Multiparameter Spectral Analysis for Aeroelastic Instability Problems

Article *in* Journal of Applied Mechanics · August 2017

DOI: 10.1115/1.4039671

2 authors:

Arion Pons
Hebrew University of Jerusalem
15 PUBLICATIONS **25** CITATIONS

SEE PROFILE

Stefanie Gutschmidt
University of Canterbury
59 PUBLICATIONS **417** CITATIONS

SEE PROFILE

Multiparameter spectral analysis for aeroelastic instability problems

Arion Pons[1,*] and Stefanie Gutschmidt[2]

[1]Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK.
[2]Department of Mechanical Engineering, University of Canterbury, Christchurch 8140, New Zealand.

Key words multiparameter eigenvalue problems; instability; aeroelasticity.

This paper presents a novel application of multiparameter spectral theory to the study of structural stability, with particular emphasis on aeroelastic flutter. Methods of multiparameter analysis allow the development of significant new solution and analysis algorithms for aeroelastic flutter problems; including direct solvers for polynomial problems of arbitrary order and size, and for characterising the nature of the flutter point and its local modal damping gradient. Two major variants of the flutter point direct solver are presented, and their computational characteristics are compared. Both are well suited for analysing and characterising the flutter points of problems arising in reduced-order modelling and preliminary design optimization. Extensions and improvements to this new conceptual framework and solution method are then discussed.

1. Introduction

Predicting and controlling aeroelastic instability forms a major part of the discipline of aeroelasticity. Many different physical systems show flutter instability, and many models exist to describe them. In a linear system, or the linearization of a nonlinear system, the onset of flutter or divergence can be formulated in the well-known stability criterion:

$$\mathrm{Im}(\chi) > 0 \text{ for stability} \tag{1}$$

where χ are the time-eigenvalues of the system according to the Fourier transform $q(t) = \hat{q}e^{\imath\chi t}$ for the system coordinate q [1]. These eigenvalues may be nondimensionalised. Flutter occurs when the system parameters are such that the system is on the stability boundary, $\mathrm{Im}(\chi_\mathrm{f}) = 0$. The flutter point may then be described as a tuple which includes the set of system parameters (particularly, an airspeed parameter) and the modal frequency of instability, χ_f. As a subclass, a static instability point (at zero frequency, $\chi_\mathrm{f} = 0$) is termed a divergence point. Typically only one or two flutter or divergence points are of industrial relevance.

Eq. 1 is not however the only criterion that can be used to characterize instability. This leads us into the study of aeroelastic methods. Apart from the four established methods – the *p-method*, *classical flutter analysis*, the *k-method* (or *U-g method*, or *V-g method*) and the *p-k method*; all of which are detailed and discussed in a number of reference works [1–3] – recent years have seen a proliferation of new aeroelastic methods. Several authors have refined the existing established methods for particular scenarios or applications [4–6]. The application of concepts from robust control theory have yielded a series of methods, including the μ-method by Lind and Brenner [7,8], the μ-k method by Borglund [9–12], and others [13–15]. The

* Corresponding author, adp53@cam.ac.uk,

prime advantage of these μ-type methods is that they facilitate the propagation of uncertainty distributions through the system, allowing a worst-case flutter speed estimate to be made in a system with high uncertainty. Other developments have come from other fields: Afolabi [16,17] characterized coupled-mode flutter as a loss of eigenvector orthogonality, using methods from catastrophe theory. Irani and Sazesh [18] used stochastic methods, while Gu et al. [19] devised a genetic algorithm, and a number of authors [20–23] have applied neural networks to the detection of flutter points.

It is in the context of these developments that we propose our method of analyzing flutter problems. The central methodological contribution of this paper is the concept that the solution of an aeroelastic system for its flutter points is nothing other than a multiparameter eigenvalue problem. We will show the simple link between the aeroelastic stability problem and multiparameter spectral theory, and how this allows for direct solution of a variety of flutter problems. The purpose of this paper is to detail this link and to explore the mechanisms by which this direct solution may be accomplished. For this purpose we will apply our analysis to a simple example – however, the method does extend to problems that are significantly more complex; these are considered in Pons and Gutschmidt [24].

2. Aeroelastic flutter as a multiparameter eigenvalue problem (MEP)

Consider a linear finite-dimensional system with eigenvector $\mathbf{x} \in \mathbb{C}^n$ and continuous dependence on both an eigenvalue parameter $\chi \in \mathbb{C}$, and another structural or environmental parameter $p \in \mathbb{R}$:

$$\mathrm{A}(\chi, p)\mathbf{x} = \mathbf{0} \tag{2}$$

where $\mathrm{A} \in \mathbb{C}^{n\times n}$. Any complex structural parameter can of course be split into two real parameters. The stability problem for this system (with respect to parameter p) is to find p such that an eigenvalue of the problem (χ) has zero imaginary part. This point is the 'stability boundary': for a system with multiple structural parameters, the stability boundary may be a line or other higher-dimensional surface. We then note that the condition $\mathrm{Im}(\chi) = 0$ is equivalent to modifying the original definition of the problem such that $\chi \in \mathbb{R}$ and not $\chi \in \mathbb{C}$. Such a manoeuvre does not seem to be immediately useful: under $\chi \in \mathbb{R}$, a solution to Eq. 2 only exists on the stability boundary, and nowhere else. In order to develop, for example, iterative methods for flutter point calculation, we need to be able to define some form of solution in the subcritical and supercritical areas (above and below the stability boundary, respectively). There is an easy way of doing this. Following [25], we take the complex conjugate of Eq. 2 as another equation:

$$\begin{aligned} \mathrm{A}(\chi, p)\mathbf{x} &= \mathbf{0}, \\ \overline{\mathrm{A}}(\chi, p)\bar{\mathbf{x}} &= \mathbf{0}. \end{aligned} \tag{3}$$

As $p \in \mathbb{R}$ and $\chi \in \mathbb{R}$ are unaffected by the conjugation, this operation enforces these conditions. This procedure has been utilized before in the analysis of delay differential equations [25], and (in a limited form) in the context of Hopf bifurcation prediction [26], but has never been applied to aeroelastic or other structural stability problems. Equation 3 is nothing other than a multiparameter eigenvalue problem (MEP): an eigenvalue problem in which the eigenvalue point is not simply defined by a scalar and an eigenvector, but by an n-tuple and an eigenvector. A number of methods of analysis have been developed for such problems, and in this paper we will explore some of these. However, as the methods that are available depend strongly on the structure of matrix A, we first define an example system.

3. An Example Section model

Consider first the simple section model shown in Figure 1. This model has two degrees of freedom: plunge h and twist θ. The governing equations for this model are easy to derive; they are:

$$\begin{aligned} m\ddot{h} + d_h\dot{h} + k_h h - mx_\theta\ddot{\theta} &= -L(t), \\ I_P\ddot{\theta} + d_\theta\dot{\theta} + k_\theta\theta - mx_\theta\ddot{h} &= M(t), \end{aligned} \tag{4}$$

where m and I_P are the section mass and polar moment of inertia, k_h and k_θ are the section plunge and twist stiffnesses, d_h and d_θ are the section plunge and twist damping coefficients, and x_θ is the section's dimensional static imbalance – defined as the distance along the x-axis from the pivot point to the centre of mass. For our numerical experiments we take parameter values as per Table 1, representing a system with relatively heavy damping. Taking the Fourier transform, $[h(t), \theta(t)] = \left[\hat{h}, \hat{\theta}\right]e^{\iota\chi t}$, of this model, we obtain:

$$\begin{aligned} (-m\chi^2 + \iota d_h\chi + k_h)\hat{h} + mx_\theta\chi^2\hat{\theta} &= L\left(\chi, \hat{h}, \hat{\theta}\right), \\ mx_\theta\chi^2\hat{h} + (-I_P\chi^2 + \iota d_\theta\chi + k_\theta)\hat{\theta} &= M\left(\chi, \hat{h}, \hat{\theta}\right). \end{aligned} \tag{5}$$

To model the aerodynamic loads in the frequency domain we use Theodorsen's unsteady aerodynamic theory [3]:

$$L = -\chi^2\left(L_h\hat{h} + L_\theta\hat{\theta}\right), \quad M = \chi^2\left(M_h\hat{h} + M_\theta\hat{\theta}\right). \tag{6}$$

The aerodynamic coefficients $\{L_h, L_\theta, M_h, M_\theta\}$ are complex functions of κ – the reduced frequency; an aerodynamic parameter related to the airspeed (U) by $\kappa = b\chi/U$. Other structural and environmental parameters involved are the air density (ρ), the semichord (b) and the distance along the x-axis from the midchord to the pivot point, as a fraction of the semichord (a). See Pons [27] or Hodges and Pierce [3] for details. We assume that the flow over the airfoil is quasisteady: that is, that Theodorsen's function takes a value of 1 universally [1,3]. We will deal with general Theodorsen aerodynamics in a later paper, as this requires iterative or approximate multiparameter solution methods which we will not cover here. We also assume without loss of generality a lift-angle of attack coefficient of $C_{L\alpha} = 2\pi$, as per thin-airfoil theory [28].

