Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 218 (2006) 324–332

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Basis selection in LOBPCG

U. Hetmaniuk *, R. Lehoucq

Sandia National Laboratories, Computational Mathematics and Algorithms Department, P.O. Box 5800, MS 1110, Albuquerque, NM 87185, United States[1]

Received 21 October 2005; received in revised form 19 January 2006; accepted 14 February 2006
Available online 31 March 2006

Abstract

The purpose of our paper is to discuss basis selection for Knyazev's locally optimal block preconditioned conjugate gradient (LOBPCG) method. An inappropriate choice of basis can lead to ill-conditioned Gram matrices in the Rayleigh–Ritz analysis that can delay convergence or produce inaccurate eigenpairs. We demonstrate that the choice of basis is not merely related to computing in finite precision arithmetic. We propose a representation that maintains orthogonality of the basis vectors and so has excellent numerical properties.
Published by Elsevier Inc.

PACS: 65F15; 65N25

Keywords: Symmetric generalized eigenvalue problem; Preconditioned eigensolver; Orthonormalization; LOBPCG

1. Introduction

Knyazev's locally optimal block preconditioned conjugate gradient (LOBPCG) method [7] is a recent approach for the numerical solution of the large-scale generalized symmetric positive definite eigenvalue problem

$$\mathbf{Au} = \mathbf{Mu}\lambda \quad (\mathbf{A}, \mathbf{M} \in \mathbb{R}^{n\times n}), \tag{1}$$

where **A** and **M** are symmetric positive definite matrices. In combination with a symmetric positive definite preconditioner, recent papers [2,10] showed that LOBPCG is a powerful algorithm for computing approximations to the smallest eigenvalues and eigenvectors.

LOBPCG minimizes the Rayleigh quotient of (1) by performing a Rayleigh–Ritz analysis with the subspace $\mathscr{S}$ spanned by $\mathbf{X}$ and $\mathbf{X}^{-}$, the blocks of current and previous iterates, and **H**, the block of preconditioned residuals. A Rayleigh–Ritz analysis computes the new block of iterates $\mathbf{X}^{+}$ whose span provides approximation to the smallest eigenvectors of (1).

* Corresponding author.
E-mail addresses: ulhetma@sandia.gov (U. Hetmaniuk), rblehou@sandia.gov (R. Lehoucq).

[1] Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2006.02.007

The choice of basis for $\mathscr{S}$ is important numerically. Early versions of LOBPCG used the representation $[\mathbf{X},\mathbf{H},\mathbf{X}^{-}]$. This basis often led to ill-conditioned Gram matrices in the Rayleigh–Ritz analysis that produced, according to Knyazev [7], *spurious* eigenpairs. This ill-conditioning stems from the convergence of $\mathbf{X}$ and $\mathbf{X}^{-}$ towards the same eigenvectors of $(\mathbf{A},\mathbf{M})$. The goal of our paper is to discuss the impact of basis selection. We demonstrate that the choice is not merely related to computing in finite precision arithmetic. We also propose a representation that appears to have excellent numerical properties.

Our paper is organized as follows. In Section 2, we review the LOBPCG algorithm and we justify why, through the Rayleigh–Ritz analysis, the choice of basis for $\mathscr{S}$ is important. Then Section 3 presents examples that require special care in the basis selection. Finally, Section 4 proposes a representation and we illustrate its efficiency on a numerical problem.

2. Overview of LOBPCG

First we introduce some notation:

- $\mathbf{N}$ is a symmetric positive definite preconditioner for the matrix $\mathbf{A}$.
- nev denotes the number of eigenpairs to compute.
- $(\mathbf{Y},\Theta) = \mathrm{RR}(\mathbf{S},b)$ performs a Rayleigh–Ritz analysis where the pencil $(\mathbf{S}^{\mathrm{T}}\mathbf{AS},\mathbf{S}^{\mathrm{T}}\mathbf{MS})$ has eigenvectors $\mathbf{Y}$ and eigenvalues Θ, i.e.,

$$\mathbf{S}^{\mathrm{T}}\mathbf{ASY} = \mathbf{S}^{\mathrm{T}}\mathbf{MSY}\Theta \quad \text{and} \quad \mathbf{Y}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{MSY} = \mathbf{I}_{b\times b},$$

where $\mathbf{I}_{b\times b}$ is the identity matrix of size $b \times b$. The first b pairs with smallest Ritz values are returned in $\mathbf{Y}$ and in the diagonal matrix Θ in a non-decreasing order.

