A Sylvester equation approach for the computation of zero-group-velocity points in waveguides

Bor Plestenjak[1,2*], Daniel A. Kiefer[3] and Hauke Gravenkamp[4]

[2*]Faculty of Mathematics and Physics, University of Ljubljana, Jadranska 19, 1000 Ljubljana, Slovenia.
[3*]Institute of Mathematics, Physics and Mechanics, Jadranska 19, 1000 Ljubljana, Slovenia.
[3]Institut Langevin, ESPCI Paris, Université PSL, CNRS, 75005 Paris, France.
[4]International Centre for Numerical Methods in Engineering (CIMNE), 08034 Barcelona, Spain.

*Corresponding author(s). E-mail(s): bor.plestenjak@fmf.uni-lj.si;
Contributing authors: daniel.kiefer@espci.fr; gravenkamp.research@gmail.com;

Abstract

Eigenvalues of parameter-dependent quadratic eigenvalue problems form eigencurves. The critical points on these curves, where the derivative vanishes, are of practical interest. A particular example is found in the dispersion curves of elastic waveguides, where such points are called zero-group-velocity (ZGV) points. Recently, it was revealed that the problem of computing ZGV points can be modeled as a multiparameter eigenvalue problem (MEP), and several numerical methods were devised. Due to their complexity, these methods are feasible only for problems involving small matrices. In this paper, we improve the efficiency of these methods by exploiting the link to the Sylvester equation. This approach enables the computation of ZGV points for problems with much larger matrices, such as multi-layered plates and three-dimensional structures of complex cross-sections.

Keywords: parameter-dependent quadratic eigenvalue problem, eigencurve, zero-group-velocity point, Sylvester equation, method of fixed relative distance, two-parameter eigenvalue problem

1 Introduction

In many physics and engineering applications, we encounter parameter-dependent quadratic eigenvalue problems (QEP) of the form

$$W(k,\omega)u := \left((\mathrm{i}k)^2 L_2 + \mathrm{i}k L_1 + L_0 + \omega^2 M\right) u = 0, \tag{1}$$

where L_0, L_1, L_2, M are real $n \times n$ matrices, which are usually obtained by a (semi-)discretization of a boundary value problem. The solutions (k,ω) form *eigencurves* $\omega(k)$, and we are interested in locating the critical points on these curves, where $\omega'(k) = \frac{\partial \omega}{\partial k} = 0$. Although solutions of (1) can be complex, we consider the important case where ω and k are both real.

This work is motivated by the study of (anisotropic) elastic waveguides (see, e.g., [23, 35]), where ω denotes the angular frequency and k the wavenumber. In this context, the eigencurves are referred to as *dispersion curves*. The slope $c_{\mathrm{g}} = \omega'$ is called *group velocity*, which is of practical relevance, as it describes the propagation of energy. Points (k_*, ω_*) on the dispersion curves where the group velocity vanishes are called *zero-group-velocity (ZGV)* points. Often, the term is used exclusively for solutions at finite wavenumber k_*, as this is the non-trivial case, but here we use the designation for solutions at any k_*. In the light of this motivating practical application, we will generally refer to points on the curves formed by eigenvalues of parameter-dependent eigenvalue problems that satisfy $\omega'(k) = 0$ as *ZGV points*, irrespective of their physical interpretation.

Recently, a numerical algorithm for the computation of ZGV points in anisotropic elastic waveguides was introduced [23] that can be applied to a general problem of the form (1). The method is based on a generalization of the method of fixed relative distance (MFRD) from [19], which provides good initial approximations that can be refined by a locally convergent Newton-type method. Inspired by the Sylvester-Arnoldi method from [28], we show in this contribution that sophisticated tools from linear algebra substantially speed up the algorithm and reduce its memory requirements. This enables us to solve problems with larger matrices and tackle more complex problems such as multi-layered plates as well as waveguides of arbitrary two-dimensional cross-sections.

In the following, we first discuss properties of ZGV points in Section 2. In Section 3, we introduce several tools we will use in the following section; the presentation is intertwined with their application to the computation of ZGV points: the Sylvester equation, multiparameter eigenvalue problems, and the MFRD. Our main contributions are included in Section 4, where we show how we can exploit the structure of the Sylvester equation to apply the MFRD more efficiently, and in Section 5, where we present a scanning algorithm for the computation of ZGV points that combines the MFRD and a locally convergent Gauss-Newton method. In Section 6, we introduce a waveguide model that is used in the numerical experiments in the following section, where we demonstrate the strength of the proposed method. Finally, we discuss possible generalizations and give a conclusion in Sections 8 and 9.

2 Theory on ZGV points

If we assume that $u = u(k)$ and $\omega = \omega(k)$ are differentiable, then, by differentiating (1), we obtain

$$(-2kL_2 + \mathrm{i}L_1 + 2\omega(k)\omega'(k)M)u(k) + W(k, \omega(k))u'(k) = 0,$$

where, in general, $\bullet' = \frac{\partial \bullet}{\partial k}$. If (k_*, ω_*) is a ZGV point, then $\omega'(k_*) = 0$, and it follows that

$$(-2k_* L_2 + \mathrm{i}L_1)u_* + W(k_*, \omega_*)v_* = 0, \tag{2}$$

where $u_* = u(k_*)$ and $v_* = u'(k_*)$. Let z_* be the corresponding left eigenvector of (1) at (k_*, ω_*), i.e., $z_*^{\mathrm{H}} W(k_*, \omega_*) = 0$. By multiplying (2) by z_*^{H} from the left, we get the following necessary condition for a ZGV point:

$$z_*^{\mathrm{H}}(-2k_* L_2 + \mathrm{i}L_1)u_* = z_*^{\mathrm{H}} W'(k_*, \omega_*)u_* = 0. \tag{3}$$

Lemma 2.1. *If (k_*, ω_*) is a ZGV point of the parameter-dependent QEP (1), then k_* is a multiple eigenvalue of the QEP*

$$Q(k)u := \left((\mathrm{i}k)^2 L_2 + \mathrm{i}k L_1 + L_0 + \omega_*^2 M\right)u = 0 \tag{4}$$

that we get by fixing ω in (1) to ω_*.

Proof. If z and u are the left and right eigenvector of a simple eigenvalue k of the QEP $Q(k)u = 0$, then it is well-known that $z^{\mathrm{H}} Q'(k)u \neq 0$, see, e.g., [31, Prop. 1] or [1, Thm. 3.2]. But, since (3) holds at a ZGV point, it thus follows that k_* is a multiple eigenvalue of (4). $\square$

Lemma 2.1 gives a necessary condition, but not every point (k_*, ω_*) such that k_* is a multiple eigenvalue of (4) for a fixed $\omega = \omega_*$, corresponds to a ZGV point. In addition, (2) must hold as well, and this means that v_* is a root vector of height two[1]. This is possible only if the algebraic multiplicity m_a of

---

[1]An eigenvalue λ_* of a quadratic matrix polynomial $Q(\lambda)$ is defective if and only if there exist an eigenvector (a root vector of height one) $v \neq 0$ and a root vector of height two w such that $Q(\lambda_*)v = 0$ and $Q(\lambda_*)w + Q'(\lambda_*)v = 0$, see, e.g., [12, Sec. 2.2].

k_* as an eigenvalue of (4) is strictly greater than the geometric multiplicity $m_g = \dim(\mathrm{Ker}(Q(k_*)))$. Also, to make sure that $\omega(k)$ is analytic in a neighborhood of k_*, we require that ω_* is a simple eigenvalue of $W(k_*, \omega)$, i.e., the generalized eigenvalue problem (GEP) that we get by fixing k to k_* in (1). Note that in some cases, it is possible to extend the dispersion curves so that they remain analytical also in points where the curves cross and multiple eigenvalues appear, see, e.g., [27]. To keep things concise, we will keep the requirement that ω_* is a simple eigenvalue of $W(k_*, \omega)$ and thus exclude points from candidates for ZGV points where two or more dispersion curves cross.

Example 2.2. We take

$$
L_2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; L_1 = \begin{bmatrix} 0 & 3 & 0 \\ -3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; L_0 = \begin{bmatrix} -1.75 & 1 & 0 \\ 1 & -1.75 & 0 \\ 0 & 0 & -0.25 \end{bmatrix}, \; M = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 3.5 \end{bmatrix}.
$$

We selected the matrices so that L_0 is symmetric, L_1 is skew-symmetric, and L_2, M are symmetric positive definite. This way, the matrices have the same properties as the larger matrices in [23], where ZGV points of Lamb waves in an austenitic steel plate are computed.

The corresponding eigenvalue problem (1) has five real ZGV points $(0, 0.2673)$, $(0, 0.4074)$, $(0, 1.0628)$, and $(\pm 1.0642, 0.2393)$, such that $\omega > 0$, which are shown together with the real dispersion curves in Fig. 1. We consider only the solutions with $\omega > 0$ since each dispersion curve $\omega(k)$ has its counterpart $-\omega(k)$ and the same holds for the ZGV points. Note that the points $(\pm 0.4236, 0.3503)$, where the dispersion curves cross, are not ZGV points, although $k_* = \pm 0.4236$ is a double eigenvalue of (4) for a fixed $w_* = 0.3503$.