It is then customary to nondimensionalise Eq. 5. Further details of this are given in Pons [27]. The final result is a flutter problem of the form

$$\left(\left(\mathrm{M}_0 + \mathrm{G}_0 + \mathrm{G}_1\frac{1}{\kappa} + \mathrm{G}_2\frac{1}{\kappa^2}\right)\chi^2 - \mathrm{D}_0\chi - \mathrm{K}_0\right)\mathbf{x} = \mathbf{0}. \tag{7}$$

with eigenvector $\mathbf{x} = [h/b\,, \theta]$, dimensionless parameters defined as in Table 1 and the nomenclature, and the matrix coefficients

$$\begin{aligned} \mathrm{G}_0 &= \frac{1}{\mu}\begin{bmatrix} 1 & a \\ a & \left(\frac{1}{8} + a^2\right) \end{bmatrix}, \qquad \mathrm{G}_1 = \frac{1}{\mu}\begin{bmatrix} -2\iota & 2\iota(1-a) \\ -\iota(1+2a) & \iota a(1-2a) \end{bmatrix}, \\ \mathrm{G}_2 &= \frac{1}{\mu}\begin{bmatrix} 0 & 2 \\ 0 & 1+2a \end{bmatrix}, \qquad \mathrm{M}_0 = \begin{bmatrix} 1 & -r_\theta \\ -r_\theta & r^2 \end{bmatrix}, \end{aligned} \tag{8}$$

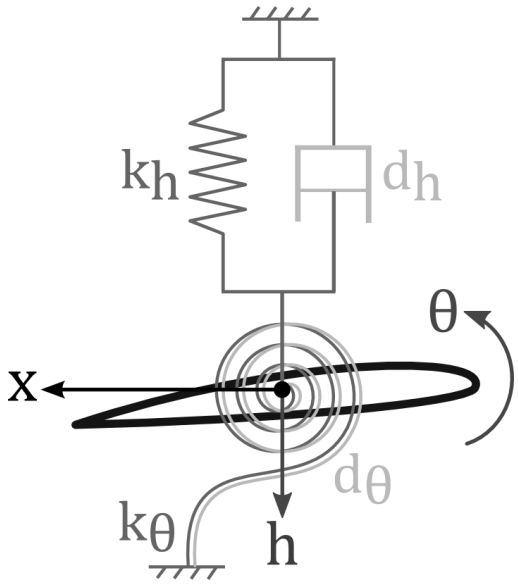

Figure 1: Diagram of section model

Table 1: Dimensionless parameter values for the section model

Parameter	Value
mass ratio – μ	20
radius of gyration – r	0.4899
bending nat. freq. – ω_h	0.5642 rad/s
torsional nat. freq. – ω_θ	1.4105 rad/s
bending damping – ζ_h	14.105 %
torsional damping – ζ_θ	11.75 %
static imbalance – r_θ	-0.1
pivot point location – a	-0.2

$$\mathrm{D}_0 = \begin{bmatrix} 2\imath\zeta_h\omega_h & 0 \\ 0 & 2\imath r^2\zeta_\theta\omega_\theta \end{bmatrix}, \qquad \mathrm{K}_0 = \begin{bmatrix} \omega_h^2 & 0 \\ 0 & r^2\omega_\theta^2 \end{bmatrix}, \tag{8}$$

Note that this nondimensionalisation is a convenience and not necessary prerequisite for using multiparameter solution methods. However something that will be of great use is to rearrange Eq. 7 into two polynomial forms by defining new eigenvalue parameters: $\Upsilon = U/b$, $\tau = 1/\kappa$, and $\lambda = 1/\chi$. These two forms are then

$$\left((\mathrm{M}_0 + \mathrm{G}_0)\chi^2 + \mathrm{G}_1\Upsilon\chi + \mathrm{G}_2\Upsilon^2 - \mathrm{D}_0\chi - \mathrm{K}_0\right)\mathbf{x} = \mathbf{0}, \tag{9}$$

which we term the Υ-χ form, and

$$\left((\mathrm{M}_0 + \mathrm{G}_0) + \mathrm{G}_1\tau + \mathrm{G}_2\tau^2 - \mathrm{D}_0\lambda - \mathrm{K}_0\lambda^2\right)\mathbf{x} = \mathbf{0}, \tag{10}$$

which we term the τ-λ form. Both are preferable to the k-χ form, Eq. 7. Via the method introduced in Section 2, we obtain MEPs for these two flutter problems:

$$\begin{aligned} \left((\mathrm{M}_0 + \mathrm{G}_0) + \mathrm{G}_1\tau + \mathrm{G}_2\tau^2 - \mathrm{D}_0\lambda - \mathrm{K}_0\lambda^2\right)\mathbf{x} &= \mathbf{0}, \\ \left((\overline{\mathrm{M}}_0 + \overline{\mathrm{G}}_0) + \overline{\mathrm{G}}_1\tau + \overline{\mathrm{G}}_2\tau^2 - \overline{\mathrm{D}}_0\lambda - \overline{\mathrm{K}}_0\lambda^2\right)\bar{\mathbf{x}} &= \mathbf{0}, \end{aligned} \tag{11}$$

and

$$\begin{aligned} \left((\mathrm{M}_0 + \mathrm{G}_0)\chi^2 + \mathrm{G}_1\Upsilon\chi + \mathrm{G}_2\Upsilon^2 - \mathrm{D}_0\chi - \mathrm{K}_0\right)\mathbf{x} &= \mathbf{0}, \\ \left((\overline{\mathrm{M}}_0 + \overline{\mathrm{G}}_0)\chi^2 + \overline{\mathrm{G}}_1\Upsilon\chi + \overline{\mathrm{G}}_2\Upsilon^2 - \overline{\mathrm{D}}_0\chi - \overline{\mathrm{K}}_0\right)\bar{\mathbf{x}} &= \mathbf{0}. \end{aligned} \tag{12}$$

These systems are all polynomial multiparameter eigenvalue problems [29]. A number of solution methods are known for such systems.

4. Direct solution via linearization

4.1. Linearization

Any polynomial MEP can be made into a linear one (consisting of a zeroth-order term and first-order terms in each eigenvalue) via the process of linearization [30]. This process bears resemblance to the linearization of single-parameter polynomial eigenvalue problems; a

process which is well-known [31]. Multiparameter linearization is relevant because a number of direct solution methods exist for linear MEPs. It is carried out by expanding the system eigenvector by concatenating appropriate products of the eigenvalue polynomials and the original eigenvector. For example, any MEP equation of the form

$$(\mathrm{A} + \mathrm{B}\lambda + \mathrm{C}\tau + \mathrm{D}\lambda\tau + \mathrm{E}\lambda^2 + \mathrm{F}\tau^2)\mathbf{x} = \mathbf{0}, \tag{13}$$

can be linearized as

$$\left(\begin{bmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} \\ 0 & -I_n & 0 \\ 0 & 0 & -I_n \end{bmatrix} + \begin{bmatrix} 0 & \mathrm{D} & \mathrm{E} \\ I_n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\lambda + \begin{bmatrix} 0 & 0 & \mathrm{F} \\ 0 & 0 & 0 \\ I_n & 0 & 0 \end{bmatrix}\tau\right)\begin{bmatrix} \mathbf{x} \\ \lambda\mathbf{x} \\ \tau\mathbf{x} \end{bmatrix} = \mathbf{0}. \tag{14}$$

The upper block row of Eq. 14 represents Eq. 13, and the lower rows identities such as $I_n(\tau\mathbf{x}) = \tau I_n\mathbf{x}$.. Note that other linearizations of similar form are possible. For Eq. 13 we have

$$\begin{array}{ll} \mathrm{A} = \mathrm{M}_0 + \mathrm{G}_0 & \mathrm{D} = -\mathrm{K}_0 \\ \mathrm{B} = \mathrm{G}_1 & \mathrm{E} = 0 \\ \mathrm{C} = -\mathrm{D}_0 & \mathrm{F} = \mathrm{G}_2 \end{array} \tag{15}$$

and for Eq. 14, changing $\lambda \to \chi$ and $\tau \to \Upsilon$,

$$\begin{array}{ll} \mathrm{A} = -\mathrm{K}_0 & \mathrm{D} = \mathrm{M}_0 + \mathrm{G}_0 \\ \mathrm{B} = -\mathrm{D}_0 & \mathrm{E} = \mathrm{G}_1 \\ \mathrm{C} = 0 & \mathrm{F} = \mathrm{G}_2 \end{array} \tag{16}$$

These linearised systems have matrix coefficients triple the size of the original problem. Note that it is not actually necessary to linearize the second (i.e. conjugate) equation of the aeroelastic system, because the linearized conjugate equation will be the conjugate of the linearized first equation. This is a property of this method of linearization.

4.2. Direct solution

Consider Eq. 11. Post-multiplying the first equation in this system by $\bar{\mathrm{C}}\mathbf{y}$ and premultiplying the second by $\mathrm{C}\mathbf{x}$, we have

$$\begin{array}{l} (\mathrm{A} + \mathrm{B}\Lambda + \mathrm{C}\tau)\mathbf{x} \otimes (\bar{\mathrm{C}}\mathbf{y}) = 0, \\ (\mathrm{C}\mathbf{x}) \otimes (\bar{\mathrm{A}} + \bar{\mathrm{B}}\Lambda + \bar{\mathrm{C}}\tau)\mathbf{y} = 0. \end{array} \tag{17}$$

Equating these two equations and cancelling terms in τ we obtain:

$$\Delta_1\mathbf{z} = \Lambda\Delta_0\mathbf{z} \tag{18}$$

with an enlarged eigenvector $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$ and the three possible operator determinants

$$\begin{array}{l} \Delta_0 = \mathrm{B} \otimes \bar{\mathrm{C}} - \mathrm{C} \otimes \bar{\mathrm{B}} \\ \Delta_1 = \mathrm{C} \otimes \bar{\mathrm{A}} - \mathrm{A} \otimes \bar{\mathrm{C}} \\ \Delta_2 = \mathrm{A} \otimes \bar{\mathrm{B}} - \mathrm{B} \otimes \bar{\mathrm{A}}. \end{array} \tag{19}$$

Eq. 18 is a generalized eigenvalue problem (GEP), in the single parameter λ. Solvers for the generalized eigenvalue problem are very widely available. If the linear system has square coefficients of size m then the operator determinants are of size m^2.

