A Padé-Based Factorization-Free Algorithm for Identifying the Eigenvalues Missed by a Generalized Symmetric Eigensolver

P. Avery[1], C. Farhat[2], and U. Hetmaniuk[3]

[1] *Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, 42 Avenue G. Coriolis, 31057 Toulouse Cedex 01, France. (avery@cerfacs.fr)*
[2] *Department of Mechanical Engineering and Institute for Computational and Mathematical Engineering, Stanford University, Mail Code 3035, Stanford, CA 94305, U.S.A. (cfarhat@stanford.edu)*
[3] *Department of Applied Mathematics, University of Washington, Box 352420, Seattle, WA 98195-2420, U.S.A. (hetmaniu@u.washington.edu).*

SUMMARY

When computing the solution of a generalized symmetric eigenvalue problem of the form $\mathbf{Ku} = \lambda\mathbf{Mu}$, the Sturm sequence check is the most popular method for reporting the number of missed eigenvalues within a range $[\sigma_L, \sigma_R]$. This method requires the factorization of the matrices $\mathbf{K} - \sigma_L\mathbf{M}$ and $\mathbf{K} - \sigma_R\mathbf{M}$. When the size of the problem is reasonable and the matrices $\mathbf{K}$ and $\mathbf{M}$ are assembled, these factorizations are possible. When the eigensolver is equipped with an iterative solver, which is nowadays the preferred choice for large-scale problems, the factorization of $\mathbf{K} - \sigma\mathbf{M}$ is not desired or feasible and therefore the Sturm sequence check cannot be performed. To this effect, the purpose of this paper is to present a factorization-free algorithm for detecting and identifying the eigenvalues that were missed by an eigensolver equipped with an iterative linear equation solver within an interval of interest $[\sigma_L,\ \sigma_R]$. This algorithm constructs a scalar, rational, transfer function whose poles are exactly the eigenvalues of the symmetric pencil $(\mathbf{K}, \mathbf{M})$, approximates it by a Padé expansion, and computes the poles of this approximation to detect and identify the missed eigenvalues. The proposed algorithm is illustrated with an academic numerical example. Its potential for real engineering applications is also demonstrated with a large-scale structural vibrations problem. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: eigensolver, missed eigenvalue, Padé, Sturm sequence

1. INTRODUCTION

The numerical solution of a generalized large-scale eigenvalue problem of the form

$$\mathbf{Ku} = \lambda\mathbf{Mu}, \tag{1}$$

where $\mathbf{K}$ and $\mathbf{M}$ are real symmetric positive definite matrices of size n, arises in many engineering problems. For example in structural dynamics, the circular frequencies $\lambda_k = \omega_k^2$ and

*Correspondence to: U. Hetmaniuk, Department of Applied Mathematics, University of Washington, Box 352420, Seattle, WA 98195-2420, U.S.A.

Received January 17, 2009
Revised January 17, 2009

corresponding eigenmodes $\mathbf{u}_k$ are typically determined to prevent resonant excitations, design a controller, or simply build a reduced-order computational model to predict the transient linear response of the structure. Confidence in any of these applications requires, among others, verifying that all relevant eigenpairs have been computed by the eigensolver.

A corollary of Sylvester's inertia theorem [11] implies that the number of eigenvalues lying in a real interval $[\sigma_L,\ \sigma_R]$ can be determined by factoring the matrices $\mathbf{K} - \sigma_R\mathbf{M}$ and $\mathbf{K} - \sigma_L\mathbf{M}$, counting their number of negative pivots, and computing the difference between these two numbers. For large-scale three-dimensional problems with ten million or more degrees of freedom, factoring the aforementioned matrices can be computationally prohibitive. In this case, experts often re-apply their eigensolvers in the space that is $\mathbf{M}$-orthogonal to the previously computed eigenmodes in order to identify any missed eigenpair. This idea was recently exploited in [9] to implement an iterative scheme for computing *all* p lowest eigenpairs of a symmetric positive definite pencil $(\mathbf{K}, \mathbf{M})$, for a specified p. This scheme relies on the monotonic convergence of the eigenvalues sequenced in the increasing order. As such, it can be applied to identify the eigenvalues missed in the range $[\sigma_L,\ \sigma_R]$ only if $\sigma_L < \lambda_1$, where λ_1 denotes the smallest eigenvalue of the symmetric positive definite pencil $(\mathbf{K}, \mathbf{M})$. Unfortunately, this can be restrictive.

To the best of the authors' knowledge, no post-processing technique that does not require the factorization of a matrix related to $\mathbf{K}$ and/or $\mathbf{M}$ is currently available for checking whether an eigensolver applied to the solution of problem (1) has missed some eigenvalues in an arbitrary range of interest $[\sigma_L,\ \sigma_R]$. Hence, the main objective of this paper is to fill this gap in the computational literature.

Since $\mathbf{K} - \lambda\mathbf{M}$ is singular only when λ is an eigenvalue of $(\mathbf{K}, \mathbf{M})$, it follows that the poles of a scalar rational function such as $\mathbf{l}^T(\mathbf{K} - \lambda\mathbf{M})^{-1}\mathbf{r}$, where $\mathbf{l}$ and $\mathbf{r}$ are two real vectors and the superscript T designates the transpose, are the eigenvalues of $(\mathbf{K}, \mathbf{M})$. It has been shown that such a rational function can be accurately and efficiently approximated by an appropriate Padé expansion [5, 6, 7, 4, 10, 1]. The numerical algorithm proposed in this paper for identifying the eigenvalues missed in a given range of interest builds on the aforementioned observations. It is described and illustrated in the remainder of this paper which is organized as follows.

