Directional algorithms for the frequency isolation problem in undamped vibrational systems

Julio Moro* and Juan C. Egaña †

Abstract

A new algorithm is presented to solve the frequency isolation problem for vibrational systems with no damping: given an undamped mass-spring system with resonant eigenvalues, the system must be re-designed, finding some close-by non-resonant system at a reasonable cost. Our approach relies on modifying masses and stiffnesses along directions in parameter space which produce a maximal variation in the resonant eigenvalues, provided the non-resonant ones do not undergo large variations. The algorithm is derived from first principles, implemented, and numerically tested. The numerical experiments show that the new algorithms are considerably faster and more robust than previous algorithms solving the same problem.

Keywords: frequency isolation, resonance, eigenvalue perturbation, partial pole assignment

1 Introduction

It is well known [13] that whenever the natural frequencies of a vibrating structure are close to the frequencies of some external force, these vibrations may be amplified to the point of becoming dangerous. This is the so-called phenomenon of *resonance*. The external forces may be, for instance, those produced by the waves affecting an off-shore oil platform [1, p. 146], an earthquake acting on a building [16, p. xv] or, maybe the most notorious example recently, the steps of pedestrians walking on the London Millenium Bridge [18, p. 235].

To model resonance in mathematical terms, some interval on the real line is typically identified as the *resonance band*, i.e., the region which should be free of natural frequencies in order to guarantee non-resonance. One example of this is the earthquake band

*Departamento de Matemáticas, Universidad Carlos III de Madrid, Madrid (Spain); e-mail: jmoro@math.uc3m.es. The research of this author was partially supported by the spanish Ministerio de Economía y Competitividad under grant MTM2014-54692-P.

†Departamento de Matemáticas, Universidad Católica del Norte, Antofagasta (Chile); e-mail: jegana@ucn.cl. The research of this author was partially supported by FONDECYT under grant FONDECYT 10050026, and by Universidad Católica del Norte internal grant DGIP-UCN.

prescribed by the California State Building Department: in order to minimize damage, the natural frequencies of any new building constructed in California must be outside that band [16, p. xv].

In this paper we propose new algorithms for a *frequency isolation problem*, which occurs whenever an initial design for a vibrational structure has some of its natural frequencies within the resonance band. In order to avoid resonance, the system must be re-designed in such a way that all new natural frequencies lie outside the resonance band, and this should be done in such a way that the impact (or the cost) of re-design is small, i.e., the new non-resonant structure is close in some sense to the initial one. In our case we focus on undamped mass-spring systems

$$Mx'' + Kx = F(t), \tag{1}$$

which appear frequently in structural engineering problems. The unknown $x = x(t)$ is an n-vector, and the $n \times n$ matrices M and K are

$$M = \mathrm{diag}(m_1, \ldots, m_n), \qquad m_i > 0, \quad i = 1, \ldots, n, \tag{2}$$

with K symmetric positive definite and tridiagonal. We consider fixed-free boundary conditions[1], i.e.,

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_n \end{bmatrix} \tag{3}$$

with $k_i > 0,\ i = 1, \ldots, n$. This corresponds to a configuration with n masses connected by n springs of stiffnesses $k_i > 0$, attaching the first spring to a wall and leaving the last mass free.

If periodic solutions $x(t) = ue^{i\omega}t$ of the unforced equations $Mx'' + Kx = 0$ are sought, then the vector u and the scalar ω satisfy $(\omega^2 M - K)u = 0$. In other words, the natural frequencies of (1) are the square roots of the eigenvalues, and the natural modes of vibration are the corresponding eigenvectors of the generalized eigenvalue problem

$$(K - \lambda M)u = 0. \tag{4}$$

Sometimes (see e.g. [10]) it is convenient to rewrite this generalized eigenvalue problem as a conventional one

$$Jv = \lambda v,$$

[1]We focus on these boundary conditions for the sake of concretion, but other boundary conditions allow for a similar treatment

for the matrix

$$J = M^{-1/2} K M^{-1/2} =$$

$$= \begin{bmatrix} \frac{k_1+k_2}{m_1} & -\frac{k_2}{\sqrt{m_1 m_2}} & & & \\ -\frac{k_2}{\sqrt{m_1 m_2}} & \frac{k_2+k_3}{m_2} & -\frac{k_3}{\sqrt{m_2 m_3}} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{k_{n-1}}{\sqrt{m_{n-2} m_{n-1}}} & \frac{k_{n-1}+k_n}{m_{n-1}} & -\frac{k_n}{\sqrt{m_{n-1} m_n}} \\ & & & -\frac{k_n}{\sqrt{m_{n-1} m_n}} & \frac{k_n}{m_n} \end{bmatrix}. \tag{5}$$

The natural frequencies are the square roots of the eigenvalues of J, and the natural modes of vibration are $u = M^{-1/2} v$, where v is the corresponding eigenvector. Notice that the matrix J is also symmetric positive definite and tridiagonal.

In this setting, the frequency isolation problem can be posed as follows:

Frequency isolation problem: *Given a resonance band* $(c - r, c + r) \subset \mathbb{R}$ *and matrices* $\overline{M}, \overline{K}$ *as in* (2), (3) *such that some eigenvalues of* $(\overline{K} - \lambda \overline{M})u = 0$ *lie inside the resonance band, find new matrices* M^* *and* K^**, also as in* (2), (3)*, and close, respectively, to* $\overline{M}$ *and* $\overline{K}$*, such that no eigenvalue of* $(K^* - \lambda M^*)u = 0$ *lies inside the resonance band.*

The frequency isolation problem is conceptually close to other well-known problems in the theory of vibrating systems (and control theory), like the *partial pole placement problem* [5, 6] and the *eigenvalue embedding problem* [4] or, more generally, *model updating problems* [12]. In all of them a potentially dangerous subset of the spectrum must be moved elsewhere by appropriately updating the system parameters. However, there are important differences between them: while in the frequency isolation problem no restriction is imposed on the 'non-dangerous' part of the spectrum, in the other three problems this part of the spectrum and the corresponding eigenvectors must remain fixed. In fact, in model updating problems, both a part of the spectrum and the corresponding eigenvectors must be preserved. Also, in the frequency isolation problem, the main concern is to find a reasonably small update, and structure must be preserved, i.e., the update must keep stringent symmetry, bandedness and definiteness constraints on the matrix coefficients of (4).

One comment to be made is that, of course, one can pose a mathematically more demanding question by asking M^* and K^* to be *as close as possible*, for some appropriate metric in space (M, K), to $\overline{M}$ and $\overline{K}$ within the class of matrices of the forms (2), (3). However, this makes the problem considerably more difficult, since then it becomes an optimization problem *with constraints on interior eigenvalues*. Whereas constraints on the largest (or smallest) eigenvalues can be treated by using, for instance, semidefinite programming, in our case the resonant frequencies can be placed anywhere in the spectrum, and this makes the optimization problem notoriously harder to deal with. Therefore, we will be deliberately vague, not specifying the degree of closeness between (M^*, K^*) and $(\overline{M}, \overline{K})$. As a matter of fact, in practice, the decision on what is close and what is not depends ultimately on the real-world constraints on the particular application which (4) comes from.

The frequency isolation problem has been previously addressed by Joseph [14], who proposed a Newton-type method for structures vibrating at low frequencies. The computational cost of this algorithm, however, was very high, and the reconstructed system was frequently far away from the initial one. Later, Egaña et al. proposed in [10] a less costly inverse eigenvalue method: a target spectrum away from the resonance band is fixed in advance. Although there is an infinity of tridiagonal matrices J, as in (5), with that particular target as their spectrum, this infinite isospectral family can be parametrized in terms of all interlaced spectra of cardinal $n-1$, i.e., if $\lambda_1 \geq \ldots \geq \lambda_n$ is the chosen target spectrum, each interlaced spectrum is a set $\{\mu_1, \ldots, \mu_{n-1}\}$ with $\lambda_1 \geq \mu_1 \geq \ldots \geq \mu_{n-1} \geq \lambda_n$. Once an interlaced spectrum and the total mass $\sum m_i$ are fixed, only one symmetric tridiagonal J exists whose spectrum is the set of λs, and such that the μs are the spectrum of the leading principal submatrix of order $n-1$ of J. Furthermore, this unique matrix can be reconstructed at cost $O(n^3)$ via a method of Lanczos-type [3]. In order to make the distance in $\mathbb{R}^{2n}$ between the new masses and stiffnesses and the initial ones small, solutions are optimized over the set of interlaced spectra using a trust region algorithm for bound constrained optimization, the so-called *Box method* [11]. Once the optimal J^* is obtained, M^* and K^* are easily recovered. This optimization procedure guarantees that the solutions computed by the algorithm are close to the original configuration. However, since each step in the optimization process invokes the $O(n^3)$ reconstruction algorithm and convergence typically requires well over n steps, the algorithm in [10] is still computationally quite expensive. Furthermore, the optimization code runs into numerical problems whenever the interlaced spectrum becomes clustered, so the overall algorithm only works satisfactorily when the n initial eigenvalues are more or less evenly spaced.

Our goal in this paper is to describe a new method, which produces solutions of the same quality as in [10] (in the sense that they are roughly at the same distance from the original design), but at a significantly lower computational cost. Furthermore, the algorithm is more robust, in the sense that it is not affected by the distribution of eigenvalues. One of the ideas underlying the new method is that it is somehow unnatural to prescribe a target spectrum from the beginning: for simplicity, Egaña et al. keep all nonresonant eigenvalues fixed, and move only the resonant ones to the boundary of the resonance interval. There is no reason why this should correspond to any optimal configuration, and in fact it artificially restricts the computed solution to be in a low-dimensional, arbitrarily chosen isospectral set. Instead, we will allow the whole spectrum to move anywhere, with only one restriction: we do not want the nonresonant part of the spectrum to move too much, in order to prevent it from entering the resonance band. Once this is taken care of, the resonant part is steered out of the resonance band as fast as possible using first derivatives of eigenvalues as a guide: taking the eigenvalues of (4) as a function of $(m,k) = (m_1, \ldots, m_n, k_1, \ldots, k_n) \in \mathbb{R}^{2n}$, we will make use of the gradients $\nabla\lambda_j(m,k)$ in order to find directions of maximal variation for the appropriate eigenvalues in parameter space (m,k). Notice that translating directions in parameter space (m,k) into directions in matrix space (M,K), and viceversa, is trivial. Therefore, from now on, we will work only in parameter space (m,k), in the understanding that once the solution (m^*,k^*) is found, we transform it into the corresponding solution (M^*,K^*) in matrix space. Once the special directions mentioned above are identified, a bisection procedure is used to monitor the critical crossings when the resonant eigenvalues leave the resonance

band. In this sense, our method is in the spirit of [17], where a similar bisection method is employed.

