

IN-64
181589
P-21

NASA Contractor Report 191491
ICASE Report No. 93-35

ICASE



A ROBUST MULTILEVEL SIMULTANEOUS EIGENVALUE SOLVER

Sorin Costiner
Shlomo Ta'asan

N94-13202

Unclas

G3/64 0181589

NASA Contract Nos. NAS1-19480, NAS1-18605
June 1993

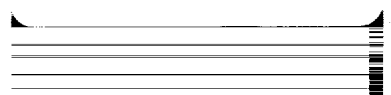
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association



National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

(NASA-CR-191491) A ROBUST
MULTILEVEL SIMULTANEOUS EIGENVALUE
SOLVER Final Report (NASA) 21 P





A Robust Multilevel Simultaneous Eigenvalue Solver

Sorin Costiner and Shlomo Ta'asan[†] *

Department of Applied Mathematics and Computer Science

The Weizmann Institute of Science, Rehovot, Israel

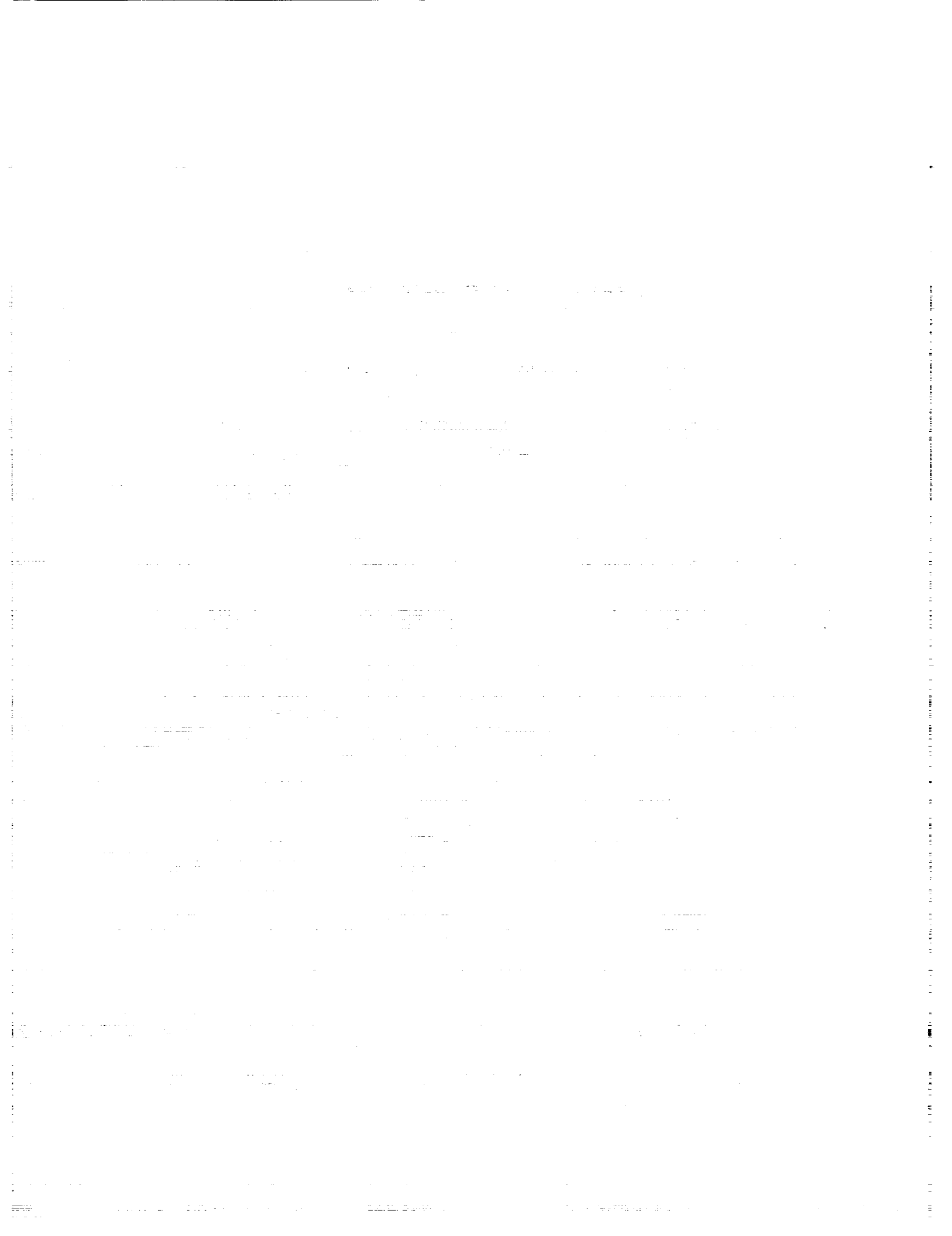
[†]Institute for Computer Applications in Science and Engineering

NASA Langley Research Center, Hampton, Va 23681, USA

Abstract

Multilevel (ML) algorithms for eigenvalue problems are often faced with several types of difficulties such as: the mixing of approximated eigenvectors by the solution process, the approximation of incomplete clusters of eigenvectors, the poor representation of solution on coarse levels and the existence of close or equal eigenvalues. Algorithms that do not treat appropriately these difficulties usually fail, or their performance degrades when facing them. These issues motivated the development of a robust adaptive ML algorithm which treats these difficulties, for the calculation of a few eigenvectors and their corresponding eigenvalues, presented in this paper. The main techniques used in the new algorithm include: the adaptive completion and separation of the relevant clusters on different levels, the simultaneous treatment of solutions within each cluster, and the robustness tests which monitor the algorithm's efficiency and convergence. The eigenvectors' separation efficiency is based on a new ML projection technique generalizing the Rayleigh Ritz projection, combined with a novel technique, the backrotations. These separation techniques, when combined with an FMG formulation, in many cases lead to algorithms of $O(qN)$ complexity, for q eigenvectors of size N on the finest level. Previously developed ML algorithms are less focused on the mentioned difficulties. Moreover, algorithms which employ fine level separation techniques are of $O(q^2N)$ complexity and usually do not overcome all these difficulties. Computational examples are presented where Schrödinger type eigenvalue problems in 2-D and 3-D, having equal and closely clustered eigenvalues, are solved with the efficiency of the Poisson multigrid solver. A second order approximation is obtained in $O(qN)$ work, where the total computational work is equivalent to only a few fine level relaxations per eigenvector.

*This research was made possible in part by funds granted to the second author through a fellowship program sponsored by the Charles H. Revson Foundation and in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 and NAS1-18605 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, Va 23681



1 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 EP can be approximated on several levels; only few eigenvalues and eigenvectors are sought. These features can be exploited naturally by multilevel (ML) solvers as has been done successfully by several authors (see for example [3] and [12]). Such solvers generally involve the discretization of the problem on a sequence of levels, relaxations employed on all levels, transfers of solutions and residuals from fine to coarse levels, interpolation of correction from coarse to fine levels, and often a fine level eigenvector separation technique.