In Section 2, the overall approach proposed in this paper for identifying the missed eigenvalues is presented and justified. A key component of this approach is a scalar rational function that is approximated by a Padé expansion whose computation is detailed in Section 3. The proposed method is laid out in Section 4. It is illustrated in Section 5 with several examples, including a large-scale problem from structural dynamics. Its performance is also assessed and its potential for real engineering applications is highlighted. Finally, conclusions are offered in Section 6.

2. OVERALL APPROACH AND NUMERICAL ALGORITHM

Consider the following scalar transfer function

$$H(\sigma) = \mathbf{b}^T(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{b}, \tag{2}$$

where $\mathbf{K}$ and $\mathbf{M}$ are two symmetric positive definite matrices, and $\mathbf{b}$ is an arbitrary vector. The poles of H are exactly the eigenvalues of the generalized symmetric eigenvalue problem

$$(\mathbf{K} - \lambda\mathbf{M})\mathbf{u} = \mathbf{0}.$$

Hence, checking the eigenvalues of $(\mathbf{K},\mathbf{M})$ is equivalent to checking the poles of H. Therefore, the approach proposed in this paper for identifying the eigenvalues missed by an eigensolver in an interval of interest $[\sigma_L,\ \sigma_R]$ consists in first computing a good approximation of H, then computing the poles of this approximation, and finally analyzing these poles to identify the sought-after missed eigenvalues. To this effect, it is first shown that H is a rational function.

2.1. Rational expression of H

Since $\mathbf{K}$ and $\mathbf{M}$ are symmetric, the pencil $(\mathbf{K},\mathbf{M})$ is diagonalizable. The diagonalization of this pencil can be written as

$$\mathbf{K}\mathbf{U} = \mathbf{M}\mathbf{U}\boldsymbol{\Lambda},$$

where $\mathbf{U}^T\mathbf{M}\mathbf{U} = \mathbf{I}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix whose entries are the eigenvalues of $(\mathbf{K},\mathbf{M})$. The columns of the matrix $\mathbf{U}$ are the corresponding eigenvectors.

Observing that $\mathbf{U}^{-1} = \mathbf{U}^T\mathbf{M}$, one obtains

$$\mathbf{K} = \mathbf{M}\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T\mathbf{M}$$

and

$$\mathbf{K} - \sigma\mathbf{M} = \mathbf{M}\mathbf{U}(\boldsymbol{\Lambda} - \sigma\mathbf{I})\mathbf{U}^T\mathbf{M}.$$

It follows that

$$(\mathbf{K} - \sigma\mathbf{M})^{-1} = \mathbf{U}(\boldsymbol{\Lambda} - \sigma\mathbf{I})^{-1}\mathbf{U}^T$$

and therefore

$$H(\sigma) = \mathbf{b}^T\mathbf{U}(\boldsymbol{\Lambda} - \sigma\mathbf{I})^{-1}\mathbf{U}^T\mathbf{b}. \tag{3}$$

The above result can be re-written as

$$H(\sigma) = \sum_{k=1}^{n} \frac{\left(\mathbf{U}^T\mathbf{b}\right)_k^2}{\lambda_k - \sigma}, \tag{4}$$

where $(\cdot)_k$ denotes the k-th component of a vector and λ_k is the k-th diagonal entry in $\boldsymbol{\Lambda}$. This shows that H is a rational function of σ with a numerator that is a polynomial of degree at most equal to $n-1$ and a denominator of degree at most equal to n, where n denotes the size of the matrix $\mathbf{K}$. Eq. (4) also emphasizes that the poles of H are exactly the eigenvalues of the pencil $(\mathbf{K},\mathbf{M})$.

2.2. One-point Padé approximation for local accuracy

A rational function such as H is typically well represented by a Padé approximant. The poles of this approximant can be expected to converge towards the poles of H, which are exactly the eigenvalues of the pencil $(\mathbf{K},\mathbf{M})$. A Padé approximation of type (l/m) of the function H around σ_0, an arbitrary real number for which however the matrix $\mathbf{K} - \sigma_0\mathbf{M}$ is non singular, is defined as a rational function of the form

$$H_{l,m}(\sigma) = \frac{p_0 + p_1(\sigma - \sigma_0) + \cdots + p_l(\sigma - \sigma_0)^l}{1 + q_1(\sigma - \sigma_0) + \cdots + q_m(\sigma - \sigma_0)^m} \tag{5}$$

whose Taylor expansion around σ_0 matches the first $l + m + 1$ terms of the Taylor expansion of H around the same point — that is,

$$H(\sigma) = H_{l,m}(\sigma) + \mathcal{O}\left((\sigma - \sigma_0)^{l+m+1}\right).$$

Since the numerator and denominator of H are polynomials of degrees equal at most to $n-1$ and n, respectively, a Padé approximation of type $\big((m-1)/m\big)$ is considered here and simply denoted by H_m. The coefficients of the polynomials are computed so that the approximant matches the first $2m$ terms of the following Taylor expansion around σ_0,

$$H_m(\sigma_0) = H(\sigma_0) \quad \text{and} \quad \frac{d^j H_m}{d\sigma^j}(\sigma_0) = \frac{d^j H}{d\sigma^j}(\sigma_0), \quad \forall\, j < 2m.$$

For the sake of completeness, the derivatives of H at σ_0 are computed. The transfer function H introduced in Eq. (2) can be re-written as follows

$$\begin{aligned} H(\sigma) &= \mathbf{b}^T \left[\mathbf{K} - \sigma_0\mathbf{M} - (\sigma - \sigma_0)\mathbf{M}\right]^{-1} \mathbf{b} \\ &= \mathbf{b}^T \left\{(\mathbf{K} - \sigma_0\mathbf{M})[\mathbf{I} - (\sigma - \sigma_0)(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{M}]\right\}^{-1} \mathbf{b} \\ &= \mathbf{b}^T \left[\mathbf{I} - (\sigma - \sigma_0)(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{M}\right]^{-1} (\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{b}. \end{aligned} \tag{6}$$