2.1. Algorithmic description

Algorithm 1 provides a pseudocode for LOBPCG. For an efficient implementation of LOBPCG, we refer the reader to [10]. In particular, the matrices $\mathbf{A}$, $\mathbf{M}$, and $\mathbf{N}$ can be accessed only once per iteration by storing the blocks of vectors: $\mathbf{X}_k$, $\mathbf{AX}_k$, $\mathbf{MX}_k$, $\mathbf{H}_I$, $\mathbf{AH}_I$, $\mathbf{MH}_I$, $\mathbf{P}_I$, $\mathbf{AP}_I$, $\mathbf{MP}_I$, and $\mathbf{R}_I$. For a version where the blocksize of $\mathbf{X}_k$ is independent from nev, we refer the reader to [2].

Algorithm 1 (*LOBPCG*).

```
1. Select an initial guess $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times \mathrm{nev}}$.
2. $\mathbf{X}_0 = \tilde{\mathbf{X}}\mathbf{Y}$ where $(\mathbf{Y},\Theta_0) = \mathrm{RR}(\tilde{\mathbf{X}},\mathrm{nev})$.
3. $\mathbf{R}_I = \mathbf{AX}_0 - \mathbf{MX}_0\Theta_0$.
4. $\mathbf{P}_I = [\,]$.
5. For $k = 0,1,2,\cdots$ do
6.    Solve the preconditioned linear system $\mathbf{NH}_I = \mathbf{R}_I$.
7.    Let $\mathbf{S} = [\mathbf{X}_k,\mathbf{H}_I,\mathbf{P}_I]$ and compute $(\mathbf{Y},\Theta_{k+1}) = \mathrm{RR}(\mathbf{S},\mathrm{nev})$.
8.    $\mathbf{X}_{k+1} = [\mathbf{X}_k,\mathbf{H}_I,\mathbf{P}_I]\mathbf{Y}$.
9.    $\mathbf{R} = \mathbf{AX}_{k+1} - \mathbf{MX}_{k+1}\Theta_{k+1}$.
10.   Set $\mathbf{R}_I$ with the unconverged columns of $\mathbf{R}$.
11.   Set $\mathbf{Y}_I$ with the columns of $\mathbf{Y}$ associated with the unconverged columns of $\mathbf{R}$.
12.   $\mathbf{P}_I = [\mathbf{0},\mathbf{H}_I,\mathbf{P}_I]\mathbf{Y}_I$.
13. end For.
```

At the kth iteration, LOBPCG minimizes the Rayleigh quotient on the subspace $\mathscr{S}$ spanned by $[\mathbf{X}_k,\mathbf{H}_I,\mathbf{X}_{k-1}]$. Knyazev [7] noticed that the span of $[\mathbf{X}_k,\mathbf{H}_I,\mathbf{P}_I]$ is a subspace in the span of $[\mathbf{X}_k,\mathbf{H}_I,\mathbf{X}_{k-1}]$. The columns of the former matrix are better conditioned than the columns of the latter matrix. So, in Algorithm 1, the subspace $\mathscr{S}$ uses the representation $[\mathbf{X}_k,\mathbf{H}_I,\mathbf{P}_I]$. We remark that the number of columns in $\mathbf{H}_I$ and $\mathbf{P}_I$ decrease as the columns of $\mathbf{X}_k$ converge. Knyazev and Argentati [8] call this technique *soft-locking*.

Through the function RR, the Rayleigh–Ritz analysis computes approximations to the eigenvalues of $(\mathbf{A}, \mathbf{M})$. This analysis also guarantees the **M**-orthonormality of the block $\mathbf{X}_{k+1}$. Indeed, we have

$$\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{M}\mathbf{X}_{k+1} = \mathbf{Y}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}\mathbf{Y} = \mathbf{I}_{b\times b}$$

by definition of the function RR. So the Rayleigh–Ritz analysis is important for LOBPCG. In the following section, we will show that the choice of basis for $\mathscr{S}$ is important for the Rayleigh–Ritz analysis.

2.2. Comments on the Rayleigh–Ritz analysis

When the basis for $\mathscr{S}$ is ill-conditioned, inaccurate eigenpairs originate in the Rayleigh–Ritz step. We give two explanations for the origin of this inaccuracy.

The first one is theoretical. For any full-column rank matrix **S**, we can pair any Ritz value θ_j for the pencil $(\mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{S}, \mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S})$ with an eigenvalue $\lambda_{j'}$ of $(\mathbf{A}, \mathbf{M})$ such that

$$|\lambda_{j'} - \theta_j| \leqslant \frac{\|\mathbf{A}\mathbf{S} - \mathbf{M}\mathbf{S}\mathbf{L}^{-1}(\mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{S})\mathbf{L}^{-\mathrm{T}}\|_{\mathbf{M}^{-1}}}{\sqrt{\lambda_{\min}(\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S})}}, \tag{2}$$

where **L** is the Cholesky factor of $\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}$ and $\lambda_{\min}(\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S})$ is the smallest eigenvalue of $\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}$. This result is proved in Parlett [12, see Theorem 11.10.1]. In other words, the bound (2) shows that the accuracy of a Ritz value degrades as the columns of **S** depart from **M**-orthonormality.

