the cheaper correction (5) can be used instead of (22) when solutions are sufficiently accurate and using backrotations. This is shown by the computational examples too. The correction (22) can be used on coarse levels and when the solutions are not well enough approximated.

(4) Computational difficulties may occur for degenerate subspaces when any matrix E is a solution of GRRP. In such cases, during an MG combined cycle, E will mix the coarse solutions and destroy the fine ones after interpolation (see, for example, that orthogonality will be destroyed). Similar or worse difficulties are obtained for clusters of eigenvalues since the algorithms act on approximated clusters as on degenerate spaces, i.e., mixing solutions. These difficulties are treated by the backrotations, as shown in the computational examples.

(5) In adaptive MGP the clusters are treated sequentially and within each cluster the solutions are treated simultaneously by a combined MG cycle solve MGP.

(6) A *simultaneous* cycle for several clusters is obtained by grouping the clusters into a single larger cluster and applying adaptive MGP to it. This can be used to improve the separation between clusters and it is particularly useful on coarse levels at initial stages of the FMG when clusters are not separated well enough.

(7) If for each cluster the GRR-BR projection is performed on the finest level, the algorithm still requires less work than an algorithm performing the fine level projection for all clusters simultaneously.

(8) If mixing occurs on coarse levels (as often happens since here the solutions are poorly represented), one may expect an algorithm using fine level separation to have a poor efficiency or even not to converge. A coarse level separation usually restores the convergence or improves the efficiency in such cases.

(9) For well separated eigenvalues the projection may not be needed except at initial coarse level stages of the FMG, later the eigenvalues determine the separation of eigenvectors via the MG solver cycles. The same holds for well separated clusters which do not need a simultaneous separation. This is especially useful for a larger number of eigenvectors, belonging to well separated clusters (e.g., already for 10 eigenvectors the improvement can be noticeable).

(10) The number of relaxations can vary with level and cluster. In the computational tests one or two relaxations per fine level passing were performed.

(11) In particular cases, parameters of subroutines such as number of relaxations and parameters of relaxations can be obtained by Fourier analysis. Robustness tests allow us to find such parameters in general cases.

(12) The computation of eigenvalues and eigenvectors may be simplified if additional information on the problem or on the solutions is known. For example, such information may include information on the problem, e.g., symmetry of the problem, ways to precondition the problem, or to reduce the problem to a simpler one, information on eigenvalues such as the cluster structure, the multiplicity of eigenvalues, approximate eigenvalues, largest or smallest eigenvalue, and information on eigenvectors, e.g., space spanned by eigenvectors, approximate eigenvectors, and solutions of a related problem. Clusters of close or equal eigenvalues may appear due to symmetries of the problem. Often in such cases one can take advantage of the symmetry and reduce the problem to one in which mixing of eigenvectors can be avoided. We refer to [6], where symmetry and other information on the problem were successfully used in solving real problems by an MG approach. The reduction based on symmetry may be problematic when the cluster structure is not known, when the cluster structure differs on different levels, or when other difficulties such as eigenvalue cross-correspondences are present (see, for example, Table I). Mixing may appear for well separated eigenvalues also. Clusters of close eigenvalues may appear in nonsymmetric cases also, as we show in [11], where real problems are solved. In the presented MG approach, we consider the cluster structure unknown, the problems are not reduced based on symmetry, and other additional information is generally not needed. We tried to develop a general adaptive algorithm which will be able to handle difficulties common to a large class of eigenvalue problems.

(13) In the computational examples the potentials have the form $V = c + W$, where the constant c is a shift introduced for convenience to make V positive in the Gauss-Seidel relaxation. Adding c to the computed eigenvalues we obtain the eigenvalues of the operator $\Delta - W$. In all examples the first eigenvalue of $\Delta - W$ is very close to 0 (i.e., 0.045, 0.065, 0.000 025, 0.000 86), showing that there is no additional difficulty in computing eigenvalues close to 0 (we can always use a shift to bring an eigenvalue to 0 or to shift it from 0). No changes of the algorithm are needed for computing eigenvalues close to 0 (e.g., the case $V = 0$, which implies the 0 eigenvalue, is treated in the same way). The singularity or close to singularity of the operators $A - \lambda I$ are treated by Kaczmarz type relaxations on coarse levels. Kaczmarz type relaxations have the advantages of being local and simple. Another coarse level treatment may be used also; see, for example, [6]. The difficulties mentioned in [6], i.e., divergence of smooth components and large interpolation errors on coarse levels, relate to mixing introduced by relaxations and transfers. The MGP and the adaptive treatment did

overcome these difficulties in the computational examples, as well as in other hard cases [11,10].