Due to the structure of the matrices, the dispersion curves are also symmetric with respect to the ω-axis and there exist trivial ZGV points at $k = 0$, which can be computed from the GEP $(L_0 + \omega^2 M)u = 0$. Nontrivial ZGV points come in pairs $(\pm k_*, \omega_*)$ and we are interested in solutions where $k_* > 0$.
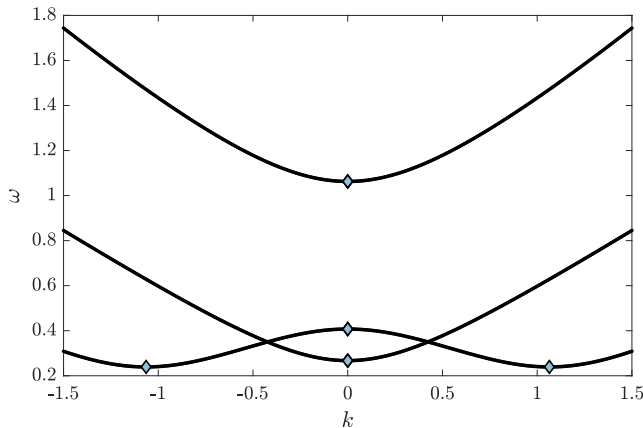

Fig. 1: Real dispersion curves $\omega(k)$ and ZGV points of Example 2.2.

3 Auxiliary results

In this section, we introduce some related results and numerical methods that we will use in the following sections to construct an efficient numerical method for finding the ZGV points of (1).

3.1 Sylvester equation

The Kronecker product $A \otimes B$ *of matrices* $A \in \mathbb{C}^{n \times m}$ *and* $B \in \mathbb{C}^{p \times q}$ *is a matrix of size* $np \times mq$ *of the block form*

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{bmatrix}.$$

For a matrix $X \in \mathbb{C}^{m \times n}$, $\text{vec}(X) \in \mathbb{C}^{mn}$ is a *vectorization* of matrix X, i.e, the vector obtained by stacking all columns of X on top of each other. Our results are based on the well-known equality (see, e.g., [16, Lem. 4.3.1]):

$$\text{vec}(AXB) = (B^{\text{T}} \otimes A)\,\text{vec}(X), \tag{5}$$

which holds for $A \in \mathbb{C}^{m \times m}$, $B \in \mathbb{C}^{n \times n}$, and $X \in \mathbb{C}^{m \times n}$. Suppose that we are looking for a matrix $X \in \mathbb{C}^{m \times n}$ that satisfies *the Sylvester equation*

$$AX + XB = C \tag{6}$$

for given matrices $A \in \mathbb{C}^{m \times m}$, $B \in \mathbb{C}^{n \times n}$, and $C \in \mathbb{C}^{m \times n}$. It follows from (5) that (6) is equivalent to

$$(I_n \otimes A + B^{\text{T}} \otimes I_m)\text{vec}(X) = \text{vec}(C). \tag{7}$$

The Sylvester equation is therefore uniquely solvable when $I_n \otimes A + B^{\text{T}} \otimes I_m$ is nonsingular, which is true if and only if $\lambda + \mu \neq 0$ for all possible pairs (λ, μ), where λ is an eigenvalue of A and μ is an eigenvalue of B, see, e.g., [16, Thm. 4.4.6]. We could apply (7) to numerically solve (6), but this is not efficient since it, in general, leads to complexity $\mathcal{O}(m^3 n^3)$ due to a matrix of size $mn \times mn$ in (7).

There exist more efficient numerical methods for the Sylvester equation, for instance, the Bartels-Stewart algorithm [3], which is appropriate in our setting, where we have to solve many Sylvester equations with the same matrices A, B and different right-hand sides C. In the Bartels-Stewart algorithm, we first compute two Schur decompositions

$$A = QRQ^{\text{H}}, \quad B = USU^{\text{H}},$$

where matrices Q, U are unitary, and matrices S, R are upper triangular. Applying the above Schur decompositions to (6), we obtain a new Sylvester equation with upper triangular matrices

$$RY + YS = D, \tag{8}$$

where $D = Q^{\text{H}}CU$ and $Y = Q^{\text{H}}XU$. The columns of $Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$ can now be computed from left to right as solutions of upper triangular linear systems

$$(R + s_{ii}I)y_i = d_i - \sum_{k=1}^{i-1} s_{ki}y_k, \quad i = 1, \ldots, n,$$

and then $X = QYU^{\text{H}}$. If the Sylvester equation is nonsingular, then $R + s_{ii}I$ is nonsingular for all $i = 1, \ldots, n$. With the above approach, we can efficiently solve the Sylvester equation (6) in complexity $\mathcal{O}(m^3 + n^3)$, which is much less than $\mathcal{O}(m^3 n^3)$, the complexity of solving (7) as a large linear system.

3.2 Multiparameter eigenvalue problems

A *d-parameter eigenvalue problem* has the form

$$A_{10}x_1 = \lambda_1 A_{11}x_1 + \cdots + \lambda_d A_{1d}x_1$$

$$\vdots \tag{9}$$

4

$$A_{d0}x_d = \lambda_1 A_{d1}x_d + \cdots + \lambda_d A_{dd}x_d,$$

where A_{ij} is an $n_i \times n_i$ complex matrix, and $x_i \neq 0$ are vectors for $i = 1, \ldots, d$. If (9) holds, then $(\lambda_1, \ldots, \lambda_d) \in \mathbb{C}^d$ is an *eigenvalue* and $x_1 \otimes \cdots \otimes x_d$ is the corresponding *eigenvector*. A generic multiparameter eigenvalue problem (MEP) (9) has $N = n_1 \cdots n_d$ eigenvalues, which are roots of a system of d multivariate characteristic polynomials

$$p_i(\lambda_1, \ldots, \lambda_d) := \det(A_{i0} - \lambda_1 A_{i1} - \cdots - \lambda_d A_{id}) = 0, \quad i = 1, \ldots, d. \tag{10}$$

The problem (9) is related to a system of GEPs

$$\Delta_1 z = \lambda_1 \Delta_0 z, \quad \ldots, \quad \Delta_d z = \lambda_d \Delta_0 z, \tag{11}$$

where $z = x_1 \otimes \cdots \otimes x_d$, and the $N \times N$ matrices

$$\Delta_0 = \begin{vmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & & \vdots \\ A_{d1} & \cdots & A_{dd} \end{vmatrix}_{\otimes} := \sum_{\sigma \in S_d} \mathrm{sgn}(\sigma)\, A_{1\sigma_1} \otimes A_{2\sigma_2} \otimes \cdots \otimes A_{d\sigma_d} \tag{12}$$

$$\Delta_i = \begin{vmatrix} A_{11} & \cdots & A_{1,i-1} & A_{10} & A_{1,i+1} & \cdots & A_{1d} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ A_{d1} & \cdots & A_{d,i-1} & A_{d0} & A_{d,i+1} & \cdots & A_{dd} \end{vmatrix}_{\otimes}, \quad i = 1, \ldots, d,$$

are called *operator determinants*, which uses the Kronecker product for multiplication. For details see, e.g., [2]. If Δ_0 is nonsingular, then we say that (9) is *regular*. In such cases, the matrices $\Gamma_i := \Delta_0^{-1}\Delta_i$, $i = 1, \ldots, d$, commute, and the eigenvalues of (9) are the joint eigenvalues of commuting matrices $\Gamma_1, \ldots, \Gamma_d$. Hence, if N is not too large, a standard numerical approach to computing the eigenvalues of (9) is to explicitly compute $\Gamma_1, \ldots, \Gamma_d$ and then solve a joint eigenvalue problem. Alternatively, if we prefer not to multiply by Δ_0^{-1}, we may solve a joint system of GEPs (11), see, e.g., [15].

If all linear combinations of matrices $\Delta_0, \Delta_1, \ldots, \Delta_d$ are singular, then (9) is a *singular MEP*, which is much more difficult to solve. In such case, it is still possible that the polynomial system (10) has finitely many roots that are the eigenvalues of (9). Then, (11) is a joint system of d singular matrix pencils whose regular eigenvalues are the solutions of (9). For more details, see, e.g., [24]. To solve a singular MEP numerically, we can apply a generalized staircase-type algorithm [30], which returns matrices Q and Z with orthonormal columns that yield projected smaller matrices $\widehat{\Delta}_i = Q^*\Delta_i Z$ for $i = 0, \ldots, d$ such that $\widehat{\Delta}_0$ is nonsingular, matrices $\widehat{\Delta}_0^{-1}\widehat{\Delta}_i$, $i = 1, \ldots, d$, commute, and their joint eigenvalues are the eigenvalues of (9).

The above approach for singular problems is used in [23], where it is shown that ZGV points of (1) correspond to the eigenvalues of a singular three-parameter eigenvalue problem (3EP)

$$\begin{aligned} (\eta C_2 + \lambda C_1 + C_0)w &= 0 \\ (\eta L_2 + \lambda L_1 + L_0 + \mu M)u &= 0 \\ (\eta \widetilde{L}_2 + \lambda \widetilde{L}_1 + \widetilde{L}_0 + \mu \widetilde{M})v &= 0, \end{aligned} \tag{13}$$

where $\lambda = \mathrm{i}k$, $\mu = \omega^2$, $\eta = (\mathrm{i}k)^2$,

$$\widetilde{L}_2 = \begin{bmatrix} L_2 & 0 \\ 0 & L_2 \end{bmatrix}, \ \widetilde{L}_1 = \begin{bmatrix} L_1 & 0 \\ 2L_2 & L_1 \end{bmatrix}, \ \widetilde{L}_0 = \begin{bmatrix} L_0 & 0 \\ L_1 & L_0 \end{bmatrix}, \ \widetilde{M} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix},$$

and

$$C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \ C_1 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \ C_0 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{14}$$

Note that the matrices (14) incorporate the relation between λ and η since $\det(\eta C_2 + \lambda C_1 + C_0) = \eta - \lambda^2$. This gives the first known numerical method that can compute all ZGV points without any initial approximations. However, since we first have to explicitly compute the corresponding Δ matrices of size $4n^2 \times 4n^2$ and then use expensive numerical methods for singular problems, this approach is feasible only for problems (1) with small matrices.

