On correction equations and domain decomposition for computing invariant subspaces

Bernard Philippe * Yousef Saad †

February 1, 2006

Abstract

By considering the eigenvalue problem as a system of nonlinear equations, it is possible to develop a number of solution schemes which are related to the Newton iteration. For example, to compute eigenvalues and eigenvectors of an $n \times n$ matrix A, the Davidson and the Jacobi-Davidson techniques, construct 'good' basis vectors by approximately solving a "correction equation" which provides a correction to be added to the current approximation of the sought eigenvector. That equation is a linear system with the residual r of the approximated eigenvector as right-hand side.

One of the goals of this paper is to extend this general technique to the "block" situation, i.e., the case where a set of p approximate eigenpairs is available, in which case the residual r becomes an $n \times p$ matrix. As will be seen, solving the correction equation in block form requires solving a Sylvester system of equations. The paper will de£ne two algorithms based on this approach. For symmetric real matrices, the £rst algorithm converges quadratically and the second cubically. A second goal of the paper is to consider the class of substructuring methods such as the Component Mode Synthesis (CMS) and the Automatic Multi-Level Substructuring (AMLS) methods, and to view them from the angle of the block correction equation. In particular this viewpoint allows us to de£ne an iterative version of well-known one-level substructuring algorithms (CMS or one-level AMLS). Experiments are reported to illustrate the convergence behavior of these methods.

1 Introduction

A number of schemes have been developed in recent years for enhancing the convergence of subspace-based methods for computing eigenvalues and eigenvectors of large matrices. These approaches essentially take the viewpoint that the eigenvalue problem is a nonlinear system of equations and attempt to £nd a good way to correct a given approximate eigenpair $\tilde{\lambda}, \tilde{u}$, by introducing to the most recent subspace of approximants, an information that is not redundant with this subspace. In practice, this means that we need to solve the correction equation, i.e., the equation which updates the current approximate eigenvector, in a subspace that is orthogonal to the most current approximate eigenvectors.

Several methods can be mentioned including the Trace Minimization method [14, 13], the Davidson method [6, 11] and the Jacobi-Davidson approach [16, 15, 18]. Most of these methods update an existing approximation by a step of Newton's method and this was illustrated in a number of papers, see, e.g., [10], and in [19].

One can think of the problem as that of solving $(A - \lambda I)u = 0$, but since there are $n + 1$ unknowns, a constraint must be added, for example, $\|u\|_2 = 1$. If the current approximate eigenpair is $(\tilde{\lambda}, \tilde{u})$, it is assumed that $\|\widetilde{u}\|_2 = 1$, and that $\widetilde{\lambda}$ is the Rayleigh quotient of $\widetilde{u}$. We de£ne the residual $r = A\widetilde{u} - \widetilde{\lambda}\widetilde{u}$. If a

*INRIA/IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France.

†Department of Computer Science and Engineering, University of Minnesota, 200 Union Street S.E., Minneapolis, MN 55455. email: saad@cs.umn.edu. Work supported by NSF grant ITR-0428774, by DOE under Grant DE-FG02-03ER25585, and by the Minnesota Supercomputing Institute

correction δ, z to $\tilde{\lambda}, \tilde{u}$ is sought, then we write the equations to be satis£ed as

$$\begin{aligned}[(A-\tilde{\lambda}I) \ - \ \delta I](\widetilde{u}+z) &= 0 \\ (\widetilde{u}+z)^T(\widetilde{u}+z) &= 1\,.\end{aligned}$$

Ignoring second order terms, this yields the system of equations,

$$(A-\tilde{\lambda}I)z \ - \ \widetilde{u} \ \delta = -r \tag{1}$$

$$-\widetilde{u}^T z = 0\,. \tag{2}$$

The above equations can be solved in a number of ways, for example as an $(n+1)\times(n+1)$ linear system. A simpler solution is to invoke the orthogonal projector $P = I - \widetilde{u}\widetilde{u}^T$. Multiplying the £rst equation by P, and observing that $P\widetilde{u} = 0, Pr = r$, yields,

$$P(A-\tilde{\lambda}I)z = -r. \tag{3}$$

Note that this system is degenerate - i.e., it has an in£nite number of solutions z. Among all the solutions of this system, we require one that is orthogonal to $\widetilde{u}$, i.e., one such that $Pz = z$. This can be enforced by solving $P(A-\widetilde{\lambda}I)Pz_0 = -r$, for z_0, and de£ning the solution to be $z = Pz_0$ (instead of z_0). This z will satisfy the equation (1). Indeed,

$$(A-\tilde{\lambda}I)z = P(A-\tilde{\lambda}I)z + (I-P)(A-\tilde{\lambda}I)z = -r + \widetilde{u}\widetilde{u}^T(A-\tilde{\lambda}I)z = -r + \widetilde{u}r^T z$$

and therefore, (1) is satis£ed with $\delta = r^T z$. In addition (2) is trivially satis£ed because $z = Pz_0$. Note that the correction δ to $\widetilde{\lambda}$ can be ignored since the new approximate eigenvalue will just be de£ned as the new Rayleigh quotient. So we are left with the equation,

$$P(A-\widetilde{\lambda}I)Pz = -r. \tag{4}$$

The Jacobi-Davidson scheme [16], the Trace Minimization method [13] and a number of related algorithms are based on the above development. In other methods, the projection is not considered since the matrix A is not used exactly. Instead, A is replaced by a "preconditioner" when solving the system $(M-\widetilde{\lambda}I)z = -r$ in place of the system (3). This viewpoint is most useful in a Davidson approach to build a subspace for a projection technique [19].

The Newton-type framework just described determines one vector at a time and it is interesting to explore situations where a block of vectors must be corrected. This is important in many practical applications. We will explore a few block correction schemes which are derived in a manner that is similar to what was done above for the one-dimensional case.

One of the possible applications of these schemes lies in domain decomposition methods. In these methods, one can consider the invariant subspaces obtained from subdomains as approximately invariant for the global operator. Such approximations can be very rough and one may be tempted to correct them in some way. This technique is taken to the limit and exploited in a quite effective way in the Automated Multi-Level Substructuring (AMLS) algorithm [2, 8]. In AMLS, the subspaces are corrected by adding correction terms from the interface variables.

The main goal of this paper is to explore block versions of the correction equation. By considering the correction equation as a nonlinear system of equations we can derive Newton-type iterations whose convergence is quadratic or cubic. We will also consider the Automatic Multi-Level Substructuring algorithm, and domain-decomposition methods, from the point of view of a block correction.

2 Block correction

This section examines a few schemes for "correcting" a given approximate invariant subspace. We are given a subspace in the form of a certain basis $U = [u_1, u_2, \cdots . u_m]$ and would like to £nd a correction W of the same dimensions as U, such that $U + W$ is a better invariant subspace than U. Schemes of this type are well-known for the case when $m = 1$, and they lead to the standard Olsen's method or Jacobi-Davidson scheme.

2.1 Correction of an orthonormal basis

Let us assume that $U \in \mathbb{R}^{n\times m}$ is an orthonormal basis of an approximate invariant subspace of $A \in \mathbb{R}^{n\times n}$. In particular, $U^TU = I$. Let $D = U^TAU$ be the interaction matrix whose eigenvalues are approximations of eigenvalues of A. The residual of the corresponding subspace is :

$$\begin{aligned} R &= AU - UD \quad (5)\\ &= (I - UU^T)AU. \quad (6)\end{aligned}$$

The last expression shows that R lies in a space that is orthogonal to U, i.e.,

$$U^TR = 0. \quad (7)$$

The goal is to obtain $(W, \Delta) \in \mathbb{R}^{n\times m}, \mathbb{R}^{m\times m}$ respectively, which will correct (U, D) so that the perturbed pair of matrices $(U + W, D + \Delta)$ satisfy the (nonlinear) equation :

$$A(U + W) = (U + W)(D + \Delta). \quad (8)$$

This equation involves mn equations and $mn + m^2$ unknowns. In order to close the system, m^2 equations must be added. We consider the additional constraint :

$$U^TW = 0. \quad (9)$$

The above constraint may seem arbitrary but it can be interpreted as follows. It can be viewed as a means of restricting the information that is being added to the current system (W) to being non redundant. Another condition we could have imposed is that the new system $U + W$ should be orthonormal. This would have m^2 constraints as desired, but these constraints are nonlinear. However, up to second order approximation these constraints will imply the requirement (9). Indeed,

$$(U + W)^T(U + W) = I \quad \rightarrow U^TW + W^TU + W^TW = 0\,.$$

Neglecting second order terms from the system of equations (8) and (9), yields the equations:

$$\begin{cases} AW - WD - U\Delta = -R \\ U^TW = 0. \end{cases} \quad (10)$$

This system is actually a general expression of equations (1) and (2) when vectors are replaced by blocks of vectors. By multiplying the £rst equation on the left side by U^T, and using relation (7) we obtain the following expression for Δ,

$$\Delta = U^TAW. \quad (11)$$

Therefore, system (10) is equivalent to solving

$$(I - UU^T)AW - WD = -R \quad (12)$$

and then computing $\Delta = U^TAW$. It can be easily shown that the obtained solution W satis£es $U^TW = 0$ as required. Formula (12) generalizes formula (4). To prove (4), we preferred to use a more classical derivation.