The second reason is numerical. In practice, the generic function RR calls the LAPACK routine DSYGV [1]. This routine computes a Cholesky factorization,

$$\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S} = \mathbf{L}\mathbf{L}^{\mathrm{T}},$$

and the eigenpairs $(\mathbf{z}_j, \theta_j)$ of the transformed matrix $\mathbf{L}^{-1}(\mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{S})\mathbf{L}^{-\mathrm{T}}$. The algorithm DSYGV is known to be numerically unstable when $\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}$ is ill-conditioned. The computed eigenvalues θ_j can differ from the true eigenvalues $\hat{\theta}_j$ by at most

$$|\hat{\theta}_j - \theta_j| \leqslant C\varepsilon\|(\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S})^{-1}\|_2\Big(\|\mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{S}\|_2 + |\hat{\theta}_j|\|\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}\|_2\Big) \tag{3}$$

(see [1] for further details) and the Ritz vectors $\mathbf{y}_j$,

$$\mathbf{y}_j = \mathbf{L}^{-\mathrm{T}}\mathbf{z}_j,$$

may not be orthogonal to machine precision. Indeed, even if the eigenvectors $\mathbf{z}_j$ are orthonormal to machine precision,

$$\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_j = \mathcal{O}(\varepsilon) \quad (i \neq j),$$

the vectors $\mathbf{y}_j$ satisfy

$$\mathbf{y}_i^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}\mathbf{y}_j = \mathcal{O}\big(\varepsilon\kappa(\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S})\big) \quad (i \neq j), \tag{4}$$

where $\kappa(\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S})$ is the condition number of the Gram matrix $\mathbf{S}^{\mathrm{T}}\mathbf{M}\mathbf{S}$.

To illustrate the sharpness of (4), we consider the pencil $(\mathbf{I}_{n\times n}, \mathbf{Hilb}(n))$, where $\mathbf{Hilb}(n)$ is the Hilbert matrix of size n. We compute the associated eigenpairs $(\mathbf{y}_j, \theta_j)$ with the routine eig of Matlab [11], using the Cholesky factorization of the Hilbert matrix. In Fig. 1, we plot the maximum error in orthogonality,

$$\max_{i\neq j} |\mathbf{y}_i^{\mathrm{T}}\mathbf{Hilb}(n)\mathbf{y}_j|,$$

and this maximal error scaled by the condition number of the Hilbert matrix,

$$\frac{\max_{i\neq j}|\mathbf{y}_i^{\mathrm{T}}\mathbf{Hilb}(n)\mathbf{y}_j|}{\kappa(\mathbf{Hilb}(n))},$$

when the size n increases. We note that the maximal error scaled by the condition number remains bounded when the size n increases.

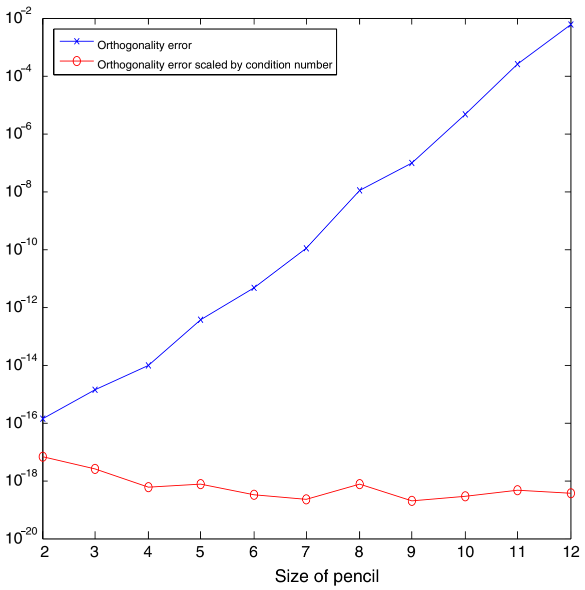

Fig. 1. Orthogonality error of eigenvectors for the pencil $(\mathbf{I}_{n\times n}, \mathbf{Hilb}(n))$.

So ill-conditioning in the matrix $\mathbf{S}$ can produce, through the routine DSYGV, inaccurate eigenpairs. One solution is to selectively discard columns of $\mathbf{S}$ to improve the conditioning of the basis. Unfortunately, when doing so, the convergence of LOBPCG is delayed because the Rayleigh quotient is not minimized on the largest available subspace. Furthermore, the criterion for selectively discarding is empirical—there exists no theoretical justification. In [10], the authors suggest discarding the search directions $\mathbf{P}_I$. However, the block of vectors $[\mathbf{X}_k, \mathbf{H}_I]$ may remain ill-conditioned.