Multilevel methods for EP sometimes encounter difficulties which make their robustness and efficiency questionable. Often such difficulties are due to several reasons which we classify as: (i) mixing of eigenvectors by the used procedures, (ii) incompleteness of a treated cluster, (iii) incompatible fine-coarse level representation of a cluster. Procedures such as relaxations, transfers and ML cycles can introduce or amplify an eigenvector in the error component of an approximated eigenvector. This we refer as eigenvector mixing. For example, it is known that ML procedures mix Fourier components which are the eigenvectors of many discretized differential operators. If the eigenvectors of an iteration operator do not coincide with the eigenvectors of the problem to be solved, then the iteration may mix the problem's eigenvectors. This difficulty occurs especially when clusters of eigenvectors with close or equal eigenvalues are approximated. Usually it is treated by simultaneous separation techniques, e.g., by a Rayleigh Ritz type projection. However, if not all eigenvectors which are mixed by the involved procedures are approximated, e.g. the clusters are not complete, then one can expect the separations to be inaccurate and inefficient. Thus a cluster will be called complete relative to a procedure if it contains a whole set of eigenvectors which are mixed by the procedure. Difficulties related to incomplete clusters may be treated by completing the clusters and processing them simultaneously. The incompatibility of clusters representation on different levels is another difficulty which has to be identified and taken care of. In particular, not all clusters can be approximated on an arbitrarily coarse level and the eigenvalues clustering may differ on various levels. These difficulties and their remedies suggest the following conclusions: (i) Clusters of eigenvectors should be treated simultaneously using a separation technique like the Rayleigh-Ritz projection, (ii) Clusters have to be completed, (III) Different clusters should be differently treated on different levels. This suggests that previously developed algorithms may fail in some standard situations. These include nonadaptive algorithms, algorithms which treat the eigenvectors sequentially in clusters, which do not complete the relevant clusters, or which do not take into account the inter-level eigenvector mixing. Even in cases when such algorithms work, their efficiency may be improved by an adaptive treatment or by coarse level separation techniques as used in this work.

This paper focuses on a more robust and efficient algorithm for the calculation of a few eigenvalues and their corresponding eigenvectors. Its development was guided by the above mentioned difficulties and their remedies. Beside its robustness, the algorithm achieves a better computational complexity than previously known ML eigenvalue algorithms which use fine level projections. The robustness of the present approach is based on the adaptive completion and separation of the relevant clusters on different levels; the simultaneous treatment of solutions for each cluster; and robustness tests which monitor the algorithm's convergence and efficiency. A central efficiency feature of the algorithm presented here results from the newly developed Generalized Rayleigh Ritz (GRR) projection and backrotation (BR), which employ the projection on coarse levels, adaptively with the cluster involved. This reduces in many cases the most time consuming part of the algorithms, namely, the $O(q^2N)$ fine level separation work, to $O(qN)$ for q eigenvectors of size N on finest level.

These ideas are combined in an FMG algorithm which first solves the problem on coarse levels, then interpolates the solutions to finer levels where they serve as an initial approximation to the corresponding finer level problems. On the currently finest level, clusters are identified and tested for completeness, completed if necessary, and improved by ML cycles using coarser levels. The eigenvalue equations is relaxed on each level followed by FAS transfers. Generalized Rayleigh Ritz projection and backrotations (GRR-BR) are employed on coarse levels usually, in order to separate eigenvectors within their clusters, and to keep the coarse level representation of the solutions as close as possible to the fine level solutions. This is done adaptively for different clusters on appropriate levels. On the level on which the algorithm starts, only a part of the sought eigenvectors are approximated usually, and more eigenvectors with their corresponding eigenvalues are added on finer levels.

In the examples presented here for the Schrödinger eigenvalue problem in 2-D and 3-D, two to four fine level relaxations per eigenvector were performed. Equal eigenvalues were calculated with more than ten decimal places and accurate results were obtained for very close eigenvalues as well. A second order approximation is obtained by 1-FMG-V(1,1) in $O(qN)$ work, for q eigenvectors of size N on the finest level.

The present approach can be extended to nonlinear eigenvalue problems, an example being presented in Costiner and Ta'asan [6].

We refer to the early works of Hackbusch [9] McCormick [17], Bank [1] for theory on multilevel eigenvalue solvers and first algorithms. A sequential ML algorithm for linear eigenvalue problems performing the projection on fine levels is presented in Brandt McCormick and Ruge [3]. More theory and algorithms on ML EP may be found in Hackbusch [12]. Zaslavski in [23], [24], uses an adaptive algebraic correction scheme cycle to compute the first eigenvector and its eigenvalue for the multigroup neutron diffusion equation. The elements of such a ML cycle, modified to the FAS form, can be used in the algorithm presented here. Our approach differs from previous ML approaches mainly by the following issues: emphasis on robustness, simultaneous

cluster processing, cluster completion, ML separation by GRR and BR, treatment of eigenvector mixing, treatment of close and equal eigenvalues. The ML projection idea was first introduced by Ta'asan [21]; backrotations were introduced in Costiner [5]. The combination of our technique with domain decomposition techniques is natural but was not analysed yet. For some domain decomposition techniques for eigenvalue problems see for example Bourquin and d'Hennezel [2], and Luo [14]. A review article on single level large-scale complex eigenvalue problems, containing many references, is Kerner [13]. For a theory on Ritz projections and on algebraic eigenvalue problems we refer to Parlett [19], Wilkinson [22], Golub and Van Loan [8]. The single level technique to obtain the eigenvectors by relaxations and projections is referred in different places as subspace, simultaneous or Ritz iterations. We refer to Nikolai [18], Rutishauser [20] and McCormick [17] for a single level algorithm and mathematical foundations.

The paper is organized as follows. Section 2 presents the GRR ML projection, the backrotations and the multilevel cycle. Section 3 presents and discusses the adaptive techniques such as the robustness tests, the cluster completion, and the adaptive FMG. Section 4 presents computational examples.

2 Multilevel Projection Techniques

One of the key elements of our algorithm is the ML projection used to separate eigenvectors corresponding to closely clustered and equal eigenvalues. To motivate the new projection method consider a fine level problem

$$A_h \bar{U}_h - \bar{U}_h \Lambda = 0 \quad (2.1)$$

where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_q)$ contains on diagonal the q sought eigenvalues corresponding to the desired eigenvectors which are the columns of \bar{U}_h . Assume that U_h consists of linearly independent combinations of eigenvectors belonging to the sought subspace. In this case, a Rayleigh-Ritz (RR) projection provides a Λ and a $q \times q$ invertible matrix E such that Λ and $\bar{U}_h = U_h E$ are solutions of (2.1). The FAS coarse level equation after performing the projection on the fine level becomes

$$A_{2h} U_{2h} E - U_{2h} E \Lambda = T_{2h}^h E \quad (2.2)$$

where $T_{2h}^h = A_{2h} I_h^{2h} U_h - I_h^{2h} A_h U_h$ is the usual FAS right hand side. The form of the coarse level solution $U_{2h} E = I_h^{2h} U_h E$ suggests that one may obtain E and Λ on the coarse level from 2.2. This can be done by a generalized version of the (RR) projection, presented next.