The operator determinants can also be used to define a second GEP in τ. By multiplying the first and second equations of Eq. 11 by $\bar{\mathrm{B}}\mathbf{y}$ and $\mathrm{B}\mathbf{x}$ respectively, we can also show that:

$$\Delta_2\mathbf{z} = \tau\Delta_0\mathbf{z}. \tag{20}$$

However, it is only necessary to solve one of the operator determinant GEPs: once one has been solved, then its solutions can be substituted back into the original polynomial system, which yields another smaller GEP. The modeshapes at flutter are also given by the eigenvectors of this GEP; or alternatively they may be decomposed from the solutions in $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$. The eigenvalue, if simple, may also be computed via Rayleigh quotients in these eigenvectors; $\mathbf{z}$ or $\mathbf{x}$ and $\mathbf{y}$. The system is thus completely solved for its flutter points. This direct solution method is known in mathematical literature as the *operator determinant method*. Its computational complexity is $\mathcal{O}(n^6)$, irrespective of whether n is the size of the linear coefficients (A, B, etc.) or the polynomial coefficients (M_0, G_0, etc.) [30,32,33]. This large complexity arises from solving the generalized eigenvalue problem, an $\mathcal{O}(m^3)$ process by the QZ algorithm [34], with operator determinants of size $m = \mathcal{O}(n^2)$. The operator determinant method has not previously been used in aeroelasticity, and has only rarely seen engineering application in the study of dynamic model updating [35,36].

4.3. Singularity

One important caveat of the operator determinant approach is that the matrix Δ_0 must not be singular. If it is, then the eigenvalues of the original polynomial system (e.g. Eq. 11-12) will not generally coincide with those of the GEPs of the linearized problem (Eq. 18 and 20) [30,37,38]. A linear MEP with singular Δ_0 is said to be singular MEP. A large proportion of the linear flutter problems that arise in the study of aircraft aeroelasticity are singular. This is largely because the linearization of polynomial problems tends to generate singular linear problems, even if the original polynomial problem has all its matrix coefficients at full rank – compare Eq. 14. In many circumstances we will be dealing with a singular problem, and for some time the lack of a working solver for such singular problems has been an obstacle to the application of multiparameter methods to real-world problems.

However, recently a solution has been proposed. Muhič and Plestenjak [29] proved that the eigenvalues of a polynomial system are equivalent to the finite regular eigenvalues of the pair of singular operator determinant GEPs constructed via linearization. The finite regular eigenvalues of a linear two-parameter system are the pairs $(\lambda,\ \mu)$ such that $i \in \{1,2\}$ [37]:

$$\operatorname{rank}(\mathrm{A}_i + \mathrm{B}_i\lambda + \mathrm{C}_i\mu) < \max_{(s,t)\in\mathbb{C}^2} \operatorname{rank}(\mathrm{A}_i + \mathrm{B}_i s + \mathrm{C}_i t), \tag{21}$$

that is, the finite regular eigenvalues are the set of points that cause the singular problem to have its maximum rank, even though this is not full rank. On the basis of this proof, Muhič and Plestenjak [29] devised a set of algorithms which would extract the common regular part of the singular matrix pencils (i.e. matrix-valued functions polynomial in a variable λ) $\Delta_{1\mathrm{s}} - \lambda\Delta_{0\mathrm{s}}$ and $\Delta_{2\mathrm{s}} - \lambda\Delta_{0\mathrm{s}}$. This common regular part is represented by two smaller nonsingular matrix pencils ($\Delta_1 - \lambda\Delta_0$ and $\Delta_2 - \lambda\Delta_0$), the eigenvalues of which are the finite regular eigenvalues of the singular problem and thus the eigenvalues of the polynomial problem. In practical terms, this can be seen as a compression of the singular operator determinant matrices into smaller full-rank matrices. The operator determinant method, as presented in Section 4.2, can be applied to these compressed pencils. The algorithms involved in the extraction of the common regular part are presented in [29] and published also in MATLAB code [39]. We will not detail them here as they are complex.

5. Direct solution via Quasi-linearization

5.1. Quasi-linearization

Hochstenbach et al. [30] recently presented another method of linearization for polynomial MEPs. Instead of increasing the size of the coefficient matrices, this method of linearization increases the number of parameters and equations in the system. To differentiate it from strict linearization, Hochstenbach et al. [30] term this new method *quasi-linearization*. The process is as follows.

Considering Eq. 11, we define two new eigenvalue parameters, $\alpha = \tau^2$ and $\beta = \lambda^2$.The equations then become

$$\begin{aligned} \big((\mathrm{M}_0 + \mathrm{G}_0) + \mathrm{G}_1\tau + \mathrm{G}_2\alpha - \mathrm{D}_0\lambda - \mathrm{K}_0\beta\big)\mathbf{x} = \mathbf{0}, \\ \big((\overline{\mathrm{M}}_0 + \overline{\mathrm{G}}_0) + \overline{\mathrm{G}}_1\tau + \overline{\mathrm{G}}_2\alpha - \overline{\mathrm{D}}_0\lambda - \overline{\mathrm{K}}_0\beta\big)\bar{\mathbf{x}} = \mathbf{0}, \end{aligned} \tag{22}$$

which is now a linear four-parameter eigenvalue problem, but with only two equations. To constrain the system, we recast the nonlinear relations $\alpha - \tau^2 = 0$ and $\beta - \lambda^2 = 0$ as determinants:

$$\det\left(\begin{bmatrix} \alpha & \tau \\ \tau & 1 \end{bmatrix}\right) = 0, \qquad \det\left(\begin{bmatrix} \beta & \lambda \\ \lambda & 1 \end{bmatrix}\right) = 0. \tag{23}$$

These determinant conditions correspond to simple linear MEPs, which can be used to constrain the system. The result is the four-parameter eigenvalue problem:

$$\begin{aligned} &\big((\mathrm{M}_0 + \mathrm{G}_0) + \mathrm{G}_1\tau + \mathrm{G}_2\alpha - \mathrm{D}_0\lambda - \mathrm{K}_0\beta\big)\mathbf{x} = \mathbf{0}, \\ &\big((\overline{\mathrm{M}}_0 + \overline{\mathrm{G}}_0) + \overline{\mathrm{G}}_1\tau + \overline{\mathrm{G}}_2\alpha - \overline{\mathrm{D}}_0\lambda - \overline{\mathrm{K}}_0\beta\big)\bar{\mathbf{x}} = \mathbf{0}, \\ &\left(\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\tau + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\alpha\right)\mathbf{y_1} = \mathbf{0}, \\ &\left(\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\lambda + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\beta\right)\mathbf{y_2} = \mathbf{0}, \end{aligned} \tag{24}$$

In a similar way, we can linearize Eq. 12 with the definitions $\alpha = \chi^2$, $\beta = \Upsilon^2$, $\gamma = \Upsilon\chi$:

$$\begin{aligned} &\big((\mathrm{M}_0 + \mathrm{G}_0)\alpha + \mathrm{G}_2\beta + \mathrm{G}_1\gamma - \mathrm{D}_0\chi - \mathrm{K}_0\big)\mathbf{x} = \mathbf{0}, \\ &\big((\overline{\mathrm{M}}_0 + \overline{\mathrm{G}}_0)\alpha + \overline{\mathrm{G}}_2\beta + \overline{\mathrm{G}}_1\gamma - \overline{\mathrm{D}}_0\chi - \overline{\mathrm{K}}_0\big)\mathbf{x} = \mathbf{0}, \\ &\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\alpha + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\chi + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right)\mathbf{y_1} = \mathbf{0}, \\ &\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\alpha + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\beta + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\gamma\right)\mathbf{y_2} = \mathbf{0}, \end{aligned} \tag{25}$$

These systems can be solved via the operator determinant method, in a more general form than that presented in Section 4.2.

5.2. The general operator determinant method

The general form of a nonhomogeneous MEP is

$$W_i(\boldsymbol{\eta})\mathbf{x}_i = \mathrm{A}_{i0}\mathbf{x}_i + \sum_{j=1}^{N} \eta_j \mathrm{A}_{ij}\mathbf{x}_i\,, \qquad i = 1, \dots, n \tag{26}$$

where $\boldsymbol{\eta}$ is the vector of eigenvalues (η_j being the individual eigenvalues), A_{ij} are the coefficient matrices, which can be complex and of different sizes for each equation, and $\mathbf{x}_i$ are the eigenvectors. Eq. 26 can be visualized as a non-square array of matrices:

$$\begin{bmatrix} \mathrm{A}_{10} & \mathrm{A}_{11} & \mathrm{A}_{12} & \cdots & \mathrm{A}_{1N} \\ \mathrm{A}_{20} & \mathrm{A}_{21} & \mathrm{A}_{22} & \cdots & \mathrm{A}_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{A}_{N0} & \mathrm{A}_{N1} & \mathrm{A}_{N2} & \cdots & \mathrm{A}_{NN} \end{bmatrix}. \tag{27}$$

In a process analogous to that presented in Section 4.4, we can construct a series of operator determinants for this system. There are $N+1$ such operator determinants (Δ_0 through to Δ_N), of size n^N for constant coefficient size n. They correspond to taking determinants of Eq. 27, with certain columns removed or inserted, and with the normal scalar multiplication operation replaced by a Kronecker product between matrices. Definitions and derivations of these determinants may be found in [40–44], and one software implementation in [39]. In the two-parameter case this analysis collapses into that of Section 4.2.

These general operator determinants allow us to compute the eigenvalues of Eq. 26 via generalized eigenvalue problems. Providing Δ_0 is nonsingular, it holds that

$$\Delta_i \mathbf{x} = \eta_i \Delta_0 \mathbf{x}. \tag{28}$$

In this way we are able to solve the quasi-linearized systems given in Section 5.1. However, note that already by using this quasi-linearization process we have made the operator determinants smaller than they were with standard linearization. For Eq. 11-12, the operator determinants are square and of size $4n^2$. With standard linearization they are of size $9n^2$.

