Multiparameter Solution Methods for Semi-Structured Aeroelastic Flutter Problems

Article *in* AIAA Journal · April 2017

DOI: 10.2514/1.J055447

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 Solution Methods for Semi-Structured Aeroelastic Flutter Problems

Arion Pons[1]
University of Cambridge, Cambridge CB2 1PZ, UK

Stefanie Gutschmidt[2]
University of Canterbury, Christchurch 8041, New Zealand

This paper presents several new methods for the solution of aeroelastic flutter problems with partial polynomial structure – problems consisting of a mix of polynomial and more complex nonlinear components. We focus particularly on systems that use Theodorsen aerodynamics: for such systems we devise four new solution algorithms. Two of these are direct but yield approximate results, and two are iterative. These algorithms are tested on an example system, and their computational characteristics are investigated and discussed. Three of them are suitable for practical implementation; the fourth is too computationally intensive to be of great practical use. Extensions and improvements to these algorithms, and the overall methods used, are also discussed.

Nomenclature

ι	=	imaginary unit
$\bar{x}$	=	complex conjugate of x
∂_x	=	partial differentiation w.r.t x
κ	=	reduced frequency
χ	=	time-eigenvalue, rad/s
τ	=	reciprocal reduced frequency
Υ	=	air frequency parameter, Hz
λ	=	reciprocal time-eigenvalue, s/rad
$\hat{\kappa}$	=	fractional-order reduced frequency
θ	=	torsional deformation variable, rad

[1] PhD student, Department of Engineering, adp53@cam.ac.uk
[2] Senior Lecturer, Department of Mechanical Engineering, stefanie.gutschmidt@canterbury.ac.nz

h	=	plunge deformation variable, m
μ	=	dimensionless mass ratio
r	=	dimensionless radius of gyration
r_θ	=	dimensionless static imbalance
ω_h	=	uncoupled bending natural frequency, rad/s
ω_θ	=	uncoupled torsional natural frequency, rad/s
ζ_h	=	uncoupled bending damping ratio
ζ_θ	=	uncoupled torsional damping ratio
$H_n^{(2)}$	=	n-th Hankel function of the second kind
$C(\cdot), C_\tau(\cdot)$	=	Theodorsen's function
υ	=	Theodorsen's function variable
d	=	flutter determinant
$\mathrm{G}_i, \mathrm{D}_0, \mathrm{K}_0$	=	nonlinear matrix coefficients
A_i	=	polynomial matrix coefficients
$\mathrm{M}_{\{0,\kappa,\hat{\kappa},\chi\}}$	=	linearized matrix coefficients
$\boldsymbol{i}_\mathrm{X}$	=	block row indices describing X
$\boldsymbol{j}_\mathrm{X}$	=	block column indices describing X
c_i	=	rational approximation coefficients
β	=	fractional-order approximation exponent
F	=	fractional-order approximation coefficient
ϵ_i	=	convergence tolerances
;	=	vertical concatenation
n	=	matrix coefficient size
$\mathrm{I}_{n\times n}$	=	identity matrix of size $n \times n$
$0_{n\times n}$	=	zero matrix of size $n \times n$

I. Introduction

Previous research by Pons [1] has shown that multiparameter spectral theory may be applied to the solution to aeroelastic instability problems. The multiparameter approach to instability problems was presaged by Bisplinghoff [2], who referred to the aeroelastic flutter problem as a "double eigenvalue problem". It occurs in the wider context of recent advances in methods for computing flutter points, such as the development of the μ-type methods [3–5] and extensions to the traditional k- and p-k methods [6,7], among others. The multiparameter approach – involving the transformation of the instability problem into a multiparameter eigenvalue problem (MPEVP) – is particularly powerful in that it can be used to derive direct solvers for the flutter problem; solvers that will compute an exact solution for the flutter points in a finite number of operations. This is in contrast to existing methods which are based around performing a parameter sweep over an airspeed-based parameter (e.g. the reduced frequency).

In Pons [1], direct solvers were devised for polynomial stability problems, and these were applied to aeroelastic flutter problems of this form. These solvers are applications of the more general multiparameter direct solvers presented by Hochstenbach et al. [8] and Muhič and Plestenjak [9,10]. Of course, not all aeroelastic stability problems are polynomial, and so the question arises of whether this multiparameter solution method may be extended to more general systems. In this work we consider what we have previously [1] termed *semi-structured* problems: stability problems with some polynomial or other analytical structure but also one or more black-box (general nonlinear) functions which must be treated numerically. This is in contrast to the *structured* systems, which are fully polynomial. The systems we are particularly interested in are those which utilize Theodorsen's aerodynamic theory. Given the definition of Theodorsen's function is in terms of the transcendental Hankel functions [11], such systems are necessarily semi-structured. It is however even possible to ignore the polynomial structure of these systems entirely, and use solvers for unstructured (i.e. generally nonlinear) systems such as those presented in Pons [1] and Plestenjak [12]. However, the utilization of the structured component of the system opens us new and potentially more effective methods of solution.

In this paper we present two strategies and four specific procedures for the solution of semi-structured aeroelastic flutter problems via multiparameter methods. The first strategy is to approximate the black-box function to a high degree of accuracy with a convenient expression (e.g. a polynomial or rational function), after which the system can be linearized and solved using linear solvers [8–10]. The second strategy is to

approximate the black-box function to a low level of accuracy with a very simple expression (e.g. a truncated Taylor series) and then solve the system iteratively, updating the approximation with data from the previous iterate. We will apply the these two strategies that we have outlined to a specific example of a semi-structured system – a section model with Theodorsen aerodynamics – and this will yield a number of algorithms for computing the flutter points of this system.