2.1 Generalized Rayleigh Ritz Projections and Backrotations

Solutions (E, Λ) of (2.2) may not exist if U_h is not a basis of $\text{span}(\bar{U}_h)$. However, as in the usual (RR) projection, if U_h approximates a basis of $\text{span}(\bar{U}_h)$, one finds E and Λ such that the projection of the residual of (2.2) on $\text{span}(U_{2h})$ is minimized, i.e., solving a $q \times q$ generalized eigenvalue problem as:

$$(U_{2h})^T (A_{2h} U_{2h} - T_{2h}^h) E = (U_{2h})^T U_{2h} E \Lambda \quad (2.3)$$

The process of solving for (E, Λ) given $(A_{2h}, U_{2h}, T_{2h}^h)$ is denoted by

$$(E, \Lambda) \leftarrow \mathbf{GRR}(A_{2h}, U_{2h}, T_{2h}^h) \quad (2.4)$$

The above projection is referred later on as generalized Rayleigh Ritz Projection (GRR) or as multilevel ML projection when several levels are involved. The GRR projection cannot be directly combined with the usual FAS correction $U_h = U_h + I_{2h}^h (U_{2h} - I_h^{2h} U_h)$, since it will change an exact fine level solution U_h , e.g. if E is not the identity but rather a permutation matrix. This difficulty can be solved by using a modified FAS correction such as

$$U_h = U_h E + I_{2h}^h (U_{2h} E - I_h^{2h} U_h E) \quad (2.5)$$

Note that (2.5) would lead to $O(q^2 N)$ operations, equivalent to a fine level projection work. Thus it is desirable to replace (2.5) with more efficient techniques. Other difficulties may occur for degenerate subspaces when any matrix can serve as a solution for E , thus, mixing and destroying orthogonality of fine level solutions for example.

A natural technique to fix these difficulties is introduced and used in combination with the multilevel projection. It is referred to as backrotation because of its geometrical meaning. As suggested by the above discussion, for degenerate subspaces the backrotation should produce block diagonal submatrices in E which are close to the identity matrix of the appropriate dimension and eliminate permutations of eigenvectors. A backrotation step will be further denoted by

$$(E, \Lambda) \leftarrow \mathbf{BackRotation}(E, \Lambda) \quad (2.6)$$

A particular backrotation algorithm is:

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 by
permuting the columns of E ,
and correspondingly the diagonal of Λ .
 - 4) Let F be a block diagonal matrix
whose diagonal blocks are the diagonal blocks of E ,
corresponding to the 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)

2.2 Multilevel Combined Cycles

In general, not all required eigenvectors can be well approximated on an arbitrarily coarse level. For example, fine level discretized Laplace operator eigenvectors corresponding to larger eigenvalues may not be representable on coarse levels. Moreover, the eigenvector cluster structure may differ on various levels. However, the efficient solution of the eigenvectors corresponding to small eigenvalues may often be done using coarse levels. A major difficulty is that solutions belonging to a cluster are often mixed by procedures, i.e. the procedures may regard linear combinations of solutions as a solution. This obstacle can be overcome by simultaneously treating all the cluster's eigenvectors and separating them on different levels.

Efficiency and convergence considerations require that the GRR projection should be done for different clusters on different levels. Moreover, the coarsest level used to treat a given cluster may not coincide with the level on which the GRR projection is done. Thus, the full algorithm depends on some parameters associated with each cluster, that determine the flow of the algorithm for that cluster. These parameters are determined adaptively during the solution process (as explained in the next section).

Following is a description of a basic ML cycle used in the adaptive algorithm presented in the next section.

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

$$U_k = (U_k^1, \dots, U_k^j) \quad (2.1)$$

where, each U_k^i approximates \bar{U}_k^i , the solution of

$$A_k \bar{U}_k^i = \bar{U}_k^i \Lambda^i + T_k^i \quad i = 1, \dots, j \quad (2.2)$$

where each \bar{U}_k^i is a matrix whose columns are eigenvectors and Λ^i is a diagonal matrix whose diagonal elements are the corresponding eigenvalues. Usually, on the finest level, $k = m$, $T_k = (T_k^1, \dots, T_k^j) = (0, \dots, 0)$. Denote $\Lambda = \text{diag}(\Lambda^1, \dots, \Lambda^j)$. For each cluster U_k^j let l_p^j be the level on which the GRR-BR projection is done, and l_c^j the coarsest level used in the ML process for this cluster. Here it is assumed that $l_c^j \leq l_p^j$. Denote $l_p = (l_p^1, \dots, l_p^j)$, $l_c = (l_c^1, \dots, l_c^j)$ and by I_k^j a function transfer from level k to level j . For improving a given approximation (U_m, Λ, T_m) , a multilevel cycle consisting in a sequence of cycles for each cluster in turn, is:

```

 $(U_m, \Lambda, T_m, l_p, l_c, q') \leftarrow \text{CL-MLP}(m, A_m, U_m, \Lambda, T_m, l_p, l_c, q')$ 
  For  $i = 1, \dots, j$  do:
    For  $k = m, \dots, l_c^i$  do:
      Repeat  $N_k^i$  Times:
        If  $l_p^i = k$  then  $(U_k^i, \Lambda^i, T_k^i) \leftarrow \text{GRR-BR}(m, A_k, U_k^i, \Lambda^i, T_k^i, l_p^i, k)$ 
           $U_k^i \leftarrow \text{Relax}(m, A_k, U_k^i, \Lambda^i, T_k^i, k, l_c^i)$ 
        If  $k > l_c^i$  then:
          Set  $k = k - 1$ ,
           $U_k^i = I_{k+1}^k U_{k+1}^i$ ,
           $T_k^i = I_{k+1}^k (T_{k+1}^i - A_{k+1} U_{k+1}^i) + A_k U_k^i$ 
      End
    For  $k = l_c^i, \dots, m$  do:
      If  $(k > l_p^i)$  Correct  $U_k^i = U_k^i + I_{k-1}^k (U_{k-1}^i - I_k^{k-1} U_k^i)$ 
      Repeat  $N_k^i$  Times
         $U_k^i \leftarrow \text{Relax}(m, A_k, U_k^i, \Lambda^i, T_k^i, k, l_c^i)$ 
        If  $l_p^i = k$  then  $(U_k^i, \Lambda^i, T_k^i) \leftarrow \text{GRR-BR}(m, A_k, U_k^i, \Lambda^i, T_k^i, l_p^i, k)$ 
      End
    End
  End

```

The GRR-BR algorithm used above is the following:

```

 $(U_k^i, \Lambda^i, T_k^i) \leftarrow \text{GRR-BR}(m, A_k, U_k^i, \Lambda^i, T_k^i, l_p^i, k)$ 
  Perform
     $(E, \Lambda') \leftarrow \text{GRR}(A_k, U_k^i, T_k^i)$ 
     $(E, \Lambda^i) \leftarrow \text{BackRotation}(E, \Lambda')$ 
     $U_k^i = U_k^i E$ 
     $T_k^i = T_k^i E$ 

```

Observe that in CL-MLP the clusters are treated sequentially and within each cluster the solutions are treated simultaneously in the ML cycle. A simultaneous cycle for several clusters is obtained by grouping the clusters into a single larger

cluster and applying CL-MLP to it. This may be used to improve separation between clusters. Observe that 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. Moreover, if mixings occur on coarse levels, one may expect an algorithm using fine level separation to have a poor efficiency; a coarse level separation usually improves the efficiency in such cases. 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 multilevel 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). Accurate normalization, if needed, can be performed as the last step only on the finest level.