2.2 Non orthonormal systems and the projection viewpoint

We now adapt what was developed above to the correction of a non orthonormal basis $X \in \mathbb{R}^{n\times m}$ of an approximation of an invariant subspace. Along with X is a certain representation of A in the subspace in the form of a matrix $D \in \mathbb{R}^{m\times m}$ such that A, X, and D satisfy the relation

$$AX = XD + R,$$

where R is a certain residual matrix. The only requirement on X, D, and R is that $X^T R = 0$. This implies in particular that $D = (X^T X)^{-1}(X^T AX)$.

We seek $(W, \Delta) \in \mathbb{R}^{n\times m} \times \mathbb{R}^{m\times m}$ such that $X^T W = 0$ and

$$A(X + W) \quad = \quad (X + W)(D + \Delta). \tag{13}$$

When the above equations are satis£ed, then $(X+W)$ spans an invariant subspace of A and the eigenvalues of $(D + \Delta)$ are eigenvalues of A. By neglecting the second order terms, the equation becomes :

$$AW - WD - X\Delta = -R, \tag{14}$$

which implies that $\Delta = (X^T X)^{-1}(X^T AW)$. Let $Q = X(X^T X)^{-1}X^T$ be the orthogonal projection onto X and $P = I - Q$ the dual projector. The £nal equation which generalizes equation (12) is :

$$PAW - WD \quad = \quad -R \tag{15}$$

with $(I - P)W = W$.

It is clear that if X and an orthonormal system U span the same subspace, then the resulting subspace obtained by the above process should be identical with the one resulting from the treatment of Section 2.1. In other words the matrices $W, \tilde{W}$ obtained in both cases are related by a nonsingular $p \times p$ transformation S, i.e., $W = \tilde{W}S$.

2.3 Nonlinear correction

Clearly the original correction equation is nonlinear and its exact solution will yield an exactly invariant subspace. It is possible to solve the non-linear equations iteratively and this section explores this possibility. In this section we go back to the case when U is orthonormal. Equation (8) is expanded as

$$AW - WD = -R + U\Delta + W\Delta \tag{16}$$

We still need to impose m^2 constraints in the form of (9). Multiplying both sides of (16) by U^T yields the same expression for Δ, i.e., Δ is again given by (11). This means that we have again removed the unknown Δ from the system (8) and we can write:

$$AW - WD = -R + (U + W)U^T AW \tag{17}$$

Grouping like terms leads to the nonlinear system,

$$(I - UU^T)AW - W(D + U^T AW) + R = 0\,. \tag{18}$$

This system can be solved with some form of iteration and then Δ can be obtained from $\Delta = U^T AW$. The solution W obtained also satis£es $U^T W = 0$ as required.

It is interesting to examine a Newton approach for solving (18). Newton's method is equivalent to starting from a certain W_0 (for example $W_0 = 0$) and then iterating as $W_{k+1} = W_k + Z_k$ where $W_k + Z_k$ is made to satisfy (18) for up to £rst order terms in Z_k, This yields after a little calculation,

$$(I - (U + W_k)U^T)AZ_k - Z_k(D + U^T AW_k) = -R - (I - UU^T)AW_k + W_k(D + U^T AW_k) \tag{19}$$

The above system is a Sylvester equation in Z_k and with the notation $U_k = U + W_k = U_{k-1} + Z_{k-1}$ and $D_k = D + U^T AW_k = U^T AU_k$ it becomes

$$(I - U_k U^T)AZ_k - Z_k D_k = -R_k \tag{20}$$

where R_k is the right-hand side seen above

A few observations will provide some insight. The matrix $D_k = D + U^T AW_k$ is the right-corrected projection matrix since $D_k = D + U^T AW_k = U^T A(U + W_k)$. There is some loss of symmetry in this process. In the end the Newton scheme would be as follows:

0. Select W_0 (e.g., $W_0 = 0$)
1. For $k = 0, \cdots$, until convergence Do:
2. Set $D_k = D + U^T A W_k$; $R_k = (I - UU^T)AW_k - W_k D_k + R$
3. Solve (for Z_k): $(I - (U + W_k)U^T)AZ_k - Z_k D_k = -R_k$
4. Set $W_{k+1} = W_k + Z_k$
5. EndDo

Observe that when $W_0 = 0$, then the £rst step corresponds simply to the previous linearized scheme and this is expected. From the de£nition of U_k and D_k, the iteration can be rewritten in terms of U_k. Note £rst that the expression for R_k can be simpli£ed :

$$\begin{aligned} R_k &= (I - UU^T)AW_k - W_k D_k + R \\ &= (I - UU^T)A(U_k - U) - (U_k - U)D_k + R \\ &= (I - UU^T)AU_k - (I - UU^T)AU - U_k D_k + UD_k + R \\ &= AU_k - U(U^T AU_k) - R - U_k D_k + UD_k + R \\ &= AU_k - UD_k - U_k D_k + UD_k \\ &= AU_k - U_k D_k \end{aligned}$$

This gives the following alternative expression of the previous algorithm

ALGORITHM **2.1** *Newton-Sylvester iteration*

0. *Select U_0 s.t. $U_0^T U = I$ (e.g., $U_0 = U$)*
1. *For $k = 0, \cdots$, until convergence Do:*
2. *Compute $D_k = U^T AU_k$, and $R_k = AU_k - U_k D_k$*
3. *Solve (for Z_k): $(I - U_k U^T)AZ_k - Z_k D_k = -R_k$*
4. *Set $U_{k+1} = U_k + Z_k$*
5. *EndDo*

An important note here concerns the solution of the Sylvester equation in Line 3. Since we would like to solve the correction equation with the constraint that $W_k^T U = 0$, we would like the relation $Z_k^T U = 0$ should be satis£ed in general. The relation is trivially satis£ed for $k = 0$ a consequence of the choice made in Line 0. For a general k, the relation $Z_k^T U = 0$ is equivalent to $U_k^T U = I$. We can use a little induction argument. Assume that the relation is satis£ed for k. Then multiplying the Sylvester equation by U^T yields $U^T Z_k D_k = 0$. This will imply that $U^T Z_k = 0$ when D_k is nonsingular, which may be true under certain assumptions. However, in order to avoid dif£culties, *we will always assume that the system (20) is solved for a Z that is orthogonal to U*. When D_k is nonsingular, then as was seen, $U^T Z_k = 0$ is automatically satis£ed. When D_k is singular, we can for example shift both A and D_k by the same shift σ so that $D_k - \sigma I$ is nonsingular. Then we solve, instead of (20), its shifted variant:

$$(I - U_k U^T)(A - \sigma I)Z_k - Z_k(D_k - \sigma I) = -R_k \tag{21}$$

Now since $D_k - \sigma I$ is nonsingular, $Z_k^T U = 0$ and this Z_k is also solution of (20). In practice this issue does not arise.

When convergence takes place it is quadratic at the limit, a characteristic of Newton's method. A relation which will establish this fact independently can also be proved.