The paper is organized as follows: Section 2 reviews two fundamental tools, namely derivatives of eigenvalues and the bisection method, which are central to the subsequent discussion. In Section 3 we derive what we call the *basic algorithm*, which is expected to work when the initial system is close to nonresonance. If that is not the case, we present in Section 4 a more general algorithm, consisting in repeated applications of the basic algorithm in Section 3. Finally, Section 5 contains numerical experiments showing that the new algorithm produces solutions comparable to those in [10], but more robustly and at a much lower computational cost.

2 Preliminaries

As mentioned in the Introduction, the idea of the algorithms in Sections 3 and 4 below is to modify the masses M and the stiffnesses K along directions in matrix space (M, K) such that the variation of the eigenvalues inside the resonance band is maximal, and the variation of the remaining eigenvalues is small. Actually, for simplicity we will do this in parameter space (m, k) instead, where $(m, k) = (m_1, \ldots, m_n, k_1, \ldots, k_n)$. We begin by reviewing some basic properties of derivatives of eigenvalues of symmetric matrices, our main tool to determine the direction of maximal variation. Then we briefly describe the bisection method, which will be employed by the algorithms in Sections 3 and 4 to monitor the crossing of eigenvalues through the boundaries of the resonance band.

2.1 Derivatives of eigenvalues of symmetric matrices

Let $\mathcal{S}_n$ be the set of real symmetric n by n matrices. We assume the eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ of any $A \in \mathcal{S}_n$ to be ordered decreasingly, i.e.

$$\lambda_1 \geq \ldots \geq \lambda_n. \tag{6}$$

For each $j = 1, \ldots, n$ we denote by

$$\begin{array}{rccc} \Lambda_j : & \mathcal{S}_n & \longrightarrow & \mathbb{R} \\ & A & \longrightarrow & \Lambda_j(A) = \lambda_j \end{array}$$

the operator on $\mathcal{S}_n$ which assigns to each matrix $A \in \mathcal{S}_n$ its i-th eigenvalue $\Lambda_j(A) = \lambda_j$ according to the order (6). Given $A \in \mathcal{S}_n$ with a *simple* eigenvalue $\lambda_j = \Lambda_j(A)$, it is well known [15, §II.6] that the corresponding operator Λ_j is differentiable at A, and the action of the differential $d\Lambda_j(A)$ on an arbitrary symmetric matrix E is given by

$$d\Lambda_j(A)(E) = \lim_{t \to 0} \frac{\Lambda_j(A + tE) - \Lambda_j(A)}{t} = u_j^T E u_j,$$

where u_j is any normalized eigenvector of A associated with the simple eigenvalue $\Lambda_j(A)$.

In our case, we are interested in the eigenvalues of the parameter-dependent matrix $J(M,K)$ given in (5). Therefore, the eigenvalues themselves can be written explicitly as a function of masses and stiffnesses

$$\begin{array}{rccc} \lambda_j: & \mathbb{R}^{2n} & \longrightarrow & \mathbb{R} \\ & (m,k)=(m_1,\ldots,m_n,k_1,\ldots,k_n) & \longrightarrow & \lambda_j=\lambda_j(m,k). \end{array}$$

Hence, each λ_j is just the composition $\Lambda_j \circ J$ of two differentiable operators: the operator Λ_j defined above and the matrix function J given in (5). Consequently, λ_j is differentiable, and the direction of maximal variation for the initial configuration $(\overline{m},\overline{k})$ is the direction of the gradient

$$\nabla\lambda_j(\overline{m},\overline{k}) = \left(\frac{\partial\lambda_j}{\partial m_1},\ldots,\frac{\partial\lambda_j}{\partial m_n},\frac{\partial\lambda_j}{\partial k_1},\ldots,\frac{\partial\lambda_j}{\partial k_n}\right)(\overline{m},\overline{k}).$$

Using the chain rule, we obtain, for instance,

$$\frac{\partial\lambda_j}{\partial m_i}(\overline{m},\overline{k}) = d\Lambda_j(\overline{J})\left(\frac{\partial J}{\partial m_i}(\overline{m},\overline{k})\right) = \overline{u}_j^T\frac{\partial J}{\partial m_i}(\overline{m},\overline{k})\overline{u}_j, \tag{7}$$

where $\overline{J}=J(\overline{m},\overline{k})=\overline{M}^{-1/2}\overline{K}\,\overline{M}^{-1/2}$, and likewise for the derivatives with respect to k_j and $\overline{u}_j$ is any normalized eigenvector of $\overline{J}$ associated with the simple eigenvalue $\Lambda_j(\overline{J})$. We stress that the partial derivatives of J with respect to both m_j and k_j can be easily computed by hand, and there are at most four nonzero elements in each derivative matrix. For instance,

$$\frac{\partial J}{\partial m_1} = \begin{bmatrix} -\frac{k_1+k_2}{m_1^2} & \frac{k_2}{2\sqrt{m_1^3 m_2}} & & & \\ \frac{k_2}{2\sqrt{m_1^3 m_2}} & 0 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & 0 & 0 \\ & & & 0 & 0 \end{bmatrix}.$$

Therefore, derivatives can be computed off-line, saving a considerable amount of arithmetic operations and, even if they must be computed online, each of them costs $O(1)$ flops once the eigenvectors $\overline{u}_j$ are known.

2.2 Bisection

Given a symmetric matrix A and any $x\in\mathbb{R}$, the number of eigenvalues of A which are smaller (resp. larger) than x is equal to the number of negative (resp. positive) eigenvalues of $A-xI$. Therefore, if $x_1<x_2$, the number of eigenvalues of A contained in the interval $(x_1,x_2]$ is equal to the difference between the number of positive eigenvalues of $A-x_1I$ and the number of positive eigenvalues of $A-x_2I$. This is the basis of the bisection method to compute the eigenvalues of symmetric matrices (see e.g. [7, §10]). If A is dense, finding the inertia of $A-xI$ via symmetric Gaussian elimination with pivoting costs $O(n^3)$, but

when A is symmetric *tridiagonal*, there is a much less expensive way of doing it: in that case the matrix $A - xI$ can be factorized as

$$A - xI = \begin{bmatrix} a_1 - x & b_1 & & \\ b_1 & a_2 - x & b_2 & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n - x \end{bmatrix} = LDL^T,$$

where $D = \mathrm{diag}(d_1, \ldots, d_n)$ is diagonal and L is unit lower bidiagonal. One can easily check that the diagonal elements of the matrix D in the factorization above can be recursively computed as

$$d_1 = a_1 - x$$

and

$$d_i = (a_i - x) - \frac{b_{i-1}^2}{d_{i-1}}, \quad i = 2, \ldots, n.$$

Since, due to Sylvester's Theorem, the inertia of D coincides with the inertia of $A - xI$, this means that the latter can be computed in just $O(n)$ arithmetic operations.

3 The basic isolation algorithm

As explained above, each eigenvalue λ_j of the generalized eigenvalue problem (4) is considered as a function of the data $(m, k) = (m_1, \ldots, m_n, k_1, \ldots, k_n) \in \mathbb{R}^{2n}$. In order to better understand the different stages of the algorithm we propose, it will be useful to illustrate some aspects of the general description in this section with a generic three-dimensional system (3D) of the form (4), say,

$$M = \mathrm{diag}(m_1, m_2, m_3), \qquad K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}, \tag{8}$$

where $m_i > 0$, $k_i > 0$, $i = 1, 2, 3$. In this setting, $(m, k) = (m_1, m_2, m_3, k_1, k_2, k_3) \in \mathbb{R}^6$. For the sake of simplicity, we assume that there is only one resonant eigenvalue, namely the middle one, λ_2, in the list (6) of eigenvalues decreasingly ordered.

Starting from the initial configuration $(\overline{m}, \overline{k})$, we intend to find a modification $(\delta m, \delta k)$ as small as possible such that all the eigenvalues are outside the resonance band for the modified data $(\overline{m} + \delta m, \overline{k} + \delta k)$. We do this by moving in space (m, k) along directions such that the variation of the eigenvalues *inside* the resonance band is maximal and the variation of eigenvalues *outside* the resonance band is minimal. For the sake of notational ease we denote by $\overline{\lambda_i}$ the eigenvalues corresponding to the initial configuration $(\overline{m}, \overline{k})$, and denote by I_{in} and I_{out} the subsets of $\{1, \ldots, n\}$ corresponding, respectively, to the eigenvalues inside and outside of the resonance band. In the 3D example above, for instance,

$$I_{\mathrm{in}} = \{2\}, \qquad I_{\mathrm{out}} = \{1, 3\}.$$

Thinking in terms of Taylor expansions, for each (m, k) close to $(\overline{m}, \overline{k})$ we have

$$\lambda_j(m, k) = \overline{\lambda}_j + < \nabla \lambda_j(\overline{m}, \overline{k}), (\delta m, \delta k) > + \ldots, \tag{9}$$

where $\delta m = m - \overline{m}$, $\delta k = k - \overline{k}$ and $< \cdot, \cdot >$ stands for the euclidean scalar product. Obviously, the maximal variation for each eigenvalue λ_i with $i \in I_{\text{in}}$ is achieved by taking $(\delta m, \delta k)$ equal to the gradient vector $\nabla\lambda_i(\overline{m}, \overline{k})$. However, even if there is only one eigenvalue inside the resonance band, this choice does not prevent some other eigenvalue from entering the resonance band. To avoid this, we will choose the increment $(\delta m, \delta k)$ in such a way that the first order term in the expansions (9) is zero for the eigenvalues outside the resonance region.