Let us remark that solvers for MEPs are not included in standard numerical packages for problems in linear algebra, but several numerical methods, also for singular problems, are implemented in the Matlab toolbox MultiParEig [33].

3.3 Method of fixed relative distance

Instead of solving (13), which gives exact solutions, we can solve a simpler regular 3EP that returns approximations of candidates for ZGV points. From each individual candidate, we can then compute the exact ZGV point by applying the locally convergent Gauss-Newton method that we provide in Section 5.1. This approach, presented first in [23] for Hermitian problems, is based on Lemma 2.1 and the method of fixed relative distance (MFRD) from [19].

We know from Lemma 2.1 that at ω_* corresponding to a ZGV point, the QEP in the variable $\lambda = \mathrm{i}k$ with fixed $\mu_* = \omega_*^2$, i.e.,

$$\left(\lambda^2 L_2 + \lambda L_1 + L_0 + \mu_* M\right) u = 0, \tag{15}$$

has a multiple (generically double) eigenvalue $\lambda_* = \mathrm{i}k_*$. Therefore, for certain $\widetilde{\mu} \neq \mu_*$ but close to μ_*, the QEP

$$\left(\lambda^2 L_2 + \lambda L_1 + L_0 + \widetilde{\mu} M\right) u = 0 \tag{16}$$

has at least two different solutions close to λ_*. The MFRD, adapted to (15) in [23], introduces the 3EP

$$(\eta C_2 + \lambda C_1 + C_0)w = 0$$
$$(\eta L_2 + \lambda L_1 + L_0 + \mu M)u = 0 \tag{17}$$
$$\left(\eta(1+\delta)^2 L_2 + \lambda(1+\delta)L_1 + L_0 + \mu M\right) v = 0$$

in $\mu = \omega^2$, $\lambda = \mathrm{i}k$, $\eta = \lambda^2$, and C_0, C_1, C_2 as in (14). Therein, $\delta > 0$ specifies the relative distance between the sought λ and serves as a regularization parameter. The 3EP (17) has an eigenvalue $(\widetilde{\lambda}, \widetilde{\mu}, \widetilde{\eta})$, where $\left(\sqrt{\widetilde{\mu}}, -\mathrm{i}\widetilde{\lambda}\right)$ is close to a ZGV point, such that $\widetilde{\lambda}$ and $\widetilde{\lambda}(1+\delta)$ are eigenvalues of the initial problem (16).

Solutions of (17) can be obtained from a transformation into the corresponding system of GEPs (11). The problem (17) is regular since the corresponding $2n^2 \times 2n^2$ matrix

$$\Delta_0 = \begin{vmatrix} C_2 & C_1 & 0 \\ L_2 & L_1 & M \\ (1+\delta)^2 L_2 & (1+\delta)L_1 & M \end{vmatrix}_\otimes \tag{18}$$

is nonsingular for $\delta > 0$. Hence, we can solve the GEP given by

$$\Delta_1 z = \lambda \Delta_0 z, \tag{19}$$

where

$$\Delta_1 = (-1) \begin{vmatrix} C_2 & C_0 & 0 \\ L_2 & L_0 & M \\ (1+\delta)^2 L_2 & L_0 & M \end{vmatrix}_\otimes, \tag{20}$$

by a standard numerical method for GEPs. In the ensuing, the obtained eigenvector z is used in the GEP associated with μ, namely,

$$\Delta_M z = \mu \Delta_0 z, \tag{21}$$

with Δ_0 defined before and

$$\Delta_M = (-1) \begin{vmatrix} C_2 & C_1 & C_0 \\ L_2 & L_1 & L_0 \\ (1+\delta)^2 L_2 & (1+\delta)L_1 & L_0 \end{vmatrix}_\otimes, \tag{22}$$

6

to obtain μ from the Rayleigh quotient

$$\mu = \frac{z^{\mathrm{H}} \Delta_M z}{z^{\mathrm{H}} \Delta_0 z}. \tag{23}$$

Even for small values of n, computing all eigenvalues of (19) is very demanding. Instead, we can apply a subspace iterative method (for instance, `eigs` in Matlab) to find eigenvalues of (19) close to a target ik_0. We can apply this several times using different targets k_0 and, thus, scan an interval $[k_a, k_b]$ for ZGV points (k_*, ω_*); for more details, see Section 5 and [23]. In the next section, we will show how we can exploit the structure of the matrices (18) and (20) to find eigenvalues of (19) close to a target ik_0 much more efficiently. This improvement enables the computation of ZGV points for much larger matrices than in the original algorithm from [23].

We remark that it is not possible to apply a similar approach with a subspace iteration to the 3EP (13) because this problem is singular.

4 Exploiting the structure

When employing a subspace iterative method such as the Krylov-Schur method [36] or the implicitly restarted Arnoldi method [25] for the solution of the GEP (21), the computational bottleneck in each step is the solution of a linear system of the form

$$(\Delta_1 - \sigma \Delta_0)z = \Delta_0 y. \tag{24}$$

Even for sparse matrices L_0, L_1, L_2, M, which we obtain using, e.g., the finite element method, this makes the computation slow already for modest matrix size n. By exploiting the structure of the matrices Δ_0 and Δ_1 in a similar way as in [28], we can solve the linear system (24) much more efficiently.

First, we note the block structure

$$\Delta_0 = \begin{bmatrix} G_1 & G_2 \\ G_2 & 0 \end{bmatrix}, \quad \Delta_1 = \begin{bmatrix} -G_0 & 0 \\ 0 & G_2 \end{bmatrix}, \tag{25}$$

where

$$\begin{aligned} G_0 &= L_0 \otimes M - M \otimes L_0, \\ G_1 &= L_1 \otimes M - (1+\delta)M \otimes L_1, \\ G_2 &= L_2 \otimes M - (1+\delta)^2 M \otimes L_2. \end{aligned} \tag{26}$$

Introducing the block notation $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, we can rewrite (24) as

$$\begin{bmatrix} -G_0 - \sigma G_1 & -\sigma G_2 \\ -\sigma G_2 & G_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} G_1 y_1 + G_2 y_2 \\ G_2 y_1 \end{bmatrix}. \tag{27}$$

If we add the second block row, multiplied by σ, to the first block row, we get an equivalent lower block triangular system

$$\begin{bmatrix} -G_0 - \sigma G_1 - \sigma^2 G_2 & 0 \\ -\sigma G_2 & G_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} (G_1 + \sigma G_2)y_1 + G_2 y_2 \\ G_2 y_1 \end{bmatrix}. \tag{28}$$

Hence, we can compute z_1 using the first block row

$$(G_0 + \sigma G_1 + \sigma^2 G_2)z_1 = -(G_1 + \sigma G_2)y_1 - G_2 y_2, \tag{29}$$

and it follows from the second equation and nonsingularity of G_2 that $z_2 = y_1 + \sigma z_1$.

To solve (29) efficiently, we transform it into a Sylvester equation using the equalities from Subsection 3.1. First, let $w := -(G_1 + \sigma G_2)y_1 - G_2 y_2$. If $y_1 = \mathrm{vec}(Y_1)$ and $y_2 = \mathrm{vec}(Y_2)$, then it follows from

(26) that the right-hand side of (29) is $w = \text{vec}(W)$, where

$$W = MY_1(L_1 + \sigma L_2)^{\text{T}} - ((1+\delta)L_1 + (1+\delta)^2 L_2)Y_1 M^{\text{T}} - MY_2 L_2 + (1+\delta)^2 L_2 Y_2 M^{\text{T}}.$$

In a similar way, we get from (26) that

$$G_0 + \sigma G_1 + \sigma^2 G_2 = L(0) \otimes M - M \otimes L(\delta),$$

where $L(\delta) := L_0 + (1+\delta)\sigma L_1 + (1+\delta)^2 \sigma^2 L_2$. Thus, (29) is equivalent to

$$MZ_1 L(0)^{\text{T}} - L(\delta)Z_1 M^{\text{T}} = W, \tag{30}$$

where $z_1 = \text{vec}(Z_1)$. Since M is nonsingular, we can write the above as a Sylvester equation

$$Z_1 L(0)^{\text{T}} M^{-\text{T}} - M^{-1}L(\delta)Z_1 = \widetilde{W}, \tag{31}$$

where $\widetilde{W} = M^{-1}WM^{-\text{T}} = Y_1(L_1 + \sigma L_2)^{\text{T}}M^{-\text{T}} - M^{-1}((1+\delta)L_1 + (1+\delta)^2 L_2)Y_1 - Y_2 L_2 M^{-\text{T}} + M^{-1}(1+\delta)^2 L_2 Y_2$. As we can solve the above Sylvester equation in complexity $\mathcal{O}(n^3)$, this is also the complexity of solving (24).