Lemma 2.1 *At each step of the Newton-Sylvester iteration the following relations hold:*

$$U_k^T U = I \tag{22}$$
$$R_k = (I - U_k U^T)AU_k \tag{23}$$
$$U^T R_k = 0 \tag{24}$$
$$R_{k+1} = -Z_k U^T A Z_k \tag{25}$$

Proof. The £rst relation was discussed in detail above. It comes from the fact that at each step $U^T Z_k = 0$. The second relation follows immediately from the de£nition of D_k and the 3rd is obtained by multiplying the second by U^T to the left and making use of (22). For the 4th relation we write,

$$\begin{aligned} R_{k+1} &= A(U_k + Z_k) - (U_k + Z_k)(D_k + U^T A Z_k) \\ &= AU_k + AZ_k - U_k D_k - Z_k D_k - U_k U^T A Z_k - Z_k U^T A Z_k) \\ &= \underbrace{R_k + (I - U_k U^T) A Z U_k - Z_k D_k}_{=0} - Z_k U^T A Z_k) = -Z_k U^T A Z_k \end{aligned}$$

■

Equations (22) and (23) show that the method is equivalent to £nding a block U_* such that $(I - U_* U^T) A U_* = 0$ subject to the condition that the system U_*, U be bi-orthogonal. Note that from (23), we can easily infer that $\|R_k\| = O(\|U_k - U_*\|)$ where U_* is the limit. Indeed, using the fact $(I - U_* U^T) A U_* = 0$

$$\begin{aligned} R_k &= (I - U_k U^T) A U_k - (I - U_* U^T) A U_* \\ &= (I - U_k U^T) A U_k - (I - U_k U^T) A U_* + (I - U_k U^T) A U_* - (I - U_* U^T) A U_* \\ &= (I - U_k U^T) A (U_k - U_*) - (U_k - U_*) U^T A U_* \end{aligned}$$

The following relationship will con£rm this and provide some additional insight:

$$R_{k+1} = -Z_k U^T A Z_k$$

In the case when $m = 1$ we get the following iteration for $k = 0, 1, \cdots$, starting with $u_0 = u$:

$$\begin{cases} d_k &= u^T A u_k \\ r_k &= A u_k - d_k u_k \\ (I - u_k u^T) A z_k - d_k z_k &= -r_k \\ u_{k+1} &= u_k + z_k \end{cases}$$

Note that the scalar d_k is an approximate eigenvalue. Indeed, $u_k^T u = (u + w_k)^T u = u^T u = 1$ and therefore $d_k = (u^T A u_k)/(u^T u_k)$. When (and if) u_k converges to an eigenvector, then d_k will converge to the corresponding eigenvalue.

The residual system in the 3rd line of the above algorithm can be transformed. The system is

$$(I - u_k u^T) A z_k - z_k d_k = -r_k \quad \rightarrow \quad (A - d_k I) z_k - u_k u^T A z_k = -r_k$$

and it can be solved in block form by setting $\xi_k = -u^T A z_k$ and putting the unknowns z_k, ξ_k in one vector:

$$\begin{pmatrix} A - d_k I & u_k \\ u^T A & 1 \end{pmatrix} \begin{pmatrix} z_k \\ \xi_k \end{pmatrix} = \begin{pmatrix} -r_k \\ 0 \end{pmatrix}$$

The advantage of writing the system in this form is that we can better exploit any sparsity in A. In contrast the matrix $(I - u_k u^T) A - d_k I$ is generally not sparse, though the Shermann-Morrisson formula can also be invoked with a similar result.

A more traditional way of invoking the Newton approach is for the case $m = 1$, by solving directly the equation $Au - u(u^T A u) = 0$, with the constraint $\|u\|_2 = 1$. Extending this to the m dimensional case is doable but the constraint that U be unitary yields a more complicated iteration. The scheme shown above avoids the problem of orthogonalization - but it yields an iteration that is somewhat nonsymmetric.

So far symmetry has not been exploited. When A is symmetric, the matrix $U^T A Z_k$ in the above expression is the transpose of $Z_k^T A U = Z_k^T (AU - UD) = Z_k^T R_0$ so that

$$A = A^T \quad \rightarrow \quad R_{k+1} = -Z_k^T R_0^T Z_k \ .$$

with $R_0 = R$. As a result one step of the process yields a residual which satis£es $R_1 = -Z_0 R_0^T Z_0$. Since $\|Z_0\|_2$ is of the order as $\|R_0\|$, an error of order ϵ should be cubed in the next iteration. Thus, one should expect to obtain cubic convergence by a process of this type. Speci£cally, if one step is performed, and the process restarted (i.e., U_1 is orthogonalized and U is set to U_1, etc.) then one should expect a cubic convergence according to this formula. This is explored next.

2.4 Iterative correction

In this section, we consider the following modi£cation of the Newton-Sylvester scheme discussed in the previous section.

ALGORITHM **2.2** *Iterative correction*

0. *Select* U_0 *(e.g.,* $U_0 = U$*)*
1. *For* $k = 0, \cdots,$ *until convergence Do:*
2. *Compute* $D_k = U_k^T A U_k$*, and* $R_k = AU_k - U_k D_k$
3. *Solve (for* W_k*):* $(I - U_k U_k^T)AW_k - W_k D_k = -R_k$
4. *Orthogonalize :* $[U_{k+1}, S_k] = \mathrm{qr}(U_k + W_k)$.
5. *EndDo*

In line 4, Matlab notation is used so that U_{k+1} is the result or orthonormalizing $U_k + W_k$.

Theorem 2.1 *When the process converges, it exhibits a cubic convergence at the limit, as expressed by the following relation :*

$$R_{k+1} \quad = \quad -W_k {R_k}^T W_k {S_k}^{-1} + O(\|R_k\|^4). \tag{26}$$

Proof. We £rst remark that $(U_k + W_k)^T (U_k + W_k) = {S_k}^T S_k = I + {W_k}^T W_k$.
Let us denote $\overline{D}_k = (U_k + W_k)^T A (U_k + W_k)$ and $\overline{R}_k = R_{k+1} S_k$; hence

$$\overline{R}_k = A(U_k + W_k) - (U_k + W_k)({S_k}^T S_k)^{-1} \overline{D}_k \ .$$

Therefore, with $\Delta_k = {U_k}^T A W_k$, the correction implies that

$$A(U_k + W_k) - (U_k + W_k)(D_k + \Delta_k) = -W_k \Delta_k, \tag{27}$$

and remembering that ${U_k}^T W_k = 0$,

$$\overline{D}_k - (I + {W_k}^T W_k)(D_k + \Delta_k) \quad = \quad -{W_k}^T W_k \Delta_k,$$

which can be rewritten as

$$\overline{D}_k = D_k + \Delta_k + {W_k}^T W_k D_k. \tag{28}$$

For the residual, we estimate $\overline{R}_k = R_{k+1} S_k$ as follows,

$$\begin{aligned}
\overline{R}_k &= A(U_k + W_k) - (U_k + W_k)(I - {W_k}^T W_k)\overline{D}_k + O(\|W_k\|^4), \\
&= A(U_k + W_k) - (U_k + W_k)\overline{D}_k + (U_k + W_k){W_k}^T W_k \overline{D}_k + O(\|W_k\|^4), \\
&= -W_k \Delta_k - (U_k + W_k){W_k}^T W_k D_k + (U_k + W_k){W_k}^T W_k (D_k + \Delta_k) + O(\|W_k\|^4), \\
&= -W_k \Delta_k - (U_k + W_k){W_k}^T W_k \Delta_k + O(\|W_k\|^4), \\
&= (-W_k - U_k {W_k}^T W_k)\Delta_k + O(\|W_k\|^4), \\
&= (-W_k - U_k {W_k}^T W_k){U_k}^T A W_k + O(\|W_k\|^4), \\
&= (-W_k - U_k {W_k}^T W_k)({R_k}^T W_k + D_k {U_k}^T W_k) + O(\|W_k\|^4), \\
&= -W_k {R_k}^T W_k + O(\|W_k\|^4).
\end{aligned}$$