Several parameters in this algorithm have to be specified, such as coarsest levels l_c and l_p , and numbers of iterations. These parameters are chosen by the adaptive algorithm discussed in next section. The choice of l_p and l_c depends on cluster and stage of the algorithm. The number of iterations N_k can be deduced for simple cases based on Fourier or 2-level cycle analysis or in general cases, by robustness tests performed during execution. The number of relaxations can vary with level. In the computational tests one or two relaxations per fine level passing were performed.

sought

3 Adaptive Multilevel Algorithms

For robustness, the construction of an adaptive version of the CL-MLP is essential since the grouping of eigenvectors into clusters, the sizes of the clusters, the coarsest level corresponding to a given cluster are not known in advance, usually. Eigenvectors belonging to clusters usually are mixed by different procedures. This deteriorates the algorithms' efficiency and often prevents convergence. A typical difficulty occurs when a procedure approximates only several eigenvectors of a cluster. Then the nonapproximated eigenvectors usually are the dominant components of the errors which are hard to eliminate due to mixing. This suggests to complete the clusters and separate the solutions within clusters whenever necessary. Simultaneous techniques treating at a time all solutions belonging to a complete cluster, can be easier coupled with separation techniques at different stages, thus acquiring better efficiency than sequential techniques which hardly avoid difficulties due to mixing.

In the adaptive algorithm, the clusters are tested for completion and completed. The cluster completion is tested on all current finest levels and performed on several levels since the structure of clusters can differ on different levels.

The full multilevel solver described below starts on coarse levels, and solves there for as many eigenvectors as possible. Then it uses those as an initial approximation for finer level solutions where more eigenvectors are added if needed.

Two important parts of this algorithm are the completion and addition of clusters. The completion of a cluster is done by adding in turn a new vector and improving it by multilevel cycles. An approximate eigenvalue is associated with this eigenvector, by a Rayleigh quotient. If the eigenvalue is close to the cluster then the new vector is added to the cluster. If the found eigenvalue 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 CL-MLP cycles.

The addition of a new clusters is usually done in the first stages of the algorithm when not enough complete clusters are found. This is performed by the cluster completion algorithm described next.

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

```

(j, U_k^j, Λ^j, T_k^j, l_p, l_c, q') ← Complete-Cluster(j, U_k^j, Λ^j, T_k^j, l_p^j, l_c^j, q')
  Until (Cluster-Completion-Test = TRUE) Do
    Choose random φ
    Until < A_k φ, φ > / < φ, φ > and residuals stabilize Do:
      (φ, Λ_max^j, T_k^j, 0, l_c^j, 1) ← CL-MLP(k, φ, Λ_max^j, T_k^j, 0, l_c^j, 1)
      Separate φ from (U_k^1, ..., U_k^j)
      Set λ = < A_k φ, φ > / < φ, φ >
      U_k^j ← (U_k^j, φ)
      Λ^j ← diag(Λ^j, λ)
      q' = q' + 1,   d_j = d_j + 1
  End
  Perform (U_k^j, Λ^j, T_k^j, l_p^j, l_c^j, d_j) ← CL-MLP(k, U_k^j, Λ^j, T_k^j, l_p^j, l_c^j, d_j)

(j, U_k, Λ, T_k, l_p, l_c, q') ← Add-Cluster(j, U_k, Λ, T_k, l_p, l_c, q')
  Set j = j + 1
  (j, U_k^j, Λ^j, T_k^j, l_p, l_c, q') ← Complete-Cluster(j, U_k^j, Λ^j, T_k^j, l_p^j, l_c^j, q')
  Set U_k = (U_k^1, ..., U_k^j),   Λ = (Λ^1, ..., Λ^j)

```

The Separation of ϕ from the other eigenvectors may be performed by orthogonality, projection or by a ML cycle.

Another crucial part of the algorithm are the robustness-tests. Robustness tests are techniques which find parameters to be used in a certain procedure for given data such that the procedure will be convergent and will be as efficient as possible. Typically, the procedure is tested over a set of data, or information from intermediate results is used in order to reduce the work involved in testing. Next, for simplicity,

we consider only the robustness test which provides the parameters (l_p, l_c) for the CL-MLP cycle. This ensures the proper values for the parameters l_c, l_p . in order to get convergence and the best attainable efficiency for CL-MLP. This is done by examining for clusters the different possible values for l_c, l_p . The use of an FMG algorithm suggests that these parameters for a given cluster will stabilize with the refinement process. Thus, a notion of a stabilized cluster is introduced in order to save unnecessary work in looking for new values for l_c, l_p for stabilized clusters.

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. For stabilized cluster, corresponding to a coarser level stabilized cluster, we take the l_c, l_p values from the corresponding coarser level cluster. For not stabilized clusters, which would exist usually on coarse levels only, we perform a search to obtain best values for l_c, l_p . This is done by performing ML cycles with different choices of these parameters, and choosing the ones that perform best. This is summarized in the next algorithm. Such tests are inexpensive when performed on coarse levels, and often lead to significant fine level work savings. Moreover, such tests are essential to ensure the algorithm's convergence.

Convergence is always obtained since at least the single level cycle converges, being a subspace iteration algorithm [8].

Denote by (l_p^j, l_c^j, m) the l_p and l_c parameters, for the level m cycle for the cluster j , and by $\mu(l_p^j, l_c^j, m) := \mu(CL - MLP(m, A_m, U_m, \Lambda, T_m, l_p^j, l_c^j, q'))$ the convergence rate (measured by the residual decrease) of the CL-MLP cycle for cluster j on level m , using the parameters (l_p, l_c) . The following algorithm updates (l_p, l_c) on level m :

```

 $(l_p, l_c, m) \leftarrow$  Robustness-Test  $(m, A_m, U_m, \Lambda, T_m, l_p, l_c, q')$ 
For j Do:
  If  $(|\Lambda_{m-1}^j - \Lambda_{m-2}^j| \leq \epsilon)$ 
  then
     $(l_p^j, l_c^j, m) = (l_p^j, l_c^j, m - 1)$ 
  else
    If  $(|\Lambda_m^j - \Lambda_{m-1}^j| \geq \epsilon)$  or if  $(\Lambda_m^j)$  is not approximated
    then
      Solve for  $(l_c^j, l_p^j, m)$ 
       $\min_{l_c^j, l_p^j} \mu(l_p^j, l_c^j, m) : l_c^j \leq l_p^j \leq m,$ 
    else
       $(l_p^j, l_c^j, m) = (l_p^j, l_c^j, m - 1)$ 
    endif
  endif

```

Observe that since the initial values for l_c^j, l_p^j on any level are taken from the next coarser level, the search needed in the Robustness-Test is over just a few choices of

the parameters.

