IN-64

34044

p- 41

ICASE

MULTIGRID TECHNIQUES FOR NONLINEAR EIGENVALUE PROBLEMS; SOLUTIONS OF A NONLINEAR SCHRÖDINGER EIGENVALUE PROBLEM IN 2D AND 3D

Sorin Costiner
Shlomo Ta'asan

Multigrid Techniques for Nonlinear Eigenvalue Problems; Solutions of a Nonlinear Schrödinger Eigenvalue Problem in 2D and 3D *

Sorin Costiner   and   Shlomo Ta'asan

Department of Applied Mathematics and Computer Science
The Weizmann Institute of Science, Rehovot, Israel, 76100
and
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center, Hampton, Va 23665, USA
e-mail: na.scostiner@na-net.ornl.gov

ABSTRACT

Algorithms for nonlinear eigenvalue problems (EP), often require solving selfconsistently a large number of EP. Convergence difficulties may occur if the solution is not sought in a right neighborhood; if global constraints have to be satisfied; and if close or equal eigenvalues are present. Multigrid (MG) algorithms for nonlinear problems and for EP obtained from discretizations of partial differential EP, have often shown to be more efficient than single level algorithms.

This paper presents MG techniques for nonlinear EP and emphasizes an MG algorithm for a nonlinear Schrödinger EP. The algorithm overcomes the mentioned difficulties combining the following techniques: an MG projection coupled with backrotations for separation of solutions and treatment of difficulties related to clusters of close and equal eigenvalues; MG subspace continuation techniques for the treatment of the nonlinearity; an MG simultaneous treatment of the eigenvectors at the same time with the nonlinearity and with the global constraints. The simultaneous MG techniques reduce the large number of selfconsistent iterations to only a few or one MG simultaneous iteration and keep the solutions in a right neighborhood where the algorithm converges fast.

Computational examples for the nonlinear Schrödinger EP in 2D and 3D, presenting special computational difficulties, which are due to the nonlinearity and to the equal and closely clustered eigenvalues, are demonstrated. For these cases, the algorithm requires $O(qN)$ operations for the calculation of q eigenvectors of size N and for the corresponding eigenvalues. One MG simultaneous cycle per fine level was performed. The total computational cost is equivalent to only a few Gauss-Seidel relaxations per eigenvector. An asymptotic convergence rate of 0.15 per MG cycle is attained.

---

i

1 Introduction

Multigrid (MG) techniques for nonlinear problems and for eigenvalue problems (EP) such as many large scale problems from physics, chemistry and engineering, have often shown to be more efficient than single level techniques, [1], [2], [3], [4]. MG techniques can use efficiently features which are generally not used by single level techniques, such as: the problems can be approximated on several discretization levels; the solutions can be well approximated by solutions of coarse level problems; only a few eigenvalues and eigenvectors are sought; and the solutions are dominated by smooth components [2]. Moreover, MG techniques have powerful solving capabilities, for example they can approximate well the efficient inverse power iteration for eigenvalue problems [5].

MG techniques involve, in general, the processing of the problem on a sequence of discretization levels. Usually, these levels are finite dimensional function spaces defined on increasingly finer grids, [3], [4].

To treat nonlinear problems or systems of coupled problems, as in our case, algorithms often involve a large number of selfconsistent iterations. The iterations may be inefficient or may not converge if the approximated solution is not in a right neighborhood. The treatment of these difficulties becomes harder when combined with eigenvalue difficulties. Algorithms for eigenvalue problems face severe difficulties especially when close or equal eigenvalues are present, as usually in Schrödinger and in electromagnetism problems. Instead of approximating an eigenvector, procedures such as relaxations approximate a linear combination of eigenvectors with close or equal eigenvalues. This we refer as eigenvector mixing. Mixing is especially severe when not all eigenvectors with close eigenvalues are computed, i.e., incomplete clusters of eigenvectors are treated. In such cases usually, the dominant components of the errors, hard to eliminate, consist of the not-approximated eigenvectors of the cluster. The nonlinear Schrödinger eigenvalue problem treated in the computational examples is ill posed when defined on incomplete clusters. Global constraints imposed on the solutions, such as norms, orthogonality, given average, introduce additional difficulties in MG algorithms since these constraints are not conserved by inter-level transfers of solutions, e.g., the transfers alter the norms and orthogonality of solutions.

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 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 methods approached some of the mentioned difficulties. In no previous approach all of these difficulties were treated together. The treatment

1

of the nonlinearity and of the constraints should be done simultaneously with the update of eigenvectors, for keeping the approximate solution in a right neighborhood of the exact solution where the algorithm is efficient.

This paper focuses on MG techniques for overcoming the mentioned difficulties and presents an MG robust and efficient algorithm for the calculation of a few eigenvalues and their corresponding eigenvectors for a nonlinear Schrödinger eigenvalue problem.

The problem used for illustration is the computation of the first q eigenvectors $u^1, ..., u^q$, and the corresponding smallest eigenvalues (in modulus) $\lambda_1, ..., \lambda_q$, of a discretized Schrödinger Nonlinear Eigenvalue problem of Hartree-Fock type:

$$\begin{cases} \Delta u^i - (V + \epsilon W)u^i = \lambda_i u^i, & i = 1, \ldots, q \\ \Delta W = -c_1 \sum_{i=1}^q (u^i)^2 + c_2 \\ \|u^i\| = 1 \\ \int W = 0 \end{cases} \tag{1}$$

Periodic boundary conditions are assumed. Eigenvectors in degenerate eigenspaces are required to be orthogonal. The problem has to be solved in 2D and 3D. V is a given linear potential operator, W is a nonlinear potential, also to be calculated, c_1, c_2, ϵ are constants. If ϵ is zero the problem is linear else it is nonlinear since the potential W depends on the solutions. It is assumed that the clusters containing the desired q eigenvectors are complete.

The problem is represented and solved on a sequence of coarse to fine levels. The algorithm is based on separation of eigenspaces and of eigenvectors in eigenspaces, simultaneously treated with the nonlinearity and constraints on all levels. Transfers between levels are used to reduce as much as possible the heavy computational tasks from fine levels to inexpensive tasks on coarse levels. The algorithm may be outlined by three steps: 1) get an approximation of the solution on a coarse level; 2) interpolate the solution to a finer level; 3) improve the fine solution by few MG cycles. Repeat steps 2) and 3) until finest level is reached. The approximation on the coarse level at step 1) solves first the linear problem ($\epsilon = 0$), then the nonlinear one by a continuation procedure. An MG cycle at step 3) starts on the fine level, transfers successively the problem down to coarser levels and then up, returning to the fine level. On each level, the eigenvectors and the nonlinear potential are updated, and on a coarse level, the eigenvectors are separated by projections and backrotations. The separation of fine level eigenvectors by transfers coupled with coarse level projections is called here *multigrid projection (MGP)* [6], [7].

The simultaneous MG schemes reduce the many selfconsistent iterations to solve the nonlinearity to a single simultaneous iteration. Due to MGP, the algorithm achieves a better computational complexity and a better convergence rate than previous MG eigenvalue algorithms which use only fine level projections. Increased robustness is obtained due to the MGP coupled with backrotations;

2

the simultaneous treatment of eigenvectors with the nonlinearity and with the global constraints.

In an adaptive version of the algorithm, [7], on each new fine level, clusters are identified, tested for completeness, completed if necessary and improved by MG cycles using coarser levels. Robustness tests control the algorithm's convergence and efficiency. These are done adaptively for different clusters on different levels.

It will be observed that the presented techniques are applicable to a much larger class of problems. In particular, the algorithms without the treatment of nonlinearity were used for linear eigenvalue problems too, see [8].

The computational examples were chosen to include special difficulties such as very close and equal eigenvalues. The algorithm uses a few (1-4) fine level cycles, and in each cycle, two fine level Gauss-Seidel 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. Exact orthogonality of fine level eigenvectors is obtained by the coarse level MGP. A second order approximation is obtained in $O(qN)$ work, for q eigenvectors of size N on the finest level. An asymptotic convergence rate of 0.15 per MG cycle is obtained.

For early works, theory and more references on MG eigenvalue algorithms, see [9], [10], [5], [2], [3], [11], [4], [12]. The sequential MG algorithm performing the separation on finest level [2], combined with a conjugate residual method, is applied to a Hartree-Fock nonlinear eigenvalue problem in 2D in [13]. A previous version of the results presented here is given in the report [6]. The linear adaptive techniques presented in [14], [7], can be combined with the presented techniques and are especially useful for the completion of clusters. Algorithms and more references for single level large scale complex eigenvalue problems can be found in [15]. We refer to [16], [17], [18], for theory on algebraic eigenvalue problems; and to [19], [20], [21], for aspects of the single level technique used here, of obtaining a few eigenvectors and their eigenvalues for linear EP.