Since W_k is the solution of a non singular Sylvester equation of which right-hand side is R_k, clearly $\|W_k\| = O(\|R_k\|)$ and (26) is established. Near convergence $U_k + W_k$ is close to an orthonormal basis and therefore matrix S_k is close to identity. Hence (26) proves that $R_{k+1} = O(\|R_k\|^3)$. ■

2.5 Inverse iteration and Rayleigh Quotient Iteration

The above developments lead to Olsen's scheme and the Jacobi-Davidson approach, see, e.g., [19]. A simpler scheme is often used in practice which consists of ignoring the projection step. In the situation of a single vector iteration, this scheme is simply the inverse iteration algorithm, which computes a new direction by solving

$$(A - \mu I)u_{new} = u \tag{29}$$

in which μ is typically a £xed shift close to an eigenvalue. Note that the right-hand side is an approximate eigenvector instead of a residual. A block generalization can be written from the scheme (12) by using a different right-hand side, namely, we solve

$$AU_{new} - U_{new}D = U \ .$$

If $R \equiv AU - UD$, and $U_{new} = U + W$ then the above condition, can be rewritten as $AW - WD = U - R$. Note that now D is no longer de£ned as $D = U^T AU$ but can be a diagonal matrix of shifts. Also a normalization must be added to the basic step mentioned above, in the form of a QR factorization of U_{new}.

In Rayleigh Quotient Iteration, the shift μ in (29) is changed to the Rayleigh quotient at each step. This, however is not practical as it requires an expensive re-factorization of the matrix at each step. Of course the same is true of Algorithm 2.1, where a change in the matrix D_k would require expensive refactoring in the method used to solve the Sylvester equation.

3 Domain decomposition: CMS and AMLS

This section explores applications of what was just learned on block corrections to Domain Decomposition (DD) methods. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric real matrix, partitioned as

$$A = \begin{pmatrix} B & E \\ E^T & C \end{pmatrix}, \tag{30}$$

where $B \in \mathbb{R}^{(n-p)\times(n-p)}$, $C \in \mathbb{R}^{p\times p}$ and $E \in \mathbb{R}^{(n-p)\times p}$. Assume that the above matrix arises from the discretization of a certain self-adjoint operator (e.g., a Laplacean) on a domain Ω which is then partitioned into several subdomains with an interface Γ, see Figure 1 for the simplest case of two subdomains. The subdomains, which may overlap, are separated by an interface Γ. The unknowns in the interior of each subdomain Ω_i are completely decoupled from the unknowns of all other subdomains. Coupling among subdomains is through the unknowns of the interface Γ and the unknowns in each Ω_i that are adjacent to Γ. With the situation just described, the matrix B is block diagonal, consisting of two diagonal blocks corresponding to the unknowns that are interior to each subdomains. The C block corresponds to the variables on the interface.

The eigenvalue problem $Au = \lambda u$, can be written as,

$$\begin{pmatrix} B & E \\ E^T & C \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \lambda \begin{pmatrix} u \\ y \end{pmatrix}, \tag{31}$$

where $u \in \mathbb{C}^{n-p}$ and $y \in \mathbb{C}^p$. A method for computing eigenvalues of matrices partitioned in this manner was introduced in structural dynamics by [5, 8]. Referred to as the method of Component Mode Synthesis (CMS), this method begins by solving the problem $Bv = \mu v$. This amounts to solving each of the decoupled smaller eigenvalue problems corresponding to each subdomain Ω_i separately. The method then injects additional vectors to account for the coupling among subdomains. This is done by invoking a carefully selected operator for the interface nodes. AMLS is a multilevel approach which exploits recursivity by continuing this approach into lower levels recursively, see e.g., [2, 1] for details.

In the following, the main steps of CMS - i.e., one level of AMLS, will be reviewed. Consider the matrix

$$U = \begin{pmatrix} I & -B^{-1}E \\ 0 & I \end{pmatrix}. \tag{32}$$

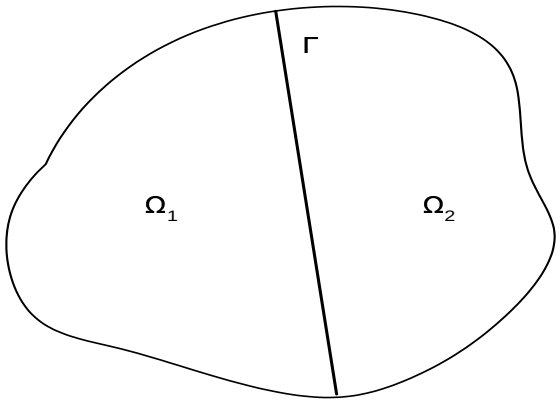


Figure 1: The simple case of two subdomains Ω_1, Ω_2 and an interface Γ.

This is a block Gaussian eliminator for matrix (30), which is selected so that

$$U^T A U = \begin{pmatrix} B & 0 \\ 0 & S \end{pmatrix},$$

where S is the Schur complement

$$S = C - E^T B^{-1} E. \tag{33}$$

The original problem (31) is equivalent to the generalized eigenvalue problem $U^T A U u = \lambda U^T U u$, which becomes

$$\begin{pmatrix} B & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \lambda \begin{pmatrix} I & -B^{-1}E \\ -E^T B^{-1} & M_S \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix}, \tag{34}$$

where $M_S = I + E^T B^{-2} E$. The next step of CMS is to neglect the coupling matrices (blocks in positions (1,2) and (2,1)) in the right-hand side matrix of (34). This yields the uncoupled problem

$$Bv = \mu\, v \tag{35}$$

$$Ss = \eta\, M_S s. \tag{36}$$

Once the desirable eigenpairs have been obtained from (35–36), they are utilized in a projection method (Rayleigh-Ritz) applied to the original problem (34). The basis used for this projection is of the form

$$\left\{ \hat{v}_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix} \quad i = 1, \ldots, m_B; \qquad \hat{s}_j = \begin{pmatrix} 0 \\ s_j \end{pmatrix} \quad j = 1, \ldots, m_S \right\}, \tag{37}$$

where $m_B < (n-p)$ and $m_S < p$ and where the v_i's ans s_j's are eigenvectors of the problems (35) and (36) respectively.

It is important to note that the projection is applied to (34) rather than to the original problem (31). There is an inherent change of basis between the two and, for reasons that will become clear shortly, the basis $\{\hat{v}_i\}_i, \{\hat{s}_j\}_j$, is well suited for the transformed problem rather than the original one. In fact let us consider this point in detail. We could also think of using the transformed basis

$$\left\{ \hat{v}_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix} \quad i = 1, \ldots, m_B; \qquad \hat{u}_j = \begin{pmatrix} -B^{-1} E s_j \\ s_j \end{pmatrix} \quad j = 1, \ldots, m_S \right\}, \tag{38}$$

for solving the original problem (31), instead of the basis (37). As can be easily seen, these two options are mathematically equivalent.

Lemma 3.1 *The Rayleigh-Ritz process using the basis (37) for problem (34) is mathematically equivalent to the Rayleigh-Ritz process using the basis (38) for problem (31).*

Proof. For a given matrix A, and a given basis (not necessarily orthogonal) consisting of the columns of a certain matrix Z, the Rayleigh Ritz process can be written as

$$Z^T A Z \underline{v} = \underline{\lambda} Z^T Z \underline{v}$$

If Z is the basis (37) then the basis (38) is nothing but UZ. Comparing the two projection processes gives the result. ■

In the rest of the paper we will use the basis (38) on the original problem (31) for describing the CMS projection.

It should be mentioned that the original CMS method as described in [5, 8] does not use selected eigenvectors s_j from the Schur complement as was done here, but instead uses a technique which amounts to taking all of them (i.e., $m_S = p$). Since this can be a problem when the interfaces are large Bourquin [4] in particular suggested a technique based on Schur complements which is similar to the more general approach described above.