5.3. Computing operator determinants

In the two-parameter case, the computation of the operator determinants is trivial. However in the case of larger systems we must find some other approach. In the general case, the operator determinants may be computed by modifying the Leibniz formula for the determinant [40]:

$$\Delta(\mathrm{M}) = \sum_{\mathbf{s} \in S_N} \mathrm{sgn}(\mathbf{s}) \bigotimes_{i=1}^{N} \mathrm{M}_{i,\mathbf{s}_i}, \tag{29}$$

where S_N is the set of permutations of the set $\{1, \ldots, N\}$, $\mathrm{sgn}(\mathbf{s})$ is the sign of the permutation vector $\mathbf{s} \in S_N$. M is the square matrix array (a modification of Eq. 27) corresponding to the operator determinant desired. The notation $\bigotimes_i^N$ denotes the repeated application of the Kronecker product. Note that the tensor determinant definition in [40] is slightly erroneous as the factor $(-1)^{\mathrm{sgn}(\sigma)}$ in their tensor determinant expressions should be either $\mathrm{sgn}(\sigma)$ or $(-1)^{N(\sigma)}$, where $N(\sigma)$ is the number of inversions in σ. Alternatively, Muhič and Plestenjak [39] devised an operator determinant Laplace expansion [45] that is able to compute the operator determinants of a system of arbitrary size, via a process of recursion:

$$\Delta(\mathrm{M}) = \sum_{i=1}^{N} (-1)^{i+1} \mathrm{M}_{i1} \otimes m_{i1}, \tag{30}$$

where m_{i1} denotes the minor entry corresponding to M_{i1}. The summation in Eq. 30 follows the first column of M, though, of course, many other summation paths could be used. It should be noted that the use of the Laplace or Leibniz methods for computing the determinant of an ordinary matrix have very high non-polynomial computational complexity costs – $\mathcal{O}(n!)$ and $\mathcal{O}(n!\, n)$, respectively [46,47]. Factorization approaches with lower computational cost are not yet available for operator determinants, and whether they are possible has not yet been established.

5.4. Singularity

One major advantage of the direct solver based on quasi-linearization is that it generates linearized problems that are not always singular – e.g. for full rank coefficient matrices, Eq. 24-25 are in general nonsingular. However in our system these linearizations are still nonsingular, as G_2 is not full rank. While the singularity theory of Muhič and Plestenjak [29] has never been extended to MEPs with $N > 2$, we do have a practical method of solving such systems. Numerical experiments (some of which are detailed in Section 7) indicate that, although the compression process of Section 4.2 takes only Δ_0, Δ_1 and Δ_2 as an input, it will successfully compress these three operator determinants for a general MEP, ignoring all the others. In the case of Eq. 28-30, with these three operator determinants we can solve for all the eigenvalue parameters of the system, irrespective of the eigenvalue ordering.

There is as yet no theoretical justification for this procedure, other than the experimental evidence – confirmed also by the nonrigourous $N > 2$ compression process utilized in [39]. The common regular part relationship proved by [29] applies only to two-parameter eigenvalue problems arising from linearization, and indeed it is not clear how the concept of the common regular part would extend to an $N > 2$ problem with more than two pencils. For the purposes of this paper, however, we will use the compression algorithm for the quasi-linearized Δ_0, Δ_1 and Δ_2 without proof.

6. Instability boundary characterization

So far we have considered methods for the direct computation of a flutter point or more general stability boundary. However in many industry problems further details about the stability boundary are required. In aeroelasticity, two local characteristics are particularly important: the direction of the stability change (destabilizing / restabilizing) and the 'hardness' of the flutter event (the rate of modal damping change in the flutter point vicinity, representing the suddenness of flutter onset) [48]. We can analyze both of these via an extension to our multiparameter analysis. In the original generalized problem, Eq. 2 with $\chi \in \mathbb{C}$, we split $\chi = \chi_R + \iota\chi_I$, where χ_R is the damped modal frequency and χ_I the modal damping (for the convention $\hat{\mathbf{x}}(t) = \mathbf{x}\exp(\iota\chi t)$). This yields an equation

$$\mathrm{A}(\chi_R, \chi_I, p)\mathbf{x} = \mathbf{0} \tag{31}$$

which, for a given χ_I, may transformed into an MEP in χ_R and p and solved. Solutions in χ_R-p for $\chi_I = 0$ are exact flutter points, the spectrum of the MEP; but those for small χ_I represent near-flutter points along the fluttering mode. Note these near-spectra are similar in concept to the pseudospectra of Trefethen [49], applied to MEPs by Hochstenbach and Plestenjak [38], but have several significant practical differences. For one, ours are generally discrete and exist only on mode paths, whereas Trefethen's are continuous and are unrestricted in domain. Ours also allow a characterization of a flutter point by two evaluations of the pseudospectral MEP: at $\chi_I = +\delta\chi$ and at $\chi_I = -\delta\chi$, for some small positive $\delta\chi$. The arrangement of pseudospectral points at p-values immediately above or below a flutter point then characterizes it as destabilizing, restabilizing or stable / unstable borderline; as given in Figure 2. The existence of alternate pseudospectra ($\pm\delta\chi$) on alternate sides of a flutter point implies a destabilizing or restabilizing point, as determined by the order of the points; whereas the existence of only one pseudospectrum in the locality implies a flutter point which only touches the stability boundary momentarily. We term the latter stable or unstable borderline points, depending on the predominant stability throughout.

For restabilizing or destabilizing flutter points it is then possible to estimate the local modal damping gradient using a central difference:

$$\frac{\partial \chi_I}{\partial p} = \frac{2\delta\chi}{p(\chi_I + \delta\chi) - p(\chi_I - \delta\chi)} \tag{32}$$

This gradient then allows a characterization of the flutter point's 'hardness', relative to the other flutter points in the system: a larger gradient represents increased hardness and thus an increased likelihood of damage and decreased level of warning beforehand. Note that it is possible to compute this gradient (and characterize the flutter point) via any perturbation in the pseudospectral MEP: in p, χ_R or χ_I. However the perturbation in χ_I has the advantage of only requiring two evaluations for the entire system ($\chi_I = \pm\delta\chi$), whereas the other parameters require two evaluations per flutter point analyzed (e.g. $p = p_F \pm \delta p$ for each flutter value p_F).

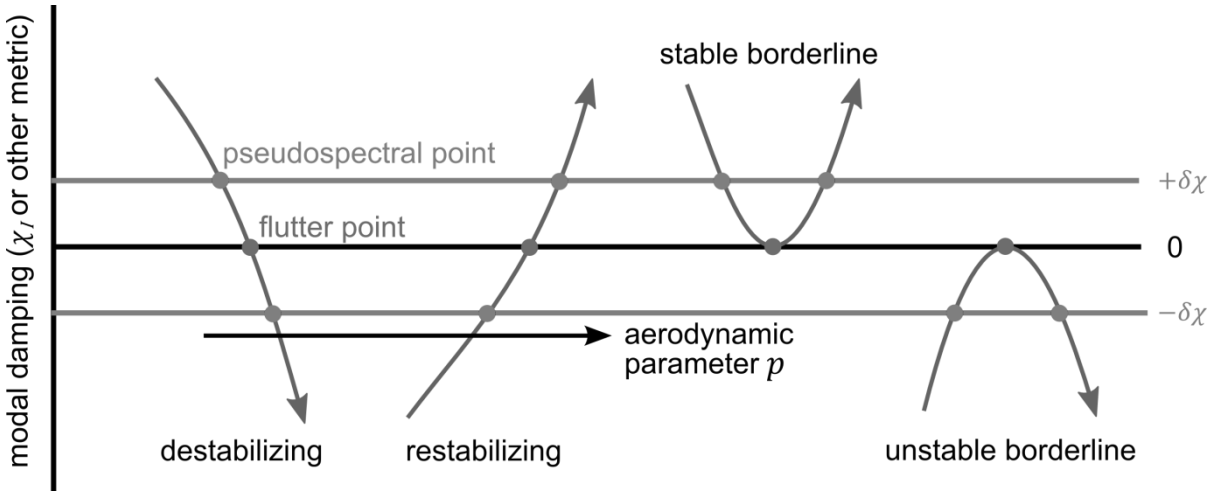

Figure 2: Characterization of flutter points based on two pseudospectral evaluations.

7. Numerical experiments

7.1. System flutter points

We are now in a position to compute the flutter points of the section model introduced in Section 3. We have devised two direct solvers to do so: a direct solver with linearization and a direct solver with quasi-linearization. To validate these solution methods, we first produce a modal damping plot (Figure 3), in the Υ-χ form. Three stability boundaries may be observed: the destabilizing flutter point at $\Upsilon = 2.79$ Hz, $\chi = 1.01$ rad/s; a destabilizing divergence point at $\Upsilon = 3.99$ Hz ($\chi = 0$); and a restabilizing flutter point $\Upsilon = 10.7$ Hz, $\chi = 0.508$ rad/s. Figure 3 also shows the modeshape bending / torsion amplitude ratio, $|h/b\theta|$, corresponding to the ratio of the eigenvector ($\mathbf{x}$) components in Eq. 9-10. The modeshape trends are interesting; the higher-frequency tends from a torsion (θ) to a plunge (h) dominance, and the lower-frequency mode the reverse. The first flutter point is mildly torsionally dominant; whereas the divergence point is plunge dominant and the restabilizing point occurs almost entirely in plunge.