The Taylor expansion around σ_0 is

$$H(\sigma) = \sum_{q=0}^{+\infty} (\sigma - \sigma_0)^q \; \mathbf{b}^T \left[(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{M}\right]^q (\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{b}. \tag{7}$$

Since

$$H(\sigma_0) = \mathbf{b}^T(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{b}$$

and

$$\frac{d^j H}{d\sigma^j}(\sigma_0) = j! \; \mathbf{b}^T \left[(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{M}\right]^j (\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{b}, \text{ for } j < 2m,$$

it follows that

$$H_m(\sigma_0) = \mathbf{b}^T(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{b} \tag{8}$$

and

$$\frac{d^j H_m}{d\sigma^j}(\sigma_0) = j! \; \mathbf{b}^T \left[(\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{M}\right]^j (\mathbf{K} - \sigma_0\mathbf{M})^{-1}\mathbf{b}, \text{ for } j < 2m. \tag{9}$$

2.3. Multi-point Padé approximation for global accuracy

The Padé approximant H_m can be expected to converge first locally around the expansion point σ_0. However, the polynomial degree m can be expected to grow large before H_m becomes a globally accurate approximation of H. Since the main focus here is on checking for missing eigenvalues of $(\mathbf{K}, \mathbf{M})$ in a range of interest $[\sigma_L, \; \sigma_R]$, a global approximation of H over the entire interval $[\sigma_L, \; \sigma_R]$ is preferrable. Such a global approximation can be provided by the multi-point Padé method which matches the moments of H at I discrete points $(\sigma_i)_{1 \le i \le I} \in [\sigma_L, \; \sigma_R]$. If one chooses to fit the value of H and its $2J-1$ derivatives at each point σ_i, the resulting multi-point Padé approximation of H satisfies

$$\forall\, i \in \{1, \cdots, I\}, \quad H_m(\sigma_i) = H(\sigma_i) \quad \text{and} \quad \frac{d^j H_m}{d\sigma^j}(\sigma_i) = \frac{d^j H}{d\sigma^j}(\sigma_i), \quad \forall\, j < 2J.$$

In this case, one can write

$$H(\sigma) = H_m(\sigma) + \mathcal{O}\left(\prod_{i=1}^{I} (\sigma - \sigma_i)^{2J}\right),$$

where $m = IJ$. The distribution of the I points σ_i within the interval $[\sigma_L,\ \sigma_R]$ is arbitrary; it can be a uniform or non uniform one. The multi-point Padé approximation can also consider a variable number of derivatives at the points σ_i. However for the sake of simplicity, it is chosen here to consider the same number of derivatives at each point σ_i.

2.4. General form of the proposed algorithm

Let $\mathbf{U}_c$ denote the matrix storing the $n_c < n$ computed eigenvectors whose associated computed eigenvalues lie within the interval of interest $[\sigma_L,\ \sigma_R]$. If $\mathbf{b}$ is a random vector, it can be assumed that

$$\left(\mathbf{U}_c^T\mathbf{b}\right)_k \neq 0, \quad \forall\ k \leq n_c.$$

On the other hand if $\mathbf{b}$ is chosen in the null space of $\mathbf{U}_c^T$ — that is, if

$$\mathbf{U}_c^T\mathbf{b} = \mathbf{0}, \tag{10}$$

it follows from Eq. (4) that

$$H(\sigma) = \sum_{k\ |\ \lambda_k \in \Lambda_{nc}} \frac{\left(\mathbf{U}^T\mathbf{b}\right)_k^2}{\lambda_k - \sigma}, \tag{11}$$

where Λ_{nc} is the union of the set of missed eigenvalues within $[\sigma_L,\ \sigma_R]$ and the set of eigenvalues that are outside this interval. Hence, if $\mathbf{b}$ is chosen to satisfy Eq. (10), H identifies with the expression given in (11) and its poles become exactly the union of the eigenvalues that were missed in $[\sigma_L,\ \sigma_R]$ and those that lie outside this interval. This result leads to the following algorithm for identifying the eigenvalues of the pencil $(\mathbf{K},\mathbf{M})$ that were missed by an eigensolver in the interval $[\sigma_L,\ \sigma_R]$.

1. Gather in $\mathbf{U}_c$ the computed eigenvectors associated with the computed eigenvalues that lie in the interval $[\sigma_L,\ \sigma_R]$.
2. Generate a random vector $\tilde{\mathbf{b}}$.
3. Compute $\mathbf{b}$, the projection of $\tilde{\mathbf{b}}$ onto the space orthogonal to $\mathbf{U}_c$; $\mathbf{b}$ satisfies $\mathbf{U}_c^T\mathbf{b} = \mathbf{0}$.
4. Compute H_m, a multi-point Padé approximation of H in $[\sigma_L,\ \sigma_R]$.
5. Identify the poles of H_m in the interval $[\sigma_L,\ \sigma_R]$.

Next, the computation of the multi-point Padé approximation H_m is discussed.

3. COMPUTATION OF THE MULTI-POINT PADÉ APPROXIMATION

3.1. Relationship with Krylov subspaces

In [5], a non symmetric Lanczos process was developed for computing a single-point Padé approximant of a transfer function such as H and was shown to be computationally efficient. In [6], an approach based on the rational non symmetric Lanczos algorithm was proposed for computing a multi-point approximation of a rational function. Unfortunately, the non symmetric Lanczos algorithm can suffer a loss of orthogonality and may even break down. For this reason, a robust alternative based on a two-sided rational Arnoldi algorithm was proposed in [7] and [2]. In the specific case of the rational function H given in (2) where the matrices are symmetric and the vectors are the transpose of each other, the two-sided Arnoldi algorithm reduces to a symmetric Lanczos algorithm.