The MG projection and backrotations were first introduced in [22], and in the reports [6], [14]. An outline of a related computational approach presented here is given in [23].

The paper is organized as follows. The next two Subsections 1.1, 1.2, describe the MG discretization of the Nonlinear Schrödinger eigenvalue problem and the general FAS inter-level transfers. Section 2 presents the central MG eigenvector linear separation techniques, i.e., the MG-solver-cycle, the MGP, the backrotations, the MG-combined-cycles, and the linear-cluster-FMG algorithm. Section 3 presents the MG nonlinear techniques, i.e., the MG cycle for the nonlinear potential W, the simultaneous updating of eigenvectors and potential, the treatment of global constraints, the subspace continuation procedures, and the FMG nonlinear eigenvalue algorithm. Section 4 presents computational examples for the nonlinear Schrödinger problem. Section 5 describes the adaptive

techniques, i.e., the adaptive-MG-cycle, the cluster-completion, the robustness-tests, the adaptive-FMG. In Subsection 5.6 are presented computational examples for the linear adaptive algorithm. A final subsection contains details and observations on the linear and adaptive techniques which were included there to keep the rest of the presentation simpler. Conclusions are presented in Section 6.

1.1 The Discretization of the Nonlinear Schrödinger Eigenvalue Problem

Assume that Ω is a domain in R^d, and let $G_1, G_2, ..., G_m$ be a sequence of increasingly finer grids, that extend over Ω. The space of functions defined on grid G_k is called level k. I_k^l denote transfer operators from level k to level l, e.g., I_k^l can be interpolation operators. The discretization of problem (1), on finest grid G_k, has the following form:

$$\begin{cases} \Delta_k u_k^i - (V_k + \epsilon W_k)u_k^i = \lambda_i u_k^i \\ \Delta_k W_k = -c_1 \sum_{i=1}^q (u_k^i)^2 + c_2 \\ \|u_k^i\|_k = 1 \\ \sum W_k^j = 0 \end{cases} \tag{2}$$

If G_k is not the finest grid then relations (2) include FAS right hand sides as shown in the next sections. Here Δ_k is a discrete approximation to the Laplacian. It is desired that, on finest level, the eigenvectors in degenerate eigenspaces be orthogonal. Periodic boundary conditions are assumed for Ω – a box in R^d. The W_k^j denotes the j'th component of W_k on level k, V_k is a transfer of the finest level V_m to coarser level k, i.e., $V_k = I_m^k V_m$.

1.2 FAS Transfers

The following is a general formulation of the FAS, (Full Approximation Scheme [1]), which is applied to the eigenvalue equations, to the separation of eigenvectors, to the nonlinear equation, and to the global constraints. Let

$$F_i U_i = T_i \tag{3}$$

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

$$F_j U_j = T_j \tag{4}$$

is an FAS transfer of the level i problem (3) if

$$T_j = I_i^j(T_i - F_i U_i) + F_j I_i^j U_i \tag{5}$$

The level j problem (4) is used in solving the level i problem (3). The level i solution U_i^{old} is corrected with the level j solution U_j by the FAS correction:

$$U_i^{new} = U_i^{old} + I_j^i(U_j - I_i^j U_i^{old}) \tag{6}$$

4

If U_i is the exact solution of (3) then its transfer to level j, $I_i^j U_i$, is the exact solution of (4). In this case the correction (6) does not change the exact solution U_i.

2   On Multigrid Separation Techniques

The introduced algorithm combines MG linear eigenvalue techniques with techniques for nonlinear problems. This section presents the MG linear eigenvalue techniques, i.e., the MG-solver-cycle, the MG Projection (MGP), and the Cluster-FMG [7]. 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 this section, the problem

$$AU = U\Lambda \tag{7}$$

is defined on a sequence of levels. The U denotes an eigenvector associated to the eigenvalue Λ. The matrix A corresponding to the level i problem is denoted by A_i. For example, A_i may be the matrices obtained by discretizing a continuous eigenvalue problem, on a sequence of grids.

2.1   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 [18].

To motivate the MG-solver-cycle, consider the eigenvalue problem (7), 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

$$U^{n+1} = (A - \Lambda'I)^{-1}U^n, \quad U^{n+1} = U^{n+1}/\|U^{n+1}\| \tag{8}$$

will converge fast (in a few iterations) to U (since the U component in U^n will be multiplied at each iteration by $1/(\Lambda - \Lambda') \approx \infty$, [18]). For large A it is too expensive to compute $(A - \Lambda'I)^{-1}$, but one can approximate (8) by solving iteratively:

$$(A - \Lambda'I)U^{n+1} = U^n, \quad U^{n+1} = U^{n+1}/\|U^{n+1}\| \tag{9}$$

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

$$(U^n)^T A U^n = (U^n)^T U^n \Lambda'. \tag{10}$$

5

For large A, the iteration (9) is impractical for single level algorithms, but it can be approximated by MG cycles, which have often shown to be efficient [2], [3].

Relation (7) 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 (5) and (6) can be considered in block form in the same way. In a simultaneous MG-solver-cycle, the problem (7) is represented on the different levels in the FAS form:

$$F_i U_i := A_i U_i - U_i \Lambda_i = T_i \tag{11}$$

where $T_m = 0$ on the initial level m (finest usually) and T_j are computed by (5) for $j < m$, with $j = i - 1$. Equation (11) is relaxed on each level and the solutions are corrected by (6).

An MG-solver-cycle from 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 $k = m, \dots, l$ **(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 $k = l, \dots, m$ **(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 [1]. In this MG-Solver-Cycle, the Λ is kept constant on all levels.

2.2 Generalized Rayleigh Ritz Projections

This subsection presents a generalization of the Rayleigh Ritz Projection [18], 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:

$$AV = V\Lambda \tag{12}$$

where $\Lambda = diag(\lambda_1,\ldots,\lambda_q)$ contains on the diagonal the q sought eigenvalues corresponding to the sought eigenvectors which are the columns of V. Assume that U which satisfies

$$V = UE \tag{13}$$

is given instead of V, where E is a $q \times q$ invertible matrix to be found. Substituting (13) into (12) gives

$$AUE = UE\Lambda \tag{14}$$

An FAS transfer (5) of (14) to another level yields an equation of the form

$$AUE = UE\Lambda + TE \tag{15}$$

where the product TE is the FAS right hand side of (15) with known T. Solutions E and Λ for (15) can be computed by solving the $q \times q$ generalized eigenvalue problem

$$U^T(AU - T)E = (U^TU)E\Lambda \tag{16}$$

obtained by multiplying (15) 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

$$(E,\Lambda) \leftarrow \mathbf{GRRP}(A,U,T) \tag{17}$$

and is refered later as the Generalized Rayleigh Ritz Projection (GRRP).

2.3 Multigrid Projections

The solutions E and Λ of (15) can be obtained by an FAS MG procedure. Consider (15) written as a level i problem:

$$A_iU_iE - U_iE\Lambda = T_iE \tag{18}$$

Then the FAS transfer of (18) to level j is

$$A_jU_jE - U_jE\Lambda = T_jE \tag{19}$$

where $U_j = I_i^jU_i$. T_jE is computed by (5), and results in

$$T_j = I_i^j(T_i - A_iU_i) + A_jI_i^jU_i \tag{20}$$

A solution (E,Λ) of (18) is a solution of (19). The solutions (E,Λ) of (19) can be obtained by a GRRP.

Problems (18) and (19) have the same form. Hence problem (19) 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 (20) for the problem (19) is the same as the transfer used in the MG-solver-cycle for the problem $A_j U_j - U_j \Lambda = T_j$. This makes possible to perform the MGP simultaneously with the MG-solver-cycle, in MG-combined-cycles, as presented in Section 2.5.

2.4   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 algorithm presented next.

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)

A backrotation step will be denoted by

$$(E, \Lambda) \leftarrow \textbf{Backrotation}(E, \Lambda) \tag{21}$$

2.5 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)

9

$$U_k = U_k E$$
$$T_k = T_k E$$

The MG-combined-cycle, Solve-MGP, is the central building element of the adaptive algorithms presented in Section 5.

2.6   The Cluster-FMG Algorithm

The Linear-Cluster-FMG algorithm starts on a coarsest level, for simplicity $l_p = l_c = 1$, peforms γ cycles Solve-MGP on each level and interpolates the solutions to the next level. In this way, the fine level initial solutions are in a close neighbourhood of the exact solutions, due to the coarser level solutions computed. In this neighbourhood, the nonlinear algorithm is usually as efficient as the linear algorithm.