We then compute the flutter points of the system via our two direct solution methods, yielding flutter points identical to those detailed above for the presented accuracy. With the solver based on linearization, the uncompressed operator determinants are of size 36 and the compressed ones of size 12. With the solver based on quasi-linearization the size of the uncompressed determinants can be reduced significantly – down to 16 – while producing

compressed determinants of the same size. The quasi-linearization method thus reduces the time required for the compression process, as the finite-regular part extraction algorithm works by successively increasing the rank of the system (not all at once). Further investigation of this and other computation time effects will be carried out in Section 6.2. Figure 4 shows the Υ-χ system flutter points, computed by the two direct solvers (yielding identical results) and superimposed on a contour plot [27]. The flutter points computed with direct solvers agree exactly with those seen on the contour plot (the intersections of the real and imaginary contours), and the results of the modal damping plot in Figure 3. Via the MEP eigenvectors (Section 4.2) the first two flutter points can be identified as being slightly torsionally dominant ($|h/b\theta| = 0.855,\ 5.00$) and the restabilizing point as almost entirely plunging ($|h/b\theta| = 18.3$). This agrees with the results from the modal damping plot.

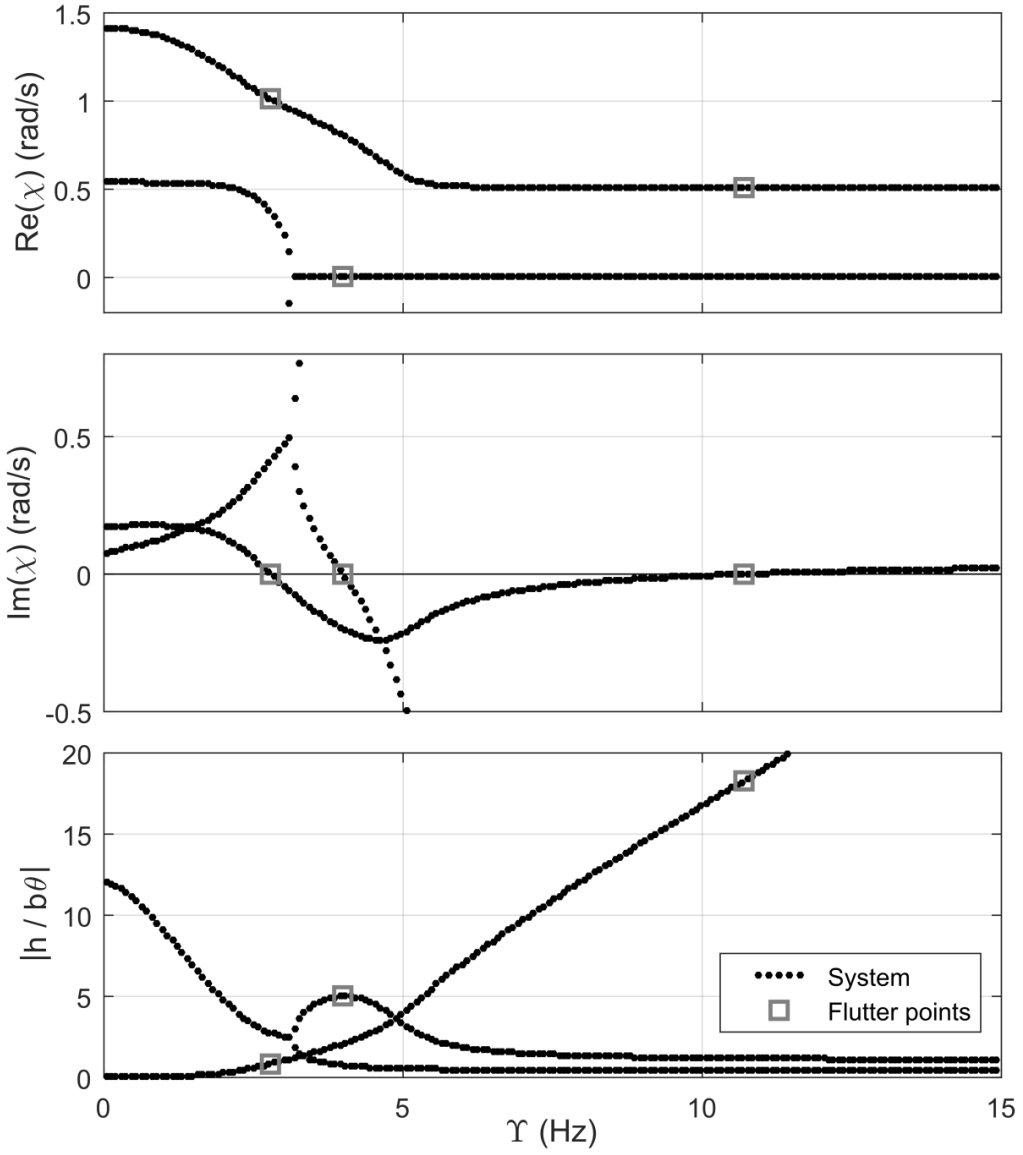

Figure 3: Modal damping plot for the Υ-χ section model, including also the modeshape bending / torsion ratio, $|h/b\theta|$.

We may analyze the three computed points via further the pseudospectral procedure in Section 6, requiring two further evaluations of the MEP. This characterizes the $\Upsilon = 2.79$ Hz flutter point as destabilizing with gradient $\partial\chi_I/\partial\Upsilon = 0.40$ rad, the $\Upsilon = 3.99$ Hz divergence point as destabilizing with gradient 0.65, and the $\Upsilon = 10.7$ Hz flutter point as restabilizing with gradient 0.015. As would be physically expected, the divergence event is more sudden that the first flutter point. The restabilizing point is very soft, implying that this mode will be only barely stable for a significant airspeed range afterwards. This gives us as much information about the local conditions of these flutter points as the modal damping plot itself (Figure 3), at significantly less computational expense.

Figure 5 shows the system in the τ-λ form, with the flutter points again computed via the multiparameter direct solvers. The first flutter point can be observed at $\tau = 2.76$, $\lambda = 0.990$ s/rad, and the restabilisation point at $\tau = 21.1$, $\lambda = 1.97$ s/rad, both of which agree with the Υ-χ results ($\lambda = 1/\chi$ and $\tau = \Upsilon/\lambda$). Note that the divergence point of this system does not appear on this plot, as it occurs at infinite τ and λ. However, the computation of divergence points does not require multiparameter methods anyway, as if the frequency χ is assumed to be zero in the governing equation (e.g. Eq. 2) then the problem becomes a single parameter eigenvalue problem in the airspeed parameter. This problem can then be solved with single-parameter solvers.

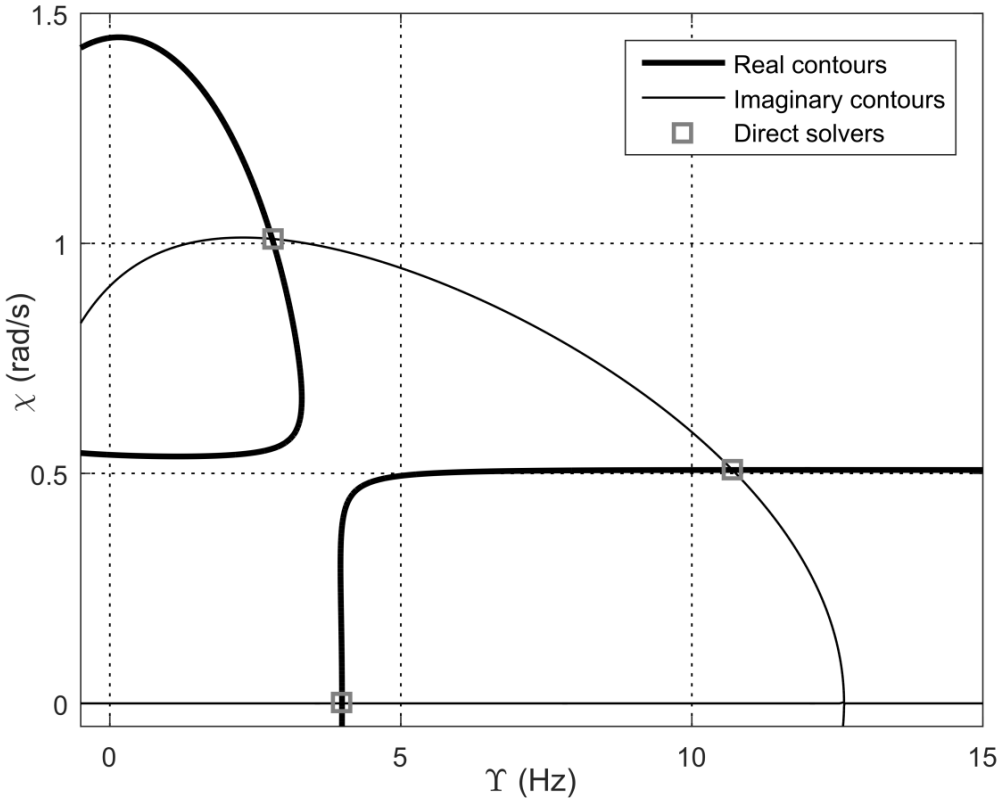

Figure 4: Contour plot for the section model (Υ-χ form), showing identical solutions from the two direct solvers