The main idea of the algorithm proposed below is to choose the new configuration (m, k) in such a way that it satisfies two conditions: on the one hand, the difference vector $(\delta m, \delta k)$ must belong to the orthogonal complement of the subspace spanned by all gradients $\nabla\lambda_j(\overline{m}, \overline{k})$, $j \in I_{\text{out}}$, i.e. by all gradients of eigenvalues outside the resonance band. Notice that there is always some leeway to do this: in the 3D example (8), for instance, the dimension of the orthogonal complement is 4, since the complement is with respect to the span of the two gradients, $\lambda_1(\overline{m}, \overline{k})$ and $\lambda_3(\overline{m}, \overline{k})$ in $\mathbb{R}^6$. On the other hand, $(\delta m, \delta k)$ should be chosen as the vector $w_{\max}$ in that orthogonal complement which maximizes, in some sense, the scalar products $< \nabla\lambda_i(\overline{m}, \overline{k}), w >$ for $i \in I_{\text{in}}$. Therefore, the algorithm can be organized in two stages:

- **Stage 1 (choice of direction):** Determine a unit vector $w_{\max}$ in

$$V^{\perp} = \{w \in \mathbb{R}^{2n} \ :< \nabla\lambda_j(\overline{m}, \overline{k}), w >= 0, \quad j \in I_{\text{out}}\} \tag{10}$$

  which maximizes, in some sense, the scalar products $< \nabla\lambda_j(\overline{m}, \overline{k}), w >$, $j \in I_{\text{in}}$.

  In the 3D example, this amounts to finding the unit vector $w_{\max}$ in the 4-dimensional subspace

$$V^{\perp} = \big(\text{span}\{\nabla\lambda_1(\overline{m}, \overline{k}),\ \nabla\lambda_3(\overline{m}, \overline{k})\}\big)^{\perp}$$

  of $\mathbb{R}^6$ attaining

$$\max\{< \nabla\lambda_2(\overline{m}, \overline{k}), w > :\ w \in V^{\perp}, \|w\| = 1\}.$$

- **Stage 2 (isolation):** Determine the first value α^* such that $J((\overline{m}, \overline{k}) + \alpha^* w_{\max})$ has exactly one of its eigenvalues on the boundary of the resonance band, and the remaining eigenvalues lie outside the band.

We now describe the implementation of each of the two stages.

Stage 1: Suppose there are q eigenvalues $\overline{\lambda_i}, \ldots, \overline{\lambda_{i+q-1}}$ inside the resonance band, i.e. $I_{\text{in}} = \{i, \ldots, i+q-1\}$. Then, the subspace $V^{\perp}$ defined in (10) is the orthogonal complement of the subspace

$$V_{n-q} = \text{span}\{\nabla\lambda_j(\overline{m}, \overline{k}) \ :\ j \in I_{\text{out}}\}.$$

We first specify in which sense the vector $w_{\max}$ will maximize the scalar products $< \nabla\lambda_k(\overline{m}, \overline{k}), w >$, $k = i, \ldots, i+q-1$: if for each $w \in V^{\perp}$ we set

$$\pi_k =< \nabla\lambda_k(\overline{m}, \overline{k}), w >, \quad k = i, \ldots, i+q-1$$

and denote

$$\pi = [\pi_i, \ldots, \pi_{i+q-1}]^T \in \mathbb{R}^q, \tag{11}$$

we will choose $w_{\max} \in V^\perp,\ \|w_{\max}\| = 1$ in such a way that the euclidean norm $\|\pi\|_2$ in $\mathbb{R}^q$ is maximized.

To do this, we first compute an orthonormal basis of $V^\perp$. We assume that the gradient vectors, which will usually be computed off-line[2], are stored in two different matrices, a matrix $\nabla_{\text{in}} \in \mathbb{R}^{2n\times q}$, with columns $\nabla\lambda_j(\overline{m},\overline{k}),\ j \in I_{\text{in}}$, and a matrix $\nabla_{\text{out}} \in \mathbb{R}^{2n\times(n-q)}$, with columns $\nabla\lambda_j(\overline{m},\overline{k}),\ j \in I_{\text{out}}$. In the 3D system (8) above, $\nabla_{\text{in}} \in \mathbb{R}^6$ has $\nabla\lambda_2(\overline{m},\overline{k})$ as its only column, while the columns of $\nabla_{\text{out}} \in \mathbb{R}^{6\times 2}$ are $\nabla\lambda_1(\overline{m},\overline{k})$ and $\nabla\lambda_3(\overline{m},\overline{k})$.

Thus, an orthonormal basis $B = \{w_1, \ldots, w_p\}$ of $V^\perp$ is computed via the function `basis`[3].

`function [W] = basis(`∇_{out}`)`

`INPUT:` $2n \times (n-q)$ matrix ∇_{out} with columns $\nabla\lambda_j(\overline{m},\overline{k}),\ j \in I_{\text{out}}$.
`OUTPUT:` $2n \times p$ matrix W whose columns form an orthonormal basis of $V^\perp$.

In the 3D system (8), the input to `basis` would be the 6×2 matrix ∇_{out}, and the output would be a 6×4 matrix whose columns are a basis of the subspace $V^\perp$.

The computational cost of the function `basis` is $O(n^3)$, corresponding to a $2n \times 2n$ QR factorization: we form a $2n \times 2n$ matrix S whose $n-q$ first columns are the columns of ∇_{out} (the $n+q$ last columns can be filled at random, or they can be just columns of identity). The last $2n - \text{rank}(\nabla_{\text{out}})$ columns of the Q factor in a QR factorization of S will form an orthonormal basis of $V^\perp$.

Once we have an orthonormal basis B of $V^\perp$, any vector $w \in V^\perp$ has unique coordinates $\alpha = [\alpha_1, \ldots, \alpha_p]^T$ with respect to B, and one can easily check that the vector π of scalar products in (11) corresponding to w can be written as

$$\pi = \Pi\,\alpha, \tag{12}$$

where $\Pi \in \mathbb{R}^{q\times p}$ is the matrix with

$$\pi_{k,l} = < \nabla\lambda_{i+k-1}(\overline{m},\overline{k}), w_l > \tag{13}$$

in the position (k,l) for each $k \in \{1,\ldots,q\},\ l \in \{1,\ldots,p\}$. Therefore, the problem of finding a unit vector $w_{\max} \in V^\perp$ maximizing $\|\pi\|_2$ is transformed into the problem of finding a vector $\alpha \in \mathbb{R}^p$ with unit norm maximizing $\|\Pi\,\alpha\|_2$. Clearly, the solution is $\alpha = v_{\max}(\Pi)$, the right singular vector of Π associated with its largest singular value $\sigma_{\max}(\Pi) = \|\Pi\|_2$. Hence, finding $w_{\max}$ amounts to computing the right singular vector corresponding to the largest singular value of the $q \times p$ scalar product matrix Π above.

In the simplest case $q = 1$, when just one eigenvalue $\overline{\lambda_i}$ is inside the resonance band, the quantity to maximize is simply the scalar product $< \nabla\lambda_i(\overline{m},\overline{k}), w >$. For any $w \in V^\perp$,

$$< \nabla\lambda_i(\overline{m},\overline{k}), w > = < P_\perp(\nabla\lambda_i(\overline{m},\overline{k})), w >,$$

[2]If not, see the function `in_out` in section 4.

[3]Generically, the dimension p of $V^\perp$ will be $p = 2n - (n-q) = n+q$, but we allow for linear dependencies among the gradient vectors

where $P_\perp$ stands for the orthogonal projector onto the subspace $V^\perp$. Therefore, the maximizer with unit euclidean norm when $q = 1$ is just

$$w_{\max} = \frac{P_\perp(\nabla\lambda_i(\overline{m}, \overline{k}))}{\|P_\perp(\nabla\lambda_i(\overline{m}, \overline{k}))\|}.$$

We summarize this discussion in the function `get_wmax`:

`function` [$w_{\max}$] = `get_wmax`(∇_{in}, W)

`INPUT:` $\nabla_{\text{in}} \in \mathbb{R}^{2n\times q}$ with columns $\nabla\lambda_j(\overline{m}, \overline{k})$, $j \in I_{\text{in}}$, $W \in \mathbb{R}^{2n\times p}$ whose columns form an orthonormal basis of $V^\perp$.

`OUTPUT:` $w_{\max} \in V^\perp$ maximizing the spectral norm of $\pi \in \mathbb{R}^q$ given by (11).

1. $q =$ `number of columns of` ∇_{in}
2. `if` $q = 1$ `then`
3. $\quad P_\perp = WW^T$
4. $\quad w_{\max} = P_\perp\nabla\lambda_i(\overline{m}, \overline{k})$
5. $\quad w_{\max} = \dfrac{w_{\max}}{\|w_{\max}\|}$
6. `else`
7. $\quad$ `form the matrix` $\Pi \in \mathbb{R}^{q\times p}$ `with entries given by` (13)
8. $\quad \alpha = v_{\max}(\Pi)$
9. $\quad w_{\max} = \displaystyle\sum_{i=1}^{p} \alpha_i w_i$
10. `end`

The most expensive steps in get_wmax are the SVD computation, and either forming the matrix Π if $q > 1$ or computing $P_\perp$ if $q = 1$. Forming Π costs $O(pqn)$, while forming $P_\perp$ costs $O(p^2n)$. Therefore, get_wmax is computationally inexpensive compared with the QR factorization performed in the function basis, since the dominant singular vector can be obtained at a cost significantly lower than $O(n^3)$ by any of the algorithms, like Lanczos in SVDPACK, for instance, specifically designed to compute a few selected singular values and vectors (see, for instance, the discussion in [2, §6.3.4]).