$(U_m, \Lambda, T_m) \leftarrow$ **Linear-Cluster-FMG** $(m, A_m, U_m, \Lambda, T_m, l_1, l_1, q)$

    For $k = 1, \ldots, m$ **do:**

        Repeat γ **Times:**

            $(U_k, \Lambda, T_k) \leftarrow$ **Solve-MGP** $(k, A_k, U_k, \Lambda, T_k, l_1, l_1, q)$

        If $k < m$ **Transfer:**

            $U_{k+1} = I_k^{k+1} U_k,$

    End

3 MG Techniques for the Treatment of the Nonlinearity

The central techniques for nonlinear problems are illustrated on the nonlinear Schrödinger eigenvalue problem (1). The treatment of the nonlinearity is performed by updating the nonlinear potential W simultaneously with the eigenvectors as well as with the global constraints. The following MG techniques are presented: an MG-Potential-Solver cycle for W, a Simultaneous-FAS cycle for W and eigenvectors, the treatment of the global constraints, the subspace continuation procedures and the Simultaneous-Nonlinear-FMG algorithm.

3.1 An MG Solver Cycle for the Nonlinear Potential

In an MG cycle for updating W, we have two options: 1) to keep the u^i's fixed, and 2) to update also the u^i's. The first case leads to sequential cycles where separate cycles are performed for W and for u. The second case leads to simultaneous cycles. The two cases lead to different FAS transfers. In this section the u^i's are considered fixed, while in Section 3.2 the u^i's are updated together with W. The equation to be solved for the nonlinear potential W is:

$$\Delta_k W_k = p_k \tag{22}$$

Here, for $k < m$, p_k is the FAS right hand side of (22)

$$p_k = I_{k+1}^k (p_{k+1} - \Delta_{k+1} W_{k+1}) + \Delta_k I_{k+1}^k W_{k+1} \tag{23}$$

On finest level, $k = m$,

$$p_k = -c_1 \sum_{i=1}^{q} (u_k^i)^2 + c_2 \tag{24}$$

An MG-Potential-Solver cycle for W, is:

$(W_m) \leftarrow$ **MG-Potential-Solver** (m, W_m, p_m, l)

    For $k = m, \ldots, l$ (step by -1) do:

      $W_k \leftarrow$**Relax** (m, W_k, p_k, k, l)

      If $k > l$ Transfer:

        $W_{k-1} = I_k^{k-1} W_k,$

        $p_{k-1} = I_k^{k-1}(p_k - \Delta_k W_k) + \Delta_{k-1} W_{k-1}$

    End

    For $k = l, \ldots, m$ (step by 1) do:

If $(k > l)$ **Correct** $W_k = W_k + I_{k-1}^k(W_{k-1} - I_k^{k-1}W_k)$

$W_k \leftarrow$ **Relax** (m, W_k, p_k, k, l)

End

This is a usual V type cycle from fine level m to coarse level l. Other cycles can be defined as well which involve a different sequence of visiting the levels. The work involved by such a cycles is several times (about 4 times) the finest level relaxation work. Such a cycle can be used in the next algorithms instead relaxations for W, but in the numerical tests this was not necessary. Similar solver cycles can be defined for the u^i.

3.2   The MG Simultaneous Updating of Nonlinear Potential and Eigenvectors

In the MG-Potential-Solver, the u_is are fixed. An MG Simultaneous-FAS cycle is obtained by combining the updating of u^is with the updating of W. The nonlinear equations in FAS form are:

$$\Delta_k u_k^i - (V_k + \epsilon W_k)u_k^i - \lambda_i u_k^i = \tau_k^i \tag{25}$$

$$\Delta_k W_k + c_1 \sum_{i=1}^q (u_k^i)^2 - c_2 = p_k \tag{26}$$

Denote by L_k the operator

$$L_k = \Delta_k - V_k - \epsilon W_k - \lambda_i \tag{27}$$

Both W_k and u_k are considered variables. The τ_k^i and p_k, in (25) and (26), are zero on the finest level and equal to the FAS right hand sides on the other levels, namely:

$$\tau_k^i = I_{k+1}^k(\tau_{k+1}^i - L_{k+1}u_{k+1}^i) + L_k I_{k+1}^k u_{k+1}^i \tag{28}$$

$$p_k = I_{k+1}^k(p_{k+1} - \Delta_{k+1}W_{k+1} - c_1 \sum_{i=1}^q (u_{k+1}^i)^2) + \Delta_k I_{k+1}^k W_{k+1} + c_1 \sum_{i=1}^q (I_{k+1}^k u_{k+1}^i)^2 \tag{29}$$

The u_k^i's are updated by relaxations, using (25) while W_k is considered constant. W_k is updated by relaxations using (26) while $u_k^1, ..., u_k^q$ are considered constants. The u_k^i's are updated by projections and backrotations on coarse levels. The Simultaneous-FAS cycle in Section 3.5 describes this algorithm.

12

3.3 The MG Treatment of Global Constraints

The FAS treatment of global constraints are needed to keep the approximate solutions in a right neighborhood of the exact solutions, where the algorithm is efficient. Keeping the solutions in a right neighborhood is accomplished in conjunction with the simultaneous techniques, the subspace continuation techniques, and the FMG algorithm. The solutions should satisfy several global constraints. The parameter c_1 is set arbitrarily to $c_1 = 1$ but it can be also used as a parameter in a continuation technique. The potential V is periodic and the solutions u_k^i are periodic. Thus W is periodic, therefore

$$\int \Delta W = 0 \tag{30}$$

The integral is computed ower the whole domain. Discretizing (30) and using (26), c_2 must satisfy on the current finest grid:

$$c_2 = \sum_{j=1}^{N_m} \sum_{i=1}^{q} (u_{m,j}^i)^2 / N_m \tag{31}$$

where N_m is the number of nodes on grid m. Since on current finest level

$$\|u_m^i\| = 1 \tag{32}$$

c_2 results independent of u and it is kept constant on all levels.

If W is a solution then for any constant C, the $W+C$ is also a solutions for the same eigenvectors and the eigenvalues $\lambda_i - C$. The constant C is fixed by the condition on W:

$$\int W = 0 \tag{33}$$

The FAS formulation of the discretized condition (33) is

$$\sum_{j=1}^{N_m} W_m^j = 0 \tag{34}$$

on all levels, if the fine to coarse grid transfers conserve zero sums, e.g., as the full weighting transfer which is often used. Else the appropriate FAS condition should be set using (5).

The FAS formulation of the norm condition $\|u_k\| = 1$ becomes

$$\|u_{k-1}\| = \rho_{k-1} := \|I_k^{k-1} u_k\| + \rho_k - \|u_k\| \tag{35}$$

The norms are set to 1 after interpolating first time the solution to a current finest level and are set to the ρ_k values, on coarsest levels, at the end of the backrotations. In (35) the same norm notation has been used for the different norms on the different levels.

13

3.4 MG Subspace Continuation Techniques

The central idea of the subspace-continuation techniques is to use a stable subspace of solutions of a given eigenvalue problem to approximate the subspace of solutions of the problem perturbed. It is important that the subspace of the perturbed problem is well approximated and not the solutions of the perturbed problem. The solutions inside the stable subspace may be very sensitive to perturbations. Subspace continuation procedures can depend on one, on several, or on a continuum of parameters, e.g., the continuation can be performed by the parameter μ varied from 0 to 1 for μW; or by two parameters α, μ for $\alpha V + \mu W$; or the parameters may be the elements of W.

The continuation process on coarsest level which we used most in our tests is the following. First the linear problem is solved by a sequence of relaxations, orthogonalizations and projections for $W = 0$ fixed. This is to approximate first the subspace of the eigenvectors and not the eigenvectors themselves. Then the problem with the potential

$$V_1 = V + \mu W \tag{36}$$

is considered, where μ is a parameter. In the continuation procedure, the μ increases in steps, from 0 to ϵ. At each step, the linear problem is resolved, considering W fixed, and afterwards W is recomputed. Thus the subspace is updated first. This would mean to perform the continuation on μW. A continuation using two parameters is to solve first the linear problem for $V = 0$, then perform a continuation on μW until $\mu = \epsilon$ is reached and only after that to start a continuation process on the linear part of the potential αV. The justification to do these comes from the fact that V may split degenerate eigenspaces in clusters with very close eigenvalues. The continuation having all elements of W as parameters, consists in the selfconsistent iterations in which the linear problem is solved in turns with the updating of W.