The full ML algorithm uses as building blocks the CL-MLP, Add-Cluster, Complete-Cluster and Robustness-Test algorithms described before. It is defined as:

```

Adaptive-FMG( $m, q, A$ )
  Set  $k = 1, q' = 0, j = 0, l_p^j = k, l_c^j = k$ 
  Until ( $q' \geq q$  or  $q' \geq \alpha \dim_k$ ) Perform
    ( $j, U_k, \Lambda, T_k, l_p, l_c, q'$ )  $\leftarrow$  Add-Cluster( $j, U_k, \Lambda, T_k, l_p, l_c, q'$ )
    ( $U_k, \Lambda, T_k, l_p, l_c, q'$ )  $\leftarrow$  CL-MLP( $k, U_k, \Lambda, T_k, l_p, l_c, q'$ )
  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
    If ( $q' \geq q$ ) then:
      If (Cluster-Completion-Test=TRUE) then:
        ( $U_k, \Lambda, T_k, l_p, l_c, q'$ )  $\leftarrow$  CL-MLP( $k, U_k, \Lambda, T_k, l_p, l_c, q'$ )
      Else
        ( $j, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, q'$ )  $\leftarrow$  Complete-Cluster( $j, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, q'$ )
        ( $U_k, \Lambda, T_k, l_p, l_c, q'$ )  $\leftarrow$  CL-MLP( $k, U_k, \Lambda, T_k, l_p, l_c, q'$ )
      endif
    Else
      Until ( $q' \geq q$  or  $q' \geq \alpha \dim_k$ ) Perform
        ( $j, U_k, \Lambda, T_k, l_p, l_c, q'$ )  $\leftarrow$  Add-Cluster( $j, U_k, \Lambda, T_k, l_p, l_c, q'$ )
        ( $U_k, \Lambda, T_k, l_p, l_c, q'$ )  $\leftarrow$  CL-MLP( $k, U_k, \Lambda, T_k, l_p, l_c, q'$ )
    endif

```

3.1 Storage and Complexity

For the Adaptive-FMG algorithm, storage is required for the q eigenvectors of size N on the finest level, the potential and the corresponding right hand sides, on all levels, giving an overall estimate of memory of order $O(3(N + 1))$ for problems in 2-D and 3-D. The FMG work requires $O(N)$ operations per eigenvector. The work performed on coarsest levels should be added to these estimates. In the performed tests where a few eigenvectors were sought, the coarse level work was usually a fraction of the finest level work. If accurately zero scalar products are needed on finest levels then orthonormalizations or projections may be required within the finest level degenerate or close clustered eigenspaces. 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 $O(qN)$ is obtained.

4 Computational Examples and Discussions

We have chosen several examples in which we focus on different aspects of the method. The first example is of an adaptive algorithm in which we take a case of special difficulties in terms of clustering of eigenvalues, and correspondence of these clusters between levels. The second example shows that it is enough to treat the clusters in a sequential manner, and separation has to be done within each cluster only to obtain good convergence and accurate separation.

The third example shows that the new multilevel projection technique may be performed on coarsest levels, even in cases with closely clustered or equal eigenvalues, thus reducing the computational work significantly. A last example shows that the same efficiency is obtained for problems in 3-D as for problem in 2-D.

In all examples the periodic boundary conditions Schrödinger eigenvalue problem

$$(\Delta - V)u = \lambda u \quad (4.1)$$

defined on $\Omega = [0, a]^d$ ($d=2$ or 3) where $a = 2\pi/10$, was considered. The potentials are chosen such that the distribution of eigenvalue present special difficulties. A finite difference discretization on cartesian grids is used.

Example I: Adaptive Algorithm

As an example for the adaptive algorithm we have chosen a potential which determines a distribution of eigenvalues which present special difficulties. Not only closely clustered eigenvalues are present but also the correspondence of the clusters between levels is not appropriate, as explained later.

The potential for this case is $V(x, y) = 5 + 3\sin(10x)$ and the results are presented in Tables 1 and 2. The first $q = 12$ eigenvalues were required, and have been approximated using an adaptive 1-FMG-V(1,1) algorithm which uses as its coarsest level a employs 4×4 coarsest grid.

The i th eigenvalue and eigenvector will be denoted next by λ_i and v_i . The boxes in Table 2 show the clusters of close or equal eigenvalues (with (-) sign) found by the algorithm (the formats are chosen to outline the equal digits). As can be seen there the cluster structure on the different levels is not the same. Particularly, level 2 cluster structure differs from the level 1 cluster structure. Observe that the cluster of 6 eigenvalues on level 1 ($\lambda_6 - \lambda_{11}$), with multiplicities 1 - 4 - 1, has no correspondence on level 2.

For demonstrating the adaptive flow of the algorithm we give a full history of one specific run, for the potential discussed here.

The particular algorithm described in section 3 is used.

The algorithm started on level 1 adding eigenvectors until the cluster containing λ_{12} was completed. Note that λ_{16} , the last eigenvector that was found belongs to the next cluster, confirming the completeness of the last sought cluster. Observe that 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 $O(10^{-4})$ relative difference.

The relevant eigenvectors v_1, \dots, v_{15} were interpolated to level 2 where they provided initial guesses for the level 2 algorithm. 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 one or 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 cycle robustness tests and the cluster completion tests). This resulted in few cycles per eigenvector. The parameters l_c and l_p were found in the following way: 1) for cluster 1, (v_1), the values were obtained from previous level since this cluster was stabilized from the beginning; 2) For cluster 2 and 3 ($v_6 - v_9$ and $v_{10} - v_{13}$) l_c and l_p were taken from level 2 values since these clusters resulted stabilised after the cluster completion; 3) Robustness tests were used for cluster 4 since the eigenvalues $\lambda_{10} - \lambda_{13}$ on level L=3 and the next coarse level L=2 were not enough close. Then one cycle (here V(1,1)), was performed for each cluster.

The level 4 cluster completion test was satisfied by the first 3 clusters, eigenvalues $v_1 - v_9$ and their parameters were taken from level 3. The cluster completion algorithm was again applied to cluster 4, ($v_{10} - v_{13}$), a few cycles being sufficient, then the parameters were taken the corresponding level 3 ones since the cluster resulted stabilised. One cycle was performed for each cluster.

The level 5 cluster completion test was satisfied by all relevant eigenvectors ($v_1 - v_{13}$). One cycle (V(1,1)), 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: for v_1 : $l_c = l_p = 1$, for the other clusters, containing v_2, \dots, 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 didn't separate it from the next eigenvectors in its cluster e.g. on level 2, to separate λ_{14} from the next 7 eigenvectors with close eigenvalues.

Observe that a second order approximation and a good convergence rate of order $O(10^{-2})$ for the first cycle are obtained. A simultaneous cycle for all clusters with

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

Observe that 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 computer time and on levels 1, 2, 3, approximately 1/4 of the computer time. This is a fixed time and it would contribute only with 1/16 if level 6 would be employed too.