V. CONCLUSIONS

A robust and efficient MG algorithm to compute a few eigenvectors and the corresponding eigenvalues for large scale eigenvalue problems has been developed. The algorithm's robustness results from the adaptive completion and treatment of clusters, the simultaneous treatment of solutions in each cluster, and from tests which monitor the algorithm's convergence and efficiency. The algorithm treats central difficulties such as the poor solution representation on coarse levels, the existence of clustered eigenvalues, the approximation of incomplete clusters, and the mixing of approximated eigenvectors during the solution process. Its eigenvector separation efficiency stems from an MG projection technique which is a generalization of the Rayleigh-Ritz projection, combined with backrotations.

In the cases when the algorithm properly separates the eigenvectors on coarse levels, its complexity is of $O(qN)$ for q eigenvectors of size N on the finest level. The numerical tests showed that an accurate fine level separation was obtained by the coarse level projection, even for problems with very close or equal eigenvalues.

The results of the numerical tests for Schrödinger eigenvalue problems, in 2D and 3D, show that the algorithm achieved the same accuracy, using the same amount of work (per eigenvector), as the Poisson MG solver. A second order approximation is obtained using the five-point in 2D and nine-point in 3D discretized Laplacian, by 1-FMG-$V(1,1)$ in $O(qN)$ work. The work was of order of a few (as 8) fine level Gauss-Seidel relaxations per eigenvector. The adaptive work was only a fraction of the fine level work and enhanced the efficiency of the fine level cycles. Constant convergence rate per cycle was obtained for the presented cases. The robustness of the algorithm has been demonstrated on problems with eigenvalue distributions that present special difficulties.

ACKNOWLEDGMENTS

This research was made possible in part by funds granted to Shlomo Ta'asan, by a program sponsored by the Charles H. Revson Foundation. Both authors were supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), Hampton, Virginia.

[1] A. Brandt, S. McCormick, and J. Ruge, SIAM. J. Sci. Stat. Comput. **4**, 244 (1983).
[2] W. Hackbusch, *Multigrid Methods and Applications*, Springer Series in Computational Math. Vol. 4 (Springer-Verlag, Berlin, 1985).
[3] W. Hackbush, SIAM J. Numer. Anal., **16**, 201 (1979).
[4] S. F. McCormick, Math. Comp. **36**, 485 (1981).
[5] R. E. Bank, SIAM J. Numer. Anal. **19**, 886 (1982).
[6] K. Davstad, J. Comp. Phys., **99**, 33 (1992).
[7] J. Mandel and S. McCormick, J. Comp. Phys. **80**, 442 (1989).
[8] S. F. McCormick, *Multilevel Projection Methods for Partial Differential Equations* (SIAM, Philadelphia, 1992).
[9] S. Costiner and S. Ta'asan, Technical Report CS92-07, The Weizmann Institute of Science, Rehovot, Israel, 1992 (unpublished).
[10] S. Costiner and S. Ta'asan, ICASE Report 94-91, ICASE, NASA Langley Research Center, Hampton, VA, 1994 (unpublished).
[11] S. Costiner, F. Manolache, and S. Ta'asan, IEEE Trans. Microwave Theory Tech. **43**, 48 (1995).
[12] L.Y. Zaslavsky Applied Math. Comput. **53**, 13 (1993).
[13] L.Y. Zaslavsky, SIAM J. Sci. Comput. (to be published).
[14] F. Bourquin and F. d'Hennzel, in *Proceedings of the Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations* (SIAM, Philadelphia, 1992).
[15] J.C. Luo, in *Proceedings of the Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations* (SIAM, Philadelphia, 1992).
[16] W. Kerner, J. Comp. Phys. **85**, 1 (1989).
[17] B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice-Hall, Englewood Cliffs, NJ, 1980).
[18] J.H. Wilkinson, *The Algebraic Eigenvalue Problem* (Clarendon Press, Oxford, 1965).
[19] G. H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, MD, 1989).
[20] P.J. Nikolai, ACM Trans. Math. Software, **5**, 118 (1979).
[21] H. Rutishauser, Numer. Math. **13**, 4 (1969).
[22] S. F. McCormick and T. Noe, J. Lin. Alg. Appl. **16**, 43 (1977).
[23] S. Ta'asan, J. Sci. Comput. **3**, 261 (1988).
[24] S. Costiner and S. Ta'asan, NASA - Report 194291, ICASE Report 93-35; ICASE, NASA Langley Research Center, Hampton, VA, 1993 (unpublished).
[25] S. Costiner and S. Ta'asan, ICASE Report 94-82; ICASE, NASA Langley Research Center, Hampton, VA, 1994 (unpublished).
[26] S. Ta'asan and S. Costiner, *Multilevel Techniques for Large Scale Eigenvalue Problems*, in Proceedings of the *Lanczos Centenary Conference, 1993* (North Carolina State University Press, City, 1994).
[27] A. Brandt, *Multigrid Techniques 1984 Guide with Applications to Fluid Dynamics*, GMD Studien Nr. 85. (Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1984).