II. An Example Semi-Structured Model

A. Formulation

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:

$$\begin{aligned} m\ddot{h} + d_h\dot{h} + k_h h - mx_\theta\ddot{\theta} &= -L(t), \\ I_P\ddot{\theta} + d_\theta\dot{\theta} + k_\theta\theta - mx_\theta\ddot{h} &= M(t), \end{aligned} \tag{1}$$

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 static imbalance – defined as the distance along the x-axis from the pivot point to the centre of mass. Taking the Fourier transform, e.g. $h(t) = \hat{h}\exp(\iota\chi t)$, of this model, we obtain:

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

To model the aerodynamics loads in the frequency domain – $L(\chi, \hat{h}, \hat{\theta})$ and $M(\chi, \hat{h}, \hat{\theta})$ – we use Theodorsen's unsteady aerodynamic theory [11]:

$$L = -\chi^2(L_h\hat{h} + L_\theta\hat{\theta}), \quad M = \chi^2(M_h\hat{h} + M_\theta\hat{\theta}). \tag{3}$$

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). Note that we assume a lift-angle of attack coefficient of 2π (as per thin-airfoil theory [13]) without loss of generality. See Pons [1] or Hodges and Pierce [11] for details. Included also in the coefficients is Theodorsen's function

$$C(\kappa) = \frac{H_1^{(2)}(\kappa)}{H_1^{(2)}(\kappa) - \iota H_0^{(2)}(\kappa)}, \tag{4}$$

where $H_n^{(2)}$ is the n-th Hankel function of the second kind [14].

It is then customary to nondimensionalise Eq. (2). Further details of this are given in Pons [1]. The final result is a flutter problem of the form

$$\left(\left(\mathrm{G}_0 + \mathrm{G}_1\frac{1}{\kappa} + \mathrm{G}_2\frac{C(\kappa)}{\kappa} + \mathrm{G}_3\frac{C(\kappa)}{\kappa^2}\right)\chi^2 - \mathrm{D}_0\chi - \mathrm{K}_0\right)\mathbf{x} = \mathbf{0}. \tag{5}$$

with parameters defined as in Table 1, and the matrix coefficients

$$\begin{gathered}
\mathrm{G}_0 = \frac{1}{\mu}\begin{bmatrix} 2 & -a - r_\theta \\ -a - r_\theta & \frac{1}{8} + a^2 + r^2 \end{bmatrix}, \mathrm{G}_1 = \frac{\iota}{\mu}\begin{bmatrix} 0 & -1 \\ 0 & -\frac{1}{2} + a \end{bmatrix}, \\
\mathrm{G}_2 = \frac{\iota}{\mu}\begin{bmatrix} -2 & -1 + 2a \\ 1 + 2a & \frac{1}{2} - 2a^2 \end{bmatrix}, \mathrm{G}_3 = \frac{1}{\mu}\begin{bmatrix} 0 & -2 \\ 0 & 1 + 2a \end{bmatrix}, \\
\mathrm{D}_0 = 2\iota\begin{bmatrix} \zeta_h\omega_h & 0 \\ 0 & r^2\zeta_\theta\omega_\theta \end{bmatrix}, \mathrm{K}_0 = \begin{bmatrix} \omega_h^2 & 0 \\ 0 & r^2\omega_\theta^2 \end{bmatrix}.
\end{gathered} \tag{6}$$

Note that this (partial) nondimensionalisation is a convenience and not necessary prerequisite for using multiparameter solution methods. However something that will be a significant aid to our solution methods is to rearrange Eq. (5) into two near-polynomial forms by defining new eigenvalue parameters: $\Upsilon = U/b$, $\tau = 1/\kappa$, and $\lambda = 1/\chi$. These two forms are then

$$\begin{aligned}
\Bigg(\mathrm{G}_0\chi^2 + \Bigg(\mathrm{G}_1 + \mathrm{G}_2 C\left(\frac{\chi}{\Upsilon}\right)\Bigg)\Upsilon\chi \\
+ \mathrm{G}_3 C\left(\frac{\chi}{\Upsilon}\right)\Upsilon^2 - \mathrm{D}_0\chi - \mathrm{K}_0\Bigg)\mathbf{x} = \mathbf{0},
\end{aligned} \tag{7}$$

which we term the Υ-χ form, and

$$\begin{aligned}
\big(\mathrm{G}_0 + \big(\mathrm{G}_1 + \mathrm{G}_2 C_\tau(\tau)\big)\tau \\
+ \mathrm{G}_3 C_\tau(\tau)\tau^2 - \mathrm{D}_0\lambda - \mathrm{K}_0\lambda^2\big)\mathbf{x} = \mathbf{0},
\end{aligned} \tag{8}$$

which we term the τ-λ form. $C_\tau(\tau)$ is Theodorsen's function in τ, that is $C_\tau(\tau) = C(1/\tau)$. We should note at this point that, although we have here derived these two forms for a small 2-DOF section model, these forms

arise in many other models; often with significantly larger matrices. The solution methods we develop for this form will thus be applicable to a wide variety of problems.

B. MPEVP Representation

Before attempting to devise multiparameter solution algorithms, we must first represent Eq. (5), (7) and (8) as multiparameter eigenvalue problems. The procedure for doing so is derived in Pons [1] – we add an equation representing the conjugate of the initial equation. The three multiparameter systems can thus be written

$$\begin{gathered}\left(\left(\mathrm{G}_0+\mathrm{G}_1\frac{1}{\kappa}+\mathrm{G}_2\frac{\upsilon}{\kappa}+\mathrm{G}_3\frac{\upsilon}{\kappa^2}\right)\chi^2-\mathrm{D}_0\chi-\mathrm{K}_0\right)\mathbf{x}=\mathbf{0}\\ \left(\left(\overline{\mathrm{G}}_0+\overline{\mathrm{G}}_1\frac{1}{\kappa}+\overline{\mathrm{G}}_2\frac{\bar{\upsilon}}{\kappa}+\overline{\mathrm{G}}_3\frac{\bar{\upsilon}}{\kappa^2}\right)\chi^2-\overline{\mathrm{D}}_0\chi-\overline{\mathrm{K}}_0\right)\bar{\mathbf{x}}=\mathbf{0}\\ \upsilon=C(\kappa),\end{gathered} \tag{9}$$