Another solution is to replace the LAPACK routine DSYGV with an efficient and backward stable algorithm for the Rayleigh–Ritz analysis. Unfortunately to the best of our knowledge, such an algorithm, which would also exploit the symmetry of symmetric generalized eigenproblems, has yet to be implemented. Chandrasekaran [3] and Davies et al. [4] have proposed potential solutions but without proving the optimality of their algorithm. Consequently, for the remainder of this paper, we assume that the function RR calls the LAPACK routine DSYGV.

In the following section, we present simple examples where the representation $[\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]$ is singular or ill-conditioned.

3. Examples of problematic $[\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]$

3.1. Initial guess from a Krylov space

We assume that the preconditioner is the identity matrix. Let $\mathbf{U}$ be a full-column rank block of b vectors such that $\mathbf{U}$ and $\mathbf{A}^{-1}\mathbf{M}\mathbf{U}$ are linearly independent. Using the notation of Algorithm 1, we define the following blocks of vectors:

$$\tilde{\mathbf{X}} = [\mathbf{U}, \mathbf{A}^{-1}\mathbf{M}\mathbf{U}],$$

$$\mathbf{X}_0 = \tilde{\mathbf{X}}\mathbf{Y}, \quad \text{where } (\mathbf{Y}, \Theta_0) = \mathrm{RR}(\tilde{\mathbf{X}}, 2b),$$

$$\mathbf{R}_I = \mathbf{A}\mathbf{X}_0 - \mathbf{M}\mathbf{X}_0\Theta_0.$$

The rank of $\mathbf{X}_0$ is in general $2b$. On the other hand, we can prove that the rank of $\mathbf{R}_I$ is at most b. We rewrite the residual as follows:

$$\mathbf{R}_I = \left(\mathbf{A}\tilde{\mathbf{X}} - \mathbf{M}\tilde{\mathbf{X}}\mathbf{Y}\Theta_0\mathbf{Y}^{-1}\right)\mathbf{Y} = \left(\mathbf{A}\tilde{\mathbf{X}} - \mathbf{M}\tilde{\mathbf{X}}(\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{M}\tilde{\mathbf{X}})^{-1}(\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{A}\tilde{\mathbf{X}})\right)\mathbf{Y}.$$

With the definition of $\tilde{\mathbf{X}}$, we have

$$(\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{M}\tilde{\mathbf{X}})^{-1}(\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{A}\tilde{\mathbf{X}}) = \left[\begin{array}{c|c} \star & \mathbf{I}_{b\times b} \\ \hline \star & \mathbf{0}_{b\times b} \end{array}\right],$$

where $\mathbf{0}_{b\times b}$ is the zero matrix of size $b \times b$. And so we obtain

$$\mathbf{R}_I = [\widetilde{\mathbf{R}}, \mathbf{0}_{n\times b}]\mathbf{Y},$$

which proves that the rank of $\mathbf{R}_I$ is at most b. Therefore, the representation $[\mathbf{X}_0, \mathbf{R}_I]$ is not of full-column rank and the routine DSYGV will fail because of this rank deficiency.

We note that a similar rank deficiency appears for the initial guess $[\mathbf{U}, \mathbf{M}^{-1}\mathbf{A}\mathbf{U}]$. On the other hand, when the initial guess is set to $[\mathbf{U}, \mathbf{A}^{-1}\mathbf{M}\mathbf{U} + \varepsilon\mathbf{V}]$, where ε is a small parameter, the representation $[\mathbf{X}_0, \mathbf{R}_I]$ will be ill-conditioned.

3.2. Rank deficient $\boldsymbol{H}_I$

We assume that the matrices $\mathbf{A}$ and $\mathbf{M}$ arise from the finite element discretization of the Laplace equation with homogeneous Dirichlet boundary condition on the unit square. We use piecewise bilinear finite elements on an uniform orthogonal grid. In each coordinate direction, we define m interior grid points. The resulting matrices are of size $n = m^2$ and, with a lexicographical ordering of the grid points, have a bandwidth of m.

Let $\mathbf{I}_{n\times b}$ be the first b columns of the identity matrix. Using the notation of Algorithm 1, we define the initial guess

$$\tilde{\mathbf{X}} = \mathbf{I}_{n\times b}.$$

The rank of $\mathbf{X}_0$ is b, while the rank of $\mathbf{H}_I$ is at most $\min(b, m)$ because of the lexicographical ordering of the grid points. Therefore, the representation $[\mathbf{X}_0, \mathbf{H}_I]$ is not of full-column rank. For the Rayleigh–Ritz analysis, the Cholesky factorization in the routine DSYGV will fail.