All numerical methods outlined above for computing a Padé approximant of a rational function such as H project the matrices $\mathbf{K}$ and $\mathbf{M}$, and the vector $\mathbf{b}$, onto a carefully constructed matrix $\mathbf{V}$. As such, they are often referred to as methods of "reduction by projection". Furthermore, all of these methods rely on Krylov subspaces for constructing the appropriate matrix $\mathbf{V}$. Here, a Krylov subspace is denoted by

$$\mathcal{K}_q(\breve{\mathbf{K}}, \breve{\mathbf{b}}) = \text{span}\left\{\breve{\mathbf{b}}, \breve{\mathbf{K}}\breve{\mathbf{b}}, \cdots, \breve{\mathbf{K}}^{q-1}\breve{\mathbf{b}}\right\}, \tag{12}$$

where $\breve{\mathbf{K}}$ is a matrix of interest and $\breve{\mathbf{b}}$ is a starting vector, and the projections of $\mathbf{K}$, $\mathbf{M}$, and $\mathbf{b}$ onto $\mathbf{V}$ are denoted by

$$\hat{\mathbf{K}} = \mathbf{V}^T\mathbf{K}\mathbf{V}, \quad \hat{\mathbf{M}} = \mathbf{V}^T\mathbf{M}\mathbf{V}, \quad \text{and } \hat{\mathbf{b}} = \mathbf{V}^T\mathbf{b}. \tag{13}$$

In [7, Theorem 3.1], it is proved that if the columns of the constructed $\mathbf{V}$ are such that their span verifies the following property

$$\text{span}\{\mathbf{V}\} \supset \oplus_{i=1}^{I}\mathcal{K}_J\left((\mathbf{K} - \sigma_i\mathbf{M})^{-1}\mathbf{M}, (\mathbf{K} - \sigma_i\mathbf{M})^{-1}\mathbf{b}\right), \tag{14}$$

the reduced matrices $\hat{\mathbf{K}}$ and $\hat{\mathbf{M}}$, and the reduced vector $\hat{\mathbf{b}}$ satisfy

$$\hat{\mathbf{b}}^T\left[(\hat{\mathbf{K}} - \sigma_i\hat{\mathbf{M}})^{-1}\hat{\mathbf{M}}\right]^j(\hat{\mathbf{K}} - \sigma_i\hat{\mathbf{M}})^{-1}\hat{\mathbf{b}} = \mathbf{b}^T\left[(\mathbf{K} - \sigma_i\mathbf{M})^{-1}\mathbf{M}\right]^j(\mathbf{K} - \sigma_i\mathbf{M})^{-1}\mathbf{b}, \tag{15}$$

where $j = 0, 1, \cdots, 2J-1$. From Eqs. (8,9) and Eq. (15) above, it follows that when the property (14) is satisfied, the function

$$H_m(\sigma) = \hat{\mathbf{b}}^T(\hat{\mathbf{K}} - \sigma\hat{\mathbf{M}})^{-1}\hat{\mathbf{b}} = \left(\mathbf{V}^T\mathbf{b}\right)^T\left(\mathbf{V}^T\mathbf{K}\mathbf{V} - \sigma\mathbf{V}^T\mathbf{M}\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{b} \tag{16}$$

matches the values of H and its $2J-1$ derivatives at the points $(\sigma_i)_{1\leq i\leq I}$, and therefore is a multi-point Padé approximant of H.

3.2. Lanczos algorithm for computing the subspace of projection

Algorithm 1 describes a symmetric Lanczos algorithm with full reorthogonalization for constructing the columns of a matrix $\mathbf{V}$, of dimension $n \times IJ$, that satisfies $\mathbf{V}^T\mathbf{M}\mathbf{V} = \mathbf{I}$, $\mathbf{U}_c^T\mathbf{M}\mathbf{V} = \mathbf{0}$, and the property (14). After $\mathbf{V}$ is built, the reduced-order quantities (13) can be constructed and the sought-after multi-point Padé approximant (16) of H can be computed.

Note that the construction of $\mathbf{V}$ requires the solution of $m = IJ$ linear systems (J linear systems with $(\mathbf{K} - \sigma_i\mathbf{M})$ for I sample points σ_i). When the norm $\sqrt{\mathbf{v}^T\mathbf{M}\mathbf{v}}$ of a vector $\mathbf{v}$ is zero, **Algorithm 1** updates the columns of $\mathbf{V}$ by a new random vector while maintaining the property (14). The reduced matrix $\hat{\mathbf{M}}$ is replaced by the identity matrix. The reduced matrix $\hat{\mathbf{K}}$ and the reduced vector $\hat{\mathbf{b}}$ are built by direct projection.

4. MISSING EIGENVALUE FINDER

The discussion presented in the previous sections of this paper leads to proposing **Algorithm 2** described below for the factorization-free identification of the eigenvalues missed in an arbitrary

Algorithm 1 Symmetric Lanczos process

```
(Initialize) $p = 1$
for $i = 1, 2, \cdots, I$ do
  for $j = 1, 2, \cdots, J$ do
    if $j == 1$ then
      $\mathbf{v} = (\mathbf{K} - \sigma_i \mathbf{M})^{-1} \mathbf{b}$
    else
      $\mathbf{v} = (\mathbf{K} - \sigma_i \mathbf{M})^{-1} (\mathbf{I} - \mathbf{U}_c (\mathbf{U}_c^T \mathbf{U}_c)^{-1} \mathbf{U}_c^T) \mathbf{M} \mathbf{v}_{p-1}$
    end if
    if $p > 1$ then
      (Orthogonalize) $\mathbf{v} = \mathbf{v} - \mathbf{V}_{p-1} \mathbf{V}_{p-1}^T \mathbf{M} \mathbf{v}$
    end if
    $\gamma = \sqrt{\mathbf{v}^T \mathbf{M} \mathbf{v}}$
    if $\gamma == 0$ then
      Reset $\mathbf{v}$ to a random vector
      (Orthogonalize) $\mathbf{v} = \mathbf{v} - \mathbf{V}_{p-1} \mathbf{V}_{p-1}^T \mathbf{M} \mathbf{v}$.
      $\gamma = \sqrt{\mathbf{v}^T \mathbf{M} \mathbf{v}}$.
    end if
    (Update) $\mathbf{v}_p = \mathbf{v} / \gamma$
    $p = p + 1$
  end for
end for
```