$$\begin{gathered}(\mathrm{G}_0+(\mathrm{G}_1+\mathrm{G}_2\upsilon)\tau+\mathrm{G}_3\upsilon\tau^2-\mathrm{D}_0\lambda-\mathrm{K}_0\lambda^2)\mathbf{x}=\mathbf{0}\\ (\overline{\mathrm{G}}_0+(\overline{\mathrm{G}}_1+\overline{\mathrm{G}}_2\bar{\upsilon})\tau+\overline{\mathrm{G}}_3\bar{\upsilon}\tau^2-\overline{\mathrm{D}}_0\lambda-\overline{\mathrm{K}}_0\lambda^2)\bar{\mathbf{x}}=\mathbf{0}\\ \upsilon=C_\tau(\tau),\end{gathered} \tag{10}$$

and

$$\begin{gathered}(\mathrm{G}_0\chi^2+(\mathrm{G}_1+\mathrm{G}_2\upsilon)\Upsilon\chi+\mathrm{G}_3\upsilon\Upsilon^2-\mathrm{D}_0\chi-\mathrm{K}_0)\mathbf{x}=\mathbf{0}\\ (\overline{\mathrm{G}}_0\chi^2+(\overline{\mathrm{G}}_1+\overline{\mathrm{G}}_2\bar{\upsilon})\Upsilon\chi+\overline{\mathrm{G}}_3\bar{\upsilon}\Upsilon^2-\overline{\mathrm{D}}_0\chi-\overline{\mathrm{K}}_0)\bar{\mathbf{x}}=\mathbf{0}\\ \upsilon=C(\chi/\Upsilon).\end{gathered} \tag{11}$$

Eq. (10) and (11) would be polynomial but for Theodorsen's function. If they could be made polynomial (e.g. by approximation) then they could be solved by the polynomial direct solvers devised by Hochstenbach et al. [8] and Muhič and Plestenjak [9,10]. That these solvers are *direct* indicates that they compute the full and exact eigenspectrum of the problem within a finite number of steps. In the absence of Theodorsen's function, Eq. (11) could also be made polynomial (by multiplying by κ^2); though this results in an MPEVP that is higher-order (up to $\kappa^2\chi^2$) than is necessary for the problem. However, to deal with approximations to Theodorsen's function that are formulated in κ it will be necessary to use this form.

III. Approximate Solution Methods

A. Approximations to Theodorsen's Function

It is well known in aeroelasticity that Theodorsen's function is difficult to treat analytically. For this reason many approximations to it have been developed: Brunton and Rowley [15] give an extensive but by no

means exhaustive list. The vast majority of these are rational functions, as they have been designed for use in control loops. With them we should be able to transform our semi-structured problem into a structured one – however, they are all formulated as rational functions in κ, and so by necessity we must use the inefficient κ-χ form of our model. Here we consider two approximations in particular: a rational representation of second order in numerator and denominator, first given by Jones [16]; and a fractional-order representation by Swinney [17]. The use of the rational approximation will demonstrate that our methods can be used with any rational approximation to Theodorsen's function, and the use of the fractional-order approximation will show also that our methods can be extended to more complex approximating functions than are widely used at present.

B. Rational Approximation

The rational approximation given by Jones [16] is

$$C(\kappa) = \frac{\frac{1}{2}\kappa^2 + c_1\kappa + c_2}{\kappa^2 + c_3\kappa + c_2} \tag{12}$$

with $c_1 = -0.2808\iota$, $c_2 = -0.01365$, $c_3 = -0.3455\iota$. Consider Eq. (5). Substituting Eq. (12) into Eq. (5), multiplying by $\kappa^2(\kappa^2 + c_3\kappa + c_2)$, and expanding, we obtain the system

$$\begin{aligned}(\kappa^4\chi^2\mathrm{A}_0 + \kappa^3\chi^2\mathrm{A}_1 + \kappa^2\chi^2\mathrm{A}_2 + \kappa\chi^2\mathrm{A}_3 + \chi^2\mathrm{A}_4 + \kappa^4\chi\mathrm{A}_5 + \kappa^3\chi\mathrm{A}_6 + \kappa^2\chi\mathrm{A}_7 + \kappa^4\mathrm{A}_8 + \kappa^3\mathrm{A}_9 \\ + \kappa^2\mathrm{A}_{10})\mathbf{x} = \mathbf{0}\end{aligned} \tag{13}$$

with

$$\begin{aligned}\mathrm{A}_0 &= \mathrm{G}_0 & \mathrm{A}_5 &= -\mathrm{D}_0 \\ \mathrm{A}_1 &= c_3\mathrm{G}_0 + \mathrm{G}_1 + \tfrac{1}{2}\mathrm{G}_2 & \mathrm{A}_6 &= -c_3\mathrm{D}_0 \\ \mathrm{A}_2 &= c_2\mathrm{G}_0 + c_3\mathrm{G}_1 + c_1\mathrm{G}_2 + \tfrac{1}{2}\mathrm{G}_3 & \mathrm{A}_7 &= -c_2\mathrm{D}_0 \\ \mathrm{A}_3 &= c_2\mathrm{G}_1 + c_2\mathrm{G}_2 + c_1\mathrm{G}_3 & \mathrm{A}_8 &= -\mathrm{K}_0 \\ \mathrm{A}_4 &= c_2\mathrm{G}_3 & \mathrm{A}_9 &= -c_3\mathrm{K}_0 \\ & & \mathrm{A}_{10} &= -c_2\mathrm{K}_0\end{aligned} \tag{14}$$

Eq. (13) is a fourth-order polynomial MPEVP and thus may be solved using polynomial direct solvers [1,8,9]. Some level of choice is involved: in the solution algorithm we may choose to reduce the polynomial problem to a linear problem via the linearization or quasilinearization [9]. We choose linearization because it is significantly simpler to perform on a system of this size. To linearize Eq. (13) we define the eigenvector

$$\begin{aligned}\mathbf{q} = [\kappa^4\chi\mathbf{x};\ \kappa^3\chi\mathbf{x};\ \kappa^2\chi\mathbf{x};\ \kappa\chi\mathbf{x};\ \chi\mathbf{x};\\ \kappa^4\mathbf{x};\ \kappa^3\mathbf{x};\ \kappa^2\mathbf{x};\ \kappa\mathbf{x};\ \mathbf{x}]\end{aligned} \tag{15}$$

which, with careful arrangement, allows us to rewrite it as

$$\left(\mathrm{M}_0 + \mathrm{M}_\chi\chi + \mathrm{M}_\kappa\kappa\right)\mathbf{q} = \mathbf{0}. \tag{16}$$