3.1 Links with the correction equation

One of the purposes of this paper is to present CMS/AMLS from the angle of the correction equation. Notice at £rst that CMS does implement a correction: it corrects the eigenvectors of B to try to obtain better approximations to the eigenvectors of the whole matrix A. This is done by using the Schur complement matrix and constructing a good (improved) basis to perform the Rayleigh Ritz projection. This viewpoint (correction) is important because there are many variants for correcting eigenvectors and some are better than others. It should be noted that, in contrast with the above description, the common CMS viewpoint begins from the eigenmodes of the Schur complement that results from the so-called Guyan reduction, or static condensation, see [9]. These modes are then enriched by adding selected modes from the interior variables of each subdomain. Since we are in effect selecting a set of coupling modes and a set of internal modes, to form a basis of for the Rayleigh Ritz projection, this distinction is no longer important.

Consider eigenvectors of B associated with smallest eigenvalues.

$$Bv_i = \mu_i v_i \ .$$

One can consider the vectors

$$\hat{v}_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix}$$

as approximate eigenvectors of A. The eigenvectors obtained in this manner amount to neglecting all the couplings and are likely to yield very crude approximations. We can now think of correcting these eigenvectors via a *correction equation* as is usually done, see the previous sections.

An interesting observation is that the residuals $r_i = (A - \mu I)\hat{v}_i$ have components only on the interface variables, i.e., they have the shape:

$$r_i = \begin{pmatrix} 0 \\ w_i \end{pmatrix} \tag{39}$$

where the partitioning corresponds to the one above and where $w_i = E^T v_i$.

Consider a single vector inverse iteration correction. In this case, for each approximate eigenvector v_i we would seek a new approximation u_i by solving an equation of the type (29), $(A - \mu I)u_i = v_i$. where μ is a certain shift. In a typical inverse iteration correction, μ is constant to reduce cost of factorization.

The matrix $A - \mu I$ can be factored as

$$(A - \mu I) = \begin{pmatrix} I & 0 \\ E^T(B - \mu I)^{-1} & I \end{pmatrix} \begin{pmatrix} B - \mu I & E \\ 0 & S(\mu) \end{pmatrix} \tag{40}$$

where $S(\mu)$ is the Schur complement

$$S(\mu) = C - \mu I - E^T (B - \mu I)^{-1} E.$$

Taking the particular structure of v_i into account we £nd that

$$u_i = (A - \mu I)^{-1} \begin{pmatrix} v_i \\ 0 \end{pmatrix} = \begin{pmatrix} [I + (B - \mu I)^{-1} E S(\mu)^{-1} E^T](B - \mu I)^{-1} v_i \\ -S(\mu)^{-1} E^T (B - \mu I)^{-1} v_i \end{pmatrix}$$

In other words,

$$u_i = \begin{pmatrix} z_i - (B - \mu I)^{-1} E s_i \\ s_i \end{pmatrix} \quad \text{with} \quad z_i = (B - \mu I)^{-1} v_i \quad \text{and} \quad s_i = -S(\mu)^{-1} E^T z_i$$

There is a strong similarity between the result of this correction and that obtained from CMS. This can be seen from the nature of the basis (38) which consists of the vectors

$$\begin{pmatrix} v_i \\ 0 \end{pmatrix} \quad \begin{pmatrix} -B^{-1} E s_j \\ s_j \end{pmatrix} \tag{41}$$

where the s_i's are eigenvectors of a Schur complement, and the v_i's are eigenvectors of the B block.

3.2 Correction equations and Domain Decomposition

Consider now the application of the correction equation (12) to the Domain Decomposition framework discussed above. Speci£cally, we consider the Newton approach discussed in Section 2.3 and we will apply one step of the Newton-Sylvester algorithm, i.e., Algorithm (2.1). Note that since we are only considering one step of the process, there is no difference between this algorithm and Algorithm 2.2. The index k is removed for simplicity and U_{k+1} in Line 4 is denoted by U^{new}. We denote by $V_1 \in \mathbb{R}^{p \times m}$ ($1 \leq m \leq p < n$) an orthogonal basis of an invariant subspace of B, so that $BV_1 = V_1 D$, where $D = diag(\mu_1, \cdots, \mu_m)$ and we let $U = \begin{pmatrix} V_1 \\ 0 \end{pmatrix} \in \mathbb{R}^{n \times m}$. To simplify notation we will denote by P_1 the orthogonal projector $P_1 \equiv I - V_1 V_1^T$, so that

$$I - UU^T = \begin{pmatrix} I - V_1 V_1^T & 0 \\ 0 & I \end{pmatrix} \equiv \begin{pmatrix} P_1 & 0 \\ 0 & I \end{pmatrix} \tag{42}$$

In addition, the matrix D_k in Line 2 of Algorithm 2.1 is simply the diagonal matrix D:

$$U^T A U = [V_1^T \quad 0] A \begin{pmatrix} V_1 \\ 0 \end{pmatrix} = V_1^T B V_1 = D \, .$$

Similarly, the residual matrix R_k has a simple structure due to (39):

$$R = AU - UD = \begin{pmatrix} 0 \\ E^T V_1 \end{pmatrix}$$

Now the Sylvester system in Line 3 of Algorithm 2.1 can be written $(I - UU^T)AZ - ZD = -R$. Since D is diagonal this system decouples into the p distinct linear systems,

$$(I - UU^T) A z_i - \mu_i z_i = -r_i = - \begin{pmatrix} 0 \\ E^T v_i \end{pmatrix} \, . \tag{43}$$

Writing $z_i = \begin{pmatrix} y_i \\ s_i \end{pmatrix}$ the system (43) translates to:

$$\begin{aligned} P_1 B y_i + P_1 E s_i - \mu_i y_i &= 0 \\ E^T \, y_i + (C - \mu I) s_i &= -E^T v_i \end{aligned}$$

Note that the £rst equation is equivalent to $P_1(B-\mu_i I)y_i + P_1 E s_i = 0$ the solution of which is

$$y_i = -[P_1(B-\mu_i I)]^\dagger P_1 E s_i.$$

Substituting this solution in the second equation results in:

$$-E^T P_1 (B-\mu_i I)^\dagger P_1 \; E s_i + (C-\mu_i I)s_i = -E^T v_i$$

which gives the following equation to solve for s_i:

$$[C - \mu_i I - E^T P_1 (B-\mu_i I)^\dagger P_1 E] s_i = -E^T v_i \tag{44}$$

This leads to a natural de£nition of a *projected Schur complement*,

$$S_*(\mu_i) = C - \mu_i I - E^T P_1 (B-\mu_i I)^\dagger P_1 E \tag{45}$$

from which the solution is readily expressible. In what follows it is assumed that $S_*(\mu_i)$ is not singular (A less restrictive assumption is that $S_*(\mu_i)$ is not singular in the range of E^T.)

In the end the column-vectors of the new matrix U^{new} are given by

$$u_i^{new} = u_i + z_i = \begin{pmatrix} v_i - P_1(B-\mu_i I)^\dagger P_1 E s_i \\ s_i \end{pmatrix} \quad \text{with} \quad s_i = -S_*(\mu_i)^{-1} E^T v_i \tag{46}$$

An interesting property which can be shown is that

$$(A-\mu_i I) u_i^{new} \;=\; \begin{pmatrix} -V_1 V_1^T E s_i \\ 0 \end{pmatrix} .$$

It is, of course possible to apply additional steps of this correction process. However, these additional steps will require expensive Sylvester-like equations to be solved at each step with different shifts. Instead of considering these we can instead gain insight from the AMLS procedure and try to de£ne good subspaces for a projection process. Speci£cally, the question is: *which subspace should be added to* $span V_1$ *if the goal is to obtain a good approximation to the original eigenspace?*

A comparison between (46) and the basis (41) used by CMS suggests that we replace μ_i by zero in (46) and that we enrich the basis V_1 by the vectors

$$\begin{pmatrix} -P_1 B^{-1} P_1 E s_i \\ s_i \end{pmatrix} \quad \text{with} \quad s_i = -S_*(0)^{-1} E^T v_i \; .$$

Note that $P_1 B^{-1} P_1 = P_1 B^{-1} = B^{-1} P_1$. In other words we can consider performing a Rayleigh-Ritz process on the original problem (31) with the basis

$$\left\{ \hat{v}_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix} \; ; \; \hat{u}_i = \begin{pmatrix} -P_1 B^{-1} E s_i \\ s_i \end{pmatrix} \quad i = 1, \ldots, m \right\}, \quad \text{with} \quad s_i = -S_*(0)^{-1} E^T v_i \; . \tag{47}$$

The differences with the basis used by CMS are (1) the way in which the s_i's are de£ned and (2) the presence of the projector P_1 in the de£nition of $\hat{u}_i$. We can also de£ne a simpli£ed basis, which we will refer to as the Newton-CMS basis, in which the projectors are removed:

$$\left\{ \hat{v}_i = \begin{pmatrix} v_i \\ 0 \end{pmatrix} \; ; \; \hat{u}_i = \begin{pmatrix} -B^{-1} E s_i \\ s_i \end{pmatrix} \quad i = 1, \ldots, m \right\}, \quad \text{with} \quad s_i = -S^{-1} E^T v_i \; . \tag{48}$$

Note that $S_*(0)$ has been replaced by the standard Schur complement $S = C - E^T B^{-1} E$. Experiments indicated that at least for a one level AMLS (i.e., for a CMS algorithm), there is no difference between the two bases (47) and (48).