We note that the following initial guess:

$$\mathbf{I}_{n\times b} + \begin{bmatrix} \mathbf{0}_{b\times b} \\ \varepsilon\mathbf{V} \end{bmatrix},$$

where ε is a small parameter, results in an ill-conditioned $[\mathbf{X}_0, \mathbf{H}_I]$.

3.3. Rank deficient $[\boldsymbol{X}_{k+1}, \boldsymbol{P}_I]$

From Algorithm 1, we remark that the new iterates $\mathbf{X}_{k+1}$ and the updated search directions $\mathbf{P}_I$ are related. If we assume no convergence ($\mathbf{Y} = \mathbf{Y}_I$) and we partition $\mathbf{Y}$ as follows:

$$\mathbf{Y} = [\mathbf{Y}_{\mathbf{X}}, \mathbf{Y}_{\mathbf{H}}, \mathbf{Y}_{\mathbf{P}}]^{\mathrm{T}},$$

then the updated search directions $\mathbf{P}_I$ satisfy

$$\mathbf{P}_I = \mathbf{X}_{k+1} - \mathbf{X}_k\mathbf{Y}_{\mathbf{X}}.$$

Consequently, the block $[\mathbf{X}_{k+1}, \mathbf{P}_I]$ is ill-conditioned when the update matrix $\mathbf{Y}_{\mathbf{X}}$ has columns with small norm.

We can generate such an example with a random initial guess $\tilde{\mathbf{X}}$ (nev = 1) and the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 10^8 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix} \tag{5}$$

and the matrix $\mathbf{M}$ is equal to the identity matrix. For this particular case, the smallest singular value of $[\mathbf{X}_1, \mathbf{P}_I]$ will be of order 10^{-8} and the conditioning of

$$[\mathbf{X}_1, \mathbf{H}_I, \mathbf{P}_I]^{\mathrm{T}}\mathbf{M}[\mathbf{X}_1, \mathbf{H}_I, \mathbf{P}_I]$$

is of order 10^{16}.

3.4. An eigensolver as preconditioner

Let us drop the requirement that $\mathbf{N}$ is symmetric positive definite for this section only. A general approach to solve a linear system iteratively is to use the conjugate gradient algorithm preconditioned by a few iterations of a Jacobi or Gauss–Seidel method. Therefore, we propose in this section to use as a preconditioner a simple eigensolver.

This simple eigensolver will consist of one step of a Rayleigh–Ritz analysis on the subspace spanned by $[\mathbf{X}_k, \mathbf{R}_I]$, i.e., we set the block $\mathbf{H}_I$ to

$$\mathbf{H}_I = [\mathbf{X}_k, \mathbf{R}_I]\Psi, \quad \text{where } (\Psi, \Phi) = \mathrm{RR}([\mathbf{X}_k, \mathbf{R}_I], \mathrm{nev}).$$

The resulting representation $[\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]$ will quickly become ill-conditioned because the blocks $\mathbf{X}_k$ and $\mathbf{H}_I$ will converge to the same eigenvectors of the pencil $(\mathbf{A}, \mathbf{M})$.

All these examples demonstrate that the potential linear dependencies are not merely related to computing in finite precision arithmetic. Knyazev [7] notes also that when $\mathbf{M}$ is ill-conditioned and a high accuracy is required (i.e., small residual norm), the Rayleigh–Ritz analysis may generate an ill-conditioned Gram matrix $\mathbf{S}^{\mathrm{T}}\mathbf{MS}$. Consequently, the representation $[\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]$ is not robust. In the following section, we propose a solution that defines an $\mathbf{M}$-orthonormal basis.

4. A robust representation

4.1. Algorithmic modifications

First, we introduce the orthonormalization function ORTHO such that for any symmetric positive definite matrix $\mathscr{M}$, any input matrix $\mathbf{W}$ of size $n \times b$, and any input matrix $\mathbf{Q}$ of size $n \times q$, the output matrix $\mathbf{V}$

$$\mathbf{V} = \mathrm{ORTHO}(\mathscr{M}, \mathbf{W}, \mathbf{Q})$$

is of size $n \times b$ and satisfies:

$$\mathbf{V}^{\mathrm{T}}\mathscr{M}\mathbf{V} = \mathbf{I}_{b\times b},$$
$$\mathbf{V}^{\mathrm{T}}\mathscr{M}\mathbf{Q} = \mathbf{0}_{b\times q},$$
$$\mathrm{Range}(\mathbf{V}) \supset \mathrm{Range}(\mathbf{W}).$$

We emphasize that the matrix $\mathbf{V}$ is of full-column rank and that its range contains the range of the input matrix $\mathbf{W}$. For an efficient block implementation of ORTHO, we refer the reader to Stathopoulos and Wu [13], where they use exclusively level 3 BLAS [5] and LAPACK [1] routines.