The coefficient matrices of Eq. (16) are sparse and so we write them in a block dictionary-of-keys format [18]. That is, we define the matrices via three arrays: A, which lists the block matrix elements; and $\boldsymbol{i}_{\mathrm{A}}$ and $\boldsymbol{j}_{\mathrm{A}}$, which list the block row and column indices of the block elements (using one-based indexing). All other elements are zero. Thus we have:

$$\begin{aligned}
&\text{For } \mathrm{M}_0,\\
&\qquad \mathrm{A} = [\mathrm{A}_5, \mathrm{A}_6, \mathrm{A}_7, \mathrm{A}_8, \mathrm{A}_9, \mathrm{A}_{10}, \mathrm{I}, \mathrm{I}, \mathrm{I}, \mathrm{I}, \mathrm{I}, \mathrm{I}, \mathrm{I}, \mathrm{I}]\\
&\qquad \boldsymbol{i}_{\mathrm{A}} = [1,1,1,1,1,1,2,3,4,5,6,7,8,9,10]\\
&\qquad \boldsymbol{j}_{\mathrm{A}} = [1,2,3,6,7,8,1,2,3,4,5,6,7,8,9]\\
&\text{For } \mathrm{M}_\chi,\\
&\qquad \mathrm{A} = [\mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_4, -\mathrm{I}]\\
&\qquad \boldsymbol{i}_{\mathrm{A}} = [1,1,1,1,1,6]\\
&\qquad \boldsymbol{j}_{\mathrm{A}} = [1,2,3,4,5,10]\\
&\mathrm{M}_\kappa = \mathrm{blockdiag}\{0, -\mathrm{I}, -\mathrm{I}, -\mathrm{I}, -\mathrm{I}, 0, -\mathrm{I}, -\mathrm{I}, -\mathrm{I}, -\mathrm{I}\}
\end{aligned} \tag{17}$$

where $\mathrm{I} = \mathrm{I}_{n\times n}$ and $0 = 0_{n\times n}$ for readability. M_κ can be written as a block diagonal ($\mathrm{blockdiag}\{\cdot\}$) matrix, avoiding the more complex block dictionary of keys notation. Eq. (16) is a linear problem which can be solved with the two-parameter operator determinant method [10,19] – a direct solver. Eq .16 is a singular problem – the matrix pencil $\mathrm{M}_0 + \mathrm{M}_\chi\chi + \mathrm{M}_\kappa\kappa$ is never full rank; in the first instance due to the sparsity of the linearized coefficient matrices (Eq. 17), but additionally due to the singular aerodynamic coefficients. It thus requires also the staircase algorithm of Muhič and Plestenjak [10]. The conjugate equation for the problem (cf. Eq. 9-11) can be constructed either by conjugating all the M-matrices or by conjugating their individual entries. This system represents an interesting physical example of a high-order polynomial MPEVP; such as have been considered before in mathematical literature [9] but without practical motivation. The linearization may seem impractically large, but we will in fact show later that the solution times are not unreasonable.

C. Fractional-order Approximation

The approximation to Theodorsen's function given by Swinney [17] is

$$C(\kappa) = \frac{1 + F(i\kappa)^\beta}{1 + 2F(i\kappa)^\beta} \tag{18}$$

with $\beta = 5/6$ and $F = 2.19$. Substituting Eq. (18) into Eq. (5), multiplying by $\kappa^2\left(1 + 2F(i\kappa)^\beta\right)$, and expanding the terms we obtain

$$\Big(\Big(\kappa^2\mathrm{G}_0+\kappa(\mathrm{G}_1+\mathrm{G}_2)+\mathrm{G}_3+k^{\frac{17}{6}}2\iota^\beta F\mathrm{G}_0 \\ +k^{\frac{11}{6}}\iota^\beta F(2\mathrm{G}_1+\mathrm{G}_2)+k^{\frac{5}{6}}\iota^\beta F\mathrm{G}_3\Big)\chi^2 \\ -\kappa^2\mathrm{D}_0\chi-k^{\frac{17}{6}}2\iota^\beta F\mathrm{D}_0\chi-\kappa^2\mathrm{K}_0-k^{\frac{17}{6}}2\iota^\beta F\mathrm{K}_0\Big)\mathbf{x}=\mathbf{0}. \tag{19}$$

If we then define a new eigenvalue variable $\hat{\kappa}=\sqrt[6]{\kappa}$, the system simplifies to

$$(\hat{\kappa}^{17}\chi^2\mathrm{A}_0+\hat{\kappa}^{12}\chi^2\mathrm{A}_1+\hat{\kappa}^{11}\chi^2\mathrm{A}_2+\hat{\kappa}^6\chi^2\mathrm{A}_3+\hat{\kappa}^5\chi^2\mathrm{A}_4 \\ +\chi^2\mathrm{A}_5+\hat{\kappa}^{17}\chi\mathrm{A}_6+\hat{\kappa}^{12}\chi\mathrm{A}_7+\hat{\kappa}^{17}\mathrm{A}_8+\hat{\kappa}^{12}\mathrm{A}_9)\mathbf{x}=\mathbf{0} \tag{20}$$

with

$$\begin{aligned}&\mathrm{A}_0=2\iota^\beta F\mathrm{G}_0 && \mathrm{A}_5=\mathrm{G}_3\\ &\mathrm{A}_1=\mathrm{G}_0 && \mathrm{A}_6=-2\iota^\beta F\mathrm{D}_0\\ &\mathrm{A}_2=\iota^\beta F(2\mathrm{G}_1+\mathrm{G}_2) && \mathrm{A}_7=-\mathrm{D}_0\\ &\mathrm{A}_3=\mathrm{G}_1+\mathrm{G}_2 && \mathrm{A}_8=-2\iota^\beta F\mathrm{K}_0\\ &\mathrm{A}_4=\iota^\beta F\mathrm{G}_3 && \mathrm{A}_9=-\mathrm{K}_0.\end{aligned} \tag{21}$$