One of the main weaknesses of AMLS is that it is a one-shot algorithm, in the sense that it just provides one set of approximations that cannot (at least with the classical algorithm de£nition) be improved. Because of the close relationship between the AMLS algorithm and the correction equation, we can think of using more steps of the correction algorithms described earlier to £ll the gap. The biggest appeal of AMLS is that it is essentially a method which performs one factorization (direct) only to extract a large number of eigenvectors. In the numerical experiments we will test the following adaptation of Algorithm 2.2 in which U_0 corresponds to the initial set of CMS which is the £rst part of the basis (38). For this reason it is important to set D_k to zero throughout. In addition, following the spirit of CMS, the correction in Line 4 of Algorithm 2.2 is replaced by a projection step using the sets given by U_k and Z_k. This gives following iterative, or corrected, variant of CMS.

ALGORITHM **3.1** *Iterative CMS*

0. *Select U_0 s.t. $U_0^T U_0 = I$ from eigenvectors in each subdomain.*
1. *For $k = 0, \cdots,$ until convergence Do:*
2. *Compute $R_k = AU_k - U_k(U_k^T AU_k)$*
3. *Solve (for Z_k): $(I - U_k U_k^T)AZ_k = -R_k$*
4. *Set $V = [U_k, Z_k]$*
5. *Compute U_{k+1} from a Rayleigh-Ritz projection on A using the basis V.*
6. *EndDo*

The system $(I - U_k U_k^T)AZ_k = -R_k$ can be solved, for example, by setting $T = U_k^T AZ_k$ and then solving

$$\begin{pmatrix} A & -U_k \\ U_k^T A & -I \end{pmatrix} \begin{pmatrix} Z_k \\ T \end{pmatrix} = \begin{pmatrix} -R_k \\ 0 \end{pmatrix} .$$

Since U_k is typically of low rank, the above system can be easily solved with one factorization of A.

This algorithm will be tested for a 2-domain case in the next section.

4 Numerical Examples

All the tests are run in the MATLAB environment.

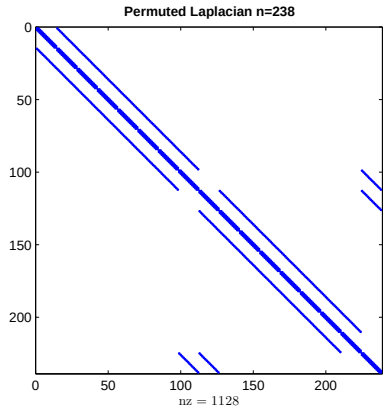
4.1 Quadratic and Cubic convergence

The £rst test considers a matrix of small order ($n = 238$). This matrix is the classical matrix obtained by the discretization of the 2D Laplacean by £nite differences on a 14×17 rectangular mesh. The twenty smallest eigenvalues are sought. The two principal diagonal blocks of order 112, the leading one and the tailing one, are separated by a block of dimension 14. When adequately permuted, the matrix pro£le is shown in Figure 2 (a). The evolution of the residual norms are displayed in Figure 2 (b) for the Newton-Sylvester iteration (Algorithm 2.1), and in Figure 2(c) for the Iterative Corr ection (Algorithm 2.2). The quadratic and cubic convergences are well illustrated by the curves. During the £rst iterations, the two methods stagnate as long as they have not yet determined a good approximation of an invariant subspace. The computed eigenvalues are not the 20 smallest ones : there are some missing eigenvalues. For Algorithm 2.1, the 12 smallest eigenvalues are computed and the last 8 computed eigenvalues correspond to eigenvalues ranking between the 14th and the 25th eigenvalue of the matrix. For Algorithm 2.2, the result is almost equivalent although a bit worse: the £rst 10 eigenvalues are computed and the last 10 computed eigenvalues range between the 12th and the 26th eigenvalue.

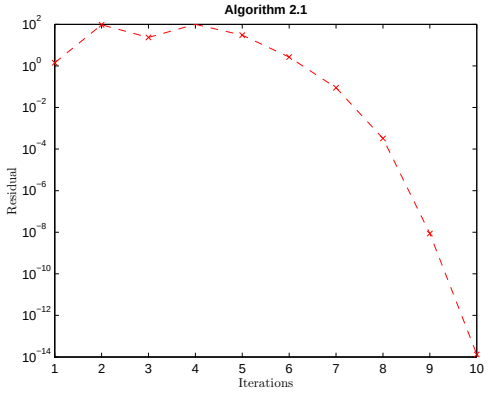
4.2 Computing inner eigenvalues

In this section, we consider as test matrix $A0$, the matrix PLAT1919 from the test suite Matrix Market [3]. The matrix is of order 1919 and its sparsity pattern is shown in Figure 3 (a). By applying a symmetric

(a) Pattern of the matrix



(b) Convergence with Algorithm 2.1



(c) Convergence with Algorithm 2.2

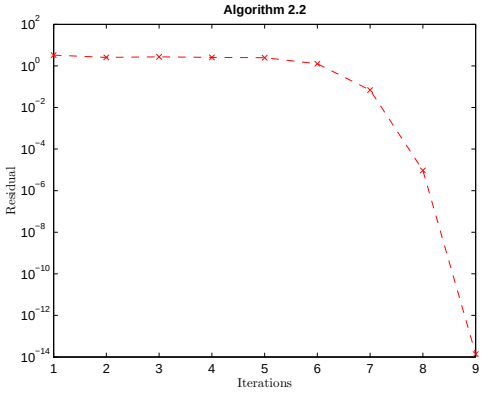
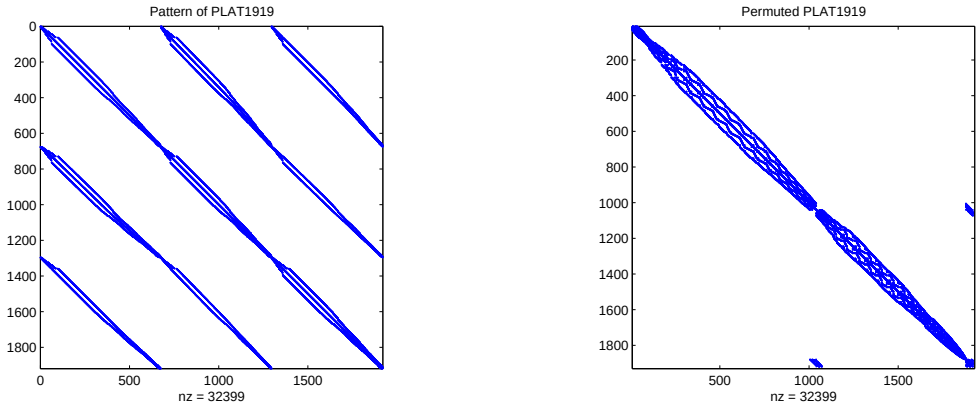


Figure 2: Test with the Laplacean

reordering P obtained from the Symmetric Reverse Cuthill-McKee algorithm, the matrix $A_P = P^T AP$ has a smaller bandwidth. Considering the permuted matrix A_P, two diagonal blocks are de£ned by the intervals of indices $I_1 = [1 : 1040]$ and $I_2 = [1081 : 1919]$. The interval $J = [1041 : 1080]$ is the separator. By renumbering symmetrically rows and columns of A_P, with the numbering de£ned by $Q = [I_1, I_2, J]$, one gets the test matrix A whose pattern is displayed in Figure 3 (b).