The modifications to build an $\mathbf{M}$-orthonormal basis in LOBPCG are described in Algorithm 2. They consist in adding two calls to the function ORTHO.

Algorithm 2 (*LOBPCG with an* ***M****-orthonormal basis*).

```
1. Select an initial guess $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times \mathrm{nev}}$.
2. $\mathbf{X}_0 = \tilde{\mathbf{X}}\mathbf{Y}$ where $(\mathbf{Y}, \Theta_0) = \mathrm{RR}(\tilde{\mathbf{X}}, \mathrm{nev})$.
3. $\mathbf{R}_I = \mathbf{A}\mathbf{X}_0 - \mathbf{M}\mathbf{X}_0\Theta_0$.
4. $\mathbf{P}_I = []$.
5. For $k = 0,1,2,\cdots$ do
6.    Solve the preconditioned linear system $\mathbf{N}\widetilde{\mathbf{H}} = \mathbf{R}_I$.
7.    $\mathbf{H}_I = \mathrm{ORTHO}(\mathbf{M}, \widetilde{\mathbf{H}}, [\mathbf{X}_k, \mathbf{P}_I])$.
8.    Let $\mathbf{S} = [\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]$ and compute $(\mathbf{Y}, \Theta_{k+1}) = \mathrm{RR}(\mathbf{S}, \mathrm{nev})$.
9.    $\mathbf{X}_{k+1} = [\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]\mathbf{Y}$.
10.   $\mathbf{R} = \mathbf{A}\mathbf{X}_{k+1} - \mathbf{M}\mathbf{X}_{k+1}\Theta_{k+1}$.
```

```
11.    Set $\mathbf{R}_I$ with the unconverged columns of $\mathbf{R}$.
12.    Set $\tilde{\mathbf{Y}} = [\tilde{\mathbf{Y}}_{\mathbf{X}}, \tilde{\mathbf{Y}}_{\mathbf{H}}, \tilde{\mathbf{Y}}_{\mathbf{P}}]^{\mathrm{T}}$ with the columns of $\mathbf{Y}$ associated with the unconverged columns of $\mathbf{R}$.
13.    $\mathbf{Y}_I = \mathrm{ORTHO}(\mathbf{S}^{\mathrm{T}}\mathbf{MS}, [\mathbf{0}, \tilde{\mathbf{Y}}_{\mathbf{H}}, \tilde{\mathbf{Y}}_{\mathbf{P}}]^{\mathrm{T}}, \mathbf{Y})$.
14.    $\mathbf{P}_I = [\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]\mathbf{Y}_I$.
15. end For.
```

The first call to ORTHO explicitly orthonormalizes the block $\mathbf{H}_I$ against the current iterate $\mathbf{X}_k$ and the search directions $\mathbf{P}_I$. This function requires a minimum of one application of the matrix $\mathbf{M}$ and floating-point operations are of order $\mathcal{O}(n \cdot \mathrm{nev}^2) + \mathcal{O}(\mathrm{nev}^3)$. In comparison to Algorithm 1, the $\mathbf{M}$-orthonormalization is an additional cost. However, Algorithm 2 can replace the matrix $\mathbf{S}^{\mathrm{T}}\mathbf{MS}$ with the identity matrix and so avoid computing $\mathbf{S}^{\mathrm{T}}\mathbf{MS}$ (which is also $\mathcal{O}(n \cdot \mathrm{nev}^2)$).

The second call to ORTHO implicitly orthonormalizes the block $\mathbf{P}_I$ against the current iterate $\mathbf{X}_{k+1}$. Indeed, we have:

$$\mathbf{X}_{k+1}^{\mathrm{T}}\mathbf{MP}_I = \mathbf{Y}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{MSY}_I = \mathbf{0}_{\mathrm{nev}\times\mathrm{nev}},$$
$$\mathbf{P}_I^{\mathrm{T}}\mathbf{MP}_I = \mathbf{Y}_I^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{MSY}_I = \mathbf{I}_{\mathrm{nev}\times\mathrm{nev}}$$

by definition of the function ORTHO. This second call to ORTHO does not involve the matrix $\mathbf{M}$ nor the blocks of vectors $\mathbf{X}_{k+1}$ and $\mathbf{P}_I$. The number of floating-point operations performed is $\mathcal{O}(\mathrm{nev}^3)$. When $n \gg \mathrm{nev}$, this additional cost is negligible in comparison, for instance, to the update step,

$$\mathbf{P}_I = [\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]\mathbf{Y}_I,$$

which requires $\mathcal{O}(n \cdot \mathrm{nev}^2)$ operations.

These modifications do not modify the subspace $\mathcal{S}$ where the Rayleigh quotient is minimized. Consequently, when the matrix $\mathbf{S}$ is well-conditioned, convergence properties remain unchanged. Algorithm 2 does not depend on a heuristic criterion but instead upon a robust and stable procedure for orthogonalizing blocks of vectors. With an $\mathbf{M}$-orthonormal basis, the Rayleigh–Ritz analysis is now performed on a subspace of maximal size because the representation $[\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I]$ is always of full-column rank.