The single level continuation procedures described above can be performed in an MG way, leading to MG sequential-selfconsistent-schemes, as the one used in [13]. A more general MG sequential-selfconsistent-scheme is the following MG-Sequential-Continuation algorithm, which iterates the simultaneous updating of the eigenvectors by MG cycles with the updating of W by MG cycles:

$(U_m, W_m, \Lambda) \leftarrow$ **MG-Sequential-Continuation**(U_m, W_m, Λ)

    Set $\mu = 0$

    While $0 \leq \mu \leq \epsilon$ **do :**

        solve until convergence:

            1) Solve the linear problem for fixed W_m and potential $V_m + \mu W_m$

$$(U_m, \Lambda, \tau_m) \leftarrow \textbf{U-Simultaneous-FAS}(m, q, U_m, W_m, \Lambda, L_m, \tau_m, \nu_1, \nu_2)$$

    2) Solve for W_m keeping U_m, Λ fixed:

$$(W_m) \leftarrow \textbf{MG-Potential-Solver } (m, W_m, p_m, l)$$

    Update W_m such that: $\sum_{j=1}^{N_m} W_m^j = 0$

Increase μ

The above U-Simultaneous-FAS algorithm is obtained by removing from the Simultaneous-FAS algorithm presented in Section 3.5 the updating of W, p, ρ. This is an algorithm for updating simultaneously the eigenve ctors, which separates the eigenvectors by projection on a coarse level. The different MG cycles for the eigenvectors and potential may have different coarsest levels.

3.5   The Simultaneous Nonlinear FMG Eigenvalue Algorithm

Assume for simplicity, in this section, that on coarsest level $k = 1$, all eigenvectors can be well approximated. Denote by $U_k = (u_k^1, ..., u_k^q)$ the matrix on level k having columns the approximations of the desired q eigenvectors, corresponding to the eigenvalues of $\Lambda = diag(\lambda_1, ..., \lambda_q)$. Assume the same type of vector notations for τ_k, p_k, ρ_k The Simultaneous-Nonlinear-FMG algorithm for q eigenvectors, m levels, reads:

Simultaneous-Nonlinear-FMG$(m, q, U_m, W_m, \Lambda, L_m, \tau_m, \rho_m, p_m, \nu_1, \nu_2, \gamma)$

    Set U_1 random, $\Lambda = 0, W_1 = 0$

    For $k = 1$ until m do:

        1) **If $k = 1$ get:**

$$(U_k, W_k, \Lambda) \leftarrow \textbf{Continuation}(U_k, W_k, \Lambda)$$

    If $k < m$ then: $k = k + 1$

        2) **Interpolate**

$$U_k = I_{k-1}^k U_{k-1} \, ,$$
$$W_k = I_{k-1}^k W_{k-1}$$

        3) **Set** $\tau_k = 0, \quad \rho_k = 1, \quad c_2 = \sum_{j=1}^{N_k} \sum_{i=1}^q (u_{k,j}^i)^2 / N_k, \quad p_k = 0$

            $\|u_k^i\| = 1 \, , \quad \lambda_i = < (\Delta_k - V_k - \epsilon W_k) u_k^i, u_k^i >$

        4) **Do γ times :**

$$(U_k, W_k, \Lambda) \leftarrow \textbf{Simultaneous-FAS}(k, q, U_k, W_k, \Lambda, L_k, \tau_k, \rho_k, p_k, \nu_1, \nu_2)$$

endif

Continuation(U_k, W_k, Λ)

    Set $\mu = 0$

    While $0 \leq \mu \leq \epsilon$ **do** :

        If $\mu = 0$ get U_k, Λ by Relaxations, Orthogonalizations and Projections

        else solve until convergence steps 1, 2:

            1) Solve the linear problem for U_k, Λ by Relaxations and Projections.

            2) Solve for W_k : $\Delta_k W_k + c_1 \sum_{i=1}^{q} (u_k^i)^2 = c_2, \quad \sum_{j=1}^{N_k} W_k^j = 0$

        endif

    Increase μ

Simultaneous-FAS$(k, q, U_k, W_k, \Lambda, L_k, \tau_k, \rho_k, p_k, \nu_1, \nu_2)$

    For $k = m, \ldots, 1$ **step** -1 **do:**

    If $k = 1$ **do:**

        1) $(U_k, W_k, \Lambda) \leftarrow$ **CoarseLevel**$(k, q, L_k, U_k, W_k, \Lambda, \tau_k, \rho_k, p_k)$

    Else

        2) Relax ν_1 times with initial guess U_k, W_k :

$$\Delta_k W_k + c_1 \sum_{i=1}^{q} (u_k^i)^2 - c_2 = p_k, \quad \sum_{j=1}^{N_k} W_k^j = 0$$

$$L_k U_k = \tau_k$$

        3) Compute the residual $r_k = \tau_k - L_k U_k$

        4) Restrict $\tau_{k-1} = L_{k-1} I_k^{k-1} U_k + I_k^{k-1} r_k$

        5) Set $p_{k-1} = \Delta_{k-1} I_k^{k-1} W_k + \sum_{i=1}^{q} (I_k^{k-1} u_k^i)^2 + I_k^{k-1}(p_k - \Delta_k W_k - \sum_{i=1}^{q} (u_k^i)^2)$

        6) Set $\rho_{k-1} = \|I_k^{k-1} U_k\| + \rho_k - \|U_k\|$

        7) Restrict :

$$U_{k-1} = I_k^{k-1} U_k,$$

$$W_{k-1} = I_k^{k-1} W_k$$

    endif

    For $k = 2, \ldots, m$ **step** 1 **do:**

        9) **If** $k < m$ Interpolate and FAS correct:

$$U_k = U_k + I_{k-1}^k (U_{k-1} - I_k^{k-1} U_k)$$

$$W_k = W_k + I_{k-1}^k (W_{k-1} - I_k^{k-1} W_k)$$

    endif

        10) Relax ν_2 times:

$$L_k U_k = \tau_k$$

$$\Delta_k W_k + c_1 \sum_{i=1}^{q}(u_k^i)^2 - c_2 = p_k, \quad \sum_{j=1}^{N_k} W_k^j = 0$$

CoarseLevel$(k, q, L_k, U_k, W_k, \Lambda, \tau_k, \rho_k, p_k)$

    Do until convergence :

        1) Update (U_k, Λ) by Projection and Backrotation

        2) Solve for W:

$$\Delta_k W_k + c_1 \sum_{i=1}^{q}(u_k^i)^2 - c_2 = p_k, \quad \sum_{j=1}^{N_k} W_k^j = 0$$

        3) Relax $L_k U_k = \tau_k$

The constant γ is the number of cycles performed on each level. The ν_1, (ν_2) is the number of relaxations performed in the simultaneous cycle, on each level in the path from fine to coarse, (coarse to fine). Such a V cycle will be denoted $V(\nu_1, \nu_2)$ and the FMG with γ cycles as above will be denoted by $\gamma - FMG - V(\nu_1, \nu_2)$.

If all desired eigenvectors cannot be well approximated on coarsest level then the Nonlinear-FMG algorithm can be used in an adaptive version in which the Nonlinear-FMG is performed for clusters of close or equal eigenvalues, each cluster having its own coarsest level. The single difference is that in the computations of p, the sums for the eigenvectors are performed not only for the eigenvectors in the cluster but for all eigenvectors in the other approximated clusters, on the common levels, (else a restriction of W can be used). The clusters of close and equal eigenvalues have to be completed in order to obtain robustness and efficiency. The constants γ, ν_1, ν_2 and the coarse level on which to perform the projection efficiently can be found adaptively. For these the adaptive techniques presented in [7] can be used.

3.6 Storage, Work and Accuracy

In the algorithm presented in the previous section, storage is required for the q eigenvectors u_k^i of size N on finest grid, the potentials and the corresponding right hand sides, on all levels, giving an overall estimate of memory of order $O(3(N + 3))$ for problems in 2-D and 3-D. The work requires $O(N)$ operations per eigenvector and $O(N)$ operations for the nonlinear potential. The work performed on coarsest grids should be added to these estimates. Usually this work does not change the complexity of the algorithm, being only a part of the fine level work. In the case of degenerate or clustered eigenvalues, if zero scalar products are needed on finest levels, inside the degenerate or clustered eigenspaces, then orthonormalizations may be required within these eigenspaces on the finest level. However, as can be seen in the computational examples, accurate orthogonality inside degenerate clusters may be obtained by coarse level separation also. The schemes presented $O(h^2)$

accuracy for the 5-point in 2-D and 9-point in 3-D Laplacian, for an 1-FMG-V(1,1) algorithm, as seen in the outputs.