(a) Original pattern (b) After symmetric renumbering

Figure 3: Sparsity patterns of PLAT1919

The full spectrum of PLAT1919, computed by the QR method, is displayed in Figure 4. The goal of

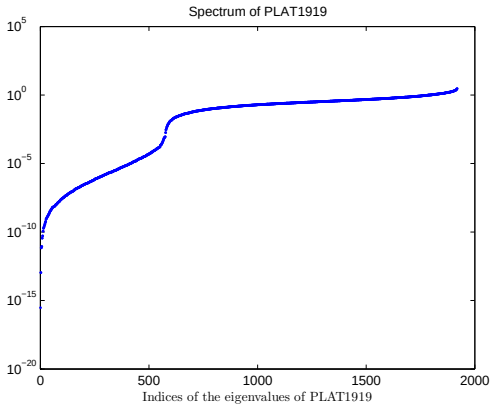


Figure 4: Spectrum of Matrix PLAT1919

the tests is to analyze the behavior of the two algorithms 2.1 and 2.2 for computing a basis of an invariant subspace corresponding to six eigenvalues in a neighborhood of $\sigma = 0.995$.

The same initial guess $U_0 \in \mathbb{R}^{1919\times 6}$ of a basis of the sought invariant subspace is considered for the two methods. It is built, consistently with Section 3.2, by the following : for each of the two blocks, the three eigenvectors corresponding to the eigenvalues which are the closest to σ are computed ; in that way, two orthonormal blocks $U^{(1)} \in \mathbb{R}^{1040\times 3}$ and $U^{(2)} \in \mathbb{R}^{849\times 3}$ are obtained and U_0 is then de£ned by

$$U_0 = \begin{pmatrix} U^{(1)} & 0 \\ 0 & U^{(2)} \\ 0 & 0 \end{pmatrix}. \tag{49}$$

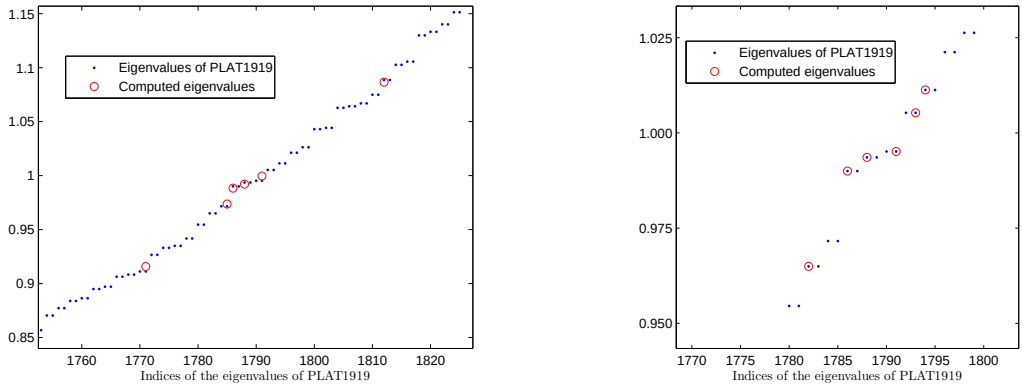
Table 1 shows, for each of the two methods, the computed eigenvalues, corresponding to the invariant

subspace de£ned by the last computed basis U_k. In each case, the eigenvalues of D_k are given, along with their respective index in the spectrum of the matrix, and their absolute error. The eigenvalues are labeled in ascending order. In Figure 5, the computed eigenvalues are located in the whole spectrum of matrix

Algorithm 2.1 After $k = 10$ iterations Residual : 5×10^{-2}			**Algorithm 2.2** After $k = 6$ iterations Residual : 3×10^{-11}		
eigenvalue	index	error	eigenvalue	index	error
0.91576	1771	4×10^{-3}	0.96497	1782	1×10^{-15}
0.97367	1785	2×10^{-3}	0.99000	1786	1×10^{-15}
0.98842	1786	2×10^{-3}	0.99359	1788	2×10^{-15}
0.99213	1788	1×10^{-3}	0.99515	1791	2×10^{-15}
0.99964	1791	4×10^{-3}	1.0053	1793	4×10^{-15}
1.0866	1812	2×10^{-3}	1.0113	1794	2×10^{-15}

Table 1: Computed eigenvalues of PLAT1919 near $\sigma = 0.995$

PLAT1919. On this example, the superiority of Algorithm 2.2 over Algorithm 2.1 is clear : the eigenvalues



(a) Algorithm 2.1

(b) Algorithm 2.2

Figure 5: Location of the computed eigenvalues in the spectrum

computed by the former are closer to the reference number σ and they are much more precise. Actually, the run of Algorithm 2.1 showed a lack of convergence. We rerun Algorithm 2.1 with U equal to the Q-factor in the QR factorization of the last estimate Uk of the £rst run. After 10 additional iterations, the residual reached 10^{-6} and the computed eigenvalues were corresponding to eigenvalues of PLAT1919 with indices from 1771 to 1815, with a precision higher than 10^{-12}. It appears that, this algorithm needs a better initial estimate than its counterpart. A drawback of Algorithm 2.1 lies in its Step 3 which corresponds to a non symmetric Sylvester equation. However, complex computation can be avoided since it can be proved that, although non symmetric, matrix D_k is similar to a symmetric matrix.

4.3 Tests with domain decomposition

We consider a Schrödinger operator of the form

$$H = -\Delta + V$$

on a rectangular domain in 2 dimensions. The potential V is a Gaussian

$$V(x, y) = -\beta e^{-(x-x_c)^2-(y-y_c)^2}$$

in which (x_c, y_c) is the center of the domain. The operator H acts on a given function u by simply adding the function Vu to the negative Laplacean of u. We selected $\beta = 100$, and discretized the domain uniformly using centered £nite differences and applied Dirichlet boundary conditions. The domain is a rectangle of dimension $(n_x + 1) \times h = 1$ and $(n_y + 1) \times h$, where n_x, n_y are the number of discrete points on the x and y directions, respectively, excluding boundary points. The domain is then split in two horizontally, in the middle of the domain. The matrix is reordered by putting the interface variables at the end as is usually done to illustrate DD orderings. The resulting matrix is shown on the left side of Figure 6. ' The £rst experiments is only for demonstrating the power of the CMS algorithm and its variants. We found in general very little difference between the different variants of the same idea. We compared the following 4 methods for computing the smallest n_{ev} eigenvalues. In the test $n_{ev} = 8$.

No correction This performs a Rayleigh Ritz procedure with eigenvectors from the two domains. The process takes $n_{ev}/2$ eigenvectors from each domain which will form the column vectors of two matrices U_1, U_2 then gathers them in a basis $W = [U_1, U_2]$ and then proceeds to a Rayleigh Ritz projection on A with the n_{ev} vectors in W.

CMS This approach consists of a CMS projection, which takes the same W from above and augments it with the set Z obtained as

$$Z = \begin{pmatrix} -B^{-1}EG \\ G \end{pmatrix}$$

where G is a matrix whose columns are eigenvectors of S associated with the smallest n_{ev} eigenvalues. This corresponds to using the basis (41) for a Rayleigh-Ritz projection. This corresponds to using the basis (41) for a Rayleigh-Ritz projection.

Newton-CMS This is similar to the previous method, but the G matrix of eigenvectors of the Schur complement is replaced by the matrix $S^{-1}E^TW$ where S is the Schur complement. Note that since $W = [U_1, U_2]$ has 2 n_{ev} columns, we end-up adding a basis of has 2 n_{ev} columns instead of n_{ev} for CMS. This scheme corresponds to using the basis (48) for a Rayleigh-Ritz projection.

Newton-CMS with projection The only difference between this and the previous process is that the projector $P_1 = I - WW^T$ is used, along with the projected Schur complement given by (45). Thus, the inverse of B is replaced by P_1B^{-1} when de£ning S and the Z matrix above. This corresponds to using the basis (47) for a Rayleigh-Ritz projection.