Let us remark that we can exploit the structure of Δ_0 and Δ_M to compute μ in (23) in complexity $\mathcal{O}(n^3)$ as well. Namely, we have

$$\Delta_M = \begin{bmatrix} G_3 & G_4 \\ G_4 & G_5 \end{bmatrix}$$

with

$$G_3 = -L_1 \otimes L_0 + (1+\delta)L_0 \otimes L_1,$$
$$G_4 = (1+\delta)^2 L_0 \otimes L_2 - L_2 \otimes L_0,$$
$$G_5 = -(1+\delta)L_2 \otimes L_1 + (1+\delta)^2 L_1 \otimes L_2.$$

To obtain (23), we compute matrices

$$T_1 = -L_0 Z_1 L_1^{\text{T}} + (1+\delta)L_1 Z_1 L_0^{\text{T}} + (1+\delta)^2 L_2 Z_2 L_0^{\text{T}} - L_0 Z_2 L_2^{\text{T}},$$
$$T_2 = (1+\delta)^2 L_2 Z_1 L_0^{\text{T}} - L_0 Z_1 L_2^{\text{T}} - (1+\delta)L_1 Z_2 L_2^{\text{T}} + (1+\delta)^2 L_2 Z_2 L_1^{\text{T}},$$
$$T_3 = MZ_1 L_1^{\text{T}} - (1+\delta)L_1 Z_1 M^{\text{T}} + MZ_2 L_2^{\text{T}} - (1+\delta)^2 L_2 Z_2 M^{\text{T}},$$
$$T_4 = MZ_1 L_2^{\text{T}} - (1+\delta)^2 L_2 Z_1 M^{\text{T}}$$

and compute μ as

$$\mu = \frac{z_1^{\text{H}} t_1 + z_2^{\text{H}} t_2}{z_1^{\text{H}} t_3 + z_2^{\text{H}} t_4}, \tag{32}$$

where $t_i = \text{vec}(T_i)$ for $i = 1, \ldots, 4$. As this computation involves only multiplications by $n \times n$ matrices, its complexity is $\mathcal{O}(n^3)$.

5 Algorithm

For large problems, we suggest applying the MFRD to provide good initial approximations, which we subsequently refine using the local convergent method presented next.

5.1 Gauss-Newton method

If we introduce $\lambda = \text{i}k$ and $\mu = \omega^2$ similarly to (15), then we know from Section 2 that for a ZGV point of (1), we have to find $\lambda, \mu \in \mathbb{C}$ and $u, z \in \mathbb{C}^n$ such that

$$(\lambda^2 L_2 + \lambda L_1 + L_0 + \mu M)u = 0$$

$$z^{\mathrm{H}}(\lambda^2 L_2 + \lambda L_1 + L_0 + \mu M) = 0$$
$$z^{\mathrm{H}}(2\lambda L_2 + L_1)u = 0 \tag{33}$$
$$(u^{\mathrm{H}}u - 1)/2 = 0$$
$$(z^{\mathrm{H}}z - 1)/2 = 0.$$

The number of equations exceeds the number of unknowns by one in (33); hence, this is an overdetermined nonlinear system. However, it is a *zero-residual system* because if (k, ω) is a ZGV point, and u and z are the corresponding right and left eigenvector, then all equations in (33) are satisfied.

For solving (33) from a good initial approximation, we apply the Gauss-Newton method [5, 32]. To overcome the obstacle that the third equation in (33) is not complex differentiable in z, we define $y = \overline{z}$, where $\overline{\bullet}$ denotes the complex conjugate, and rewrite (33) as

$$F(u, y, \lambda, \mu) := \begin{bmatrix} (\lambda^2 L_2 + \lambda L_1 + L_0 + \mu M)u \\ (\lambda^2 L_2 + \lambda L_1 + L_0 + \mu M)^{\mathrm{T}}y \\ y^{\mathrm{T}}(2\lambda L_2 + L_1)u \\ (u^{\mathrm{H}}u - 1)/2 \\ (y^{\mathrm{H}}y - 1)/2 \end{bmatrix} = 0. \tag{34}$$

Suppose that $(u_k, y_k, \lambda_k, \mu_k)$ is an approximation to the solution of (34). If $F(u, y, \lambda, \mu) = 0$ then also $F(\alpha u, \beta y, \lambda, \mu) = 0$ for arbitrary $\alpha, \beta \in \mathbb{C}$ such that $|\alpha| = |\beta| = 1$. Because of that, the vectors u and y are not uniquely defined and, although the last two equations in (34) are not complex differentiable, as explained in [26], we can obtain a correction $(\Delta u_k, \Delta y_k, \Delta\lambda_k, \Delta\mu_k)$ for the update

$$(u_{k+1}, y_{k+1}, \lambda_{k+1}, \mu_{k+1}) = (u_k, y_k, \lambda_k, \mu_k) + (\Delta u_k, \Delta y_k, \Delta\lambda_k, \Delta\mu_k)$$

as a solution of the $(2n + 3) \times (2n + 2)$ least squares problem

$$J_F(u_k, y_k, \lambda_k, \mu_k)\Delta s_k = -F(u_k, y_k, \lambda_k, \mu_k),$$

where $\Delta s_k = \begin{bmatrix} \Delta u_k^{\mathrm{T}} & \Delta y_k^{\mathrm{T}} & \Delta\lambda_k & \Delta\mu_k \end{bmatrix}^{\mathrm{T}}$, and the Jacobian $J_F(u_k, y_k, \lambda_k, \mu_k)$ is

$$\begin{bmatrix} \lambda_k^2 L_2 + \lambda_k L_1 + L_0 + \mu_k M & 0 & (2\lambda_k L_2 + L_1)u_k & Mu_k \\ 0 & (\lambda_k^2 L_2 + \lambda_k L_1 + L_0 + \mu_k M)^{\mathrm{T}} & (2\lambda_k L_2 + L_1)^{\mathrm{T}}y_k & M^{\mathrm{T}}y_k \\ y_k^{\mathrm{T}}(2\lambda_k L_2 + L_1) & u_k^{\mathrm{T}}(2\lambda_k L_2 + L_1)^{\mathrm{T}} & 2y_k^{\mathrm{T}}L_2 u_k & 0 \\ u^{\mathrm{H}} & 0 & 0 & 0 \\ 0 & y^{\mathrm{H}} & 0 & 0 \end{bmatrix}. \tag{35}$$

Besides an initial approximation (k_0, ω_0) for the ZGV point, the method requires initial approximation for the right and left eigenvector as well. If we do not have them, then usually a good choice is to use a random vector from the space spanned by the right and left singular vectors that belong to a few of the smallest singular values of $W(k_0, \omega_0)$.

The Gauss-Newton method converges locally quadratically for a zero-residual problem if the Jacobian J_F has full rank at the solution, see, e.g., [5, Section 4.3.2] or [32, Section 10.4]. We can show that the Jacobian $J_F(u_*, y_*, \lambda_*, \mu_*)$ has full rank at a generic ZGV point, where $k_* = -\mathrm{i}\lambda_*$ is a double eigenvalue of (4) for $\omega = \sqrt{\mu_*}$, and u_* and $\overline{y}_*$ are the corresponding right and left eigenvector. For the proof, see Lemma A.1 in the appendix. If ZGV points exist where the multiplicity of k_* is higher than two, then, at such points, we can expect a linear convergence.

Let us note that the above Gauss-Newton method also converges to points (λ, μ) where there exists a solution of (33), and these include the points where the dispersion curves cross. At such points, we can also expect a linear convergence. If the method has converged to a zero-residual solution, then we can verify if a computed point is a ZGV point by checking if ω_* is indeed a simple eigenvalue of $W(k_*, \omega)$.

We also remark that for a special case when $W(k, \omega)$ is Hermitian for real values k, the left eigenvector corresponding to a right eigenvector u is u^{H}, which happens, for example, in the case when the matrices

L_i are alternately symmetric/anti-symmetric and M is symmetric positive definite. For such problems, we can use a more efficient Newton's method, see [23, Sec. IV].

5.2 Scanning method

The following algorithm uses the MFRD from Section 3.3 to scan a wavenumber interval $[k_a, k_b]$ and compute ZGV points (k_*, ω_*) such that $k \in [k_a, k_b]$. The matrices Δ_0, Δ_1 and Δ_M refer to (18), (20) and (22), respectively, but we do not have to compute them explicitly.

Algorithm 1: Scanning method for ZGV points
Input: $n \times n$ matrices L_2, L_1, L_0, M, interval $[k_a, k_b]$, default step size Δk
Output: ZGV points (k_*, ω_*)

1:   set $k_0 = k_a$
2:   while $k_0 < k_b$
3:       find m eigenvalues of $\Delta_1 z = \lambda \Delta_0 z$ closest to $\lambda_0 = \mathrm{i} k_0$
4:       for each λ and eigenvector z
5:           compute $\mu = z^{\mathrm{H}} \Delta_M z / z^{\mathrm{H}} \Delta_0 z$
6:           if $|\mathrm{Re}(\lambda)|$ and $|\mathrm{Im}(\mu)|$ are both small then
7:               apply Gauss-Newton method to solve (33) with initial guess $(\mathrm{Im}(\lambda), \mathrm{Re}(\mu))$
8:               if the method converged to (λ_*, μ_*), and (3) holds then
9:                   add $(k_*, \omega_*) = (-\mathrm{i}\lambda_*, \sqrt{\mu_*})$ to the list of ZGV points
10:      set $k_0 = \max(k_0 + \Delta k, 0.95 \cdot \max\{k_* \in \text{ZGV points}\})$

In the following, we provide additional details about Algorithm 1.