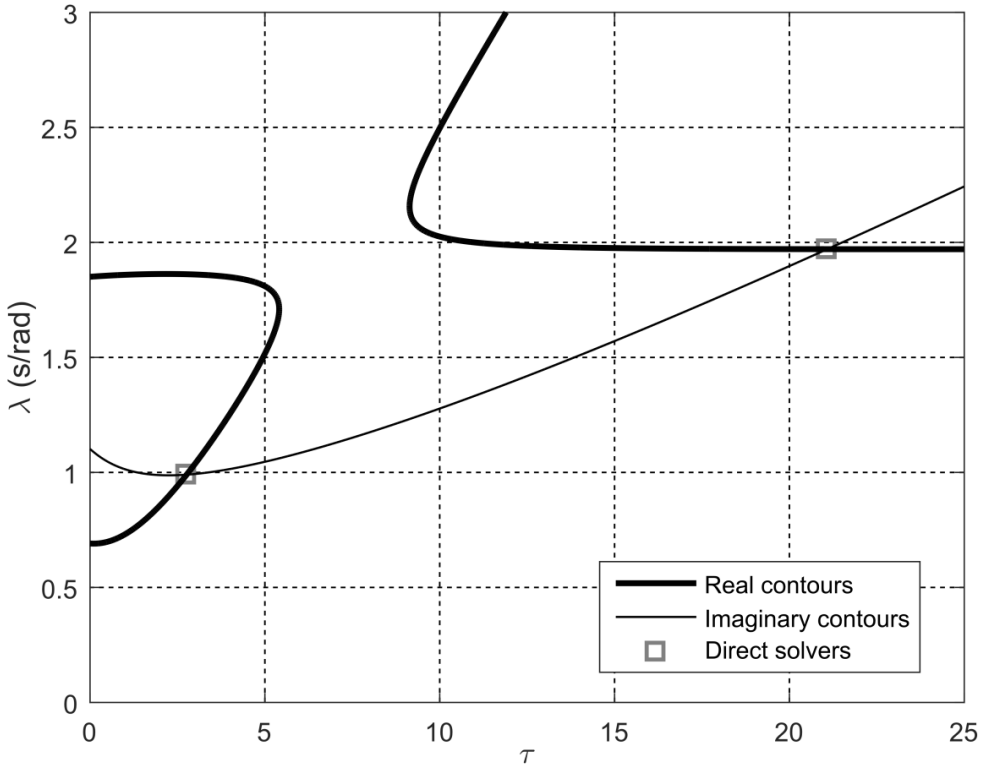

Figure 5: Contour plot for the section model (τ-λ form), showing identical solutions from the two direct solvers.

7.2. Computation time

Given the high computational complexity of these direct solution methods – $\mathcal{O}(n^6)$ – we are interested in the maximum system size for which a direct solution is practical. We have already found that for $n = 2$ the computational effort required is tiny, and so at the very least these direct solvers are useful for small reduced-order models as might be used in a preliminary design analysis. This alone is of use, as the directness of these solvers makes them ideal for use in optimization routines or other applications in which a large number of computations must be performed with limited user guidance.

To gain a better understanding of the computational complexity characteristics of our methods, we simulate a series of systems of increasing matrix coefficient size. We generate random complex-valued matrices for the polynomial coefficients (M_0, G_0, etc.), of size $n = 2^k$ with $k \in \{1, 2, \ldots, 5\}$. For robustness, we average the results for $k = 1$ over 50 random matrices, for $k = 2$ over 10 random matrices, for $k = 3$ over 5 random matrices; and for systems larger than this we generate only one matrix. Figure 6 shows the wall-clock solution time against system size for the direct solver with linearization, for a 64-bit Intel i7-4770 with 3.4 GHz processor and 16 GB RAM, running MATLAB R2014b. Note that the λ solution time denotes the time required to compute the λ components of the solution via a series of one-parameter eigenvalue problems. As can be seen, the compression process is the most expensive component of the algorithm, making up a constant fraction of about 65% of the total computation time over the entire range of n. The GEP solution time is initially completely negligible but becomes more significant as system size increases: by $n = 32$ it makes up 34% of the total computation time. The λ-solution and setup process are never of much significance.

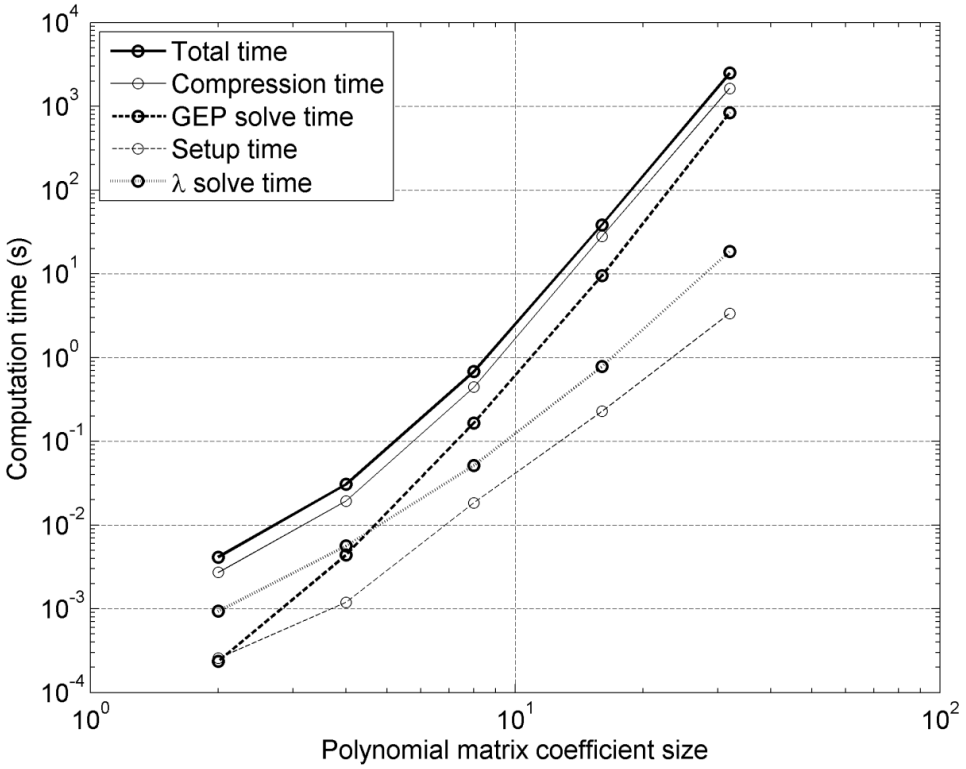

Figure 6: Wall-clock solution time against system size for the direct solver with linearization.

We expect from computational complexity theory that the gradient of the GEP-solution time curve in log-log units will be approximately 6.0 (Section 4.2). Fitting a linear curve through this data yields a gradient of 5.46, with an R^2 of 0.994. However, the GEP solution-time curve is slightly concave, with a maximum gradient of 6.47. The algorithm can then be said to have complexity $\mathcal{O}(n^6)$ overall. Figure 7 shows the wall-clock solution time against system size for the direct solver with quasi-linearization. This simulation was run on the same Intel i7-4770 platform with MATLAB R2014b. The random coefficient matrices used are identical to those in Figure 6.And although now the quasi-linearized system in no longer singular, we still apply the compression algorithm to capture some of its overhead costs.. As can be seen, these costs are not large but are still more significant than the GEP and λ solution times for some system sizes.

These GEP and λ solution times are effectively identical to those of the direct solver with linearization, as the algorithm is solving a GEP of exactly the same size, yielding exactly the same eigenvalues. However, the most striking aspect of Figure 7 is the fact that the setup time occupies the majority of the required computation time right up to $n = 16$ (after which it is surpassed by the GEP solution time). The setup process involved using defining the system array (Eq. 32) in a cell array, and computing the necessary operator determinants of this array (Δ_0 and Δ_2) using the modified Leibniz formula (Eq. 34). On investigating these two procedures we find that it is the computation of the Kronecker products within the operator determinant which occupies the greatest time. However, despite this, the actual computational complexity of the algorithm as a whole is lower than that of the direct solver with linearization. GEP solution time is still approximately $\mathcal{O}(n^6)$, though the concavity that was noted in Figure 6 is still present.

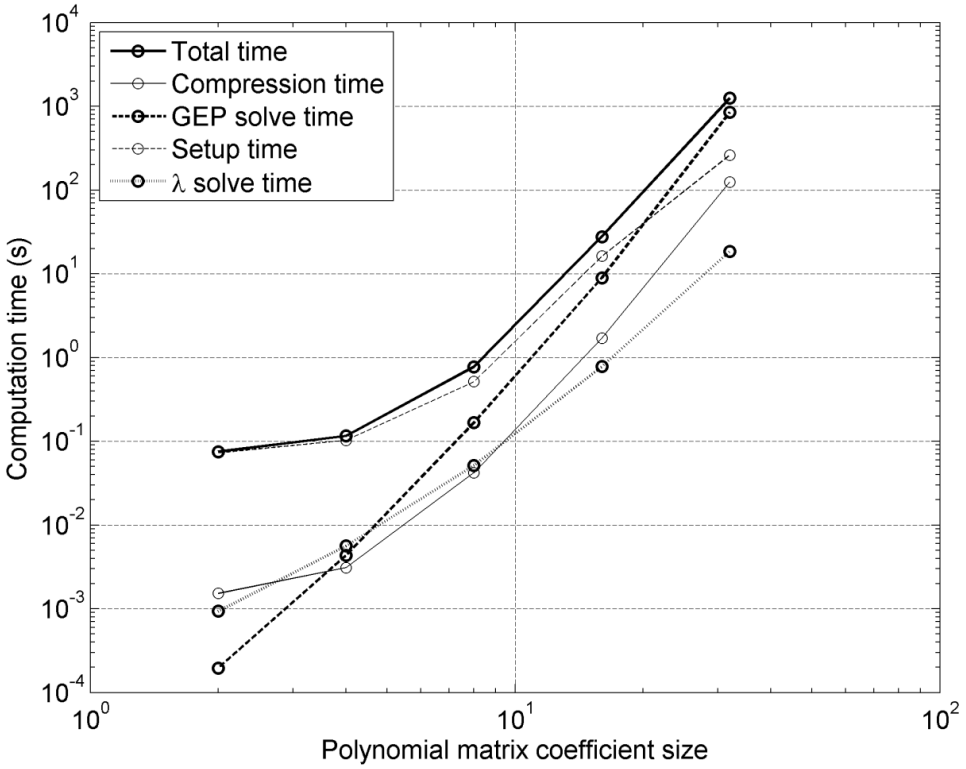

Figure 7: Wall-clock solution time against system size for the direct solver with quasi-linearization.