Table 1
1 -FMG-V(1,1), 2-D, 16-EV, 5-Lev
Residuals at Start and End of ML Cycles

E	level 1		level 2		level 3		level 4		level 5	
1	.48E+2	.37E-13	.69E+0	.97E-13	.22E+0	.30E-12	.60E-1	.64E-04	.15E-1	.14E-04
2	.53E+2	.44E-13	.30E+2	.14E-12	.11E+2	.35E-12	.30E+1	.86E-03	.76E+0	.73E-04
3	.61E+2	.38E-13	.30E+2	.80E-13	.11E+2	.29E-12	.30E+1	.86E-03	.76E+0	.73E-04
4	.66E+2	.68E-13	.30E+2	.17E-12	.11E+2	.30E-12	.30E+1	.54E-02	.76E+0	.11E-02
5	.55E+2	.39E-13	.30E+2	.24E-12	.11E+2	.45E-12	.30E+1	.54E-02	.76E+0	.11E-02
6	.52E+2	.12E-12	.11E+3	.32E-12	.16E+2	.35E-12	.44E+1	.42E-02	.11E+1	.82E-03
7	.59E+2	.31E-12	.45E+2	.54E-11	.16E+2	.32E-12	.44E+1	.42E-02	.11E+1	.82E-03
8	.61E+2	.20E-12	.45E+2	.57E-11	.16E+2	.41E-12	.44E+1	.39E-02	.11E+1	.83E-03
9	.62E+2	.17E-12	.45E+2	.71E-11	.16E+2	.33E-12	.44E+1	.16E-02	.11E+1	.93E-03
10	.73E+2	.13E-12	.45E+2	.15E-09	.12E+3	.30E-09	.43E+2	.72E-08	.12E+2	.33E-02
11	.58E+2	.34E-12	.11E+3	.41E-09	.12E+3	.16E-09	.43E+2	.20E-08	.12E+2	.33E-02
12	.54E+2	.39E-12	.12E+3	.81E-11	.12E+3	.26E-09	.43E+2	.21E-05	.12E+2	.29E-01
13	.51E+2	.70E-12	.12E+3	.50E-04	.12E+3	.50E-09	.43E+2	.16E-05	.12E+2	.29E-01
14	.44E+2	.96E-12	.12E+3	.19E-05	.12E+3	.17E-06	.44E+2	.34E-02		
15	.53E+2	.16E-12	.12E+3	.55E+01						
16	.69E+2	.19E-06								

Table 2
1 -FMG-V(1,1), 2-D, 16-EV, 5-LEV
Eigenvalues at End of ML Cycles

E	level 1	level 2	level 3	level 4	level 5
1	.496347395806E+1	.495721389176E+1	.495552134150E+1	.495509317773E+1	.495498173425E+1
2	.860204208719E+2	.999213342469E+2	.103677004418E+3	.104634633842E+3	.104874695012E+3
3	.860204208719E+2	.999213342469E+2	.103677004418E+3	.104634633842E+3	.104874695012E+3
4	.860569469139E+2	.9995 E+2	.10371 E+3	.1046 E+3	.10491 E+3
5	.860569469139E+2	.99998 E+2	.10375 E+3	.1047 E+3	.10495 E+3
6	.1670 E+3	.194919376181E+3	.202435153808E+3	.204351758395E+3	.204831900326E+3
7	.167113893828E+3	.194919376181E+3	.202435153808E+3	.204351758395E+3	.204831900326E+3
8	.167113893828E+3	.194962161804E+3	.202479632146E+3	.204395 E+3	.204876918643E+3
9	.167113893828E+3	.194962161804E+3	.202479632146E+3	.204396 E+3	.204876918643E+3
10	.167113893828E+3	.329185001547E+3	.384812002762E+3	.399841022256E+3	.403673103803E+3
11	.16715 E+3	.329185001547E+3	.384812002762E+3	.399841022256E+3	.403673103808E+3
12	.248170840742E+3	.329227787655E+3	.3848590736 E+3	.399888846 E+3	.403720980600E+3
13	.248170840742E+3	.329227787656E+3	.3848590739 E+3	.399888846 E+3	.403720980600E+3
14	.248207366784E+3	.424191908011E+3	.483580557031E+3	.499567983067E+3	
15	.248207366784E+3	.424295844705E+3			
16	.329264313697E+3				

Example II: Fine level Separation

In the next 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 9 eigenvectors, are presented in Tables 3 and 4. A 10-FMG-V(1,1) algorithm was used to show the constant convergence rate per cycle. The coarsest relaxation level for clusters 2 and 3 was level 2 and for the first eigenvector was level 1. On levels 1 and 2 the adaptive algorithm and few cycles were used. 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.

Evect.	level 1		level 2		level 3		level 4	
1	0.48E+02	0.14E-13	0.83E+00	0.12E-12	0.27E+00	0.11E-11	0.72E-01	0.41E-11
2	0.46E+02	0.83E-09	0.30E+02	0.48E-09	0.11E+02	0.42E-12	0.30E+01	0.21E-11
3	0.52E+02	0.29E-09	0.30E+02	0.12E-08	0.11E+02	0.75E-12	0.30E+01	0.52E-11
4	0.56E+02	0.56E-10	0.30E+02	0.73E-09	0.11E+02	0.93E-12	0.30E+01	0.56E-11
5	0.54E+02	0.85E-09	0.30E+02	0.55E-08	0.11E+02	0.17E-11	0.30E+01	0.12E-10
6	0.53E+02	0.57E-02	0.11E+03	0.40E-05	0.16E+02	0.61E-12	0.44E+01	0.16E-10
7	0.53E+02	0.68E-11	0.45E+02	0.57E-05	0.16E+02	0.10E-11	0.44E+01	0.39E-10
8	0.41E+02	0.13E-10	0.45E+02	0.29E-05	0.16E+02	0.82E-12	0.44E+01	0.33E-10
9	0.43E+02	0.80E-02	0.11E+03	0.14E-05	0.16E+02	0.83E-12	0.44E+01	0.48E-10

E	level 1	level 2	level 3	level 4
1	-0.494698319454E+01	-0.493789518604E+01	-0.493543833853E+01	-0.493481214576E+01
2	-0.860202443918E+02	-0.9991E+02	-0.10367E+03	-0.10463E+03
3	-0.860202443918E+02	-0.999361E+02	-0.1036931E+03	-0.104650E+03
4	-0.860406326305E+02	-0.999362E+02	-0.1036937E+03	-0.104651E+03
5	-0.860406326305E+02	-0.9997E+02	-0.1037E+03	-0.10469E+03
6	-0.1670E+03	-0.19491E+03	-0.20243E+03	-0.20434E+03
7	-0.167113893828E+03	-0.19493E+03	-0.20245+03	-0.20436E+03
8	-0.167113893828E+03	-0.19495E+03	-0.20247E+03	-0.20439E+03
9	-0.16713E+03	-0.19497E+03	-0.20249E+03	-0.2044E+03

Example III: Coarse Level Separation, 2D and 3D

In the next two runs (Tables 5, 6) we show that a coarsest level separation ($l_c = l_l = 1$), even for clusters containing very close and degenerate eigenvalues, can be enough to provide accurate finest level separation. In the degenerate clusters the eigenvectors were not orthogonalized on finest levels but resulted so from the FMG, where orthogonality was imposed on the coarsest level solution (in the FMG, not during the cycles). This implies an $O(qN)$ algorithm even for close clustered cases.

Table 5 shows results for a problem in 2-D with a potential $V(x, y) = 2 + 0.1\sin(10x + 10y)$, which produces a splitting of the first cluster of four eigenval-

ues into two degenerate clusters whose eigenvalues are close to within 10^{-4} . A second order approximation was obtained by an 1-FMG-V(1,1) algorithm and the asymptotic convergence rate per fine level cycle was 1/10. Observe the 13 equal digits of the degenerate eigenvalues, on all levels. On level 5, 8 cycles were performed to show the constant convergence rate per cycle, (see cycles 3 and 8 where the convergence rate is accurately 1/10). The eigenvectors came out accurately orthogonal, even in the degenerate eigenspaces, although the projection was performed only on the coarsest level, (the eigenvector's scalar products being of order 10^{-13} on level 5).