4 Computational Results

The Tables 1, 2, 3 present results for the 2D, nonlinear eigenvalue problem (1) with the potential $V(x, y) = 14 - (2\pi/a)^2 f(x, y)/(7 + f(x, y))$. Here $f(x, y) = sin(10x + 10y) + cos(10x + 10y)$, ($a = 2\pi/10$ is the size of the domain in both directions). V is chosen so in order to determine a cluster consisting of two clusters of two equal eigenvalues. An 1-FMG-V(1,1) algorithm was used to show that one $V(1, 1)$ cycle per level is enough to obtain a second order convergence towards the continuous solution. See for this the residuals at the beginning of the first V cycle on each level decreasing with a factor about 4 from one level to the next finer level. (The mesh size decreases with a factor of 2 from one level to the next finer one.) Seven V cycles were performed on finest level 6, to show the convergence rate for eigenvectors and potential, better than 0.15 in all cycles. The convergence rate is the same for all eigenvectors in the cluster, of order 0.15 in all cycles from 3 to 7. For the potential W, three relaxations were used, but an MG cycle for W could be employed as well instead (this was not needed in the tests performed). The separation by projection is performed on level 2 instead of 1 and the eigenvalue systems were solved exactly on coarsest level. The eigenvectors are normalized to 1 on finest level. The eigenvalues presented are computed by Rayleigh quotients on finest levels. (Generally, the fine level Rayleigh quotients are not necessary, the coarse level projection providing the accurate eigenvalues, but showed to improve the efficiency of at least the first cycle on each level. In the first cycle, the eigenvalues are improved by the quotients and used on the path down before they are recomputed by the projection. This first cycle is generally sufficiently efficient for obtaining a second order scheme so that additional cycles are not necessary at least until finest level where one may desire accurate converged solutions, thus would employ several more cycles.) The degenerate eigenvalues come out with 11 to 14 equal digits. The convergence rate of the nonlinear potential is also about 0.15 per cycle, as for the eigenvectors, see Table 2. Accurate separation is obtained in the cluster and in degenerate eigenspaces, although the separation was performed on the coarse level 2, the scalar products on level 6 being of order 10^{-12}, see Table 3.

The Tables 3 to 7 present results for problems in 3D which are similar with the 2D results. The first seven eigenvectors were sought. The problems were discretized on three levels. The cycles were $V(1, 1)$ and the projections were performed on second level.

The potential $V(x, y, z) = 14 - 100sin(10x + 10y + 10z)/(30 + sin(10x + 10y + 10z))$, provides a cluster of six degenerate eigenvalues, presented in Table 4. The approximations of the degenerate

eigenvalues present 13 equal digits, on levels 2 and 3. The results in Table 5 are for the same problem with nonsymmetric V, $V(x, y, z) = 14 - 100sin(30x+20y+10z)/(30+sin(30x+20y+10z))$. On first level, V splits the previous cluster of six eigenvalues into two degenerate clusters of two and four eigenvalues. On levels 2 and 3, the cluster of four degenerate eigenvalues splits into two clusters of two degenerate eigenvalues. The degenerate eigenvalues present 14 equal digits. The six clustered eigenvalues have the first 5 digits equal. On level 3, the eigenvectors come out exactly orthogonal, their scalar products are presented in Table 6. Table 7 shows the residuals of the nonlinear potential W. The fact that the cluster structure differs on different levels introduces special computational difficulties. The problem has to be defined on complete clusters of eigenvectors and the clusters have to be completed. These difficulties can be detected and treated by the adaptive techniques [7].

cycle	vector	start res.	end res.	eigenvalue
L E V E L 1				
5	1	0.37E-13	0.46E-13	-0.15528834591395E+02
	2	0.12E-12	0.84E-13	-0.90047054014218E+02
	3	0.85E-13	0.79E-13	-0.90047054014218E+02
	4	0.74E-12	0.26E-12	-0.10369602966161E+03
	5	0.12E-11	0.45E-12	-0.10369602966161E+03
L E V E L 2				
1	1	0.44E+01	0.50E-03	-0.15182335042395E+02
	2	0.30E+02	0.22E-01	-0.10144043667188E+03
	3	0.30E+02	0.22E-01	-0.10144043667188E+03
	4	0.32E+02	0.47E-01	-0.12014770030904E+03
	5	0.32E+02	0.47E-01	-0.12014770030904E+03
3	1	0.17E-05	0.60E-08	-0.15182335072480E+02
	2	0.43E-04	0.88E-07	-0.10144043560798E+03
	3	0.43E-04	0.88E-07	-0.10144043560798E+03
	4	0.19E-03	0.84E-06	-0.12014769418108E+03
	5	0.19E-03	0.84E-06	-0.12014769418108E+03
L E V E L 3				
1	1	0.13E+01	0.59E-01	-0.15069813064192E+02
	2	0.11E+02	0.46E-01	-0.10444903871181E+03
	3	0.11E+02	0.46E-01	-0.10444903871181E+03
	4	0.12E+02	0.88E-01	-0.12465344903258E+03
	5	0.12E+02	0.88E-01	-0.12465344903258E+03
L E V E L 4				
1	1	0.36E+00	0.22E-01	-0.15039575054851E+02
	2	0.31E+01	0.32E-01	-0.10521096070648E+03
	3	0.31E+01	0.32E-01	-0.10521096070648E+03
	4	0.33E+01	0.19E-01	-0.12580555034765E+03
	5	0.33E+01	0.19E-01	-0.12580555034763E+03
L E V E L 5				
1	1	0.95E-01	0.65E-02	-0.15031924453065E+02
	2	0.79E+00	0.13E-01	-0.10540212079295E+03
	3	0.79E+00	0.13E-01	-0.10540212079291E+03
	4	0.85E+00	0.85E-02	-0.12609530927232E+03
	5	0.85E+00	0.85E-02	-0.12609530927231E+03
L E V E L 6				
1	1	0.24E-01	0.17E-02	-0.15030004902969E+02
	2	0.20E+00	0.39E-02	-0.10544995104364E+03
	3	0.20E+00	0.39E-02	-0.10544995104306E+03
	4	0.21E+00	0.28E-02	-0.12616785302342E+03
	5	0.21E+00	0.28E-02	-0.12616785302487E+03
3	1	0.17E-03	0.18E-04	-0.15030004885985E+02
	2	0.46E-03	0.55E-04	-0.10544995101300E+03
	3	0.46E-03	0.55E-04	-0.10544995101134E+03
	4	0.34E-03	0.42E-04	-0.12616785302509E+03
	5	0.34E-03	0.42E-04	-0.12616785302485E+03
7	1	0.29E-07	0.88E-08	-0.15030004896583E+02
	2	0.92E-07	0.11E-07	-0.10544995101183E+03
	3	0.92E-07	0.11E-07	-0.10544995101118E+03
	4	0.80E-07	0.11E-07	-0.12616785302732E+03
	5	0.80E-07	0.99E-08	-0.12616785302532E+03

Table 1: The residuals and eigenvalues of the first 5 eigenvectors of the discretized Nonlinear Schrödinger eigenvalue problem in 2D, on 6 levels, computed by an 1-FMG-V(1,1) simultaneous algorithm. On first level 5 cycles were performed and on second level 3 cycles. The projection was performed on level 2. Seven cycles were performed on finest level to illustrate a constant convergence rate per MG cycle of 0.15. The residuals are computed at start and the end of the $V(1,1)$ cycles; and the eigenvalues at the end of the cycles by Rayleigh Quotients. The decrease of the residuals by a factor of four from one level to the next (the start residuals in the first cycle, on fine levels) indicate a second order convergence towards the differential solution for the eigenvectors.

cycle	start res.	end res.
L E V E L 1		
1	0.11E-09	0.10E-13
7	0.69E-14	0.70E-14
L E V E L 2		
1	0.35E+00	0.16E-03
2	0.16E-03	0.59E-06
3	0.59E-06	0.23E-08
L E V E L 3		
1	0.36E-01	0.13E-03
L E V E L 4		
1	0.69E-02	0.11E-03
L E V E L 5		
1	0.17E-02	0.37E-04
L E V E L 6		
1	0.44E-03	0.11E-04
2	0.11E-04	0.86E-06
3	0.86E-06	0.98E-07
4	0.98E-07	0.12E-07
5	0.12E-07	0.14E-08
6	0.14E-08	0.16E-09
7	0.16E-09	0.21E-10

Table 2: The residuals of the nonlinear potential W of the discretized Nonlinear Schrödinger eigenvalue problem in 2D, on 6 levels, computed by an 1 -FMG-V(1,1) simultaneous algorithm. Three relaxations were performed for W. On first level 5 cycles were performed and on second level 3 cycles. Seven cycles were performed on finest level to illustrate a constant convergence rate per MG cycle of 0.15. The residuals are computed at start and the end of the MG cycles. The decrease of the residuals by a factor of four from one level to the next (the start residuals in the first cycle, on fine levels) indicate a second order convergence towards the differential solution for W.

Vector 1	Vector 2	Scalar Product
1	1	0.10E+01
1	2	0.82E-14
1	3	-0.14E-12
1	4	0.12E-12
1	5	-0.30E-14
2	2	0.10E+01
2	3	0.12E-13
2	4	-0.12E-13
2	5	0.18E-13
3	3	0.10E+01
3	4	-0.17E-13
3	5	-0.86E-14
4	4	0.10E+01
4	5	0.14E-13
5	5	0.10E+01

Table 3: The scalar products of the first 5 eigenvectors of the discretized Nonlinear Schrödinger eigenvalue problem in 2D, on level 6, at the end of cycle 7. The projection was performed on level 2.