- In line 3, we can apply any subspace method, for instance `eigs` in Matlab. Thereby, it is important that we do not generate the matrices Δ_0 and Δ_1 explicitly. Internally, the `eigs` function iteratively solves the linear system $(\Delta_1 - \sigma\Delta_0)z = \Delta_0 y$, and it is possible to provide a pointer to a custom implementation thereof. We do so, thereby exploiting the relation with the Sylvester equation from Section 4.
- Alternatively, if n is small enough, we can compute all eigenvalues of $\Delta_1 z = \lambda \Delta_0 z$. This approach gives all ZGV points in just one run; hence, the scanning is not needed, and it could be more efficient than solving the related singular 3EP (13).
- In line 5, we compute μ using (32), which avoids generating Δ_0 and Δ_M. Note that this expression might not return correct μ if λ is a multiple eigenvalue of $\Delta_1 z = \lambda \Delta_0 z$, e.g., when there exist different ZGV points with the same $k = -\mathrm{i}\lambda$. This is very unlikely to appear, except for the trivial ZGV points at $\lambda = 0$. This makes ZGV points with k_* close to zero very difficult to compute.
- For an initial approximation for the left and right eigenvector, we take the left and right singular vector for the smallest singular value of $\lambda^2 L_2 + \lambda L_1 + L_0 + \mu M$.
- In line 10, we update the target k_0 in such a way that it is unlikely that the method will miss ZGV points in the interval. We assume that if the subspace iteration method in line 3 returns some approximations, it does not miss any of the closest ZGV points. The idea is to increase the target either for a default step Δk or use a larger step if some ZGV points were found in the last loop.

It is difficult to provide the best values for the parameters $k_a, k_b, \Delta k, m$, as they are problem-dependent. For a sensible choice when treating guided waves in plates, see the implementation in the package GEWtool [21] and the numerical examples in Section 7.

6 Waveguide model

While the proposed approach can be applied, quite generally, to compute critical points on eigencurves of parameter-dependent quadratic eigenvalue problems, this work is motivated by a particular application, namely the modeling of elastic waves propagating along structures of constant cross-section, commonly referred to as waveguides. In this context, a finite-element discretization of the cross-section yields matrices with the properties discussed above. Hence, we will briefly summarize the formulation that has been

deployed to obtain the matrices used in our numerical experiments. Consider a waveguide of linearly elastic material and arbitrary cross-section Γ as depicted in Fig. 2a. Its mass density is denoted as ρ, and its 4$^{\text{th}}$-order stiffness tensor $\mathbf{c}$ is of arbitrary anisotropy. In absence of external loads, the mechanical displacements $\tilde{\mathbf{u}}(x, y, z, t)$ are governed by the boundary-value problem

$$\nabla \cdot \mathbf{c} : \nabla \tilde{\mathbf{u}} - \rho \partial_t^2 \tilde{\mathbf{u}} = \mathbf{0} \quad \text{in} \quad \Gamma \times \mathbb{R}, \tag{36a}$$

$$\tilde{\mathbf{u}} = \mathbf{0} \quad \text{on} \quad \partial\Gamma_D \times \mathbb{R}, \tag{36b}$$

$$\mathbf{e}_{\text{n}} \cdot \mathbf{c} : \nabla \tilde{\mathbf{u}} = \mathbf{0} \quad \text{on} \quad \partial\Gamma_N \times \mathbb{R}. \tag{36c}$$

Therein, ∂_t is the partial derivative with respect to time t, and $\nabla = \mathbf{e}_i \partial_i$ is the Nabla operator. $\partial\Gamma_D$ and $\partial\Gamma_N$ denote the parts of the cross-sectional boundary where Dirichlet and Neumann boundary conditions are imposed, respectively. Lastly, $\mathbf{e}_{\text{n}}$ is the outward unit normal vector.

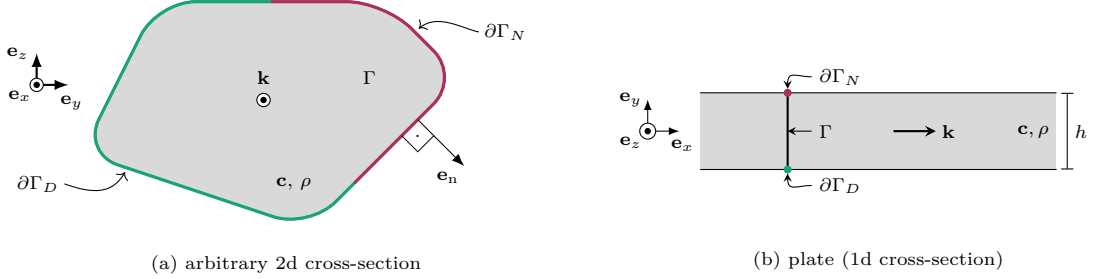(a) arbitrary 2d cross-section   (b) plate (1d cross-section)

Fig. 2: Waveguide geometries. The waveguides extend infinitely along $\mathbf{e}_x$, which corresponds to the wave vector orientation. (a) Arbitrary two-dimensional cross-section. (b) A plate confines waves only in the one-dimensional thickness direction. Γ: cross-sectional domain, $\partial\Gamma_D$ Dirichlet boundary, $\partial\Gamma_N$: Neumann boundary, $\mathbf{e}_{\text{n}}$: unit normal to the boundary, $\mathbf{c}$: stiffness tensor, ρ: density, $\mathbf{k}$: wave vector, h: thickness.

Due to the translational invariance in time t and the axial coordinate x, modal solutions are of the form $\tilde{\mathbf{u}}(x, y, z, t) = \mathbf{u}(k, y, z, \omega)\, \mathrm{e}^{\mathrm{i}(kx - \omega t)}$. Inserting into (36), we obtain the waveguide problem

$$\left[(\mathrm{i}k)^2 \mathcal{L}_2 + \mathrm{i}k\mathcal{L}_1 + \mathcal{L}_0 + \omega^2 \rho \mathbf{I} \right] \mathbf{u} = \mathbf{0} \quad \text{in} \quad \Gamma, \tag{37a}$$

$$\mathbf{u} = \mathbf{0} \quad \text{on} \quad \partial\Gamma_D, \tag{37b}$$

$$\left[\mathrm{i}k\mathcal{B}_1 + \mathcal{B}_0 \right] \mathbf{u} = \mathbf{0} \quad \text{on} \quad \partial\Gamma_N, \tag{37c}$$

which describes the plane harmonic guided wave solutions $(k, \omega, \mathbf{u})$ of interest. In the above, $\mathcal{L}_i$ and $\mathcal{B}_j$ are differential operators, which are explicitly given, using the 2$^{\text{nd}}$-order tensors $\mathbf{c}_{ij} = \mathbf{e}_i \cdot \mathbf{c} \cdot \mathbf{e}_j$, as

$$\mathcal{L}_2 = \mathbf{c}_{xx}, \tag{38a}$$

$$\mathcal{L}_1 = (\mathbf{c}_{xy} + \mathbf{c}_{yx})\partial_y + (\mathbf{c}_{xz} + \mathbf{c}_{zx})\partial_z, \tag{38b}$$

$$\mathcal{L}_0 = \mathbf{c}_{yy}\partial_y^2 + (\mathbf{c}_{yz} + \mathbf{c}_{zy})\partial_y\partial_z + \mathbf{c}_{zz}\partial_z^2, \tag{38c}$$

and

$$\mathcal{B}_1 = \mathbf{c}_{nx}, \tag{39a}$$

$$\mathcal{B}_0 = \mathbf{c}_{ny}\partial_y + \mathbf{c}_{nz}\partial_z. \tag{39b}$$

Equations (37)-(39) are also valid for the special case of the infinite plate as depicted in Fig. 2b (the reader may refer to [23] for a succinct derivation). A plane strain field should be assumed in this case, i.e., all terms in ∂_z vanish while the equations remain otherwise unaffected. The waveguide's cross-section can thereby be modeled by a one-dimensional discretization, as the displacements do not depend on z. Furthermore, by expressing the problem in cylindrical coordinates, a similar formulation can be obtained

for waves propagating along full or hollow cylinders. We also note that the above vector-field problem can be reduced to the scalar wave equation representing waves in a fluid medium, i.e., scalar acoustics.

The waveguide problem in (37) represents a differential eigenvalue problem, which we discretize using a Galerkin finite-element procedure [20, 17, 6, 14, 11].[2] This numerical formulation yields matrices $L_i, M \in \mathbb{R}^{n \times n}$ such that

$$W(k, \omega)u := \left[(\mathrm{i}k)^2 L_2 + \mathrm{i}k L_1 + L_0 + \omega^2 M \right] u = 0 \tag{40}$$

approximates the original problem (37a) and respects the boundary conditions (37b) and (37c). Therein, u is the vector of coefficients corresponding to the chosen discretization. Note that, for a nondissipative material and real-valued parameters k and ω, the waveguide operator $W(k, \omega)$ is Hermitian, and, furthermore, M is positive definite.

Elastic guided waves as described by (40) usually exhibit several ZGV points (k_*, ω_*). As the Δ-matrices scale as $4n^2 \times 4n^2$ (in eq. (13)) or $2n^2 \times 2n^2$ (in eq. (17)), large waveguide problems quickly lead to prohibitively large computational demands, effectively rendering the methods from [23] unusable. There are mainly two crucial situations where this is the case: (i) plates and cylinders with many layers, and (ii) waveguides of two-dimensional cross-section. In the following, we demonstrate that the method presented in Section 4 is capable of computing ZGV points even for such complex structures.

7 Numerical experiments

All numerical experiments are performed on an Apple M1 Pro notebook with 32 GB of memory. The regularization parameter of the MFRD method is chosen as $\delta = 10^{-2}$ for all computations.

7.1 Austenitic steel plate

An orthotropic austenitic steel plate exhibits many ZGV points. They were computed in [23] with an MFRD algorithm that uses `eigs` from Matlab to compute eigenvalues of (19) close to a target $\mathrm{i}k_0$. The matrices Δ_1 and Δ_0 were computed explicitly and are represented as sparse matrices. The method in Matlab first computes an LU sparse factorization, which is then used to solve (24) in each step of the method. The example given in the mentioned reference is of size $n = 39$, which yields Δ_i-matrices of size 3042×3042. The computational time with the old method was 12 s.