The same efficiency is obtained for problems in 3-D as can be seen in Table 6. The potential $V(x, y, z) = 2 + \sin(20x + 10y - 10z)$ determines a cluster of six close eigenvalues grouped into two clusters of two and four equal eigenvalues (with 13 digits). Observe the 1/10 convergence rate in cycle 3 and the first 6 common digits of the eigenvalues in the cluster.

These examples indicate that the reduction of complexity of the newly developed algorithm from $O(q^2N)$ to $O(qN)$ is quite general and also works for rather difficult cases.

Table 5				
1 -FMG-V(1,1), 2-D, 5-EV, 5-LEV				
cycle	vector	first res	last res	eigenvalue
L E V E L 4				
1	1	0.18E-02	0.13E-03	-0.19999752449715E+01
	2	0.30E+01	0.43E-02	-0.10162979203934E+03
	3	0.30E+01	0.43E-02	-0.10162979203934E+03
	4	0.30E+01	0.43E-02	-0.10172931140738E+03
	5	0.30E+01	0.43E-02	-0.10172931140738E+03
L E V E L 5				
1	1	0.46E-03	0.36E-04	-0.19999750026388E+01
	2	0.76E+00	0.40E-03	-0.10186970728937E+03
	3	0.76E+00	0.40E-03	-0.10186970728937E+03
	4	0.76E+00	0.40E-03	-0.10196970729590E+03
	5	0.76E+00	0.40E-03	-0.10196970729590E+03
3	1	0.35E-05	0.33E-06	-0.19999749801202E+01
	2	0.27E-04	0.26E-05	-0.10186970049930E+03
	3	0.27E-04	0.26E-05	-0.10186970049930E+03
	4	0.27E-04	0.26E-05	-0.10196970049780E+03
	5	0.27E-04	0.26E-05	-0.10196970049780E+03
8	1	0.97E-11	0.26E-10	-0.19999749799142E+01
	2	0.29E-09	0.31E-10	-0.10186970048459E+03
	3	0.29E-09	0.31E-10	-0.10186970048459E+03
	4	0.29E-09	0.31E-10	-0.10196970048302E+03
	5	0.29E-09	0.31E-10	-0.10196970048302E+03

Table 6				
1-FMG-V(1,1), 3-D, 7-EV, 3-LEV,				
cycle	vector	first res	last res	eigenvalue
L E V E L 3				
3	1	0.15E-02	0.10E-03	-0.19991341655960E+01
	2	0.25E-01	0.25E-02	-0.10072012662990E+03
	3	0.25E-01	0.25E-02	-0.10072012662990E+03
	4	0.23E-01	0.21E-02	-0.10072068269198E+03
	5	0.23E-01	0.21E-02	-0.10072068269198E+03
	6	0.23E-01	0.21E-02	-0.10072068269198E+03
	7	0.23E-01	0.21E-02	-0.10072068269198E+03

5 Conclusions

A robust and efficient ML 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 a new ML 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, the algorithm's 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.

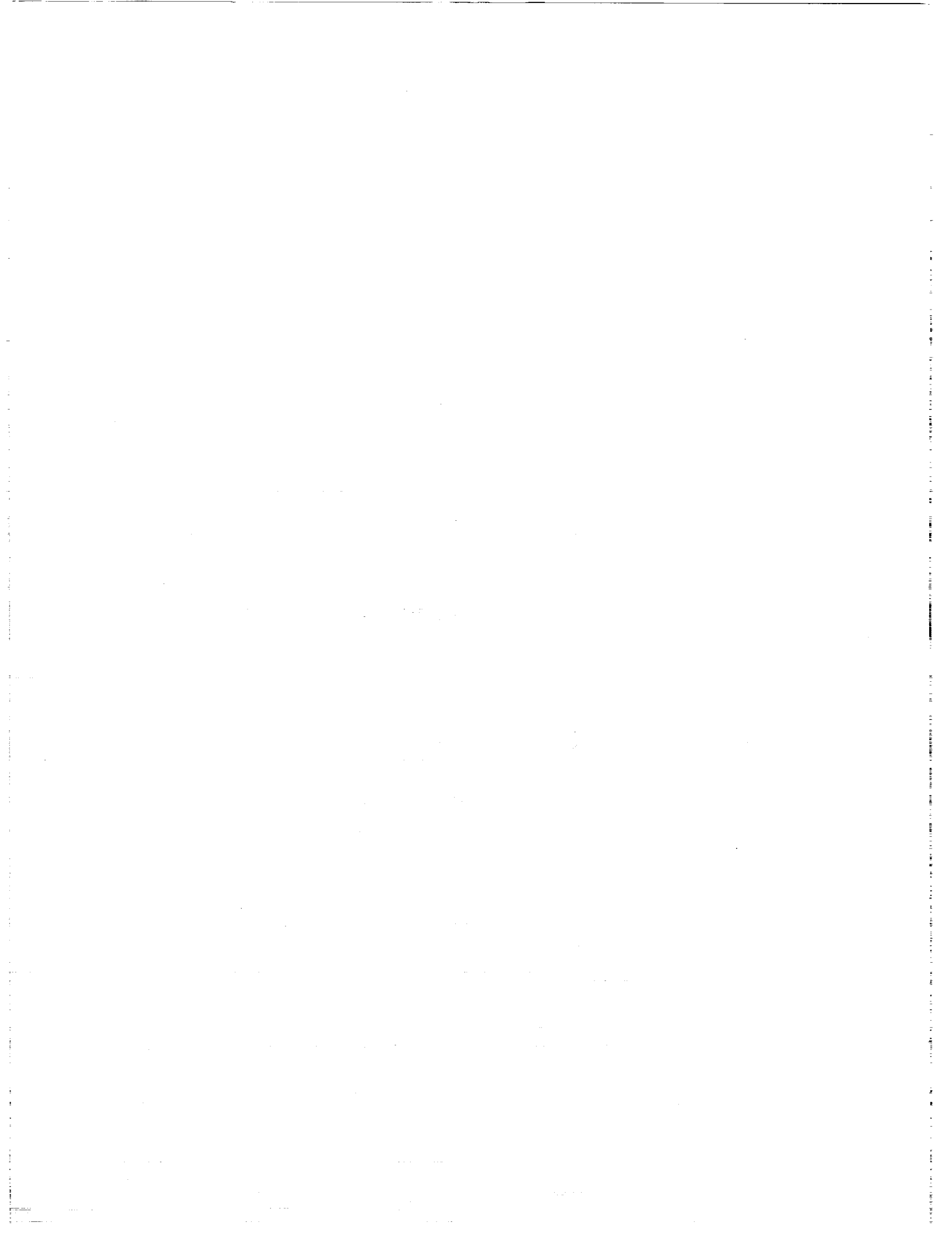
The results of the numerical tests for Schrödinger eigenvalue problems, show that the algorithm achieved the same accuracy, using the same amount of work (per eigenvector), as the Poisson multigrid solver. A second order approximation is obtained using the 5-point in 2-D and 7-point in 3-D discretized Laplaceian, by 1-FMG-V(1,1) in $O(qN)$ work. This means that the work was of order a few (2-4) fine level relaxations per eigenvector. The adaptive work was only part of the fine level work and enhanced the fine level cycle efficiency. Constant convergence rate per cycle was obtained for the presented cases. The robustness of the algorithm has been demonstrated on problems with eigenvalue distribution that present special difficulties.