Stage 2: Since the implementation of this stage is somewhat involved, we describe first the simpler case when one single eigenvalue lies inside the resonance band.

2a) One single eigenvalue in the resonance band: Let $\overline{\lambda}_i$ be the only eigenvalue in $(c-r, c+r)$. If, for each $\alpha \in \mathbb{R}$, we define

$$(m_\alpha, k_\alpha) = (\overline{m}, \overline{k}) + \alpha\, w_{\max}, \qquad J_\alpha = J(m_\alpha, k_\alpha), \qquad \lambda_i(\alpha) = \lambda_i(J_\alpha), \tag{14}$$

the goal of this stage is to find a value α^* such that all eigenvalues of J_{α^*} are outside the resonance band, and $\lambda_i(\alpha^*)$ is on the boundary of the band. Notice first that the choice of α is constrained by the fact that all masses in m_α and all stiffnesses in k_α should remain positive. Therefore, if $\overline{m} = (m_j)_{j=1}^n$, $\overline{k} = (k_j)_{j=1}^n$ and $w_{\max} = (w_j)_{j=1}^{2n}$, then any positive value of α should be smaller than

$$\tau^+ = \min\left\{ \frac{m_i}{|w_i|}, \frac{k_j}{|w_{n+j}|} \ : \ i,j \in \{1,\dots,n\} \text{ such that } w_i < 0,\, w_{j+n} < 0 \right\}, \tag{15}$$

and the modulus of any negative value of α should be smaller than the positive quantity

$$\tau^- = \min\left\{ \frac{m_i}{w_i}, \frac{k_j}{w_{n+j}} \ : \ i,j \in \{1,\dots,n\} \text{ such that } w_i > 0,\, w_{j+n} > 0 \right\}. \tag{16}$$

These constrains lead to a minimal range for α, this is, $-\tau^- \le \alpha \le \tau^+$. For instance, suppose that the sign pattern of $w_{\max}$ in the $3D$ system (8) is $(+,+,-,-,+,-)^T$. Then,

$$\tau^+ = \min\left\{ \frac{m_3}{|w_3|}, \frac{k_1}{|w_4|}, \frac{k_3}{|w_6|} \right\}, \qquad \tau^- = \min\left\{ \frac{m_1}{w_1}, \frac{m_2}{w_2}, \frac{k_2}{w_5} \right\}.$$

Now, we must choose to drive $\lambda_i(\alpha)$ out of the band either through $c-r$ or through $c+r$. To decide which is best, we take two facts into account: first, whether the spectrum can be isolated at all through each particular side of the interval; second, the distance from $\overline{\lambda}_i$ to each side.

Hence, we first determine whether or not the limit values τ^+ and τ^- for α allow the spectrum to become isolated. To check this, we use the following routine, which, given $\overline{m}, \overline{k}, \alpha \in \mathbb{R}$ and an interval $(a,b]$, updates the matrix $J_\alpha = J(m_\alpha, k_\alpha)$ as in (14) and computes the number of eigenvalues of J_α in $(a,b]$ using the procedure described in §2.2.

`function [`N`] = N_eig(`$\overline{m}, \overline{k}, w, \alpha, a, b$`)`

`INPUT:` masses and stiffnesses $\overline{m}, \overline{k}$, direction $w \in \mathbb{R}^{2n}$, scalars $\alpha \in \mathbb{R}$, $a, b \in \mathbb{R}$ with $a < b$.

`OUTPUT:` number of eigenvalues of $J_\alpha = J(\overline{m}, \overline{k}) + \alpha\, w$ contained in the interval $(a,b]$.

Since both updating J_α and computing inertias cost $O(n)$, this routine also costs $O(n)$ flops.

To have any chance of isolating the spectrum we need either

$$N^+ = N\_eig(\overline{m}, \overline{k}, w_{\max}, \tau^+, c-r, c+r) \quad \text{or} \quad N^- = N\_eig(\overline{m}, \overline{k}, w_{\max}, -\tau^-, c-r, c+r) \tag{17}$$

to be zero (if both are nonzero, then this basic algorithm should be replaced by one of the continuation algorithms in Section 4). If only one of them is zero, then the sign of α^* is determined: if, for instance, $N^+ = 0$ and $N^- \neq 0$, then we must find a positive critical value α^* such that $\lambda_i(\alpha^*)$ is an endpoint of the resonance band. If $N^+ \neq 0$ and $N^- = 0$, then we look for a negative critical value α^*. To approximate the value of α^* we use the following bisection routine: given an initial range $(\alpha_{\min}, \alpha_{\max})$ and a tolerance *tol*, the procedure locates the critical value α^* within an absolute error of size *tol*, provided that $\alpha_{\min}$ and $\alpha_{\max}$ satisfy what we may call the *bisection condition*, namely that one of the two evaluations of $\mathrm{N\_eig}(\overline{m}, \overline{k}, w_{\max}, \cdot, c-r, c+r)$ at $\alpha_{\min}$ and $\alpha_{\max}$ is zero, and the other one is nonzero.

`function` $[\alpha^*]$ `= bisect(`$\overline{m}, \overline{k}, w, \alpha_{\min}, \alpha_{\max}, tol$`)`

`INPUT:` masses and stiffnesses $\overline{m}, \overline{k}$, direction $w \in \mathbb{R}^{2n}$, scalars $\alpha_{\min}, \alpha_{\max} \in \mathbb{R}$ with $\alpha_{\min} < \alpha_{\max}$ satisfying the bisection condition, tolerance $tol > 0$.

`OUTPUT:` α^* such that $\lambda_i(\alpha)$ is on the boundary of the resonance band and the remaining eigenvalues $\lambda_k(\alpha)$ are outside the band.

1. $a = \alpha_{\min}, \quad b = \alpha_{\max}$
2. `while` $b - a > tol$
3. $\quad \alpha = \dfrac{a+b}{2}$
4. $\quad$ `if` $N\_eig(\overline{m}, \overline{k}, w, a, c-r, c+r) = 0$ `then`
5. $\quad\quad$ `if` $N\_eig(\overline{m}, \overline{k}, w, \alpha, c-r, c+r) = 0$ `then`
6. $\quad\quad\quad a = \alpha$
7. $\quad\quad$ `else` $b = \alpha$
8. $\quad\quad$ `end`
9. $\quad$ `else`
10. $\quad\quad$ `if` $N\_eig(\overline{m}, \overline{k}, w, \alpha, c-r, c+r) = 0$ `then`
11. $\quad\quad\quad b = \alpha$
12. $\quad\quad$ `else` $a = \alpha$
13. $\quad\quad$ `end`
14. $\quad$ `end`
15. `end`

If, as we assumed above, α^* is positive, then we apply bisect with $\alpha_{\min} = 0$ and $\alpha_{\max} = \tau^+$. If α^* is negative, then we apply bisect with $\alpha_{\min} = -\tau^-$ and $\alpha_{\max} = 0$. When the procedure stops we end up with an interval of maximal length *tol* containing the critical value α^*. In view of (9), an absolute error of size *tol* in α amounts roughly to an absolute error of size $< \nabla\lambda_i(\overline{m}, \overline{k}), w_{\max} > \cdot tol$ in the eigenvalue, so if one wants to locate the eigenvalue more precisely, it is enough to reduce the tolerance *tol* accordingly.

Finally, in case both quantities N^+ and N^- are zero, we denote by

$$\pi_i^{\max} =< \nabla\lambda_i(\overline{m}, \overline{k}), w_{\max} > \tag{18}$$

the approximate speed at which $\lambda_i(\alpha)$ moves as α varies. We further define

$$d^+ = \begin{cases} c + r - \overline{\lambda_i} & \text{if} \quad \pi_i^{\max} \geq 0 \\ \overline{\lambda_i} - c + r & \text{if} \quad \pi_i^{\max} < 0 \end{cases}, \qquad d^- = 2r - d^+, \quad d_{\min} = \min\{d^+, d^-\}. \tag{19}$$

The quantity d^+ (resp. d^-) is the distance which $\lambda_i(\alpha)$ must travel to reach the boundary of the resonance band for positive α (resp. for negative α). Hence, to minimize re-design, α^* should be taken as positive if $d_{\min} = d^+$, and as negative if $d_{\min} = d^-$. Once the sign of α^* is known, it only remains to apply the bisection procedure on the appropriate initial range: on $(0, \tau^+)$ if $\alpha^* > 0$, or $(-\tau^-, 0)$ otherwise.

2b) Several eigenvalues in the resonance band: Suppose now that $I_{\text{in}} = \{i, \ldots, i+q-1\}$ for arbitrary q. As in the case $q = 1$, the possible values for α are constrained to vary within the thresholds $-\tau^-$ and τ^+ defined in (16) and (15). Also as in the case $q = 1$, to have any chance of isolating the spectrum we need one of the two quantities N^+ and N^- in (17) to be zero. If only one of them is zero, then the procedure is identical to the one above. The only difference appears when $N^+ = N^- = 0$. In that case we define for each $k \in \{i, \ldots, i+q-1\}$

$$\pi_k^{\max} =< \nabla\lambda_k(\overline{m}, \overline{k}), w_{\max} >, \qquad d_k^+ = \begin{cases} c + r - \overline{\lambda_k} & \text{if} \quad \pi_k^{\max} \geq 0 \\ \overline{\lambda_k} - c + r & \text{if} \quad \pi_k^{\max} < 0 \end{cases}, \quad d_k^- = 2r - d_k^+,$$

and set

$$d^+ = \sum_{k=i}^{i+q-1} d_k^+, \qquad d^- = \sum_{k=i}^{i+q-1} d_k^-, \qquad d_{\min} = \min\{d^+, d^-\}. \tag{20}$$

Each quantity d_k^+ (resp. d_k^-) is the distance which $\lambda_k(\alpha)$ must travel to reach the boundary of the resonance band when α is taken as positive (resp. when α is taken as negative). Given a choice of sign for α, some of the eigenvalues will move to the right, and some others to the left. Since d^+ and d^- give a collective measure of the distance which the eigenvalues need to travel for each sign choice to get out of the resonance band, α^* should be taken as positive if $d_{\min} = d^+$, and negative if $d_{\min} = d^-$.