In all cases we tested for this example, Methods 3 and 4 seemed to be exactly identical. This means that the G matrices generate the same subspace in both cases. Because of this we only show the results with the £rst 3 methods.

Figure 6 shows an illustration for a case when $n_x = 35, n_y = 33$ which yields a matrix of size $n = 1,155$. The number of eigenvalues computed is 12. As it turns out it is very dif£cult for this example to £nd cases where CMS and Newton-CMS yield (signi£cantly) different results. Because n_x is relatively small, the subspace spanned by the matrices G involved above is the same or very close and this leads to the same approximations. What is remarkable is the quality of the approximation obtained from CMS-type approaches. The accuracy obtained by using eigenvectors from subdomains alone (no correction) is already quite good, considering the simplicity of this approach.

In the next test we consider the iterative CMS, Algorithm 3.1, discussed earlier. Only two correction steps (corresponding to the k loop in the algorithm) are taken. In this test we take $n_x = 45, n_y = 43$ which leads to a bigger matrix of size $n = 1,935$. Figure 7 shows the result of computing the 20 lowest eigenvalues with the three methods: Newton-CMS, 1st step of Iterative CMS, and 2nd step of Iterative CMS. The results are much improved, especially for the lowest 10 eigenvalues. Note also, that the biggest improvement is achieved by the £rst corrective step. What is important to emphasize here is that the improvements achieved by the two corrections are obtained without any additional factorizations of the matrix A.

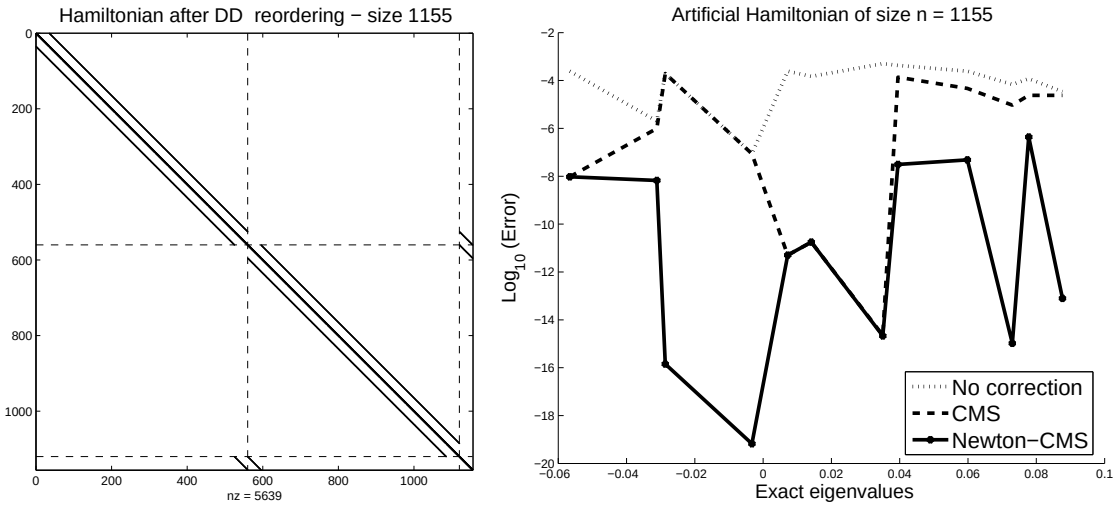


Figure 6: Left: Pattern of the Hamiltonian matrix after reordering. Right: Performance of 3 techniques for computing its 12 smallest eigenvalues

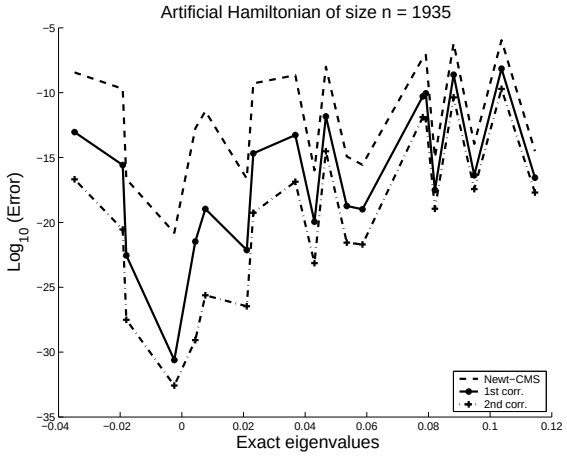


Figure 7: Performance of Newton-CMS and the results of two corrective steps of Algorithm 3.1. The 20 lowest eigenvalues are computed.

5 Summary and Conclusion

We have discussed a few variants of certain algorithms based on the correction equation for solving eigenvalue problems and we showed how they can be adapted in a domain decomposition framework. In particular, block variants of the correction equation were derived by viewing the eigenvalue problem as a system of nonlinear equations. The resulting algorithms converge cubically or quadratically but they require the solution of a different Sylvester equation at each step. In the case of CMS, experiments show that it is possible to obtain good improvements by versions of these algorithms which do not require to refactor the matrix at each step.

References

[1] K. BEKAS AND Y. SAAD, *Computation of smallest eigenvalues using spectral Schur complements*, Tech. Rep. umsi-2004-6, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2004. Appeared in sisc, volume 27, number 2, (2005) pages 458-481.

[2] J. K. BENNIGHOF AND R. B. LEHOUCQ, *An automated multilevel substructuring method for eigenspace computation in linear elastodynamics*, SIAM. J. Sci. Comput., 25 (2004), pp. 2084–2106.

[3] R. F. BOISVERT, R. POZO, K. REMINGTON, R. BARRETT, AND J. DONGARRA, *The Matrix Market: A Web repository for test matrix data*, in The Quality of Numerical Software, Assessment and Enhancement, R. Boisvert, ed., Chapman&Hall, London, 1997, pp. 125–137.

[4] F. BOURQUIN, *Component mode synthesis and eigenvalues of second order operators: discretization and algorithm*, Mathemetical modeling and numerical analysis, 26 (1992), pp. 385–423.

[5] R. R. CRAIG, JR. AND M. C. C. BAMPTON, *Coupling of substructures for dynamic analysis*, AIAA Journal, 6 (1968), pp. 1313–1319.

[6] M. CROUZEIX, B. PHILIPPE, AND M. SADKANE, *The Davidson method*, SIAM, Journal on Scienti£c and Statistical Computing, 15:1 (1994), pp. 62–76.

[7] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 3rd ed., 1996.

[8] W. C. HURTY, *Dynamic analysis of structural systems using component modes*, AIAA Journal, 3 (1965), pp. 678–685.

[9] ——, *Reduction of stifness and mass matrices*, AIAA Journal, 3 (1965), p. 380.

[10] E. LUNDSTRM AND L. ELDÉN, *Adaptive eigenvalue computations using Newton's method on the Grassmann manifold*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 819–839.

[11] R. MORGAN AND D. SCOTT, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 817–825.

[12] Y. SAAD, *Iterative Methods for Sparse Linear Systems, 2nd edition*, SIAM, Philadelpha, PA, 2003.

[13] A. SAMEH AND Z. TONG, *The trace minimization method for the symmetric generalized eigenvalue problem*, J. of Computational and Applied Math., 123 (2000), pp. 155–175.

[14] A. SAMEH AND J. WISNIEWSKI, *A trace minimization algorithm for the generalized eigenvalue problem*, SIAM J. Numer. Anal., 19 (1982), pp. 1243–1259.

[15] G. L. G. SLEIJPEN, A. G. L. BOOTEN, D. R. FOKKEMA, AND H. A. VAN DER VORST, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.

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

[17] B. SMITH, P. BJØRSTAD, AND W. GROPP, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, New-York, NY, 1996.

[18] A. STATHOPOULOS AND Y. SAAD, *Restarting techniques for the (jacobi-)Davidson symmetric eigenvalue methods*, Electron. Trans. Numer. Anal., 7 (1998), pp. 163–181.

[19] G. W. STEWART, *Matrix Algorithms II: Eigensystems*, SIAM, Philadelphia, 2001.