Algorithm 2 Lanczos-based eigencheck

```
Require: Computed $\mathbf{M}$-orthonormal eigenvectors $\mathbf{U}_c$
Require: Range of interest $[\sigma_L,\ \sigma_R]$
Require: Value for $I$ (for sampling the above range of interest in $I$ points $(\sigma_i)_{1 \leq i \leq I}$)
Require: Value for $J$ (for matching at each point $\sigma_i$ the first $2J - 1$ derivatives of the scalar
    rational function $H$)
Sample $[\sigma_L,\ \sigma_R]$ in $I$ points $(\sigma_i)_{1 \leq i \leq I}$
Generate a random vector $\tilde{\mathbf{b}}$
Compute $\mathbf{b} = \left(\mathbf{I} - \mathbf{U}_c (\mathbf{U}_c^T \mathbf{U}_c)^{-1} \mathbf{U}_c^T\right) \tilde{\mathbf{b}}$
Apply Algorithm 1 for constructing the matrix $\mathbf{V}$ of dimension $n \times m$ (where $m = IJ$).
Compute the reduced matrices $\hat{\mathbf{K}}$ and $\hat{\mathbf{M}}$ (we recall $\hat{\mathbf{M}} = \mathbf{I}$)
Compute the full solution of the reduced-order $m \times m$ generalized symmetric eigenvalue
    problem
        $$\hat{\mathbf{K}} \mathbf{w} = \lambda \hat{\mathbf{M}} \mathbf{w} \qquad (17)$$
Identify those computed eigenvalues of problem (17) which lie in the interval of interest
    $[\sigma_L,\ \sigma_R]$
```

range of interest $[\sigma_L,\ \sigma_R]$ by an eigensolver applied to the solution of the generalized symmetric eigenvalue problem (1).

Algorithm 2 is essentially a post-processing procedure. It does not depend on the eigensolver used to compute the eigenvectors $\mathbf{U}_c$ and the corresponding eigenvalues. The inputs to this proposed algorithm are the computed eigenvectors $\mathbf{U}_c$, the range of interest $[\sigma_L,\ \sigma_R]$, the desired number of sampling points I, and the integer J for which the Padé approximant of H matches its first $2J-1$ derivatives at the sampled points $(\sigma_i)_{1\leq i\leq I}$. The projection of the randomly generated vector $\tilde{\mathbf{b}}$ (Step 3) ensures that the vector $\mathbf{b}$ satisfies the desired property

$$\mathbf{U}_c^T\mathbf{b} = \mathbf{0}$$

so that the poles of the computed Padé approximant correspond to the missed eigenvalues in the range of interest $[\sigma_L,\ \sigma_R]$.

In Section 1, it was noted that experts often try to find the missed eigenpairs by re-applying their eigensolvers in the space that is $\mathbf{M}$-orthogonal to the previously computed eigenvectors. Here, it is concluded that this process is equivalent to **Algorithm 2** with $I = 1$ (a single sampling point) and therefore is unreliable for identifying all eigenvalues that have been missed in a specific range of interest $[\sigma_L,\ \sigma_R]$.

5. EXAMPLES

In this section, the proposed factorization-free **Algorithm 2** for finding the eigenvalues missed in an interval of interest by an eigensolver is illustrated with two numerical examples. The first one is of the academic type. It has the merit of being easily reproducible by the reader. It also demonstrates the main behavior and tendencies of the proposed **Algorithm 2**. The second example is associated with a large-scale structural vibrations problem. It highlights the potential of the proposed algorithm for real engineering applications. In both cases, ARPACK [8] is chosen as the eigensolver.

5.1. Matrix BCSSTK16

Here, the generalized symmetric eigenvalue problem (1) is considered with $\mathbf{K}$ set to the matrix BCSSTK16 from the Harwell-Boeing sparse matrix collection [3] whose size is $n = 4884$, and $\mathbf{M}$ set to the identity matrix. For reference, a complete solution of this problem is obtained using the MATLAB EIG routine. It is found that all 73 first eigenvalues of the pencil $(\mathbf{K}, \mathbf{M})$ are equal to 1.0, the 74th one is equal to 1.5×10^5, and the largest eigenvalue is equal to 4.9×10^9.

To compute an approximation of the first 100 eigenpairs $(\mathbf{U}_c, \mathbf{\Lambda}_c)$ of the problem specified above, ARPACK is applied within MATLAB 7 in the shift-invert mode with a shift $\sigma = 0$, a Krylov subspace of size equal to 200, and the convergence tolerance $\tau = 10^{-8}$. For this purpose, the random starting vector is generated with the MATLAB commands

```
rand('state', 1);
resid = rand(size(K,1), 1);
```

Using these parameters, 67 of the 100 computed eigenpairs are found to have their eigenvalues in the range $[0, 10]$. Since the problem considered here has exactly 73 eigenvalues in $[0, 10]$, this means that in this case, ARPACK has missed six eigenpairs in the range $[0, 10]$. The 67 computed eigenvalues in $[0, 10]$ are equal to 1.0 up to machine precision.