We can summarize the description above in the following pseudocode for the overall algorithm, which makes use of all three subroutines basis, get_wmax and bisect:

`Algorithm 1:` Basic frequency isolation algorithm

`INPUT:` initial masses and stiffnesses $\overline{m}, \overline{k} \in \mathbb{R}^{n\times n}$, index sets $I_{\text{in}}, I_{\text{out}}$, center c and radius r of resonance band, matrices $\nabla_{\text{in}} \in \mathbb{R}^{2n\times q}$, $\nabla_{\text{out}} \in \mathbb{R}^{2n\times(2n-q)}$, tolerance tol for the bisection procedure.

`OUTPUT:` boolean variable ISO, taking the value 1 if the spectrum has been isolated, and 0 otherwise; whenever ISO = 1, a new system (m^*, k^*) with its spectrum isolated from the resonance band and one of its eigenvalues on the boundary of the resonance band.

1. $W =$ `basis`(∇_{out})
2. $w_{\max} =$ `get_wmax`(∇_{in}, W)
3. `compute` τ^+, τ^- `as in` (15), (16)
4. $N^+ = N\_eig(\overline{m}, \overline{k}, w_{\max}, \tau^+, c-r, c+r)$
5. $N^- = N\_eig(\overline{m}, \overline{k}, w_{\max}, -\tau^-, c-r, c+r)$
6. `if` $N^+ \neq 0$ `and` $N^- \neq 0$ `then`
7. `ISO` $= 0$
8. `write 'The spectrum cannot be isolated'`
9. `else`
10. `ISO` $= 1$
11. `if` $N^+ = 0$ `and` $N^- \neq 0$ `then`
12. $\alpha^* =$ `bisect`$(\overline{m}, \overline{k}, w_{\max}, 0, \tau^+, tol)$
13. `if` $N^+ \neq 0$ `and` $N^- = 0$ `then`
14. $\alpha^* =$ `bisect`$(\overline{m}, \overline{k}, w_{\max}, -\tau^-, 0, tol)$
15. `if` $N^+ = 0$ `and` $N^- = 0$ `then`
16. `if` $q = 1$ `then`
17. `compute` $\pi_i^{\max}$ `as in` (18)
18. `compute` $d^+, d^-, d_{\min}$ `as in` (19)
19. `if` $d_{\min} = d^+$ `then`

20. $\alpha^* =$ `bisect`$(\overline{m}, \overline{k}, w_{\max}, 0, \tau^+, tol)$

21. `else`

22. $\alpha^* =$ `bisect`$(\overline{m}, \overline{k}, w_{\max}, -\tau^-, 0, tol)$

23. `end`

24. `else`

25. `compute` $\pi_k^{\max}$ `for each` $k \in I_{\text{in}}$ `as in` (18)

26. `compute` $d^+, d^-, d_{\min}$ `as in` (20)

27. `if` $d_{\min} = d^+$ `then`

28. $\alpha^* =$ `bisect`$(\overline{m}, \overline{k}, w_{\max}, 0, \tau^+, tol)$

29. `else`

30. $\alpha^* =$ `bisect`$(\overline{m}, \overline{k}, w_{\max}, -\tau^-, 0, tol)$

31. `end`

32. `end`

33. `update` $(m^*, k^*) = (\overline{m}, \overline{k}) + \alpha^* \, w_{\max}$

34. `end`

35. end

To end this section, we just show the performance of the basic isolation algorithm on a couple of specific instances of 3D systems like (8). This will reveal some of the limitations of the basic algorithm: consider a system with initial parameters

$$\begin{aligned} &\overline{m}_1 = 10, \quad \overline{m}_2 = 5, \quad \overline{m}_3 = 3, \\ &\overline{k}_1 = 3, \qquad \overline{k}_2 = 5, \quad \overline{k}_3 = 1. \end{aligned} \tag{21}$$

One can check that the corresponding initial eigenvalues are

$$\overline{\lambda}_1 = 1.76480917, \qquad \overline{\lambda}_2 = 0.43963768, \qquad \overline{\lambda}_3 = 0.12888647.$$

If we first take the interval $[0.4, 0.48]$ as the resonance band, and run the basic isolation algorithm on this system, it finds a new configuration

$$\begin{aligned} &m_1 = 9.65874873, \quad m_2 = 3.26258156, \quad m_3 = 2.81417415, \\ &k_1 = 6.06306403, \quad k_2 = 5.21917707, \quad k_3 = 0.33501957, \end{aligned}$$

which is away from the initial one by a relative distance of

$$\frac{\|(m,k)-(\overline{m},\overline{k})\|_\infty}{\|(\overline{m},\overline{k})\|_\infty} \approx 3\times 10^{-1},$$

measured in the infinity (i.e., max) norm, and has associated eigenvalues

$$\lambda_1 = 2.40601328, \qquad \lambda_2 = 0.48000000, \qquad \lambda_3 = 0.10351228.$$

If we enlarge the resonance band to $[0.3, 0.5]$, however, it turns out that this basic algorithm is unable to isolate, because both quantities N^+ and N^- turn out to be nonzero, i.e., either some mass or some stiffness becomes negative before we are able to isolate λ_2 as we move along the direction $w_{\max}$ in (m,k)-space. In the next section we will see a possible way to circumvent this difficulty.

4 A continuation algorithm

Algorithm 1 is a simple and inexpensive way of solving the frequency isolation problem. However, we cannot expect it to provide solutions close to the optimal in general, due to two shortcomings: first, it does not even guarantee the isolation of the spectrum, since the thresholds τ^+, τ^- may prevent the algorithm from going all the way to isolation (this is precisely what happened when we enlarged the resonance in the 3D experiment above). The second shortcoming is that, even if it isolates the spectrum, Algorithm 1 is unlikely to produce designs close to optimal unless the initial setting $(\overline{m},\overline{k})$ is close enough to some optimal configuration: first order Taylor approximations see only local information in a neighborhood of the initial design $(\overline{m},\overline{k})$, so we can only expect the approximation to be accurate for sufficiently small values of $(\delta m, \delta k)$.

These two difficulties may be overcome by taking Algorithm 1 as the basis for an iterative algorithm: instead of trying to isolate the spectrum in one single run of Algorithm 1, the idea is to repeat the procedure over and over again, updating (m_α, k_α) as in (14), but with eventually smaller values of α. This should make the Taylor approximation more accurate and, hopefully, the resulting configuration closer to the initial one. Also, in those cases in which Algorithm 1 is unable to isolate the spectrum, iterating allows us to move a certain distance in the chosen direction, and then try either to isolate, or to move closer to isolation in the next iteration.

To describe this continuation procedure, suppose that the i-th step of the iteration has been performed, and a configuration (m_i, k_i) (correspondingly, (M_i, K_i) in matrix space) has been obtained such that the spectrum of $(K_i - \lambda M_i)x = 0$ is not yet isolated. In the $(i+1)$-th step of the continuation procedure we compute the thresholds τ_i^-, τ_i^+ corresponding to the configuration (m_i, k_i), as well as the direction $w_{\max}^{(i)}$, analogous to $w_{\max}$, of maximal variation for those eigenvalues of $(K_i - \lambda M_i)x = 0$ which lie inside the resonance band. Then the output of the $(i+1)$-th step is

$$(m_{i+1}, k_{i+1}) = (m_i, k_i) + h_i\, w_{\max}^{(i)}, \tag{22}$$

for some appropriate $h_i \in (-\tau_i^-, \tau_i^+)$. If the spectrum associated with (m_{i+1}, k_{i+1}) is isolated, the algorithm stops. Otherwise, step $i+2$ follows.

Many sensible choices are possible for h_i, each one giving rise to a different kind of continuation algorithm: maybe the simplest choice for h_i is to consider a constant step $h > 0$ and take h_i equal to either $\min\{h, \tau_i^+\}$ or $-\min\{h, \tau_i^-\}$, depending on which side of the resonance band is closer to the eigenvalues we are trying to move. To be more precise, if we want to preserve the n masses strictly positive, then we should take $h_i = \tau_i^+ - \delta$, or $h_i = -\tau_i^- + \delta$, for some small $\delta > 0$, which we may call the *positivity margin*.

If h is sufficiently small, we can expect the computed solutions to be close to the optimal, since we are moving at each step along directions which are locally close to the optimal ones. This choice also keeps the computational cost low, since it makes no use of the bisection procedure bisect. This gain, however, is only marginal, since the most expensive part of each step, namely computing the basis of $V^\perp$, must be performed anyway. Therefore, each step of the continuation procedure costs $O(n^3)$ flops.

A brief description of this algorithm might be as follows: since the gradients ∇_{in} and ∇_{out} must be recomputed at each iteration, we may no longer assume that I_{in}, I_{out}, ∇_{in} or ∇_{out} are part of the input. Therefore, we need a function, say in_out, which, given (m, k) and the resonance band, computes eigenvalues and eigenvectors of the corresponding matrix pencil $(K - \lambda M)x = 0$ (equivalently, of the symmetric tridiagonal matrix $M^{1/2}KM^{1/2}$), determines I_{in} and I_{out}, and computes the gradient matrices ∇_{in} and ∇_{out} according to (7).

`function` $[(I_{\text{in}}, I_{\text{out}}, \nabla_{\text{in}}, \nabla_{\text{out}})]$ `= in_out(`m, k, c, r`)`

`INPUT:` masses and stiffnesses m, k, center c and radius r of the resonance band.

`OUTPUT:` subsets I_{in} and I_{out}, matrices ∇_{in} and ∇_{out}.