5 Adaptive Multigrid Algorithms

The techniques presented in this section were used first for linear eigenvalue problems, as we show in [14], [7]. They can be used for the nonlinear eigenvalue problem in two ways: 1) use them to solve the linear eigenvalue problems in an MG continuation procedure; and 2) use them directly as nonlinear algorithms by replacing the linear MG cycle with a nonlinear MG cycle, e.g., with the Simultaneous-FAS cycle. Their central task, to detect the cluster structure and the parameters of the algorithms is easily solved treating the linear problem first, i.e., $\epsilon = 0$ for ϵW. Then the found parameters can be used for the presented Simultaneous-Nonlinear-FMG algorithm, as shown in the computational examples in Section 4. (Note that for those examples the cluster structure had to be known in advance as well as several parameters as number of relaxations and level on which the projection should be performed efficiently. These are found by the techniques presented further.) These adaptive techniques are used mainly on coarse levels, at initial stages of the algorithm, until the cluster structure gets stabilised. It is often sufficient to use them only for the linear problem.

5.1 Motivation, Central Difficulties

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: existence of close and equal eigenvalues; unknown cluster structure; different cluster structures on different levels; inter-level 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. The nonlinear eigenvalue problem is ill posed on incomplete degenerate subspaces.

Some of the central difficulties related to the algorithms are due to: incompleteness of clusters; mixing of solutions; nonlinearity; global constraints; and unknown parameters of the algorithms,

cycle	vector	start res.	end res.	eigenvalue
L E V E L 1				
7	1	0.89E-13	0.76E-13	-0.14048591304840E+02
	2	0.93E-09	0.38E-09	-0.95098529109559E+02
	3	0.93E-09	0.38E-09	-0.95098529109559E+02
	4	0.93E-09	0.38E-09	-0.95098529109559E+02
	5	0.93E-09	0.38E-09	-0.95098529109559E+02
	6	0.93E-09	0.38E-09	-0.95098529109559E+02
	7	0.93E-09	0.38E-09	-0.95098529109559E+02
L E V E L 2				
1	1	0.90E+00	0.33E-02	-0.14040128427761E+02
	2	0.30E+02	0.18E+00	-0.10899132948707E+03
	3	0.30E+02	0.18E+00	-0.10899132948707E+03
	4	0.30E+02	0.18E+00	-0.10899132948707E+03
	5	0.30E+02	0.18E+00	-0.10899132948707E+03
	6	0.30E+02	0.18E+00	-0.10899132948707E+03
	7	0.30E+02	0.18E+00	-0.10899132948707E+03
4	1	0.22E-07	0.17E-08	-0.14040128424469E+02
	2	0.30E-03	0.23E-04	-0.10899126009610E+03
	3	0.30E-03	0.23E-04	-0.10899126009610E+03
	4	0.30E-03	0.23E-04	-0.10899126009610E+03
	5	0.30E-03	0.23E-04	-0.10899126009610E+03
	6	0.30E-03	0.23E-04	-0.10899126009610E+03
	7	0.30E-03	0.23E-04	-0.10899126009610E+03
L E V E L 3				
1	1	0.25E+00	0.46E-01	-0.14036829480230E+02
	2	0.11E+02	0.69E+00	-0.11274758485900E+03
	3	0.11E+02	0.69E+00	-0.11274758485900E+03
	4	0.11E+02	0.69E+00	-0.11274758485900E+03
	5	0.11E+02	0.69E+00	-0.11274758485900E+03
	6	0.11E+02	0.69E+00	-0.11274758485900E+03
	7	0.11E+02	0.69E+00	-0.11274758485900E+03
4	1	0.58E-03	0.65E-04	-0.14036815617277E+02
	2	0.20E-02	0.30E-03	-0.11274310146319E+03
	3	0.20E-02	0.30E-03	-0.11274310146319E+03
	4	0.20E-02	0.30E-03	-0.11274310146319E+03
	5	0.20E-02	0.30E-03	-0.11274310146319E+03
	6	0.20E-02	0.30E-03	-0.11274310146319E+03
	7	0.20E-02	0.30E-03	-0.11274310146319E+03

Table 4: The residuals and eigenvalues of the first 7 eigenvectors of the discretized Nonlinear Schrödinger eigenvalue problem in 3D, on 3 levels, computed by an 4-FMG-V(1,1) simultaneous algorithm. The linear potential is $V(x, y, z) = 14 - 100sin(10x + 10y + 10z)/(30 + sin(10x + 10y + 10z))$. On first level 7 cycles were performed. The projection was performed on level 2. The residuals are computed at start and end of the $V(1, 1)$ cycles; and the eigenvalues at the end of the cycles. Observe the 6 accurately equal eigenvalues.

cycle	vector	start res.	end res.	eigenvalue
L E V E L 1				
7	1	0.52E-12	0.23E-12	-0.14055580293076E+02
	2	0.19E-07	0.11E-07	-0.95112505605267E+02
	3	0.23E-07	0.59E-08	-0.95112505605267E+02
	4	0.47E-08	0.14E-07	-0.95112516406102E+02
	5	0.44E-07	0.12E-07	-0.95112516406102E+02
	6	0.43E-08	0.75E-09	-0.95112516406102E+02
	7	0.54E-08	0.64E-09	-0.95112516406102E+02
L E V E L 2				
1	1	0.13E+01	0.13E-04	-0.14053758492811E+02
	2	0.30E+02	0.17E+00	-0.10901746968777E+03
	3	0.30E+02	0.17E+00	-0.10901746968777E+03
	4	0.30E+02	0.17E+00	-0.10901758164786E+03
	5	0.30E+02	0.17E+00	-0.10901758164786E+03
	6	0.30E+02	0.17E+00	-0.10901781869743E+03
	7	0.30E+02	0.17E+00	-0.10901781869743E+03
4	1	0.35E-10	0.12E-10	-0.14053758492812E+02
	2	0.38E-05	0.21E-07	-0.10901741800157E+03
	3	0.38E-05	0.21E-07	-0.10901741800157E+03
	4	0.17E-04	0.83E-06	-0.10901752982146E+03
	5	0.17E-04	0.83E-06	-0.10901752982146E+03
	6	0.37E-05	0.17E-07	-0.10901776702773E+03
	7	0.37E-05	0.17E-07	-0.10901776702773E+03
L E V E L 3				
1	1	0.13E+01	0.25E+00	-0.14051499340829E+02
	2	0.11E+02	0.75E+00	-0.11277655294003E+03
	3	0.11E+02	0.75E+00	-0.11277655294003E+03
	4	0.11E+02	0.75E+00	-0.11277700995289E+03
	5	0.11E+02	0.75E+00	-0.11277700995289E+03
	6	0.11E+02	0.74E+00	-0.11277731911554E+03
	7	0.11E+02	0.74E+00	-0.11277731911554E+03
4	1	0.29E-02	0.32E-03	-0.14051251375940E+02
	2	0.64E-02	0.96E-03	-0.11277176295116E+03
	3	0.64E-02	0.96E-03	-0.11277176295116E+03
	4	0.92E-02	0.17E-02	-0.11277225319175E+03
	5	0.92E-02	0.17E-02	-0.11277225319175E+03
	6	0.55E-02	0.80E-03	-0.11277260890858E+03
	7	0.55E-02	0.80E-03	-0.11277260890858E+03

Table 5: The residuals and eigenvalues of the first 7 eigenvectors of the discretized Nonlinear Schrödinger eigenvalue problem in 3D, on 3 levels, computed by an 4-FMG-V(1,1) simultaneous algorithm. The linear potential is $V(x, y, z) = 14 - 100sin(30x + 20y + 10z)/(30 + sin(30x + 20y + 10z))$. On first level 7 cycles were performed. The projection was performed on level 2. The residuals are computed at start and the end of the $V(1, 1)$ cycles; and the eigenvalues at the end of the cycles. Observe the 6 eigenvalues with 6 common digits in the cluster of 6 consisting in 3 degenerate clusters.