Then, **Algorithm 2** is applied to find the eigenvalues missed in the range $[0, 10]$. The random vector $\tilde{\mathbf{b}}$ is generated with the MATLAB commands:

```
rand('state', 2);
btilde = rand(size(K, 1), 1);
```

The obtained results with $I = 1$ (one-point Padé approximation) are reported in Table I. More specifically, Table I reports for different matching points σ_0, the number of eigenvalues identified by **Algorithm 2** as missed eigenvalues in $[0, 10]$, as a function of the polynomial degree $m = J$ characterizing the Padé approximant H_J. The reader can observe that once it has converged, **Algorithm 2** retrieves all six eigenpairs that have been missed by ARPACK in the range $[0, 10]$. The convergence of this algorithm is shown however to depend on the matching point σ_0. Indeed, since all 73 eigenvalues lying in $[0, 10]$ are equal to 1, the closer σ_0 is to 1, the smaller can the degree m be expected.

	$\sigma_0 = 0$	$\sigma_0 = 5$	$\sigma_0 = 10$
ARPACK	67	67	67
ARPACK + poles of H_2	67 + 1	67 + 1	67 + 1
ARPACK + poles of H_4	67 + 2	67 + 2	67 + 1
ARPACK + poles of H_8	67 + 3	67 + 3	67 + 3
ARPACK + poles of H_{16}	67 + 6	67 + 5	67 + 5
ARPACK + poles of H_{17}	67 + 6	67 + 6	67 + 5
ARPACK + poles of H_{18}	67 + 6	67 + 6	67 + 6
ARPACK + poles of H_{19}	67 + 6	67 + 6	67 + 6
ARPACK + poles of H_{20}	67 + 6	67 + 6	67 + 6

Table I. Problem BCSSTK16: number of identified missed eigenvalues in $[0, 10]$ (specified after the + symbol)

To illustrate that **Algorithm 2** recovers the correct eigenvalues in $[0, 10]$, Table II reports the smallest eigenvalues of the reduced pencil $(\hat{\mathbf{K}}, \hat{\mathbf{M}})$ (up to the first six) when $\sigma_0 = 5$. The

	Recovered values by **Algorithm 2**
Poles of H_2	(1.0000, 5.0×10^7)
Poles of H_4	(1.0000, 1.0917, 3.4×10^7, 1.2×10^8)
Poles of H_6	(1.0000, 1.0000, 3.0×10^7, 4.5×10^7, 8.7×10^7, 3.7×10^8)
Poles of H_8	(1.0000, 1.0000, 1.0000, 3.0×10^7, 3.9×10^7, 5.5×10^7, $\cdots$)
Poles of H_{12}	(1.0000, 1.0000, 1.0000, 1.0000, 3.0×10^7, 3.2×10^7, $\cdots$)
Poles of H_{16}	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 3.0×10^7, $\cdots$)
Poles of H_{17}	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0716, $\cdots$)
Poles of H_{18}	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, $\cdots$)
Poles of H_{19}	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, $\cdots$)
Poles of H_{20}	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, $\cdots$)

Table II. Problem BCSSTK16: poles recovered by **Algorithm 2** when $\sigma_0 = 5$

reader can observe that once it has identified six missed eigenvalues in the range $[0, 10]$ (with H_{17}), **Algorithm 2** refines the values for the eigenpairs that have been missed by ARPACK. However, **Algorithm 2** can only guarantee that the recovered eigenvalues will be accurate up to $\tau = 10^{-8}$ as $(\mathbf{U}_c, \mathbf{\Lambda}_c)$ were computed with that tolerance.

Next, a smaller tolerance $\tau = 10^{-16}$ is used and all other ARPACK parameters are kept the same. In this case, ARPACK finds all 73 eigenvalues that belong to the range $[0, 10]$, and the post-processing **Algorithm 2** correctly finds that no eigenvalue has been missed in this range.

Finally, the post-processing **Algorithm 2** is reapplied for finding the eigenvalues missed in the range $[0, 10]$ using this time $I > 1$ matching points (multi-point Padé approximation). Table III reports for this case the minimal polynomial degree $m^* = IJ^*$ (or for a given I, the minimum value of J, J^*) that enables **Algorithm 2** to identify six missed eigenvalues in the range $[0, 10]$, as a function of I and the choice of the sampling points $(\sigma_i)_{1\leq i\leq I}$. Note that m^* is also the number of columns in $\mathbf{V}$. As expected from the global approximation property of the multi-point Padé algorithm, m^* is found to decrease with I.

I	$(\sigma_i)_{1\leq i\leq I}$	J^*	$m^* = IJ^*$
1	$\{5\}$	17	17
2	$\{0, 10\}$	9	18
3	$\{0, 5, 10\}$	6	18
4	$\{0, 10/3, 20/3, 10\}$	4	16
5	$\{0, 2.5, 5, 7.5, 10\}$	3	15
6	$\{0, 10/6, 20/6, 30/6, 40/6, 50/6, 10\}$	2	12

Table III. Problem BCSSTK16: smallest polynomial degree $m = IJ$ for which **Algorithm 2** identifies correctly all eigenvalues missed in $[0, 10]$

5.2. Large-scale structural vibrations problem

Here, the generalized symmetric eigenvalue problem (1) is considered with $\mathbf{K}$ and $\mathbf{M}$ set to the stiffness and mass matrices, respectively, arising from the finite element discretization of the bolted joint model of Figure 1 whose size is $n = 322710$. To construct an approximation of the first 100 eigenpairs $(\mathbf{U}_c, \mathbf{\Lambda}_c)$ of this model, the P(arallel)ARPACK FORTRAN library functions PDSAUPD and PDSEUPD are applied within an in-house developed finite element structural analysis code in the shift-invert mode. The shift parameter is set to $\sigma = 0$, the size of the Krylov subspace is set to 200, the convergence tolerance to $\tau = 10^{-16}$, and the FETI-DP solver [12, 13] is chosen to solve all the resulting linear systems of equations. Furthermore, every 10^{th} converged eigenpair is deliberately excluded from the approximation $(\mathbf{U}_c, \mathbf{\Lambda}_c)$ so that some (or all) of the missed eigenvalues can be known a priori for the purpose of verification.