Eq. (20) is a polynomial multi-parameter eigenvalue problem of degree 17 in $\hat{\kappa}$ and degree 2 in χ. We may again solve it via linearization. Defining the eigenvector

$$\mathbf{q}=[\hat{\kappa}^{17}\chi\mathbf{x};\ \hat{\kappa}^{12}\chi\mathbf{x};\ \hat{\kappa}^{11}\chi\mathbf{x};\ \hat{\kappa}^6\chi\mathbf{x};\ \hat{\kappa}^5\chi\mathbf{x};\ \chi\mathbf{x}; \\ \hat{\kappa}^{17}\mathbf{x};\ \hat{\kappa}^{16}\mathbf{x};\ \hat{\kappa}^{15}\mathbf{x};\ \hat{\kappa}^{14}\mathbf{x};\ \hat{\kappa}^{13}\mathbf{x}; \\ \hat{\kappa}^{12}\mathbf{x};\ \hat{\kappa}^{11}\mathbf{x};\ \hat{\kappa}^{10}\mathbf{x};\ \hat{\kappa}^9\mathbf{x};\ \hat{\kappa}^8\mathbf{x}; \\ \hat{\kappa}^7\mathbf{x};\ \hat{\kappa}^6\mathbf{x};\ \hat{\kappa}^5\mathbf{x};\ \hat{\kappa}^4\mathbf{x};\ \hat{\kappa}^3\mathbf{x};\ \hat{\kappa}^2\mathbf{x};\ \hat{\kappa}\mathbf{x};\ \mathbf{x}], \tag{22}$$

Eq. (20) becomes

$$\big(\mathrm{M}_0+\mathrm{M}_\chi\chi+\mathrm{M}_{\hat{\kappa}}\hat{\kappa}\big)\mathbf{q}=\mathbf{0}. \tag{23}$$

The coefficients of Eq. (23) are again sparse block matrices, of size 24×24 blocks, or $24n\times24n$ absolutely:

For M_0,

$$\begin{aligned}&\mathrm{A}=[\mathrm{A}_0,\mathrm{A}_1,\mathrm{A}_2,\mathrm{A}_3,\mathrm{A}_4,\mathrm{A}_5,\mathrm{A}_6,\mathrm{A}_7,\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I}],\\ &\boldsymbol{i}_\mathrm{A}=[1,1,1,1,1,1,1,1,2,3,6,7],\\ &\boldsymbol{j}_\mathrm{A}=[1,2,3,4,5,6,7,12,7,12,19,24]\end{aligned}$$

For M_χ,

$$\begin{aligned}&\mathrm{A}=-[-\mathrm{A}_8,-\mathrm{A}_9,\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\\ &\qquad\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I}],\\ &\boldsymbol{i}_\mathrm{A}=[1,1,2,3,4,5,6,7,8,9,10,\\ &\qquad11,12,13,14,15,16,17,\\ &\qquad18,19,20,21,22,23,24],\\ &\boldsymbol{j}_\mathrm{A}=[7,12,1,2,2,4,5,6,7,8,9,\\ &\qquad10,11,12,13,14,15,16,\\ &\qquad17,18,19,20,21,22,23],\end{aligned} \tag{24}$$

For $\mathrm{M}_{\hat{\kappa}}$,

$$\mathrm{A} = [\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I},\mathrm{I}],$$

$$\boldsymbol{i}_{\mathrm{A}} = [4,5,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24],$$

$$\boldsymbol{j}_{\mathrm{A}} = [3,5,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24],$$

where $\mathrm{I} = I_{n\times n}$ for readability. This is again a linear system that can be solved by the two-parameter operator determinant method [10,19]. Through a substitution of a variable representing the highest common factor of the fractional powers, $\hat{\kappa}$, we have been able to transform this to a polynomial system. This method of linearization (and hence, direct solution) for fractional-order multiparameter problems has not previously been considered, and the current mathematical literature [12] is not completely true in its statement that that MPEVPs with non-integer powers cannot be linearized. By this method, only systems with irrational powers cannot be linearized – though more complex fractions will generate larger linearizations. There is significant potential here for further research, both in the solution of fractional-order MPEVPs and their application to engineering problems.

D. Numerical Experiments

To begin our numerical study of these systems with approximate Theodorsen aerodynamics, we briefly quantify the accuracy of the two approximations to Theodorsen's function that we are using. Visual comparisons may be seen in Jones [16] and Swinney [17]. The maximum errors in the real part of both approximations are not very significant, being on the order of 2%. The error in the imaginary part of the fractional-order approximation is of similar order; however, the maximum error in the imaginary part of the rational approximation is slightly more significant at 8%. The fractional-order approximation is thus only slightly more accurate than the rational one – but it requires a much larger linearized system (Section III.B and III.C).

Figure 2 shows the direct flutter-point solutions for the both approximations, superimposed over a contour plot of the exact Theodorsen system, Eq. (5). The contour plot is a visualization and solution method for aeroelastic problems [1,20]: contours of the real and imaginary part of the flutter determinant – $d = \det(\mathbf{D})$ where **D** represents the MPEVP matrix function; e.g. the matrix coefficient of **x** in Eq. 5 – are superimposed. The intersections of the contours of $\mathrm{Re}(d)$ and $\mathrm{Im}(d)$ then represent the exact flutter points of the system. Two

of these may be located at $\kappa = 5.4 \times 10^{-4}$ and $\kappa = 0.283$. In Figure 2 the flutter points generated by the direct solvers are shown as points on the same field.