24

Vector 1	Vector 2	Scalar Product
1	1	0.10E+01
1	2	0.13E-16
1	3	0.15E-15
1	4	0.59E-15
1	5	0.28E-15
1	6	-0.20E-13
1	7	0.12E-13
2	2	0.10E+01
2	3	-0.24E-15
2	4	0.51E-14
2	5	0.23E-14
2	6	-0.12E-14
2	7	-0.26E-14
3	3	0.10E+01
3	4	-0.84E-14
3	5	0.43E-13
3	6	-0.11E-14
3	7	0.44E-14
4	4	0.10E+01
4	5	-0.26E-14
4	6	0.11E-14
4	7	-0.16E-14
5	5	0.10E+01
5	6	0.60E-14
5	7	0.86E-14
6	6	0.10E+01
6	7	-0.43E-14
7	7	0.10E+01

Table 6: The scalar products of the first 7 eigenvectors of the discretized Nonlinear Schrödinger eigenvalue problem in 3D, on level 3, at the end of cycle 4. The projection was performed on level 2.

cycle	start res.	end res.
L E V E L 1		
7	0.49E-10	0.20E-10
L E V E L 2		
1	0.38E+00	0.13E-04
2	0.13E-04	0.72E-07
3	0.72E-07	0.20E-08
4	0.20E-08	0.99E-10
L E V E L 3		
1	0.30E-01	0.39E-03
2	0.39E-03	0.52E-04
3	0.52E-04	0.72E-05
4	0.72E-05	0.10E-05

Table 7: The residuals of the nonlinear potential W of the discretized Nonlinear Schrödinger eigenvalue problem in 3D, on 3 levels, computed by an 4 -FMG-V(1,1) simultaneous algorithm. Three relaxations were performed for W. The residuals are computed at start and the end of the MG cycles.

such as iteration numbers, relaxation parameters, step sizes in continuation procedures, and levels on which to apply a given procedure.

These central difficulties can be further grouped in difficulties related to a) clusters, mixing and nonlinearity, and b) unknown parameters of subroutines. The techniques introduced for treating the difficulties related to clusters, mixing and nonlinearity are the adaptive separation and completion of clusters on different levels, the simultaneous processing of clusters, the MG projections and backrotations, the subspace continuation technique, the treatment of global constraints and the simultaneous cycles. 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.

5.2 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:

$$U_k = (U_k^1, \ldots, U_k^j) \tag{37}$$

where as before, each U_k^i approximates the solution of $A_k \bar{U}_k^i = \bar{U}_k^i \Lambda^i + T_k^i$ $\quad i = 1, \ldots, j$. 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 process for this cluster. Here it is assumed that $l_c^i \leq l_p^i$. Denote $l_p = (l_p^1, \ldots, l_p^j)$, $l_c = (l_c^1, \ldots, l_c^j)$ and by $\Lambda = diag(\Lambda^1, \ldots, \Lambda^j)$. 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 Subsection 5.4. Other Adaptive-MGP algorithms are obtained by replacing the Solve-MGP with other cycles. For example a nonlinear algorithm is obtained using the more general Simultaneous-FAS instead of Solve-MGP. On coarse levels on which W cannot be properly defined, (e.g., the sums are not defined since the corresponding eigenvectors are not defined on that levels), restrictions of a fine level W to coarse levels can be used instead. The further algorithms are for the linear case but they have the same form for the nonlinear case since Adaptive-MGP is their basic element.

5.3 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 is 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 $< A_k u, u > / < u, u >$ **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 $\lambda = < A_k u, u > / < u, u >$
            $U_k^j \leftarrow (U_k^j, u)$
            $\Lambda^j \leftarrow diag(\Lambda^j, \lambda)$
            $q = q + 1, \quad d_j = d_j + 1$
    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 $U_k = (U_k^1, \ldots, U_k^j)$, $\Lambda = (\Lambda^1, \ldots, \Lambda^j)$

Observe in the nonlinear example from Table 1 that the cluster of four eigenvalues $\lambda_2 - \lambda_5$ consists of two well enough separated eigenvalues. The four corresponding eigenvectors should be treated together since they mix during the processing. They derive from a degenerate cluster (for $V = 0$, $\epsilon = 0$) and have in a sense (of dominant Fourier components) the same smoothness. For this example, the criteria which defines the clusters based on close eigenvalues does not work. The complete cluster should include all eigenvectors which get mixed by the used procedures, in our case for which the MG cycle is efficient. When accuracy improves, a cluster may split in several clusters.

5.4 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 corresponding stabilized clusters, will require the same parameters l_c, l_p. Thus, robustness tests are applied on coarse levels until clusters get stabilized. For non-stabilized 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_q 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 $(l_{p,m}, l_{c,m})$:
                $\min_{l_P, l_C} \mu(l_P, l_C)$, $l_C \leq l_P \leq m$,
            else

$$(l_{p,m}, l_{c,m}) = (l_{p,m-1}, l_{c,m-1})$$
$\quad$ **endif**

endif

Convergence Remark 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 (for $l_c = l_p = m$ when $\mu < 1$). This was not proved for the nonlinear algorithm.

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.

For the nonlinear algorithm parameters have to be found for the continuation procedures, e.g., the continuation steps need often to be small at initial stages but becomes larger when the solution to the nonlinear problem becomes better approximated. The number of iterations decreases in these cases as the efficiency of the cycles increases and tends to rich the efficiency of the linear algorithms. When this efficiency is reached, one may consider that the approximate solution is in a right neighborhood and may continue the FMG to next levels.

5.5 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 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)
$\quad$ Set $k = 1$, $q' = 0$, $j = 0$, $l_p = k$, $l_c = k$
$\quad\quad$ **Until** $(q' \geq q$ or $q' \geq \alpha \dim_k)$ **Perform**
$\quad\quad\quad (j, U_k, \Lambda, T_k, q') \leftarrow$ **Add-Cluster**$(j, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$\quad\quad\quad (U_k, \Lambda, T_k) \leftarrow$ **Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$\quad$ **Until** $k \geq m$ **Do:**
$\quad\quad$ **If** $k < m$ **then:**
$\quad\quad\quad$ Set $k = k + 1$, $U_k = I_{k-1}^k U_{k-1}$, $T_k = 0$
$\quad\quad$ **endif**
$\quad\quad (l_p, l_c) \leftarrow$ **Robustness-Test** $(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$\quad\quad$ **If** $(q' \geq q)$ **then:**
$\quad\quad\quad$ **If** (**Cluster-Completion-Test**=**TRUE**) **then:**
$\quad\quad\quad\quad (U_k, \Lambda, T_k) \leftarrow$ **Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$\quad\quad\quad$ **Else**
$\quad\quad\quad\quad (U_k^j, \Lambda^j, T_k^j, q') \leftarrow$ **ClusterCompletion**$(j, A_k, U_k^j, \Lambda^j, T_k^j, l_p^j, l_c^j, q')$
$\quad\quad\quad\quad (U_k, \Lambda, T_k) \leftarrow$ **Adaptive-MGP**$(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$
$\quad\quad\quad$ **endif**
$\quad\quad$ **Else**
$\quad\quad\quad$ **Until** $(q' \geq q$ or $q' \geq \alpha \dim_k)$ **Perform**

$$(j, U_k, \Lambda, T_k, q') \leftarrow \textbf{Add-Cluster}(j, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$$
$$(U_k, \Lambda, T_k) \leftarrow \textbf{Adaptive-MGP}(k, A_k, U_k, \Lambda, T_k, l_p, l_c, q')$$
       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).

Our MG approach differs from previous MG approaches [9],[10], [5], [2], [13], [3], [11], [4], [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.

5.6 Computational Example for the Adaptive Algorithm for the Linear Schrödinger Problem

This section presents a numerical example illustrating the adaptive algorithm for the linear Schrödinger eigenvalue problem, shown in [7]. In this case we have $\epsilon = 0$. The example demonstrates the central difficulties related to clusters and mixing, and illustrates the efficiency of the presented techniques in overcoming these difficulties. 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 Schrödinger eigenvalue problem

$$(\Delta - V)u = \lambda u \tag{38}$$

with periodic boundary conditions, defined on $\Omega = [0, a]^d$, (d=2 or 3), where $a = 2\pi/10$, is considered. The i'th eigenvalue and eigenvector will be denoted next by λ_i and v_i. The potential V is 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 could be used as well. Richardson type extrapolations based on the sequence of solutions obtained on the different levels could 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 potential $V(x, y) = 5 + 3sin(10x)$ was considered. 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 8 and 9.

The boxes in Table 8 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 6 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 Section 5.5 is used. To clarify the adaptive flow of the algorithm, a full history of the run is given.

The algorithm started on level 1 adding eigenvectors until the cluster containing λ_{12} was completed. The last 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 $O(10^{-4})$ 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 first cluster $\{v_1\}$, the values were obtained from previous level since this cluster was stabilized from level 2; 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 since these clusters resulted 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 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 3 clusters, eigenvectors $\{v_1 - v_9\}$ resulted 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 resulted 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 didn't separate it from the next eigenvectors in its cluster, e.g., on level 2, λ_{14} was not separated from the next 7 eigenvectors with close eigenvalues.