Instead, we can use the approach proposed in Section 4 to solve (24) without explicitly constructing the matrices Δ_1 and Δ_0. In the initial phase, we compute the Schur decompositions of $L(0)^{\mathrm{T}} M^{-\mathrm{T}}$ and $M^{-1}L(\delta)$ from (31) and then use them to solve (31) and, thus, obtain the solution of (24). In this way, we never use matrices larger than $n \times n$. Applying this procedure to the numerical example of [23] with $n = 39$ finds all 18 ZGV points in 0.5 s, which is more than twenty times faster than with the previous method. Note that the computational times are difficult to compare, as the strategy to update the target wavenumber also changed. More importantly, our new procedure scales favorably with the problem size, which is demonstrated by the following examples.

7.2 Fluid-filled pipe

The following example, taken from Cui et al. [4], consists of a water-filled steel pipe with a wall thickness of $h = 0.5$ mm and an inner radius of 9.5 mm, see Fig. 3a. The steel pipe is characterized by shear and longitudinal wave speeds of $c_{\mathrm{t}} = 3200 \frac{\mathrm{m}}{\mathrm{s}}$, $c_\ell = 5900 \frac{\mathrm{m}}{\mathrm{s}}$ and a mass density of $7900 \frac{\mathrm{kg}}{\mathrm{m}^3}$. The water inside the pipe has a mass density of $1000 \frac{\mathrm{kg}}{\mathrm{m}^3}$ and a bulk wave speed of $1500 \frac{\mathrm{m}}{\mathrm{s}}$. As the fluid domain is relatively large compared to the bulk wavelength in water, this problem requires a considerable number of degrees of freedom. Specifically, we used one element of 7th order (eight nodes) to discretize the pipe wall, while the fluid domain required a polynomial degree of 140 to yield accurate results within the selected frequency range.[3] In the pipe wall, we assume displacements $\mathbf{u}(r)\, \mathrm{e}^{\mathrm{i}(kx + n_\varphi \varphi + \omega t)}$ of integer circumferential

---

[2]Specifically, we opt, in this work, for a particular type of high-order polynomial interpolation (sometimes referred to as spectral elements) for one-dimensional cross-sections [11, 7] and for nonuniform rational B-splines (NURBS) [8] to discretize complex 2D cross-sections. A review of various shape functions in the context of semi-analytical methods is given in [10].

[3]While such large polynomial degrees are generally uncommon in the Finite Element Method, they have been found to be remarkably efficient in this particular context of waveguide modeling. This is because the bottleneck in the computation of the

order n_φ (similarly for the acoustic pressure in the fluid). This enables us to discretize only a radial line as sketched in Fig. 3a. As an additional challenge, the pressure-displacement formulation leads to non-Hermitian matrices, and a complex formulation of the Newton-iteration refinement as described in Section 5.1 is required. For comparison with the literature, we choose $n_\varphi = 0$ and $\mathbf{u} = u_x \mathbf{e}_x + u_r \mathbf{e}_r$ to obtain the so-called longitudinal modes L$(0, m)$ [4]. Overall, this results in matrices L_i and M in (40) of size 157×157 and Δ-matrices of size 49298×49298. As demonstrated by Cui et al. [4], multiple ZGV points are found in the frequency region close to the backward wave of the empty pipe, i.e., the curve with negative slope in Fig. 3b. Using parameters $m = 8$, $k_a h = 0.01$, $k_b h = 2$, $\Delta k h = 0.05$, our algorithm locates all 15 ZGV points in 11 s. The result is depicted in Fig. 3b.
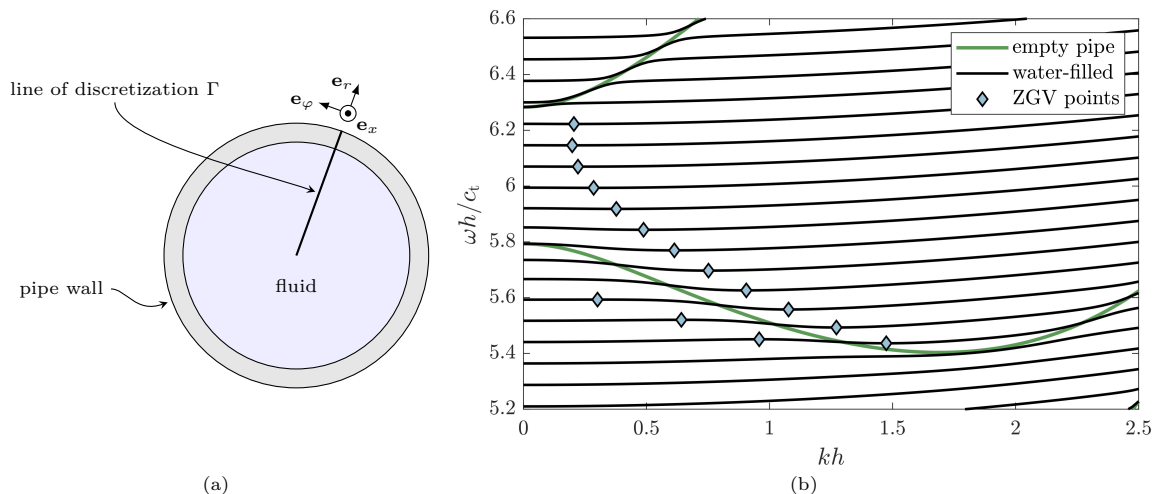

(a)

(b)

Fig. 3: Longitudinal waves in a water-filled steel pipe. (a) Geometry: cross-section of the water-filled circular steel pipe of inner radius $9.5\,\text{mm}$ and outer radius $10\,\text{mm}$. (b) Dispersion curves of longitudinal modes, i.e., u_x-u_r-polarized waves. The dispersion curves of the empty pipe are shown for comparison. The fluid-filled pipe exhibits 15 ZGV points close to the backward wave of the free pipe.

7.3 Composite plate

A plate consisting of many layers requires a large number of degrees of freedom to describe guided waves since each layer is represented by at least one finite element. We use the proposed method to compute ZGV points in a composite plate consisting of 400 layers with a total thickness of $h = 50\,\text{mm}$. Such materials are used in the aerospace industry, and this particular example is taken from [18]. The plate is composed of a symmetric layup of a T800/913 carbon fiber reinforced polymer (CFRP) as depicted in Fig. 4a. The mass density is given as $\rho = 1550\,\text{kg/m}^3$, and the stiffness in Voigt notation reads

$$\mathbf{C} = \begin{bmatrix} 154 & 3.7 & 3.7 & 0 & 0 & 0 \\ & 9.5 & 5.2 & 0 & 0 & 0 \\ & & 9.5 & 0 & 0 & 0 \\ & & & 2.15 & 0 & 0 \\ & \text{sym.} & & & 4.2 & 0 \\ & & & & & 4.2 \end{bmatrix} \text{GPa} \,.$$

In order to consistently nondimensionalize the results, we define the smallest shear wave velocity as $c_t = \sqrt{C_{44}/\rho}$. For the frequency range of interest, it is sufficient to discretize each layer with one linear

---

dispersion relations is the (complete) solution of an eigenvalue problem. The costs for this solution for moderate matrix sizes depend mainly on the matrix size rather than its sparsity. Hence, in contrast to most finite element applications, which require mainly the solution of linear systems of equations, it is, in this case, desired to obtain small matrices, even if they are dense. The advantage of such large element orders was described in [11] and discussed in more detail in [7].
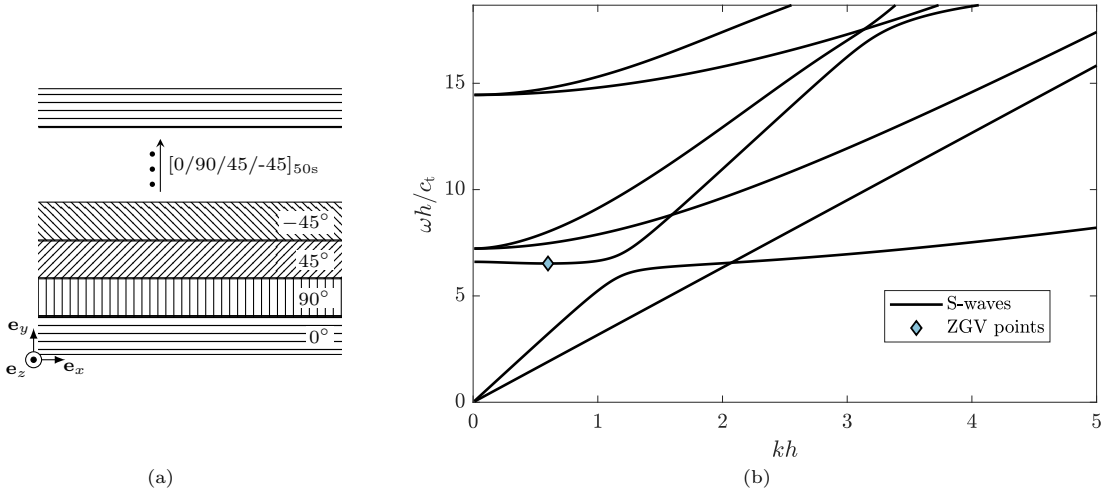
Fig. 4: Composite plate of symmetric layup $[0/90/45/\text{-}45]_{50s}$. (a) Geometry: the fibers in each layer are oriented at angle θ as indicated; thereby, θ represents the angular coordinate rotating around $\mathbf{e}_y$ and measured from $\mathbf{e}_x$. All 400 layers are of the same thickness and amount to a total of 50 mm. (b) Dispersion curves for wave vectors $\mathbf{k} = k\mathbf{e}_x$, i.e., $\theta = 0°$. One point exists where the axial group velocity component $\partial\omega/\partial k$ vanishes, and it is marked in the plot.

finite element. Symmetric and anti-symmetric waves decouple, and we consider symmetric waves only. This is achieved by representing one-half of the geometry and fixing the u_y displacement at the center node. Lamb and shear-horizontal polarizations are coupled due to anisotropy. This requires modeling all three displacement components in the equations of motion (37). Proceeding as described previously yields the matrices L_i and M from (40) of size 602×602. Note that the corresponding Δ_i-matrices are of size $724\,808 \times 724\,808$, which is considerable.