Consider the results of the fractional-order approximation. As can be seen, these are very accurate – they agree with the exact results to within 0.5% for both eigenvalues. This accuracy is easily sufficient for practical application – in which the inaccuracy of Theodorsen's aerodynamic theory in modeling the system of interest will far outweigh the relative inaccuracy of the approximation. However, the computational cost of performing the operator determinant method on the linearized fractional-order approximation is very high. The operator determinants [10] before staircase algorithm compression are of size 2304×2304, and after compression they are still of size 232×232. Running MATLAB R2014b on an Intel i7-4770 with 3.40 GHz processor, 16.0 GB RAM and a 64-bit operating system, it takes an average of 20.8 seconds to complete the operator determinant method and compute the flutter points of the system. Approximately 98% of this time is taken up by the staircase algorithm [9]. Only 1% is expended on the initialization routine (defining the system matrices and operator determinants), and less than 1% (0.08 seconds) on the solution of the generalized eigenvalue problem in the operator determinants. This is very interesting, and suggests that efforts to increase the efficiency of the operator determinant method for polynomial problems should be focused on the development of faster compression algorithms, or else on the devising of compact nonsingular linearizations that do not require compression. The actual solution of the operator determinant GEP is very fast. As a point of interest, we may compare our implementation of the two-parameter operator determinant method to that published by Muhič and Plestenjak [21]. Their implementation takes an average of 21.2 seconds to compute the flutter points, which is not perceptibly different to ours.

Consider now the rational approximation. Again, the flutter points computed by the direct solver are very accurate; also agreeing to within 0.5% for both eigenvalues. There is thus no material difference in accuracy between the two approximations (at least for this system) and thus the rational approximation is clearly preferable as the faster option. The operator determinants before compression are of size 400×400, and after compression they are of size 53×53. Running on the same Intel Pentium i7-4770, it takes an average of 0.19 seconds to complete the operator determinant method and compute the flutter points of the system. Again, the vast majority of this time (86%) is spent on extracting the finite regular part of the operator determinant pencils. Initializing the variables and defining the operator determinants takes 12%, and solving the operator determinant

pencil only 2% (0.003 seconds). The implementation of the two-parameter operator determinant method published by Muhič and Plestenjak [21] takes an average of 0.23 seconds to compute the flutter points: only slightly longer than our implementation. It is clear from these results that the rational approximation is far more appropriate for industrial aeroelastic analysis than the fractional-order one: the difference in accuracy between the approximations is negligible, but solving the system with the rational approximation is over an order of magnitude faster. The fast computation time for this approximation will allow it to be used in optimization or other parameter-search routines. The fact that the solver is direct is a big advantage in this respect: it is not necessary to provide initial guesses or search grids that have to be changed based on the parameter values. We anticipate that optimization or parameter search routines will be the most significant application of this solver – for larger once-off simulations the contour plot is available, which gives more contextual information.

IV. Picard Iteration

A. Formulation

Consider again Eq. (10) and (11). These are systems of two under-constrained polynomial MPEVPs and a nonlinear scalar constraint equation. If v is known (or guessed), then the two MPEVPs can be solved by a standard singular direct solver such as are described in Pons [1] and Muhič and Plestenjak [10]. The eigenvalues obtained in this way (τ, λ or Υ, χ) can then be used to generate the next v. This yields a solution method which is essentially a type of Picard iteration [22]: we evaluate part of the system (Theodorsen's function) at the previous timestep, and then we solve for the remaining system at the current timestep. For example, with the Υ-χ form (Eq. 11), the iterative sequence is defined implicitly as:

$$\begin{gathered}(\mathrm{G}_0\chi_k^2 + (\mathrm{G}_1 + \mathrm{G}_2 v_{k-1})\Upsilon_k\chi_k + \mathrm{G}_3 v_{k-1}\Upsilon_k^2 - \mathrm{D}_0\chi_k - \mathrm{K}_0)\mathbf{x} = \mathbf{0} \\ (\overline{\mathrm{G}}_0\chi_k^2 + (\overline{\mathrm{G}}_1 + \overline{\mathrm{G}}_2\bar{v}_{k-1})\Upsilon_k\chi_k + \overline{\mathrm{G}}_3\bar{v}_{k-1}\Upsilon_k^2 - \overline{\mathrm{D}}_0\chi_k - \overline{\mathrm{K}}_0)\bar{\mathbf{x}} = \mathbf{0} \\ v_{k-1} = C(\chi_{k-1}/\Upsilon_{k-1})\end{gathered} \tag{25}$$

Note that we may use the exact Theodorsen function here, and require no rational or fractional-order approximation – the system's non-polynomial terms are approximated with a zeroth-order Taylor series at about the previous iterate. The remaining polynomial problem (Eq. 25) can then be linearized to a singular linear MPEVP in the manner of Section III.

For the solution of this singular linear problem, there are two available options. The operator determinant method (introduced in Section III) is a direct solver, giving us a full and reliable picture of the

spectrum of the linearized problem; allowing us to select the eigenvalue best suited to proceed. However it scales very poorly ($\mathcal{O}(n^6)$ [10]) with matrix size. An alternative is to use the Jacobi-Davidson method of Hochstenbach et al. [23]: this is an iterative method for singular linear systems, which computes eigenvalues near a target. It still utilizes an operator determinant solution method at base level, but applied to a smaller projected problem. Its computational complexity for one eigenvalue of a nonsingular system is $\mathcal{O}(n^2)$ [24,25], though the effects of the problem singularity is not certain. We test both methods. In both we proceed the iteration using the multiparameter eigenvalue nearest the current iterate – the one best approximated by the Picard linearization. In the Jacobi-Davidson method we set the current iterate as the eigenvalue target, and in the operator-determinant method we select the eigenvalue closest in distance to the current iterate.

Finally, in our implementation of the Picard iteration, we use two convergence criteria: one increment-based and one residual-based. The increment-based criterion is based on the Euclidean distance; it is:

$$(\Upsilon_k - \Upsilon_{k-1})^2 + (\chi_k - \chi_{k-1})^2 < \epsilon_1^2. \tag{26}$$

To define a residual-based convergence criterion, we define the residual of the system in a manner analogous to that of a single-parameter eigenvalue problem:

$$\|\mathbf{r}_k\| < \epsilon_2, \qquad \mathbf{r}_k = \mathrm{T}(\Upsilon_k, \chi_k)\mathbf{x}_k. \tag{27}$$

where

$$\begin{aligned}\mathrm{T}(\Upsilon_k, \chi_k) = \mathrm{G}_0\chi_k^2 + \left(\mathrm{G}_1 + \mathrm{G}_2 C(\chi_k/\Upsilon_k)\right)\Upsilon_k\chi_k \\ +\mathrm{G}_3 C(\chi_k/\Upsilon_k)\Upsilon_k^2 - \mathrm{D}_0\chi_k - \mathrm{K}_0.\end{aligned} \tag{28}$$

and $\|\cdot\|$ is some suitable norm – we use the Euclidean norm again. Eq. (27) and (28) provide a complete measure of the residual of the two multiparameter equations (cf. Eq. (11)), because the residual vectors of these two equations are complex conjugates: $\mathbf{r}_{k,(2)} = \overline{\mathrm{T}}_k\bar{\mathbf{x}}_k = \overline{\mathrm{T}_k\mathbf{x}_k} = \bar{\mathbf{r}}_{k,(1)}$. Though the method has so far been defined with Υ and χ, alternate eigenvalues require no more modification that a change in system definition.