References

- [1] R. E. Bank; *Analysis of a multi-level inverse iteration procedure for eigenvalue problems*; SIAM J. Numer. Anal., 19, pp. 886-898, 1982
- [2] F. Bourquin, F. d'Hennzel; *Application of Domain Decomposition Techniques to Modal Synthesis for Eigenvalue Problems*; in D.E. Keyes, T.F. Cahn, G. Meurant, J.S. Scroggs, R.G. Voigt eds. Domain Decomposition Methods for PDE, SIAM, 1992
- [3] A. Brandt, S. McCormick and J. Ruge; *Multigrid Methods for Differential Eigenproblems*; SIAM. J. SCI. STAT. COMPUT., Vol 4, No 2, 1983.
- [4] A. Brandt; *Multigrid Techniques 1984 Guide with Applications to Fluid Dynamics*; GDM Studien Nr. 85. Gesellschaft für Mathematik und Datenverarbeitung; St. Augustin, 1984.
- [5] S. Costiner; *Simultaneous Multigrid Solver for the Nonlinear Schrödinger Eigenvalue Problem*; M.Sc. Thesis, The Weizmann Institute of Science, 1990
- [6] S. Costiner, S. Ta'asan; *Separation Techniques for Multilevel Eigenvalue Solvers Applied to a Nonlinear Schrödinger Eigenvalue Problem*; Technical Report CS92-07, The Weizmann Institute of Science, 1992
- [7] K. Davstad; *A Multigrid Conjugate Residual Method for the Numerical Solution of the Hartree-Fock Equation for Diatomic Molecules*; JCP, Vol 99, nr1, pp 33-38, 1992
- [8] G. H. Golub, C.F. Van Loan; *Matrix Computations*; The Johns Hopkins University Press, 1989
- [9] W. Hackbush; *On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method*; SIAM J. Numer. Anal., 16, pp 201-215, 1979
- [10] W. Hackbush; *Multi-grid solutions to linear and nonlinear eigenvalue problems for integral and differential equations*; Rostock Math. Colloq. 25, pp 79-98, 1984
- [11] W. Hackbush; *Multi-grid eigenvalue computation*; in *Advances in Multi-Grid Methods*, Proc. conference held in Oberwolfach, December 8-13, 1984, D. Braes, W. Hackbusch and U. Trottenberg, eds., Notes on Numerical Fluid Mechanics, Volume 11, Vieweg, Braunschweig, pp 24-32, 1985
- [12] W. Hackbusch; *Multigrid methods and applications*; Springer Series in Comp. Math. 4, Springer-Verlag, Berlin, 1985

- [13] W. Kerner; *Large-Scale Complex Eigenvalue Problems, Review Article*; JCP. 85, pp 1-85, 1989
- [14] J.C. Luo; *A Domain Decomposition Method for Eigenvalue Problems*; in D.E. Keyes, T.F. Cahn, G. Meurant, J.S. Scroggs, R.G. Voigt eds. *Domain Decomposition Methods for PDE*, SIAM, 1992
- [15] J. Mandel, S. McCormick; *A Multilevel Variational Method for $Au = \lambda Bu$ on Composite Grids*; JCP 80, pp 442-452, 1989
- [16] S. F. McCormick, T. Noe; *Simultaneous iteration for the matrix eigenvalue problem*; J. Lin. Alg. Appl. 16, pp 43-56, 1977
- [17] S. F. McCormick; *A mesh refinement method for $Ax = \lambda Bx$* ; Math. Comp., 36, pp 485-489, 1981
- [18] P.J. Nikolai; *ALGORITHM 538, Eigenvectors and Eigenvalues of Real Generalized Symmetric Matrices by Simultaneous Iteration*; ACM Trans. on Math. Software, Vol 5, No 1, pp 118-125, 1979
- [19] B. N. Parlett; *The Symmetric Eigenvalue Problem*; Prentice-Hall, Englewood Cliffs, NJ, 1980
- [20] H. Rutishauser; *Computational aspects of F.L. Bauer's simultaneous iteration method*; Numer. Math. 13, pp 4-13, 1969
- [21] S. Ta'asan; *Multigrid Method for Stability Calculations*; J. of Scientific Computing, Vol. 3. No. 3. 1988
- [22] J.H. Wilkinson; *The Algebraic Eigenvalue Problem*; Clarendon Press, Oxford, 1965
- [23] L.Y. Zaslavsky; *An Adaptive Algebraic Multigrid for Multigroup Neutron Diffusion Reactor Core Calculations*; Applied Math. and Computation 53, pp 13-26, Elsevier, 1993
- [24] L.Y. Zaslavsky; *An Adaptive Self-Stabilized Multigrid for the Multigroup Neutron Diffusion* ; in *Proceedings of Mathematical Methods and Supercomputing in Nuclear Applications Conf.*; Karlsruhe, Germany, 19-23 April 1993







REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1993	3. REPORT TYPE AND DATES COVERED Contractor Report		
4. TITLE AND SUBTITLE A ROBUST MULTILEVEL SIMULTANEOUS EIGENVALUE SOLVER		5. FUNDING NUMBERS C NAS1-19480 C NAS1-18605		
6. AUTHOR(S) Sorin Costiner and Shlomo Ta'asan		WU 505-90-52-01		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23681-0001		8. PERFORMING ORGANIZATION REPORT NUMBER ICASE Report No. 93-35		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA CR-191491 ICASE Report No. 93-35		
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Michael F. Card Final Report Submitted to Physical Review E				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 64		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) <p>Multilevel (ML) algorithms for eigenvalue problems are often faced with several types of difficulties such as: the mixing of approximated eigenvectors by the solution process, the approximation of incomplete clusters of eigenvectors, the poor representation of solution on coarse levels and the existence of close or equal eigenvalues. Algorithms that do not treat appropriately these difficulties usually fail, or their performance degrades when facing them. These issues motivated the development of a robust adaptive ML algorithm which treats these difficulties, for the calculation of a few eigenvectors and their corresponding eigenvalues, presented in this paper. The main techniques used in the new algorithm include: the adaptive completion and separation of the relevant clusters on different levels, the simultaneous treatment of solutions within each cluster, and the robustness tests which monitor the algorithm's efficiency and convergence. The eigenvectors' separation efficiency is based on a new ML projection technique generalizing the Rayleigh Ritz projection, combined with a novel technique, the backrotations. These separation techniques, when combined with an FMG formulation, in many cases lead to algorithms of $O(qN)$ complexity, for q eigenvectors of size N on the finest level. Previously developed ML algorithms are less focused on the mentioned difficulties. Moreover, algorithms which employ fine level separation techniques are of $O(q^2N)$ complexity and usually do not overcome all these difficulties. Computational examples are presented where Schrödinger type eigenvalue problems in 2-D and 3-D, having equal and closely clustered eigenvalues, are solved with the efficiency of the Poisson multigrid solver. A second order approximation is obtained in $O(qN)$ work, where the total computational work is equivalent to only a few fine level relaxations per eigenvector.</p>				
14. SUBJECT TERMS leading eigenvalues, multilevel algorithms		15. NUMBER OF PAGES 20		16. PRICE CODE A03
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	