In [10], the authors discard the search directions $\mathbf{P}_I$ to handle ill-conditioned $\mathbf{S}^{\mathrm{T}}\mathbf{MS}$ matrices. This choice appears helpful for their particular experiments but, as pointed out in Section 2.2, no theoretical justification exists. Moreover, the block of vectors $[\mathbf{X}_k, \mathbf{H}_I]$ can still remain ill-conditioned (see Section 3). Finally, discarding search directions can delay the convergence of LOBPCG because the Rayleigh–Ritz analysis is not performed with all available information.

We could reorganize Algorithm 2 so that only one call to ORTHO is made. Such a reorganization would eliminate Steps 13 and 14 and replace Step 7 as follows:

$$[\mathbf{H}_I, \mathbf{P}_I] = \mathrm{ORTHO}(\mathbf{M}, [\widetilde{\mathbf{H}}, \tilde{\mathbf{P}}], \mathbf{X}_k),$$

where $\tilde{\mathbf{P}}$ is defined as

$$\tilde{\mathbf{P}} = [\mathbf{X}_k, \mathbf{H}_I, \mathbf{P}_I][\mathbf{0}, \tilde{\mathbf{Y}}_{\mathbf{H}}, \tilde{\mathbf{Y}}_{\mathbf{P}}]^{\mathrm{T}}.$$

However, when $n \gg \mathrm{nev}$, a formulation with one call to ORTHO requires more floating-point operations than Algorithm 2 because Step 13 works only with vectors of length at most 3nev. For the remainder of the paper, we consider only Algorithm 2.

The algorithm with an $\mathbf{M}$-orthonormal basis does not fail on the examples described in Section 3. Next, we illustrate the numerical performance of this modified algorithm on a practical engineering problem.

4.2. Numerical experiment

This example stems from an homogeneous linear elastic problem. The pencil $(\mathbf{A}, \mathbf{M})$ is of order $n = 48{,}000$. These matrices result from the finite element discretization of an elastic tube. The mesh has 16,080 vertices and is depicted in Fig. 2. Homogeneous Dirichlet boundary conditions are enforced on the outer left radial face. The matrix $\mathbf{A}$ has 3,218,400 non-zero entries and so does the matrix $\mathbf{M}$.

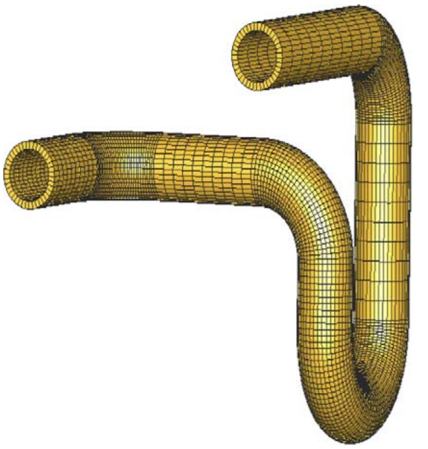

Fig. 2. Mesh for the elastic tube model.

The codes are implemented in C++, using the Trilinos [6] project. This project provides, through a collection of classes, the algebraic operations and several preconditioners.

A strength of LOBPCG is the ability to use *black-box* preconditioning of the matrix $\mathbf{A}$ (*black-box* means that the preconditioner $\mathbf{N}$ is available only as a function performing $\mathbf{N}^{-1}\mathbf{R}$—see [9] for further details). For our experiments, we choose a smoothed aggregation algebraic multigrid (AMG) preconditioner [14].

We compute the first 4 and the first 16 eigenpairs of the pencil $(\mathbf{A},\mathbf{M})$, using Algorithms 1 and 2. We use the same initial guess for both algorithms, which is generated randomly. A pair $(\mathbf{x},\theta)$ is considered converged when the criterion

$$\frac{\|\mathbf{Ax}-\mathbf{Mx}\theta\|_2}{\|\mathbf{x}\|_{\mathbf{M}}} \leqslant \theta\cdot 10^{-6}$$

is satisfied.

When computing the first 4 eigenpairs, Algorithms 1 and 2 converge with the same number of iterations. Algorithm 2 is more expensive because of the $\mathbf{M}$-orthonormalizations. For this particular example, the additional cost increased the CPU time by 20% on a 1 GHz PowerPC G4 with 1 GB of memory.