B. Numerical Experiments

We now solve our section model with Theodorsen aerodynamics using this Picard iteration, with parameter values given in Table 1. Figure 3 shows a contour plot of the Υ-χ form of the system (Eq. (11)) with sixteen different iteration paths of the Picard iteration with the Jacobi-Davidson solver. Only the top-right quadrant represents physical flutter point values: $\chi < 0$ and $\Upsilon < 0$ have no meaning. The oscillatory contours

near the χ-axis and the observable nondifferentiability along the Υ-axis are both features of Theodorsen's function when evaluated at these nonphysical values. Several iteration paths locate the first (physical) flutter point at $\Upsilon_F = 3.149$ Hz, $\chi_F = 0.8899$ rad/s: this point agrees with that of the contour plot. The convergence of the algorithm is monotonic, relatively uniform, and order 1.0; consistent with the general linear convergence of Picard iterations [26]. Paths tending towards the divergence point neither converge nor diverge but remain trapped in a stable oscillation around it. This is due to the non-differentiability of Theodorsen's function at $\chi = 0$, which invalidates the Taylor expansion used to produce the Picard iteration; see Pons [1].

The convergence behavior of the Picard iteration with Jacobi-Davidson and operator determinant linear solvers is observed to be the same. The difference is one of computation time. Figure 4 shows a logarithmic plot of the solution times for the Picard iteration (this time in the τ-λ form) for both linear solvers, applied to Theodorsen systems with random complex-valued matrix coefficients. The tests were carried out in MATLAB R2015b on a desktop computer. Surprisingly, the operator determinant method outperforms the Jacobi-Davidson method for all system sizes, and shows very similar scaling. It is also difficult to get the Jacobi-Davidson method to converge at system sizes greater than about 10; or a linearized system of size 30. The empirical computational complexities are order 3.7 (operator determinant) and 3.9 (Jacobi-Davidson). The cause of this poor performance and difficult convergence are not clear, but it may be related to other expensive solver components such as the staircase algorithm. From henceforth we use the operator determinant method as the linearized system solver.

To understand the convergence properties of the Picard method further, we map out a large mesh of initial values, and test their convergence numerically. This yields images of the basins of attraction around each flutter point; Figure 5 shows the Υ-χ form. The performance is poor, largely due to the presence of the divergence point which attracts the iterations but does not allow them to converge. However, we can eliminate the divergence point by changing the eigenvalue coordinates to τ and λ – in this frame the divergence point occurs at infinite τ. Figure 6 shows the convergence basins of the Picard method applied to the τ-λ form, with dramatically improved performance. Tests indicate that the basin of attraction to the first flutter point extends as far into upper right quarter plane as could possibly be needed. The divergence points, if required, can be solved for by an *a priori* substitution of $\chi = 0$ into the Υ-χ form, yielding a single-parameter nonlinear eigenvalue

problem solvable by well-established methods [27]. The τ-λ Picard iteration can then be used to compute the flutter points.

V. Newton Iteration

A. Formulation

In Section IV our approximation of Theodorsen's function corresponded to a zeroth-order Taylor approximation about the previous iterate. We might logically try to use a first-order Taylor approximation instead. In this case it would be sensible to use the τ-λ form of our section model, Eq. (10), because we will then only need to expand Theodorsen's function in terms of τ (and not λ). In the Υ-χ form we would need perform a multivariate expansion. So, we have

$$\left(\mathrm{G}_0 + \left(\mathrm{G}_1 + \mathrm{G}_2 C_\tau(\tau)\right)\tau + \mathrm{G}_3 C_\tau(\tau)\tau^2 - \mathrm{D}_0\lambda - \mathrm{K}_0\lambda^2\right)\mathbf{x} = \mathbf{0}. \tag{29}$$

Expanding $C_\tau(\tau_k)$ around τ_{k-1} we obtain:

$$C_\tau(\tau_k) = C_\tau(\tau_{k-1}) + (\tau_k - \tau_{k-1})\partial_\tau C_\tau|_{\tau_{k-1}}. \tag{30}$$

Denoting $C_{k-1} = C_\tau(\tau_{k-1})$ and $T_{k-1} = \partial_\tau C_\tau|_{\tau_{k-1}}$, we substitute Eq. (30) into Eq. (29), with all free variables evaluated at step k. The resulting expression can be rearranged to

$$\begin{aligned}(\mathrm{G}_0 + (\mathrm{G}_1 + \mathrm{G}_2 C_{k-1} - \tau_{k-1}\mathrm{G}_2 T_{k-1})\tau_k \\ + (\mathrm{G}_3 C_{k-1} - \tau_{k-1}\mathrm{G}_3 T_{k-1} + \mathrm{G}_2 T_{k-1})\tau_k^2 \\ + \mathrm{G}_3 T_{k-1}\tau_k^3 - \mathrm{D}_0\lambda_k - \mathrm{K}_0\lambda_k^2)\mathbf{x} = \mathbf{0}.\end{aligned} \tag{31}$$

which is a polynomial MPEVP, cubic in τ_k and quadratic in λ_k. We can linearize it via the same process used in Section III. Defining a new eigenvector $\mathbf{q}_k = [\mathbf{x}_k;\ \tau_k\mathbf{x}_k;\ \tau_k^2\mathbf{x}_k; \lambda_k\mathbf{x}_k]$, we obtain the linear problem