Since eigenvalues and eigenvectors of a symmetric tridiagonal matrix can be computed at cost $O(n^2)$ using the MRRR method devised by Dhillon and Parlett [8, 9], the computational cost of in_out is just $O(n^2)$.

Once we have this function, a pseudocode for the continuation algorithm may be as follows:

`Algorithm 2:` Continuation algorithm for frequency isolation (constant step version)

`INPUT:` initial masses and stiffnesses $\overline{m}, \overline{k} \in \mathbb{R}^{n\times n}$, center c and radius r of resonance band, step $h > 0$, positivity margin $\delta > 0$.

`OUTPUT:` masses and stiffnesses m, k such that all eigenvalues of the corresponding eigenproblem $(K - \lambda M)x = 0$ lie outside the resonance band, with one of them on the boundary.

1. $m = \overline{m}, \;\; k = \overline{k}$
2. `ISO` $= 0$

3. `while ISO` $= 0$ `do`

4. $(I_{\text{in}}, I_{\ \text{out}}, \nabla_{\text{in}}, \nabla_{\ \text{out}}) =$ `in_out`(m, k, c, r)

5. $W =$ `basis`(∇_{out})

6. $w_{\max} =$ `get_wmax`$(\nabla_{\ \text{in}}, W)$

7. `compute` τ^+, τ^- `as in` (15), (16)

8. `if` $q = 1$ `then`

9. `compute` $\pi_i^{\max}$ `as in` (18)

10. `compute` $d^+, d^-, d_{\min}$ `as in` (19)

11. `if` $d_{\min} = d^+$ `then`

12. $\alpha = \min\{h, \tau^+ - \delta\}$

13. `else`

14. $\alpha = -\min\{h, \tau^- - \delta\}$

15. `end`

16. `else`

17. `compute` $\pi_k^{\max}$ `for each` $k \in I_{\text{in}}$ `as in` (18)

18. `compute` $d^+, d^-, d_{\min}$ `as in` (20)

19. `if` $d_{\min} = d^+$ `then`

20. $\alpha = \min\{h, \tau^+ - \delta\}$

21. `else`

22. $\alpha = -\min\{h, \tau^- - \delta\}$

23. `end`

24. `end`

25. `update` $(m, k) = (m, k) + \alpha\, w_{\max}$

26. $N =$ `N_eig`$(m, k, w_{\max}, 0, c - r, c + r)$

27. `if` $N = 0$ `then ISO` $= 1$

28. `end`

29. `end`

If we run this algorithm on the 3D system (21) with the resonance band $[0.3, 0.5]$ (which the basic algorithm was unable to isolate), we obtain isolation after 14 iterations. The new masses and stiffnesses are

$$m_1 = 10.09525531, \quad m_2 = 5.52036200, \quad m_3 = 2.71740738,$$
$$k_1 = 2.44412779, \quad k_2 = 4.92615710, \quad k_3 = 1.18313460,$$

which are away from the initial ones by a relative distance

$$\frac{\|(m,k) - (\overline{m}, \overline{k})\|_\infty}{\|(\overline{m}, \overline{k})\|_\infty} \approx 6 \times 10^{-2}.$$

The new eigenvalues are

$$\lambda_1 = 1.65435229, \qquad \lambda_2 = 0.50526253, \qquad \lambda_3 = 0.11253344.$$

Notice that, as expected, the continuation algorithm is able to isolate in some situations where the basic one is not, although at the price of a significant computational load (in this case, fourteen iterations).

Although this algorithm may be convenient to produce solutions close to the initial design, the example above shows that it may take many iterates to isolate the spectrum if h is too small; and, in general, it is not easy to guess an appropriate size for h, except by trial and error. Therefore, it may make sense to somehow link the size of h_i with the value α_i^* obtained from the bisection procedure at the i-th iteration. The simplest possibility is to choose at each step the largest possible value for h_i, trying to isolate the spectrum as soon as we can: if the spectrum can be isolated, we take $h_i = \alpha_i^*$, update m and k accordingly, and the algorithm stops. If it cannot be isolated, we proceed to the next iteration after updating m and k as in (22), with $h_i = \tau_i^+ - \delta$ if $\tau_i^+ \geq \tau_i^-$, and $h_i = -\tau_i^- + \delta$ otherwise (δ is the positivity margin). This "greedy" version will usually isolate in much fewer iterations than Algorithm 2, but the final configuration might be farther from the initial one than the one obtained from Algorithm 2.

`Algorithm 3:` Continuation algorithm for frequency isolation (greedy version)

`INPUT:` initial masses and stiffnesses $\overline{m}, \overline{k} \in \mathbb{R}^{n \times n}$, center c and radius r of resonance band, tolerance tol for the bisection procedure, positivity margin $\delta > 0$.

`OUTPUT:` masses and stiffnesses m, k such that all eigenvalues of the corresponding matrix pencil $(K - \lambda M)x = 0$ lie outside the resonance band, with one of them on the boundary.

1. $m = \overline{m}, \;\; k = \overline{k}$
2. `ISO` $= 0$

3. `while ISO` $= 0$ `do`

4. $(I_{\text{in}}, I_{\text{ out}}, \nabla_{\text{in}}, \nabla_{\text{ out}}) =$ `in_out`(m, k, c, r)

5. $W =$ `basis`(∇_{out})

6. $w_{\max} =$ `get_wmax`$(\nabla_{\text{ in}}, W)$

7. `compute` τ^+, τ^- `as in` (15), (16)

8. `if` $q = 1$ `then`

9. `compute` $\pi_i^{\max}$ `as in` (18)

10. `compute` $d^+, d^-, d_{\min}$ `as in` (19)

11. `else`

12. `compute` $\pi_k^{\max}$ `for each` $k \in I_{\text{in}}$ `as in` (18)

13. `compute` $d^+, d^-, d_{\min}$ `as in` (20)

14. `end`

15. $N^+ = N\_eig(m, k, w_{\max}, \tau^+, c - r, c + r)$

16. $N^- = N\_eig(m, k, w_{\max}, -\tau^-, c - r, c + r)$

17. `if` $N^+ \neq 0$ `and` $N^- \neq 0$ `then`

18. `if` $d_{\min} = d^+$ `then`

19. $\alpha = \tau^+ - \delta$

20. `else`

21. $\alpha = -\tau^- + \delta$

22. `end`

23. `else`

24. `ISO` $= 1$

25. `if` $N^+ = 0$ `and` $N^- \neq 0$ `then`

26. $\alpha =$ `bisect`$(m, k, w_{\max}, 0, \tau^+, tol)$

27. `if` $N^+ \neq 0$ `and` $N^- = 0$ `then`

28. $\alpha =$ `bisect`$(m, k, w_{\max}, -\tau^-, 0, tol)$

29. `if` $N^+ = 0$ `and` $N^- = 0$ `then`

30. `if` $d_{\min} = d^+$ `then`

31. $\alpha =$ `bisect`$(m, k, w_{\max}, 0, \tau^+, tol)$

32. `else`

33. $\alpha =$ `bisect`$(m, k, w_{\max}, -\tau^-, 0, tol)$

34. `end`

35. `end`

36. `update` $(m, k) = (m, k) + \alpha\, w_{\max}$

37. `end`

38. `end`

A different possibility for choosing h_i corresponds to not trying to isolate at each iteration, and advancing only a portion of α_i^* in the direction of $w_{\max}^{(i)}$. For instance, we may take

$$h_i = \begin{cases} \dfrac{\alpha_i^*}{n-i+1} & \text{for } i = 1, \ldots, n, \\ \alpha_i^* & \text{for } i > n \end{cases}, \tag{23}$$

i.e., we choose a small portion of α_i^* in the first iterates, and the portion gradually increases up to the n-th iteration. Notice that in doing so we artificially prevent the algorithm from isolating the spectrum up to the n-th iteration, where n is the dimension of the system. In other words, we are forcing the algorithm to cost $O(n^4)$ operations. The pseudocode for this version would be identical to the one for Algorithm 3, only adding the choice (23) for h_i right before step 36.

Going back to our benchmark example, if we now run this greedy Algorithm 3 on the 3D system (21) with resonance band $[0.3, 0.5]$, isolation is obtained after just one single iteration. The new masses and stiffnesses are

$$\begin{aligned} &m_1 = 10.30160347, \quad m_2 = 6.53555896, \quad m_3 = 3.16423593, \\ &k_1 = 0.29281367, \quad\ \ k_2 = 4.80628769, \quad\ \ k_3 = 1.58772062, \end{aligned}$$

which are away from the initial ones by a relative distance

$$\frac{\|(m, k) - (\overline{m}, \overline{k})\|_\infty}{\|(\overline{m}, \overline{k})\|_\infty} \approx 3 \times 10^{-1},$$

and the new eigenvalues are

$$\lambda_1 = 1.46073263, \qquad \lambda_2 = 0.5, \qquad \lambda_3 = 0.01436078.$$

This shows that the main difference between the performance of Algorithms 2 and 3 is that the latter will typically isolate at a significantly lower computational cost than Algorithm 2, although the final design we obtain is likely to be farther from the initial design than the one we would obtain from Algorithm 2.

5 Numerical experiments

In this section we report the numerical performance of Algorithms 1, 2 and 3 in solving the frequency isolation problem for undamped mass-spring systems. Extensive numerical tests have been performed, showing that the algorithms are considerably faster and more robust than previous algorithms solving the same problem. All numerical tests were done using Fortran Power Station 4.0 on a Pentium 4 XP 2000+ processor, 2.6 GHz, 1 GB RAM with IEEE double precision arithmetic.

For each dimension we have generated 1000 random $n \times n$ mass-spring systems with one single eigenvalue in the resonance band in order to compare the solutions obtained either by the *basic* algorithm (Algorithm 1), or by the *continuation* algorithm – *greedy version* – (Algorithm 3), with the solutions provided by the *Box algorithm* due to Egaña et al. [10]. We chose the greedy version over the one with constant steps (Algorithm 2) since, as shown in Tables 3 and 4 below, the greedy version is considerably faster.