Using the above parameters, 92 of the 100 computed eigenpairs are found to have their eigenvalues in the range $[0, 3.9478\times10^9]$ (corresponding to the frequency range $[0, 1\times10^4]$ Hz) of which nine are deliberately excluded from the approximation. The eigenvalues of the excluded eigenpairs are $3.4939 \times 10^7, 1.7928 \times 10^8, 3.3891 \times 10^8, 8.0245 \times 10^8, 1.0672 \times 10^9, 1.4863 \times 10^9, 2.4172\times10^9, 2.8610\times10^9$, and 3.7337×10^9. The number of genuinely missed eigenvalues is unknown, but is most likely zero since the problem did not exhibit any clustering phenomenon and a tight convergence tolerance was chosen.

Figure 1. Large-scale finite element model of a bolted joint for modal analysis

Next, **Algorithm 2** is applied to find the excluded eigenvalues as well as those potentially missed in the range $[0, 3.9478418 \times 10^9]$. In **Algorithm 1**, the linear systems are solved iteratively by the FETI-DPH solver [14], a domain-decomposition based iterative solver designed for indefinite problems of the form $(\mathbf{K} - \sigma\mathbf{M})\mathbf{x} = \mathbf{f}$. The results obtained with $I = 1$ (one-point Padé approximation) are summarized in Table IV. More specifically, Table IV reports for different matching points σ_0, the number of eigenvalues identified by **Algorithm 2** as missed eigenvalues in $[0, 3.9478 \times 10^9]$, as a function of the polynomial degree $m = J$ characterizing the Padé approximant H_J. The reader can observe that once it has converged, **Algorithm 2** retrieves all nine eigenpairs excluded from the approximation in the range $[0, 3.9478 \times 10^9]$ and does not identify any other missed eigenvalue. Since all 92 eigenvalues lying in $[0, 3.9478 \times 10^9]$ are evenly spaced, a value of σ_0 in the middle of the range can be expected, and indeed is found, to be the best one.

To illustrate that **Algorithm 2** recovers the correct eigenvalues in $[0, 3.9478 \times 10^9]$, Table V reports the smallest eigenvalues of the reduced pencil $(\hat{\mathbf{K}}, \hat{\mathbf{M}})$ (up to the first nine ones) when $\sigma_0 = 1.9739 \times 10^9$. The reader can observe that once it has identified nine missed eigenvalues

	$\sigma_0 = 0$	$\sigma_0 = 1.9739 \times 10^9$	$\sigma_0 = 3.9478 \times 10^9$
PARPACK†	83	83	83
PARPACK† + poles of H_4	83 + 2	83 + 2	83 + 1
PARPACK† + poles of H_8	83 + 5	83 + 4	83 + 2
PARPACK† + poles of H_{12}	83 + 7	83 + 6	83 + 3
PARPACK† + poles of H_{16}	83 + 7	83 + 8	83 + 4
PARPACK† + poles of H_{18}	83 + 8	83 + 9	83 + 5
PARPACK† + poles of H_{20}	83 + 8	83 + 9	83 + 5
PARPACK† + poles of H_{22}	83 + 9	83 + 9	83 + 6
PARPACK† + poles of H_{24}	83 + 9	83 + 9	83 + 6
PARPACK† + poles of H_{26}	83 + 9	83 + 9	83 + 7
PARPACK† + poles of H_{28}	83 + 9	83 + 9	83 + 7
PARPACK† + poles of H_{30}	83 + 9	83 + 9	83 + 8
PARPACK† + poles of H_{32}	83 + 9	83 + 9	83 + 8
PARPACK† + poles of H_{34}	83 + 9	83 + 9	83 + 8
PARPACK† + poles of H_{36}	83 + 9	83 + 9	83 + 9
PARPACK† + poles of H_{38}	83 + 9	83 + 9	83 + 9
PARPACK† + poles of H_{40}	83 + 9	83 + 9	83 + 9

Table IV. Large-scale structural vibrations problem: number of identified missed eigenvalues in $[0, 3.9478 \times 10^9]$ (specified after the + symbol). PARPACK† denotes PARPACK results with selected eigenpairs excluded

	Recovered values by **Algorithm 2**
Poles of H_4	$(2.2807 \times 10^9,\ 2.4173 \times 10^9)$
Poles of H_8	$(9.6607 \times 10^8,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8452 \times 10^9)$
Poles of H_{12}	$(3.1697 \times 10^8,\ 8.0244 \times 10^8,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9)$
Poles of H_{16}	$(1.7875 \times 10^8,\ 3.3889 \times 10^9,\ 8.0245 \times 10^9,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9,\ 3.7349 \times 10^9)$
Poles of H_{20}	$(3.4946 \times 10^7,\ 1.7928 \times 10^8,\ 3.3891 \times 10^8,\ 8.0245 \times 10^8,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9,\ 3.7337 \times 10^9,\ \cdots)$
Poles of H_{22}	$(3.4939 \times 10^7,\ 1.7928 \times 10^8,\ 3.3891 \times 10^8,\ 8.0245 \times 10^8,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9,\ 3.7337 \times 10^9,\ \cdots)$
Poles of H_{24}	$(3.4939 \times 10^7,\ 1.7928 \times 10^8,\ 3.3891 \times 10^8,\ 8.0245 \times 10^8,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9,\ 3.7337 \times 10^9,\ \cdots)$
Poles of H_{26}	$(3.4939 \times 10^7,\ 1.7928 \times 10^8,\ 3.3891 \times 10^8,\ 8.0245 \times 10^8,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9,\ 3.7337 \times 10^9,\ \cdots)$
Poles of H_{28}	$(3.4939 \times 10^7,\ 1.7928 \times 10^8,\ 3.3891 \times 10^8,\ 8.0245 \times 10^8,\ 1.0672 \times 10^9,\ 1.4863 \times 10^9,\ 2.4172 \times 10^9,\ 2.8610 \times 10^9,\ 3.7337 \times 10^9,\ \cdots)$