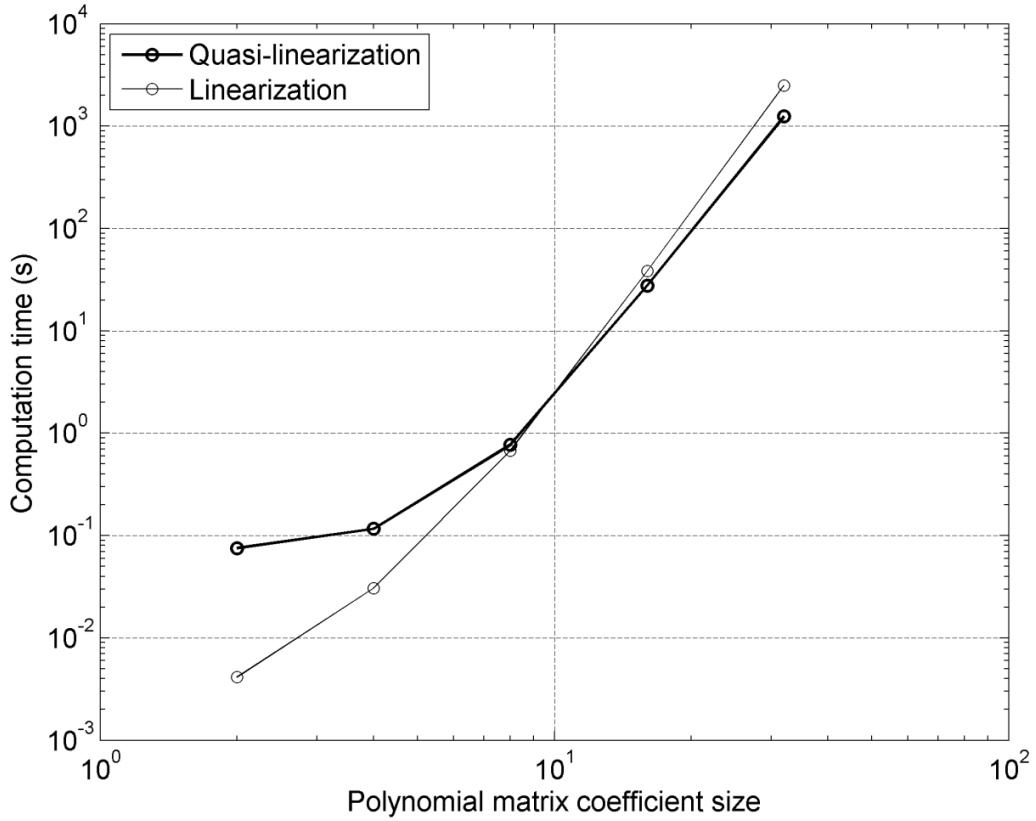

Figure 8: Total wall-clock solution time against system size for the two direct solver variants.

Finally, Figure 8 shows a comparison of the total computation times for the two algorithms. As can be seen, below a system coefficient size of approximately 10, linearisation is faster than quasi-linearisation (at $n = 2$, over an order of magnitude faster). Above this coefficient size, quasi-linearisation is faster (at $n = 32$, twice as fast). To keep computation times below 10s, the system size must be below 11; to keep it below 1s, 8, and below 0.1s, 5. As it stands, the operator determinant method is not suitable for use with finite-element models or any other systems with a large number of degrees of freedom. It is, however, useful for the solution of simpler reduced order models, e.g. in a preliminary design optimization. For such applications it provides an unparalleled direct solver for the system flutter points, with a flutter point characterization routine (describing the stability trend of the system and its gradient) at minimal extra cost.

8. Discussion

8.1. Alternative solution methods

We have so far discussed the operator determinant method – a solution method which is direct but computationally expensive. However, two other classes of method are also available. The Sylvester-Arnoldi type methods are a set of closely-related algorithms that are only valid for two-parameter problems, but are fast and can handle large systems [50]. They still use operator determinants to reformulate the system into a set of generalized eigenvalue problems, but instead of solving this set of GEPs directly, the solution procedure is optimized based on the given knowledge that the operator determinants consist of a difference of two Kronecker products. For example, each step of a Krylov subspace procedure solution procedure for the GEP reduces to the solution of a Sylvester equation. Several related algorithms may be devised this way, and in some cases the computational complexity of the solution process can be reduced all the way down to $\mathcal{O}(n^3)$ [50]. However, this technique shows little potential for generalization to N-parameter systems, as it relies on being able to reformulate the operator determinant GEP into a simple and well-known matrix equation. As the operator determinant definition becomes more complex, the resulting matrix equation changes and efficient solvers may not be available. There is also no easy extension for singular problems.

Subspace methods for one-parameter eigenvalue problems are based around generating a series of linear spaces that eventually approximate one of the system's eigenspaces (the linear space of eigenvectors corresponding to a given eigenvalue). The Jacobi-Davidson and Rayleigh-Ritz methods are well-known one-parameter subspace methods, which can be generalized to apply to two-parameter systems [25,32,51–53], though this is not without difficulties [51]. These methods do not invoke the operator determinants, and show potential for generalization both to N-parameter and to polynomial systems. The Jacobi-Davidson is applicable to singular systems and has previously been tested on singular linearized aeroelastic stability problems [24]; but its performance was observed to be poorer than the operator determinant method. However the solution of singular MEPs has only been achieved very recently (2010 [29]), and so it is likely that the next few years will bring significant developments in this area.

8.2. Extensions to the concept

The core methodology presented in this paper can be extended to a wide variety of problems. Not only more complex two-parameter flutter models – considered in Pons and Gutschmidt [24] – but also stability problems from a wide variety of fields, including systems with

entirely different eigenvalue definitions. Indeed, any combination of model parameters in a stability model can indeed be treated as multiparameter eigenvalues; for example, in aeroelastic model, given the location of a flutter point, we could identify the sets of parameter values that could generate such a flutter point via an MEP solution in those variables. Specifying a flutter point constraints two parameters and thus allows only two (e.g. plunge and torsional stiffness) to be identified; but the specification of multiple flutter points provides more identification capability. This could pave the way for a least-squares approach for overconstrained multiparameter systems. Scalar constraints between system parameters also allow an expansion of the MEP solution dimension – in fact, scalar constraints can be cast as eigenvalue problems by introducing a scalar eigenvector [27]. An example application is the specification of a flight altitude / Mach number parameter, constrained to the airspeed and air density parameters, allowing the computation of points on an aircraft aeroelastic flight envelope. The multiparameter formulation provides a versatile way of analysing stability problems in any combination of parameters.

9. Conclusion

In this paper we have demonstrated and discussed the use of multiparameter solution techniques for the solution of aeroelastic and related stability problems. We have introduced the link between multiparameter spectral theory and stability analysis, and we showed how this link can be used to reformulate stability problems with a complex-valued stability metric and a pertinent environmental parameter into a two-parameter eigenvalue problem. We demonstrated that this allows the direct solution of stability problems that are linear or polynomial in these parameters, and we discussed aspects of the solution process, including the linearization and quasi-linearization of polynomial problems, general N-parameter problems, computational costs and approaches to problem singularity. We devise methods for characterizing the local flutter point behavior, including the nature of the point (restabilizing, destabilizing, etc.), the local gradient of modal damping and the fluttering modeshape. We discuss extensions to our methods, including the generalization to more complex problems; and further applications, including parameter identification and flight envelope computation. The application of multiparameter methods to stability problems – in aeroelasticity and in other disciplines – has the potential to provide a wide variety of new methods for stability analysis.

References

[1] R. L. BISPLINGHOFF, H. ASHLEY, R. L. HALFMAN. *Aeroelasticity*. Addison-Wesley: Reading, MA, 1957.

[2] H. J. HASSIG. An approximate true damping solution of the flutter equation by determinant iteration. *Journal of Aircraft* 1971; **8**(11): 885–889. DOI: 10.2514/3.44311.

[3] D. H. HODGES, G. A. PIERCE. *Introduction to Structural Dynamics and Aeroelasticity*. 2nd ed. Cambridge University Press: New York, 2011.

[4] H. HADDADPOUR, R. D. FIROUZ-ABADI. True damping and frequency prediction for aeroelastic systems: The PP method. *Journal of Fluids and Structures* 2009; **25**(7): 1177–1188. DOI: 10.1016/j.jfluidstructs.2009.06.006.

[5] A. NAMINI, P. ALBRECHT, H. BOSCH. Finite element-based flutter analysis of cable-suspended bridges. *Journal of Structural Engineering* 1992; **118**(6): 1509–1526. DOI: 10.1061/(ASCE)0733-9445(1992)118:6(1509).

[6] P. C. CHEN. Damping perturbation method for flutter solution: the g-method. *AIAA Journal* 2000; **38**(9): 1519–1524. DOI: 10.2514/2.1171.

[7] R. LIND, M. BRENNER. *Robust aeroservoelastic stability analysis*. Springer-Verlag: London, 1999.

[8] R. LIND. Match-Point Solutions for Robust Flutter Analysis. *Journal of Aircraft* 2002; **39**(1): 91–99. DOI: 10.2514/2.2900.

[9] D. BORGLUND. Robust aeroelastic stability analysis considering frequency-domain aerodynamic uncertainty. *Journal of Aircraft* 2003; **40**(1): 189–193. DOI: 10.2514/2.3074.

[10] D. BORGLUND. The μ-k method for robust flutter solutions. *Journal of Aircraft* 2004; **41**(5): 1209–1216. DOI: 10.2514/1.3062.