The dispersion curves corresponding to propagation along $\mathbf{e}_x$ are shown in Fig. 4b. There exists a point where $\partial\omega/\partial k = 0$, and it is marked therein. Using the parameters $m = 8$, $k_a h = 0.2$, $k_b h = 2$, $\Delta k h = 0.1$, our algorithm was able to successfully locate it in 43 s. It is important to remark that the group velocity is a vector parallel to the xz-plane. For anisotropic plates, it is not necessarily collinear to the wave vector $\mathbf{k}$. The derivative $\partial\omega/\partial k$ represents the group velocity component along the wave vector $\mathbf{k}$. Since our numerical methods compute points such that $\partial\omega/\partial k = 0$, we find the waves whose group velocity is orthogonal to the wave vector or vanishes altogether. This was exploited in [22] to find waves with a power flux transverse to their wave vector. As a side note, we also remark that the dispersion curves in Fig. 4b do not exhibit crossings. Instead, they get very close and then veer apart; see [9] for details on this phenomenon.

7.4 Rail

In this numerical experiment, we compute ZGV points of a relatively complex three-dimensional structure, namely a rail with the cross-section depicted in Fig. 5a. Rails are typical examples of guided wave propagation in three-dimensional structures, and their dynamic properties are often investigated due to the relevance of acoustic emission and ultrasonic material testing, see, e.g., [6, 13, 37, 38] and the references therein. This particular geometry has been studied in [8], where dispersion curves have already been computed based on the semi-analytical formulation outlined in Section 6. Instead of conventional finite elements, the cross-section is discretized by means of non-uniform rational B-splines (NURBS), which allow for the exact description of this shape without introducing geometry approximation errors. Furthermore, NURBS are very robust at high frequencies. However, for the discussion in this paper, these differences are of lesser significance, as the obtained matrices possess the same relevant properties compared to using high-order polynomials. For computing the ZGV points, we use the discretization suggested in [8] for computing dispersion curves for the first nine modes up to a frequency of 10 kHz.
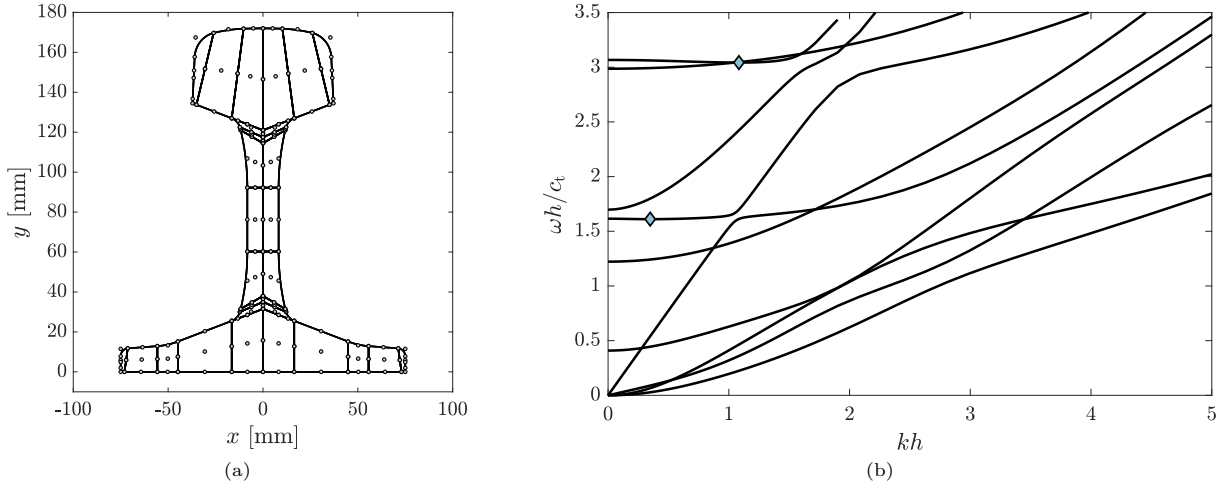
14

Fig. 5: Wave propagation along a rail. (a): Discretization of the cross-section, showing a division into 30 patches and the control points for describing the contour. (b): Dispersion curves of all propagating modes and the two ZGV points found within the chosen frequency range.

Specifically, the interpolation relies on the 30 patches shown in Fig. 5a, each of them locally refined using NURBS of the third order, resulting in matrices of size 1020×1020. For clarity, the figure only includes the minimal number of control points required to describe the geometry. A simple isotropic linearly elastic material is assumed with a Poisson's ratio of $\nu = 0.2$. To nondimensionalize the results for consistency with the other examples, we define $h = 172\,\mathrm{mm}$ as the height of the rail, i.e., the largest extent in the y-direction. The dispersion curves are displayed in Fig. 5b, together with the two ZGV points found within the selected frequency range. Using $m = 6$, $k_a h = 0.1$, $k_b h = 2$, $\Delta k h = 0.2$, Algorithm 1 locates the two ZGV points in $545\,\mathrm{s}$ ($9\,\mathrm{min}$).

8 Possible generalizations

It is possible to define ZGV points for similar two-parameter eigenvalue problems. For example, in [34], the critical points of dispersion curves for the eigenvalue problem $(A + \lambda B + \mu C)x = 0$ are discussed. The obtained numerical methods for this problem are very similar, and we can apply the MFRD and the Gauss-Newton method, suitably modified to the structure of the eigenvalue problem. Specifically, we need to modify the approach in Section 4 so that we can use a subspace method in the MFRD without explicitly constructing large Δ matrices for the related MEP. In a similar way, it would be possible to define a ZGV point and extend the theory and numerical methods for a parameter-dependent polynomial eigenvalue problem of the form

$$P(k, \omega)u := \left((\mathrm{i}k)^d L_d + (\mathrm{i}k)^{d-1} L_{d-1} + \cdots + \mathrm{i}k L_1 + L_0 + \omega^2 M\right) u = 0, \tag{41}$$

or for a nonlinear parameter-dependent eigenvalue problem $N(k, \omega)u = 0$, where $N : \mathbb{C}^2 \to \mathbb{C}^{n \times n}$.

9 Conclusion

The improved approach enables us to tackle significantly larger problems and compute more accurate solutions in cases where it was previously either impossible to construct the matrices $\Delta_0, \Delta_1, \Delta_M$ explicitly, or the computation was unfeasibly slow. For even larger n, the improved algorithm also eventually reaches its limits due to the considerable memory requirements. Namely, vectors that span the search subspace in the subspace iteration method for the eigenvalue problem (18) are of size $2n^2$. If the matrices L_i and M are large, the memory required for saving a sufficient number of vectors of this size can become prohibitively large. A possible solution that could extend the approach to large and sparse matrices L_i

and M would be to generalize the subspace iteration methods from [28], which exploit the low-rank format of the vectors and work only on vectors of size n.

Declarations

Conflict of interests   Not applicable.

References

[1] A. L. Andrew, K.-W. E. Chu, and P. Lancaster. Derivatives of eigenvalues and eigenvectors of matrix functions. *SIAM J. Matrix Anal. Appl.*, 14(4):903–926, 1993.

[2] F. V. Atkinson. *Multiparameter Eigenvalue Problems.* Mathematics in Science and Engineering, Vol. 82. Academic Press, New York-London, 1972. Volume I: Matrices and compact operators.

[3] R. H. Bartels and G. W. Stewart. Solution of the matrix equation $AX + XB = C$. *Commun. ACM*, 15(9):820–826, 1972.

[4] H. Cui, W. Lin, H. Zhang, X. Wang, and J. Trevelyan. Backward waves with double zero-group-velocity points in a liquid-filled pipe. *J. Acoust. Soc. Am.*, 139(3):1179–1194, Mar. 2016.

[5] P. Deuflhard. *Newton Methods for Nonlinear Problems*, volume 35 of *Springer Series in Computational Mathematics.* Springer, Heidelberg, 2011. Affine invariance and adaptive algorithms, First softcover printing of the 2006 corrected printing.

[6] L. Gavrić. Computation of propagative waves in free rail using a finite element technique. *J. Sound Vib.*, 185(3):531–543, Aug. 1995.

[7] H. Gravenkamp, C. Birk, and C. Song. The computation of dispersion relations for axisymmetric waveguides using the Scaled Boundary Finite Element Method. *Ultrasonics*, 54:1373–1385, 2014.

[8] H. Gravenkamp, S. Natarajan, and W. Dornisch. On the use of NURBS-based discretizations in the scaled boundary finite element method for wave propagation problems. *Comput. Methods Appl. Mech. Eng.*, 315:867–880, 2017.

[9] H. Gravenkamp, B. Plestenjak, and D. A. Kiefer. Notes on osculations and mode tracing in semi-analytical waveguide modeling. *Ultrasonics*, 135:107112, July 2023.