To generate each random system, we begin by generating both the eigenvalues $\{\lambda_i\}_{i=1}^n$ and the interlaced spectrum $\{\mu_i\}_{i=1}^{n-1}$, satisfying the interlacing property $\lambda_i < \mu_i < \lambda_{i+1}, i = 1, 2, \ldots, n-1$. The eigenvalues λ_i are either *evenly distributed* (ED) or *randomly distributed* (RD), the latter distribution being employed only in §5.2 (the Box algorithm is unable to cope with such initial configurations). The ED distribution corresponds to choosing

$$\lambda_1 = r_1, \quad \lambda_i = \lambda_{i-1} + r_{i-1}, \quad i = 2, 3, \ldots, n,$$

where $\{r_i\}_1^n$ are random numbers generated by the Fortran *rand* routine. The interlaced spectrum is then chosen as

$$\mu_i = \frac{\lambda_i + \lambda_{i+1}}{2}, \quad i = 1, 2, \ldots, n-1.$$

The RD distribution simply amounts to generating $2n-1$ random numbers r_i via the Fortran *rand* routine, and then setting $\lambda_i = r_i$ for odd i, and $\mu_i = r_i$ if i is even.

Once we have the λs and the μs, and the total mass is fixed (see below its particular value in each case), the *fast orthogonal reduction method* [3] reconstructs the tridiagonal matrix J in (5), from which the corresponding masses and stiffnesses can be easily recovered. This leads to the random initial configuration $(\overline{m}, \overline{k})$.

All experiments have been done with one single resonant eigenvalue, namely the fifth one, which is also the center of the resonance band $(\lambda_5 - r, \lambda_5 + r)$. The radius r of the resonance band is described through a parameter η, inversely proportional[4] to r: the smaller η, the larger the radius r.

The $2n$-vector (m, k) lists the values of masses and stiffnesses in the initial system, generated randomly for each value of η. The solutions computed by the Box algorithm are denoted by $(m^*, k^*)_{box}$, while the $2n$-vectors $(m^*, k^*)_{basic}$ and $(m^*, k^*)_{cont}$ denote the solutions computed by the basic and the continuation isolation algorithm, respectively.

[4]To be precise, if $d = \min\{\lambda_5 - \lambda_4, \lambda_6 - \lambda_5\}$ is the smallest distance from the resonant eigenvalue λ_5 to its closest neighbour, the radius r of the resonance band is related to η through the equality $r = d/\eta$.

5.1 Comparison of the Box algorithm with the basic and the continuation isolation algorithms

For each dimension, we generate 1000 random $n \times n$ mass-spring systems with evenly distributed (ED) initial spectrum. Also, since the cost of the Box algorithm allows it to work only for low dimension, we consider systems of dimension $n = 6, 8, 10$, and 12. For each n, the total mass of each system is set to 10^4. In all numerical tests the 5-th eigenvalue is the center of the resonance band, and the radius r of the band depends on the value of the parameter η (see the footnote above).

In order to compare the performance of the two algorithms, we make use of the following control quantities:

1) the relative distance from the initial configuration

$$Dist_{m,k}^{(\blacktriangle)} = \frac{\left\| \left(\overline{m}, \overline{k}\right) - (m^*, k^*)_{\blacktriangle} \right\|_\infty}{\left\| \left(\overline{m}, \overline{k}\right) \right\|_\infty}, \tag{24}$$

where ($\blacktriangle$) is either *box*, *basic* or *greed*, where *greed* denotes the greedy version of the continuation algorithm.

2) the distance quotients

$$Q_{m,k}^{(basic)} = \frac{Dist_{m,k}^{(box)}}{Dist_{m,k}^{(basic)}}, \tag{25}$$

$$Q_{m,k}^{(greed)} = \frac{Dist_{m,k}^{(box)}}{Dist_{m,k}^{(greed)}}. \tag{26}$$

3) the time quotients

$$Q_T^{(basic)} = \frac{\text{CPU time for Box alg.}}{\text{CPU time for basic isolation alg.}}, \tag{27}$$

$$Q_T^{(greed)} = \frac{\text{CPU time for Box alg.}}{\text{CPU time for cont. isolation alg.}}, \tag{28}$$

where CPU time is measured in seconds, using the Fortran *dtime* routine.

4) the percentage $\pi_{m,k}^{(basic)}$ (resp. $\pi_{m,k}^{(greed)}$) of cases in which $Q_{m,k}^{(basic)}$ (resp. $Q_{m,k}^{(greed)}$) is larger than 1. We omit the percentages $\pi_T^{(\blacktriangle)}$ of cases in which $Q_T^{(\blacktriangle)}$ is larger than 1 since, as can be read from the minima, in both tables the time quotients are larger than one in every single experiment.

5) the percentage $\pi_{isol}^{(basic)}$ (resp. $\pi_{isol}^{(greed)}$) of cases in which the basic (resp. the continuation) algorithm successfully isolates the resonant eigenvalue.

Table 1 shows the error and time quotients $Q_{m,k}^{(basic)}$ and $Q_T^{(basic)}$ for Algorithm 1 (basic isolation), while Table 2 shows the error and time quotients $Q_{m,k}^{(greed)}$ and $Q_T^{(greed)}$ for Algorithm 3 (continuation algorithm, greedy version). Both Tables are organized as follows: the first and second column display the dimension n and the parameter η, respectively. To display the error quotient $Q_{m,k}^{(\blacktriangle)}$ there are three columns: the first one displays the average over the 1000 tests made; the second and third display, respectively, the smallest and largest value for the error quotient among the 1000 experiments. The following two columns display the percentages $\pi_{m,k}^{(\blacktriangle)}$ and $\pi_{isol}^{(\blacktriangle)}$, respectively. For the time quotient $Q_T^{(\blacktriangle)}$ there are three columns, as before, displaying the average over the 1000 tests, the minimal and the maximal value for the time quotient.

			$Q_{m,k}^{(basic)}$					$Q_T^{(basic)}$	
n	η	mean	min	max	$\pi_{m,k}^{(basic)}$	$\pi_{isol}^{(basic)}$	mean	min	max
6	3	3.39	2.66	4.12	100	4	15.99	3.99	6.25e13
	10	3.47	9.77e-1	16.21	98.66	75	33.16	1.99	9.37e13
	20	1.93	3.54e-1	11.57	88.04	92	44.24	1.99	7.81e13
8	3	3.36	1.34	8.08	100	57.99	87.23	12.99	4.53e14
	10	1.23	1.92e-1	3.42	84.04	94	67.69	12.99	3.59e14
	20	9.01e-1	1.69e-1	2.87	72.44	98	78.04	12.99	4.68e14
10	3	2.43	9.84e-1	5.48	98.38	62	259.47	43.99	2.73e15
	10	7.44e-1	7.86e-1	2.01	52.52	99	502.46	61.99	4.59e15
	20	5.40e-1	4.83e-2	1.84	36.36	99	531.47	68.99	4.67e15
12	3	1.67	5.25e-1	7.70	91.46	82	1185.19	399.99	1.52e16
	10	6.55e-1	1.05e-1	2.61	39.17	97	900.36	268.99	1.04e16
	20	4.91e-1	1.08e-1	2.59	32.32	99	754.52	271.99	1.10e16

Table 1: Numerical results: comparison of the Box and the basic isolation algorithms.

			$Q_{m,k}^{(greed)}$					$Q_T^{(greed)}$	
n	η	mean	min	max	$\pi_{m,k}^{(greed)}$	$\pi_{isol}^{(greed)}$	mean	min	max
6	3	9.52	1.25	38.44	100	100	37.99	2.99	7.81e13
	10	2.96	3.72e-1	11.05	96	100	33.99	2.99	9.37e13
	20	1.75	1.84e-1	7.80	87	100	25.86	1.99	9.37e13
8	3	2.60	2.54e-1	11.86	94	100	95.28	16.99	4.68e14
	10	9.18e-1	1.92e-1	3.32	64	100	83.29	10.99	3.59e14
	20	8.44e-1	2.28e-1	2.67	64	100	97.66	12.99	4.68e14
10	3	1.25	1.68e-1	8.58	74	100	235.60	47.99	2.75e15
	10	6.04e-1	3.98e-2	2.01	55	100	446.90	84.99	4.60e15
	20	4.32e-1	2.01e-2	1.84	40	100	407.96	68.99	4.32e15
12	3	1.08	1.08e-1	5.59	72	100	1167.06	349.99	1.52e16
	10	4.42e-1	2.16e-2	2.28	42	100	726.93	272.99	1.04e16
	20	5.14e-1	2.95e-2	2.29	40	100	823.05	266.99	1.10e16

Table 2: Numerical results: comparison of the Box and the continuation algorithms.

As can be seen from the tables, the new algorithms find solutions which are roughly at the same distance from the initial design as the solutions obtained with the Box algorithm. However, both algorithms, basic and continuation, are consistently faster than the Box algorithm by a wide margin, and the speeding factor seems to grow with dimension.

5.2 Comparison of the continuation algorithms: constant step vs. greedy version

To compare Algorithms 2 and 3 we also generated 1000 random $n \times n$ mass-spring systems, this time for larger dimensions $n = 10, 20, 50$ and 100. To show the robustness of the algorithms, the initial eigenvalues were distributed both evenly (ED) and randomly (RD). For the systems of dimension $n = 10, 20, 50$ and 100, the total mass is set to $10^4, 10^{10}, 10^{28}$ and 10^{57}, respectively. In all numerical tests the fifth eigenvalue is the center of the resonance band, whose radius depends on the value of the parameter η. The control quantities are now:

1) the distance quotients

$$Q_{m,k}^{(cg)} = \frac{Dist_{m,k}^{(const)}}{Dist_{m,k}^{(greed)}},$$

where $Dist_{m,k}^{(const)}$ and $Dist_{m,k}^{(greed)}$ denote the relative distance from the initial system of the solutions obtained by Algorithms 2 and 3, respectively.