On the other hand, when computing the first 16 eigenpairs, Algorithm 1 builds several ill-conditioned $\mathbf{S}^{\mathrm{T}}\mathbf{MS}$ matrices during the iteration loop. In particular, the basis $[\mathbf{X}_1,\mathbf{P}_I]$ is extremely ill-conditioned $(\kappa([\mathbf{X}_1,\mathbf{P}_I])\approx 10^{18})$. The number of iterations is then much larger than when using Algorithm 2. For this particular example, the number of iterations with Algorithm 1 is almost three-times larger (64 iterations versus 189 iterations). Because of this slower rate of convergence, the CPU time of Algorithm 2 is much less than that of Algorithm 1.

We repeated the experiments with several random initial guesses. The conclusions remained unchanged. We varied also the number of eigenpairs requested. When computing up to the first 12 eigenpairs, the two algorithms converge with similar number of iterations. When computing more than the first 12 eigenpairs, Algorithm 1 builds several ill-conditioned $\mathbf{S}^{\mathrm{T}}\mathbf{MS}$ matrices during the iteration loop and requires more iterations.

When using *black-box* preconditioners for engineering problems, we cannot define relationships between $\mathbf{X}_k$, $\mathbf{H}_I$, and $\mathbf{P}_I$ that guarantee a well-conditioned matrix $\mathbf{S}$. Therefore, we believe that Algorithm 2 is a robust solution that guarantees a well-conditioned matrix $\mathbf{S}$.

5. Conclusions

For LOBPCG, the choice of basis for $\mathscr{S}$ is important numerically. An inappropriate choice of basis can lead to ill-conditioned Gram matrices in the Rayleigh–Ritz analysis that can delay the convergence or produce inaccurate eigenpairs. Practical examples can generate rank-deficient or ill-conditioned representations $[\mathbf{X}_k,\mathbf{H}_I,\mathbf{P}_I]$ for the subspace $\mathscr{S}$.

In order to perform an accurate Rayleigh–Ritz analysis on the subspace $\mathcal{S}$, we propose computing an **M**-orthonormal basis. This solution depends only upon a robust and stable procedure for orthogonalizing a block of vectors. Such an **M**-orthonormal basis provides an accurate Rayleigh–Ritz analysis because the matrix $\mathbf{S}^T\mathbf{M}\mathbf{S}$ is never singular nor ill-conditioned.

This orthonormalization has of course a cost, which can appear at first pointless for simple problems. However, this **M**-orthonormal representation is robust and theoretically justified. For challenging engineering problems, this solution results in a robust algorithm and may reduce the total number of iterations so leading to a more efficient algorithm.

Acknowledgments

We thank Dr. M. Gee (Sandia National Laboratories) for the elastic tube model. We acknowledge Professor A. Knyazev (CU Denver) for suggesting the Krylov subspace as initial guess and for numerous discussions.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, Society for Industrial and Applied Mathematics, 1999.

[2] P. Arbenz, U. Hetmaniuk, R. Lehoucq, R. Tuminaro, A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods, Int. J. Numer. Meth. Eng. 64 (2005) 204–236.

[3] S. Chandrasekaran, An efficient and stable algorithm for the symmetric definite generalized eigenvalue problem, SIAM J. Matrix Anal. Appl. 21 (2000) 1202–1228.

[4] P. Davies, N. Higham, F. Tisseur, Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem, SIAM J. Matrix Anal. Appl. 23 (2001) 472–493.

[5] J. Dongarra, J. Du Croz, I. Duff, S. Hammarling, A set of level 3 basic linear algebra subprograms, AM Trans. Math. Software 16 (1990) 1–17.

[6] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, K. Stanley, An overview of the Trilinos project, ACM Trans. Math. Software (2004).

[7] A. Knyazev, Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method, SIAM J. Sci. Comput. 23 (2001) 517–541.

[8] A. Knyazev, M. Argentati, Implementation of a preconditioned eigensolver using Hypre, University of Colorado at Denver, Technical Report UCD-CCM 220, 2005.

[9] A. Knyazev, K. Neymeyr, Efficient solution of symmetric eigenvalue problems using multigrid preconditioners in the locally optimal block conjugate gradient method, ETNA 15 (2003) 38–55.

[10] I. Lashuk, M. Argentati, E. Ovtchinnikov, A. Knyazev, Preconditioned eigensolver LOBPCG in Hypre and PETSc, in: Proceedings of the 16th Domain Decomposition Conference, 2005.

[11] The MathWorks Inc., Using Matlab, Natick, MA, 2002.

[12] B. Parlett, The Symmetric Eigenvalue Problem, Society for Industrial and Applied Mathematics, 1998.

[13] A. Stathopoulos, K. Wu, A block orthogonalization procedure with constant synchronization requirements, SIAM J. Sci. Comput. 23 (2002) 2165–2182.

[14] P. Vaněk, J. Mandel, M. Brezina, Algebraic multigrid based on smoothed aggregation for second and fourth order problems, Computing 56 (1996) 179–196.