[11] D. BORGLUND. Upper Bound Flutter Speed Estimation Using the μ-k Method. *Journal of Aircraft* 2005; **42**(2): 555–557.

[12] D. BORGLUND, U. RINGERTZ. Efficient computation of robust flutter boundaries using the μ-k method. *Journal of Aircraft* 2006; **43**(6): 1763–1769.

[13] Y. GU, Z. YANG, W. WANG, W. XIA. Dynamic Pressure Perturbation Method for Flutter Solution: the Mu-Omega Method. In *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, 2009. DOI: 10.2514/6.2009-2312.

[13] Y. GU, Z. YANG, W. WANG, W. XIA. Dynamic Pressure Perturbation Method for Flutter Solution: the Mu-Omega Method. In *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, 2009. DOI: 10.2514/6.2009-2312.

[14] Y. GU, Z. YANG. Generalized Mu-Omega Method with Complex Perturbation to Dynamic Pressure. In *Proceedings of the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Orlando, FL, 2010. DOI: 10.2514/6.2010-2799.

[15] D. BORGLUND. Robust Aeroelastic Analysis in the Laplace Domain: The μ-p Method. In *Proceedings of the 2007 International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Stockholm, Sweden, 2007.

[16] D. AFOLABI. Flutter analysis using transversality theory. *Acta mechanica* 1994; **103**(1–4): 1–15. DOI: 10.1007/BF01180214.

[17] D. AFOLABI, R. M. V. PIDAPARTI, H. T. Y. YANG. Flutter Prediction Using an Eigenvector Orientation Approach. *AIAA Journal* 1998; **36**(1): 69–74. DOI: 10.2514/2.353.

[18] S. IRANI, S. SAZESH. A new flutter speed analysis method using stochastic approach. *Journal of Fluids and Structures* 2013; **40**: 105–114. DOI: 10.1016/j.jfluidstructs.2013.03.018.

[19] Y GU, X. ZHANG, Z. YANG. Robust flutter analysis based on genetic algorithm. *Science China Technological Sciences* 2012; **55**(9): 2474–2481.

[20] P. W. BLYTHE, I. H. HERSZBERG. The Solution of Flutter Equations Using Neural Networks. In *5th Australian Aeronautical Conference: Preprints of Papers*, Institution of Engineers, Australia, 1993; 415.

[21] C. H. CHEN. Determination of flutter derivatives via a neural network approach. *Journal of Sound and Vibration* 2003; **263**(4): 797–813. DOI: 10.1016/S0022-460X(02)01279-8.

[22] C. H. CHEN, J. C. WU, J. H. CHEN. Prediction of flutter derivatives by artificial neural networks. *Journal of Wind Engineering and Industrial Aerodynamics* 2008; **96**(10): 1925–1937. DOI: 10.1016/j.jweia.2008.02.044.

[23] A. NATARAJAN, "Aeroelasticity of morphing wings using neural networks," Doctoral Dissertation, Virginia Polytechnic Institute and State University, Virginia, USA, 2002.

[24] A PONS AND S GUTSCHMIDT. Multiparameter Solution Methods for Semi-Structured Aeroelastic Flutter Problems. *AIAA Journal*, article in advance, 2017. DOI: 10.2514/1.J055447.

[25] K. MEERBERGEN, C. SCHRÖDER, H. VOSS. A Jacobi-Davidson method for two-real-parameter nonlinear eigenvalue problems arising from delay-differential equations. *Numerical Linear Algebra with Applications* 2013; **20**(5): 852–868. DOI: 10.1002/nla.1848.

[26] K. MEERBERGEN, A. SPENCE. Inverse Iteration for Purely Imaginary Eigenvalues with Application to the Detection of Hopf Bifurcations in Large-Scale Problems. *SIAM Journal on Matrix Analysis and Applications* 2010; **31**(4): 1982–1999. DOI: 10.1137/080742890.

[27] A PONS, "Aeroelastic flutter as a multiparameter eigenvalue problem," Master's Thesis, University of Canterbury, New Zealand, 2015.

[28] F. M. WHITE. *Fluid mechanics*. 6th ed. McGraw-Hill: New York, 2009.

[29] A. MUHIČ, B. PLESTENJAK. On the quadratic two-parameter eigenvalue problem and its linearization. *Linear Algebra and its Applications* 2010; **432**(10): 2529–2542. DOI: 10.1016/j.laa.2009.12.022.

[30] M. E. HOCHSTENBACH, A. MUHIČ, B. PLESTENJAK. On linearizations of the quadratic two-parameter eigenvalue problem. *Linear Algebra and its Applications* 2012; **436**(8): 2725–2743. DOI: 10.1016/j.laa.2011.07.026.

[31] F. TISSEUR, K. MEERBERGEN. The Quadratic Eigenvalue Problem. *SIAM Review* 2001; **43**(2): 235–286. DOI: 10.1137/S0036144500381988.

[32] M. E. HOCHSTENBACH, B. PLESTENJAK. A Jacobi-Davidson Type Method for a Right Definite Two-Parameter Eigenvalue Problem. *SIAM Journal on Matrix Analysis and Applications* 2002; **24**(2): 392–410. DOI: 10.1137/S0895479801395264.

[33] B. PLESTENJAK. A continuation method for a weakly elliptic two-parameter eigenvalue problem. *IMA Journal of Numerical Analysis* 2001; **21**(1): 199–216. DOI: 10.1093/imanum/21.1.199.

[34] A. QUARTERONI, R. SACCO, F. SALERI. *Numerical mathematics*. 2nd ed. Springer-Verlag Berlin: Berlin, Germany, 2007.

[35] N. COTTIN. Dynamic model updating – a multiparameter eigenvalue problem. *Mechanical Systems and Signal Processing* 2001; **15**(4): 649–665. DOI: 10.1006/mssp.2000.1362.

[36] N. COTTIN, J. REETZ. Accuracy of multiparameter eigenvalues used for dynamic model updating with measured natural frequencies only. *Mechanical Systems and Signal Processing* 2006; **20**(1): 65–77. DOI: 10.1016/j.ymssp.2004.10.005.

[37] A. MUHIČ, B. PLESTENJAK. On the singular two-parameter eigenvalue problem. *Electronic Journal of Linear Algebra* 2009; **18**: 420–437.

[38] M. E. HOCHSTENBACH, B. PLESTENJAK. Backward error, condition numbers, and pseudospectra for the multiparameter eigenvalue problem. *Linear Algebra and its Applications* 2003; **375**: 63–81. DOI: 10.1016/S0024-3795(03)00613-X.

[39] B. PLESTENJAK, A. MUHIČ. *MultiParEig 1.0*. MATLAB File Exchange: Ljubljana, Slovenia, 2015.

[40] T. KOŠIR. Finite-dimensional multiparameter spectral theory: the nonderogatory case. *Linear Algebra and its Applications* 1994; **212**–**213**: 45–70. DOI: 10.1016/0024-3795(94)90396-4.

[41] B. BINDING, P. J. BROWNE. Two parameter eigenvalue problems for matrices. *Linear Algebra and its Applications* 1989; **113**: 139–157. DOI: 10.1016/0024-3795(89)90293-0.

[42] F. V. ATKINSON. Multiparameter spectral theory. *Bulletin of the American Mathematical Society* 1968; **74**(1): 1–28. DOI: 10.1090/S0002-9904-1968-11866-X.

[43] F. V. ATKINSON. *Multiparameter Eigenvalue Problems: Matrices and Compact Operators*. Academic Press: London, 1972.

[44] B. P. RYNNE. Multiparameter spectral theory and Taylor's joint spectrum in Hilbert space. *Proceedings of the Edinburgh Mathematical Society* 1988; **31**(01): 127. DOI: 10.1017/S0013091500006635.

[45] H. ANTON. *Elementary Linear Algebra*. 2nd ed. John Wiley & Sons: New York, New York State, USA, 1977.

[46] H. A. EISELT, C. L. SANDBLOM. *Linear Programming and its Applications*. Springer-Verlag: Berlin, Germany, 2007.

[47] L. N. TREFETHEN, D. BAU. *Numerical linear algebra*. Society for Industrial and Applied Mathematics: Philadelphia, PA, 1997.

[48] E. H. DOWELL, D. COX, H. C. CURTISS, JR., J. W. EDWARDS, K. C. HALL, D. A. PETERS, R. H. SCANLAN, E. SIMIU, F. SISTO, T. W. STRGANAC, *A Modern Course in Aeroelasticity*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004

[49] L. N. TREFETHEN, 1992, "Pseudospectra of Matrices," *Numerical Analysis 1991*, D.F. GRIFFITHS, G.A. WATSON, eds., Longman Scientific and Technical, Harlow, Essex, UK, pp. 234–266.

[50] K. MEERBERGEN, B. PLESTENJAK. *A Sylvester-Arnoldi type method for the generalized eigenvalue problem with two-by-two operator determinants*. Department of Computer Science, Katholieke Universiteit Leuven: Leuven, Belgium, 2014.

[51] M. E. HOCHSTENBACH, T. KOSIR, B. PLESTENJAK. A Jacobi-Davidson Type Method for the Two-Parameter Eigenvalue Problem. *SIAM Journal on Matrix Analysis and Applications* 2004; **26**(2): 477–497. DOI: 10.1137/S0895479802418318.

[52] M. E. HOCHSTENBACH, B. PLESTENJAK. Harmonic Rayleigh–Ritz extraction for the multiparameter eigenvalue problem. *Electronic Transactions on Numerical Analysis* 2008; **29**: 81–96.

[53] M. E. HOCHSTENBACH, A. MUHIČ, B. PLESTENJAK. Jacobi–Davidson methods for polynomial two-parameter eigenvalue problems. *Journal of Computational and Applied Mathematics* 2015; **288**: 251–263. DOI: 10.1016/j.cam.2015.04.019.