2) the time quotients

$$Q_T^{(cg)} = \frac{\text{CPU time for constant step alg.}}{\text{CPU time for greedy alg.}}.$$

3) the percentage $\pi_{m,k}^{(cg)}$ of cases in which $Q_{m,k}^{(cg)} > 1$.

4) the percentage $\pi_T^{(cg)}$ of cases with $Q_T^{(cg)} > 1$.

5) the percentage π_{isol} of cases in which both the greedy and the constant step version of the continuation algorithm successfully isolated the resonant eigenvalue (these are the only cases taken into account).

As before, the first and second column of Table 3 display the dimension n and the value of η for the test systems, with the third column displaying the distribution of the generated eigenvalues (either ED or RD). Again, there are three columns for the distance quotients $Q_{m,k}^{(cg)}$, the first one displaying the average over the 1000 tests, the second and third displaying, respectively, the smallest and largest value for the quotient among the 1000 experiments. The next two columns display the percentages $\pi_{m,k}^{(cg)}$ and π_{isol}. For the time quotient $Q_T^{(cg)}$ there are three columns, again displaying the average, the minimum and the maximum among the values in all experiments. The last column displays the percentage $\pi_T^{(cg)}$.

				$Q_{m,k}^{(cg)}$					$Q_T^{(cg)}$		
n	η	distb.	mean	min	max	$\pi_{m,k}^{(cg)}$	π_{isol}	mean	min	max	$\pi_T^{(cg)}$
10	3	ED	8.04e-1	6.26e-2	6.21	61	99	7.80	6.39e-14	6.25e13	82
	10	ED	9.13e-1	4.00e-2	2.50	73	100	3.33	6.39e-14	3.12e13	56.99
	20	ED	8.22e-1	2.52e-2	3.60	79	100	1.54	6.39e-14	1.56e13	34
20	3	ED	6.08e-1	2.08e-2	18.50	72	100	15.87	2.66	77.99	100
	10	ED	4.23e-1	1.39e-1	34.68	43.29	77	6.18	6.66e-1	124.49	64.94
	20	ED	7.97e-1	1.61e-1	4.53	57.99	100	2.99	1	9.99	93
50	3	ED	8.38e-1	1.99e-1	9.32	67	98	79.16	16.84	436.69	98
	10	ED	7.17e-1	2.39e-1	12.15	74	98	23.34	3.97	100.01	98
	20	ED	9.90e-1	4.04e-1	5.27	83	98	12.43	1.98	64.39	98
100	3	ED	9.50e-1	2.53e-1	7.79	77.21	97	222.12	1.98	1036.28	97
	10	ED	8.43e-1	4.60e-1	4.51	63.53	97	55.67	1.99	212.31	97
	20	ED	7.57e-1	3.89e-1	4.05	70	96	27.38	3.98	120.31	96

Table 3: Comparison of the continuation algorithms for test systems with *evenly* distributed eigenvalues.

				$Q_{m,k}^{(cg)}$					$Q_T^{(cg)}$		
n	η	distb.	mean	min	max	$\pi_{m,k}^{(cg)}$	π_{isol}	mean	min	max	$\pi_T^{(cg)}$
10	3	RD	9.58e-1	4.19e-4	7.45	46.80	75	4.47	2.13e-14	4.84e14	48.93
	10	RD	2.03	1.47e-1	24.07	73.19	84	1.85	6.39e-14	7.81e13	27.83
	20	RD	4.45	1.53e-1	47.76	85.71	88	2.79	6.39e-14	2.18e14	30.61
20	3	RD	3.44e-1	5.97e-2	38.06	45.26	69	17.83	6.66e-1	292.49	69.47
	10	RD	1.71	2.82e-2	48.42	65.97	80	6.08	6.66e-1	125.49	60.82
	20	RD	3.25	8.69e-2	15.27	71.42	80	2.60	6.66e-1	35.49	47.95
50	3	RD	1.27e-1	9.12e-2	22.15	48.68	62	164.75	5.03e-1	2544.67	58.24
	10	RD	2.72	8.87e-2	31.57	67.17	67	120.68	9.72e-1	3085.97	52.17
	20	RD	3.27	1.01e-1	36.01	72.98	71.00	101.41	9.72e-1	2966.53	42.26
100	3	RD	9.97e-1	8.99e-2	15.42	43.00	63	215.12	1.01	5012.22	54.21
	10	RD	1.17	9.12e-2	27.63	52.01	70	200.23	9.71e-1	4919.03	53.77
	20	RD	2.48	2.13e-1	19.71	59.22	74	213.88	9.96e-1	5132.17	51.01

Table 4: Comparison of the continuation algorithms for test systems with *randomly* distributed eigenvalues.

As before, no significant difference is observed as to the quality of the approximations: both algorithms provide solutions at roughly the same distance from the original system. As to speed, the greedy version is usually faster than the constant step version, as one would expect, but not always: for random spectral distributions, for instance, the time percentage $\pi_T^{(cg)}$ shows that each algorithm is fastest about half of the times. The only case where the greedy version seems to be consistently faster is the one with a large dimension and even spectral distributions. Under those circumstances, the greedy version is faster for almost every experiment than the constant step version.

6 Conclusion

New frequency isolation algorithms are proposed and analyzed for undamped vibrational systems: whenever such a system has unwanted, resonant eigenvalues, the question arises of how to re-design in such a way that the natural frequencies of the new system lie outside the resonance band, and the cost (or impact) of re-design is reasonably small. The basic idea underlying the proposed algorithms is to identify directions in parameter space corresponding to maximal variation of the resonant eigenvalues, guaranteeing at the same time that the non-resonant ones are not changed too much. The fact that isolation is checked quite inexpensively, via bisection, greatly reduces the computational cost in comparison with previous algorithms in the literature, usually based on employing optimization routines (e.g., the Box Method).

Three versions of the algorithm are proposed: a basic one which attempts to isolate the spectrum in one single run, and two iterative versions which consist in repeatedly applying the basic algorithm with two different choices for the step sizes, either constant,

or the so-called greedy version, where the step is taken as large as possible in each iterate. Among the two latter ones, the greedy version is advisable whenever we want to keep the computational cost low, although the re-design we obtain may not be as close to the original design as the output of the constant-step version. This version, however, is usually significantly slower as to computational speed.

Extensive numerical experiments on random systems show that either of the two latter versions is considerably faster than previous algorithms for this problem, like the Box Method, with the greedy version being the fastest. Moreover, the solutions they provide are as close to the original design as the ones produced by the Box Method. Also, the new algorithms are more robust, in the sense that they can cope with situations, like those with randomly distributed initial eigenvalues, which the Box Method is unable to deal with.

References

[1] L. Åasland and P. Bjø rstad, *The generalized eigenvalue problem in ship design and offshore industry – a comparison of traditional methods with the Lanczos process, in Matrix Pencils, B. Kägstrom and A. Ruhe, eds. Proc. Conference held at Pite Havsbad, Sweden, March 22–24, 1982; Lecture Notes in Mathematics, 973, Springer Verlag, Berlin, 1983, pp. 146–155.*

[2] Z. Bai, J. W. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (eds.), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide (Software, Environments and Tools). SIAM, Philadelphia, 2000.*

[3] D. Boley and G. H. Golub, *A survey of inverse matrix eigenvalue problems. Inverse Problems,* ***3*** *(1987), pp. 595–622.*

[4] J. B. Carvalho, B. N. Datta, W. Lin and C. Wang, *Symmetry preserving eigenvalue embedding in finite-element model updating of vibrating structures, J. of Sound and Vibration,* ***290*** *(2006), pp. 839-864.*

[5] B. N. Datta, S. Elhay and Y. M. Ram, *Orthogonality and partial pole assignment for the symmetric positive definite pencil, Linear Algebra Appl.,* ***257*** *(1997), pp. 29–48.*

[6] B. N. Datta, S. Elhay, Y. M. Ram and D. R. Sarkissian, *Partial eigenstructure assignment for the quadratic pencil, J. of Sound and Vibration,* ***230*** *(2000), pp. 101–110.*

[7] B. N. Datta, *Numerical Linear Algebra and Applications, 2nd Edition. SIAM, Philadelphia, 2009.*

[8] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps, SIAM J. Matrix Anal. Appl.,* ***25*** *no. 3 (2004), pp. 858-899.*

[9] I. S. Dhillon, B. N. Parlett and C. Vömel, *The design and implementation of the MRRR algorithm, ACM Trans. Math. Softw.,* ***32*** *no. 4 (2006), pp. 533–560.*

[10] J. C. Egaña, N. M. Kuhl and L. C. Santos, *An inverse eigenvalue method for frequency isolation in spring-mass systems, Numer. Linear Algebra Appl.,* ***9*** *(2002), pp. 65–79.*

[11] A. Friedlander, J. M. Martínez and S. A. Santos, *New trust region algorithms for bound constrained minimization, Applied Mathematics and Optimization,* ***30*** *(1994), pp. 235–266.*

[12] M. Friswell and J. Mottershead, *Finite Element Model Updating in Structural Dynamics. Kluwer Academic Publishers, London (1995).*

[13] D. Inman, *Vibration with Control, Measurement and Stability. Prentice-Hall, Englewood Cliffs, 1989.*

[14] K. T. Joseph, *Inverse eigenvalue problem in structural design, AIAA J. (American Institute of Aeronautics and Astronautics Journal),* ***30*** *(1992), pp. 2890–2896.*

[15] T. Kato, *Perturbation Theory for Linear Operators, Reprint of the 1980 edition. Classics in Mathematics, Springer, Berlin, 1995.*

[16] B. N. Parlett, *The Symmetric Eigenvalue Problem, Reprint edition, Classics in Applied Mathematics Series; Vol. 20, Prentice-Hall SIAM, 1998.*

[17] D. B. Szyld, *Criteria for combining inverse and Rayleigh quotient iteration, SIAM J. Numer. Anal.,* ***25*** *(1988), pp. 1369–1375.*

[18] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem, SIAM Rev.,* ***43*** *(2001), pp. 235–286.*