Table V. Large-scale structural vibrations problem: poles recovered by **Algorithm 2** when $\sigma_0 = 1.9739 \times 10^9$

in the range $[0, 3.9478 \times 10^9]$, **Algorithm 2** refines the actual values for the identified missed eigenpairs. Note that nine poles for H_{20} and H_{22} are in the range $[0, 3.9478 \times 10^9]$ and that the smallest pole is not exact for H_{20} but more accurate for H_{22}.

Finally, the post-processing **Algorithm 2** is reapplied for finding the eigenvalues missed in the range $[0, 3.9478 \times 10^9]$ using this time $I > 1$ matching points (multi-point Padé approximation). Table VI reports for this case the minimal polynomial degree $m^* = IJ^*$ (or for a given I, the minimum value of J, J^*) that enables **Algorithm 2** to identify nine missed eigenvalues in the range $[0, 3.9478 \times 10^9]$, as a function of I and the choice of the sampling points $(\sigma_i)_{1 \leq i \leq I}$. Note that m^* is also the dimension of the reduced pencil $(\hat{\mathbf{K}}, \hat{\mathbf{M}})$ and the total number of linear solves performed in **Algorithm 1**. As expected from the global approximation property of the multi-point Padé algorithm, J^* and m^* are found to decrease with I.

I	$(\sigma_i)_{1 \leq i \leq I}$	J^*	$m^* = IJ^*$
1	$\{1.9739 \times 10^9\}$	18	18
2	$\{0, 3.9478 \times 10^9\}$	9	18
3	$\{0, 1.9739 \times 10^9, 3.9478 \times 10^9\}$	6	18
4	$\{0, 1.3159 \times 10^9, 2.6319 \times 10^9, 3.9478 \times 10^9\}$	4	16
5	$\{0, 9.8696 \times 10^8, 1.9739 \times 10^9, 2.9609 \times 10^9, 3.9478 \times 10^9\}$	3	15
6	$\{0, 7.8956 \times 10^8, 1.5791 \times 10^9, 2.3687 \times 10^9, 3.1582 \times 10^9, 3.9478 \times 10^9\}$	3	18

Table VI. Large-scale structural vibrations problem: smallest polynomial degree $m = IJ$ for which **Algorithm 2** identifies correctly all eigenvalues missed in $[0, 3.9478 \times 10^9]$

6. CONCLUSIONS

To the best of the author's knowledge, a factorization-free algorithm for detecting and identifying the eigenvalues missed by an eigensolver equipped with an iterative linear equation solver within an interval of interest $[\sigma_L,\ \sigma_R]$ has been presented for the first time. This algorithm constructs a scalar, rational, transfer function whose poles are exactly the eigenvalues of the symmetric pencil $(\mathbf{K}, \mathbf{M})$, approximates it by a Padé expansion, and computes the poles of this approximation to detect and identify the missed eigenvalues. The proposed algorithm was illustrated with an academic numerical example. Its potential for real engineering applications was also demonstrated with a large-scale structural vibrations problem.

Acknowledgments

The third author thanks Dr. R. Lehoucq (Sandia National Laboratories) for enlightening discussions about the Lanczos algorithm.

REFERENCES

1. P. Avery, C. Farhat, and G. Reese. Fast frequency sweep computations using a multi-point Padé-based reconstruction method and an efficient iterative solver. *Int. J. Numer. Meth. Engng.*, 69(13):2848–2875, 2007.
2. J. Cullum and T. Zhang. Two-sided Arnoldi and nonsymmetric Lanczos algorithms. *SIAM J. Matrix Anal. Appl.*, 24(2):303–319, 2002.
3. I. Duff, R. Grimes, and J. Lewis. Users' guide for the Harwell-Boeing sparse matrix collection (Release I). Technical Report TR/PA/92/86, CERFACS, 1992.
4. R. Djellouli, C. Farhat, and R. Tezaur. A fast method for solving acoustic scattering problems in frequency bands. *J. Comput. Phys.*, 168:412–432, 2001.
5. P. Feldmann and R. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, 14(5):639–649, 1995.
6. K. Gallivan, E. Grimme, and P. van Dooren. A rational Lanczos algorithm for model reduction. *Numer. Algorithms*, 12:33–63, 1996.
7. E. Grimme. *Krylov projection methods for model reduction.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.
8. R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods.* Society for Industrial and Applied Mathematics, 1998.
9. J. McCombs and A. Stathopoulos. Iterative validation of eigensolvers: a scheme for improving the reliability of Hermitian eigenvalue solvers. *SIAM J. Sci. Comput.*, 28(6):2337–2358, 2006.
10. K. Olsson and A. Ruhe. Rational Krylov for eigenvalue computation and model order reduction. *BIT Numerical Mathematics*, 46:S99–S111, 2006.
11. B. N. Parlett. *The symmetric eigenvalue problem.* Society for Industrial and Applied Mathematics, 1998.
12. C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7:687–714, 2000.
13. C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - Part I: a faster alternative to the two-level FETI method. *Int. J. Numer. Meth. Engng.*, 50:1523–1544, 2001.
14. C. Farhat, J. Li, and P. Avery. A FETI-DP method for the parallel iterative solution of indefinite and complex-valued solid and shell vibration problems. *Int. J. Numer. Meth. Engng.*, 63(3):398–427, 2005.