The left columns, in Table 9, show the residuals after the cubic interpolation in the FMG. These residuals decrease with a factor of four (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 9, 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.

31

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.

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				

Table 8: The first 16 eigenvalues (E) of the discretized Schrödinger eigenvalue problem in 2D, on 5 levels, computed by an 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 equal digits of eigenvalues in each cluster.

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 9: The residuals of the 16 eigenvectors (E) of the discrete Schrödinger eigenvalue problem in 2D, on 5 levels, computed by an 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 four from one level to the next (on fine levels, left column) indicate 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.

33

5.7 Observations on the Algorithms

Observations and details of the algorithms, not introduced before in order to keep the exposition 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 neighbours.

3) Different relaxations can be used in the algorithms, like damped Jacobi, SOR, Richardson, Kaczmarz, block relaxations, see for example [1]. We consider two types of relaxations 1) power type iterations (with shifts), e.g. Richardson relaxations for operator H:

$$u^{n+1} = (I - \omega(H - \mu I))u^n \tag{39}$$

and 2) solver type relaxations like Kaczmartz or Gauss-Seidel.

To approximate eigenspaces, at the initial stages when eigenvalues are poorly approximated, one can use power relaxations which are generally slow but safe. If the eigenvalues are enough separated and well approximated then solver relaxations may separate the eigenvectors. Gauss-Seidel relaxations are generally faster than power relaxations but can be unsafe, e.g. amplifying unwanted eigenvectors if the eigenvalues are not accurate. In the numerical presented tests, the red-black ordering Gauss-Seidel relaxations were performed.

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 (16) can be replaced by a more general set of vectors Y^T.

3) Solutions (E, Λ) of (15) 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 (15) 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 [1]. 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 Ritz 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 (5) 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.

34

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 eigenvectors 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, (6), 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:*

$$U_i = U_i E + I_j^i(U_j - I_i^j U_i E), \quad T_i = T_i E, \tag{40}$$

In this correction the right hand side T is updated also. In (40) the multiplication $U_i E$ is of the same order of work as needed for an Rayleigh Ritz separation for U_i. Still, the cheaper correction (6) can be used instead of (40) when solution are sufficiently accurate and using backrotations. This is shown by the computational examples too. The correction (40) 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 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 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 to find such parameters

in general cases.

6   Conclusions

An MG simultaneous algorithm for a nonlinear Schrödinger eigenvalue problem is presented. The algorithm combines the following techniques: the MG projection and backrotations; the MG subspace continuation technique; the FAS treatment of global constraints; the simultaneous processing of eigenvectors, nonlinear potential and global constraints. In the computational examples, the simultaneous MG technique reduced the large number of sequential selfconsistent iterations to one MG simultaneous iteration (1-FMG here). One simultaneous cycle involves less computations than one sequential cycle (updating eigenvectors sequentially and separating them on finest level) due to the cheap coarse level separation by the MGP and backrotations. The MG subspace continuation techniques, coupled with the simultaneous processing on all levels helped keeping the approximated solution in a right neighborhood where the algorithm is efficient. MG projections and backrotations are used to separate the eigenvectors by coarse level work and to overcome difficulties due to close or equal eigenvalues. Robustness is obtained from the adaptive completion of clusters and from tests which monitor the algorithm's convergence and efficiency.

Computational examples for the nonlinear Schrödinger eigenvalue problem in 2D and 3D having special computational difficulties, which are due to equal and closely clustered eigenvalues, are presented. For these cases, the algorithm requires $O(qN)$ operations for the calculation of q eigenvectors of size N. 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 5-point in 2D and 9-point in 3D discretized Laplacian, by 1-FMG-V(1,1) in $O(qN)$ work. The work was of order of a few (about 8) fine level Gauss-Seidel relaxations per eigenvector. Constant convergence rate per cycle of 0.15 was obtained for the presented cases. The robustness of the algorithm has been demonstrated on problems with eigenvalue distributions that present special difficulties. 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. This reduced the expensive fine level separation work of order $O(q^2N)$ of previous algorithms, to coarse level work of order $O(qN)$.

References

[1] A. Brandt; *Multigrid Techniques 1984 Guide with Applications to Fluid Dynamics;* GDM Studien Nr. 85. Gesellschaft für Mathematik und Datenverarbeitung; St. Augustin, 1984

[2] A. Brandt, S. McCormick and J. Ruge; *Multigrid Methods for Differential Eigenproblems;* SIAM. J. SCI. STAT. COMPUT., Vol 4, No 2, 1983.

[3] W. Hackbusch; *Multigrid methods and applications;* Springer Series in Comp. Math. 4, Springer-Verlag, Berlin, 1985

[4] S. F. McCormick; *Multilevel Projection Methods for Partial Differential Equations;* SIAM, 1992

[5] R. E. Bank; *Analysis of a multi-level inverse iteration procedure for eigenvalue problems;* SIAM J. Numer. Anal., 19, pp. 886-898, 1982

[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, Rehovot, Israel, 1992

[7] S. Costiner, S. Ta'asan; *Adaptive Multigrid Techniques for Large Scale Eigenvalue Problems; Solutions of the Schrödinger Problem in Two and Three Dimensions;* submitted to Physical Review E, 1994

[8] S. Costiner, F. Manolache, S. Ta'asan; *Multigrid Methods Applied to the Design of Resonant Cavities;* In print at IEEE - MTT

[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] S. F. McCormick; *A mesh refinement method for $Ax = \lambda Bx$;* Math. Comp., 36, pp 485-489, 1981

[11] J. Mandel, S. McCormick; *A Multilevel Variational Method for $Au = \lambda Bu$ on Composite Grids;* JCP 80, pp 442-452, 1989

[12] L.Y. Zaslavsky; *An Adaptive Algebraic Multigrid for Multigroup Neutron Diffusion Reactor Core Calculations;* Applied Math. and Computation 53, pp 13-26, Elsevier, 1993

[13] 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

[14] S. Costiner, S. Ta'asan; *A Robust Multilevel Simultaneous Eigenvalue Solver;* NASA CR-194291, ICASE Report 93-35; ICASE, NASA Langley Research Center, Hampton, VA 23681, USA, 1993

[15] W. Kerner; *Large-Scale Complex Eigenvalue Problems, Review Article;* JCP. 85, pp 1-85, 1989

[16] B. N. Parlett; *The Symmetric Eigenvalue Problem;* Prentice-Hall, Englewood Cliffs, NJ, 1980

[17] J.H. Wilkinson; *The Algebraic Eigenvalue Problem;* Clarendon Press, Oxford, 1965

[18] G. H. Golub, C.F. Van Loan; *Matrix Computations;* The Johns Hopkins University Press, 1989

[19] 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

[20] H. Rutishauser; *Computational aspects of F.L. Bauer's simultaneous iteration method;* Numer. Math. 13, pp 4-13, 1969

[21] S. F. McCormick, T. Noe; *Simultaneous iteration for the matrix eigenvalue problem;* J. Lin. Alg. Appl. 16, pp 43-56, 1977

[22] S. Ta'asan; *Multigrid Method for Stability Calculations;* J. of Scientific Computing, Vol. 3. No. 3. 1988

[23] S. Ta'asan, S. Costiner; *Multilevel Techniques for Large Scale Eigenvalue Problems* Proc. of the *Lanczos Centenary Conference 1993,* North Carolina State University, 1994

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE October 1994	3. REPORT TYPE AND DATES COVERED Contractor Report

4. TITLE AND SUBTITLE
MULTIGRID TECHNIQUES FOR NONLINEAR EIGENVALUE PROBLEMS; SOLUTIONS OF A NONLINEAR SCHRÖDINGER EIGENVALUE PROBLEM IN 2D AND 3D

5. FUNDING NUMBERS
C NAS1-19480
WU 505-90-52-01

6. AUTHOR(S)
Sorin Costiner
Shlomo Ta'asan

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. 94-91

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-194999
ICASE Report No. 94-91

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Michael F. Card
Final Report
To be submitted to Physical Review E, August 1994

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 64

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This paper presents MG techniques for nonlinear EP and emphasizes an MG algorithm for a nonlinear Schrödinger EP. The algorithm overcomes the mentioned difficulties combining the following techniques: an MG projection coupled with backrotations for separation of solutions and treatment of difficulties related to clusters of close and equal eigenvalues; MG subspace continuation techniques for the treatment of the nonlinearity; an MG simultaneous treatment of the eigenvectors at the same time with the nonlinearity and with the global constraints. The simultaneous MG techniques reduce the large number of selfconsistent iterations to only a few or one MG simultaneous iteration and keep the solutions in a right neighborhood where the algorithm converges fast.

14. SUBJECT TERMS
multigrid, eigenvalue problems, nonlinear problems, Schrödinger, subspace techniques

15. NUMBER OF PAGES
40

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