[10] H. Gravenkamp, A. Saputra, and S. Duczek. High-order shape functions in the scaled boundary finite element method revisited. *Arch. Comput. Methods Eng.*, 28:473–494, 2021.

[11] H. Gravenkamp, C. Song, and J. Prager. A numerical approach for the computation of dispersion relations for plate structures using the scaled boundary finite element method. *J. Sound Vib.*, 331:2543–2557, 2012.

[12] S. Güttel and F. Tisseur. The nonlinear eigenvalue problem. *Acta Numer.*, 26:1–94, 2017.

[13] T. Hayashi, W.-J. Song, and J. L. Rose. Guided wave dispersion curves for a bar with an arbitrary cross-section, a rod and rail example. *Ultrasonics*, 41(3):175–183, May 2003.

[14] A.-C. Hladky-Hennion. Finite element analysis of the propagation of acoustic waves in waveguides. *J. Sound Vib.*, 194(2):119–136, July 1996.

[15] M. E. Hochstenbach, T. Košir, and B. Plestenjak. A Jacobi-Davidson type method for the two-parameter eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26(2):477–497, 2004/05.

[16] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, Cambridge, 1991.

[17] K. H. Huang and S. B. Dong. Propagating waves and edge vibrations in anisotropic composite cylinders. *J. Sound Vib.*, 96(3):363–379, 1984.

[18] A. M. A. Huber and M. G. R. Sause. Classification of solutions for guided waves in anisotropic composites with large numbers of layers. *J. Acoust. Soc. Am.*, 144(6):3236–3251, Dec. 2018.

[19] E. Jarlebring, S. Kvaal, and W. Michiels. Computing all pairs (λ, μ) such that λ is a double eigenvalue of $A + \mu B$. *SIAM J. Matrix Anal. Appl.*, 32(3):902–927, 2011.

[20] E. Kausel and J. M. Roësset. Semianalytic Hyperelement for Layered Strata. *J. Eng. Mech.*, 103(4):569–588, Aug. 1977.

[21] D. A. Kiefer. GEWtool. https://github.com/dakiefer/GEWtool, 2024. Computer software. https://doi.org/10.5281/zenodo.10114243.

[22] D. A. Kiefer, S. Mezil, and C. Prada. Beating resonance patterns and extreme power flux skewing in anisotropic elastic plates. *Sci. Adv.*, 9(51):eadk6846, Dec. 2023.

[23] D. A. Kiefer, B. Plestenjak, H. Gravenkamp, and C. Prada. Computing zero-group-velocity points in anisotropic elastic waveguides: Globally and locally convergent methods. *J. Acoust. Soc. Am.*, 153(2):1386–1398, 2023.

[24] T. Košir and B. Plestenjak. On the singular two-parameter eigenvalue problem II. *Linear Algebra Appl.*, 649:433–451, 2022.

[25] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, 17(4):789–821, 1996.

[26] T. Lu and Y. Su. A Newton-type method for two-dimensional eigenvalue problems. *Numer. Linear Algebra Appl.*, 29(4):e2430, 2022.

[27] T. Lu, Y. Su, and Z. Bai. Variational characterization and Rayleigh quotient iteration of 2D eigenvalue problem with applications. *SIAM J. Matrix Anal. Appl.*, 45(3):1455–1486, 2024.

[28] K. Meerbergen and B. Plestenjak. A Sylvester-Arnoldi type method for the generalized eigenvalue problem with two-by-two operator determinants. *Numer. Linear Algebra Appl.*, 22(6):1131–1146, 2015.

[29] R. Mennicken and M. Möller. *Non-self-adjoint boundary eigenvalue problems*, volume 192 of *North-Holland Mathematics Studies*. North-Holland Publishing Co., Amsterdam, 2003.

[30] A. Muhič and B. Plestenjak. On the quadratic two-parameter eigenvalue problem and its linearization. *Linear Algebra Appl.*, 432(10):2529–2542, 2010.

[31] A. Neumaier. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 22(5):914–923, 1985.

[32] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[33] B. Plestenjak. MultiParEig 2.8. https://www.mathworks.com/matlabcentral/fileexchange/47844-multipareig, 2024. MATLAB Central File Exchange.

[34] B. Plestenjak. On properties and numerical computation of critical points of eigencurves of bivariate matrix pencils. *arXiv:2411.07450*, 2024.

[35] C. Prada, D. Clorennec, and D. Royer. Local vibration of an elastic plate and zero-group velocity Lamb modes. *J. Acoust. Soc. Am.*, 124(1):203–212, 2008.

[36] G. W. Stewart. A Krylov-Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.*, 23(3):601–614, 2001/02.

[37] D. J. Thompson. Wheel-rail noise generation, part III: rail vibration. *J. Sound Vib.*, 161(3):421–446, 1993.

[38] J. Zhang, H. Ma, W. Yan, and Z. Li. Defect detection and location in switch rails by acoustic emission and Lamb wave analysis : A feasibility study. *Appl. Acoust.*, 105:67–74, 2016.

A Quadratic convergence of the Gauss-Newton method

Lemma A.1. *Let $\xi \in \Omega$ be an eigenvalue of algebraic multiplicity two and geometric multiplicity one of a nonlinear eigenvalue problem $N(\lambda)u = 0$, where $N : \Omega \to \mathbb{C}^{n \times n}$ is holomorphic on a domain $\Omega \subseteq \mathbb{C}$. Let nonzero vectors $u, z, s, p \in \mathbb{C}^n$ be, respectively, the right and left eigenvector and the right and left root vector of height two such that*

$$N(\xi)u = 0, \tag{42a}$$

$$N(\xi)s + N'(\xi)u = 0, \tag{42b}$$

$$z^{\mathrm{H}}N(\xi) = 0, \tag{42c}$$

$$p^{\mathrm{H}}N(\xi) + z^{\mathrm{H}}N'(\xi) = 0. \tag{42d}$$

Then

$$z^{\mathrm{H}}N'(\xi)s + p^{\mathrm{H}}N'(\xi)u + z^{\mathrm{H}}N''(\xi)u \neq 0.$$

Proof. If we multiply (42b) by p^{H} from the left and (42d) by s from the right, we observe the equality $z^{\mathrm{H}}N'(\xi)s = p^{\mathrm{H}}N'(\xi)u$. Since it follows from [29, Thm. 1.6.5], see also [12, Thm. 2.5], that

$$z^{\mathrm{H}}N'(\xi)s + z^{\mathrm{H}}\frac{N''(\xi)}{2}u \neq 0,$$

this completes the proof. $\qquad\square$

Lemma A.2. *Let (k_*, ω_*) be a ZGV point of (1), such that the algebraic multiplicity of k_* as an eigenvalue of the QEP (4) is two, and the geometric multiplicity is one. Let u_* and z_* be the corresponding right and left eigenvectors, $y_* = \bar{z}_*$, and let a and b be such vectors that $a^{\mathrm{H}}u_* = 1$ and $b^{\mathrm{H}}y_* = 1$. The Jacobian $J_F(u_*, y_*, \lambda_*, \mu_*)$, given in (35), where $\lambda_* = \mathrm{i}k_*$ and $\mu_* = \omega_*^2$, has full rank.*

Proof. Suppose that the Jacobian $J_F(u_*, y_*, \lambda_*, \mu_*)$ is rank deficient. Then, there exist vectors s, t and scalars α, β, not all being zero, such that

$$J_F(u_*, y_*, \lambda_*, \mu_*)\begin{bmatrix} s^{\mathrm{T}} & t^{\mathrm{T}} & \alpha & \beta \end{bmatrix}^{\mathrm{T}} = 0.$$

Then

$$(\lambda_*^2 L_2 + \lambda_* L_1 + L_0 + \mu_* M)s + \alpha(2\lambda_* L_2 + L_1)u_* + \beta M u_* = 0, \tag{43a}$$

$$(\lambda_*^2 L_2 + \lambda_* L_1 + L_0 + \mu_* M)^{\mathrm{T}}t + \alpha(2\lambda_* L_2 + L_1)y_* + \beta M^{\mathrm{T}}y_* = 0, \tag{43b}$$

$$y_*^{\mathrm{T}}(2\lambda_* L_2 + L_1)s + u_0^{\mathrm{T}}(2\lambda_* L_2 + L_1)^{\mathrm{T}}t + 2\alpha y_*^{\mathrm{T}}L_2 u_0 = 0, \tag{43c}$$

$$u_*^{\mathrm{H}}s = 0, \tag{43d}$$

$$y_*^{\mathrm{H}}t = 0. \tag{43e}$$

First, we show that $\beta = 0$. If we multiply (43a) by z_*^{H}, then it follows that $\beta = 0$ because $z_*^{\mathrm{H}}(2\lambda_* L_2 + L_1)u_0 = 0$ due to a ZGV point and $z_*^{\mathrm{H}}M u_* \neq 0$ because we require that ω_* is a simple eigenvalue of $W(k_*, \omega)$.

If $\alpha \neq 0$, then it follows from (43a) and (43b) that $(1/\alpha)s$ and $(1/\alpha)\bar{t}$ are left and right root vectors of height 2 of the QEP $N(\lambda)u := (\lambda^2 L_2 + \lambda L_1 + L_0 + \mu_* M)u = 0$ for the eigenvalue λ_*. But then it follows from Lemma A.2 that (43c) is not zero. Therefore, $\alpha = 0$.

Since $\alpha = \beta = 0$ it follows from (43a) that $s = \gamma u_*$ for a scalar γ and then $s = 0$ because of (43d). In a similar way we get from (43b) and (43e) that $t = 0$. This shows that the kernel of $J_F(u_*, y_*, \lambda_*, \mu_*)$ is trivial. $\qquad\square$