$$(\mathrm{A}_k + \mathrm{B}_k\lambda_k + \mathrm{C}_k\tau_k)\mathbf{q}_k = \mathbf{0}. \tag{32}$$

with coefficients:

$$\mathrm{A}_k = \begin{bmatrix} \mathrm{G}_0 & \mathrm{G}_1 + \mathrm{G}_2 C_{k-1} - \tau_{k-1}\mathrm{G}_2 T_{k-1} & 0 & -\mathrm{D}_0 \\ 0 & -\mathrm{I} & 0 & 0 \\ 0 & 0 & -\mathrm{I} & 0 \\ 0 & 0 & 0 & -\mathrm{I} \end{bmatrix}$$

$$\mathrm{B}_k = \{\mathrm{A} = [-\mathrm{K}_0, \mathrm{I}], \boldsymbol{i}_A = [1,4], \boldsymbol{j}_A = [4,1] \tag{33}$$

$$\mathrm{C}_k = \begin{bmatrix} 0 & \mathrm{G}_3 C_{k-1} - \tau_{k-1}\mathrm{G}_3 T_{k-1} + \mathrm{G}_2 T_{k-1} & \mathrm{G}_3 T_{k-1} & 0 \\ \mathrm{I} & 0 & 0 & 0 \\ 0 & \mathrm{I} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Note that B_k is defined using a block dictionary of keys as this is more compact, and that $\mathrm{I} = \mathrm{I}_{n\times n}$. To evaluate the matrices A_k, B_k and C_k we required one value that we have not yet obtained: $T_{k-1} = \partial_\tau C_\tau|_{\tau_{k-1}}$. We can obtain this derivative in two ways: either by differentiating Theodorsen's function analytically, or by using finite-difference approximations. In the general case (when we have a system with a truly unstructured component) we will have to use finite-differences, and so we use this method here. For convenience we use a first-order accurate forward difference scheme. We solve Eq. (32) using the operator determinant method, and we use convergence criteria identical to those of Section IV.A. This completes our formulation: we can now iterate on Eq. (32), updating A_k, B_k and C_k as the iteration progresses. This algorithm is a type of Newton iteration, as it uses a local approximation that is accurate to second order. It may also be seen as a modification of the successive linear problems algorithm of Pons [1] or Plestenjak [12] which is tailored to the form of nonlinearity present in this problem.

B. Numerical Experiments

As in Section IV.B, we now simulate our section model with Theodorsen aerodynamics, with parameter values given in Table 1. However, this time we are using the τ-λ form, Eq. (10), because the algorithm is devised for this form specifically. Figure 7 shows a contour plot with nine different iteration paths in the physical quadrant ($\tau > 0$, $\lambda > 0$) superimposed. All of the iterations converge to the first flutter point at $\tau = 3.55$, $\lambda = 1.12$ s/rad. The convergence of the algorithm can be observed (again using logarithmic plots) to be monotonic, very uniform, and of order 2.0. Newton-type methods generally show second-order convergence [28]. The fact that we are using finite-differences to compute the derivatives has not noticeably affected the convergence rate. Figure 8 shows the convergence basins of the semi-structured Newton's method for the section model with Theodorsen aerodynamics. The observed convergence behavior is extremely good, and is equivalent to that of the Picard iteration. There are no points that do not converge, and the basin of attraction around the first flutter point is unbounded upwards and to the right. Overall, we obtain the excellent convergence basins of the Picard iteration, but with second-order convergence.

VI. Discussion

A. Higher-Order Methods

The previous two methods have formed a logical progression: that of increasing order in the Taylor series approximation of Theodorsen's function. We might then ask whether it is worth increasing the order of approximation any further, to second-order or beyond. For the system we have been dealing with – the section model with Theodorsen aerodynamics – there is no real motivation to do so, as the existing methods have such good convergence properties (see Section IV.B and V.B). However, for more complex systems, where there may be multiple physical flutter points, there is more reason to look at more advanced methods. We would be particularly interested in approximations that are sufficiently good that they may be able to model multiple nearby flutter points simultaneously. We have, in fact, already met such approximations in Section III. These approximations are so good that they are able to model all physical flutter points (those inhabiting the upper right quarter plane) with a high level of accuracy – though the price is a significantly enlarged problem to solve. However, there is a subtle difference between these approximations and the Taylor approximations we have used earlier. This is that these high-accuracy approximations are entirely global: they cannot be updated with data from a previous iteration. Hence they cannot be used in an iterative method like the previous two that we have presented.

We would ideally look for methods somewhere in between the two: that contained some globally-fixed terms, and also some local terms that could be iteratively refined. This would lead to an iterative solver that, while costly in terms of time per iteration, would converge very fast, and (more importantly) would be able to accurately identify specific flutter points to avoid or to home in on. This is the major disadvantage of the two iterative solvers; it is not possible to force them converge globally to a given flutter point (e.g. the first flutter point). The iterates cannot consistently be made to escape the basin of attraction that they initially find themselves in. This is because the approximation of Theodorsen's function that is being used is not accurate enough to give any information about flutter points in any location outside the immediate vicinity of the current iterate. Ideally the approximation would be formulated in one of the efficient polynomial variable combinations (Υ-χ and τ-λ) – current approximations are exclusively in κ. Another promising solution to this problem is to use a multi-resolution approach, with an initial global approximation to estimate the flutter point locations, and then iterative local approximations to refine these estimates. For example, we could estimate the flutter point with the rational approximation of Section III.B (or potentially an even simpler and less accurate

approximation), and then use the Newton method of Section V to refine these estimates. Alternatively, the fully-nonlinear solution methods from Pons [1] or Plestenjak [12] may be used for refinement. There is significant potential here for the devising of methods with global or near-global convergence – an extremely valuable property for industrial flutter-point solvers.

B. Scalability

As an overview of their computational aspects, Figure 9 shows the computation times for the three best methods presented in this work – the Picard iteration, Newton iteration, and rational approximation solvers – against size of the model coefficient matrices (Eq. 6) when randomly generated. The tests were carried out in MATLAB R2015b, running on desktop computer, averaging multiple test results for system sizes up to 7. Of all the methods, the Picard iteration scales the best. This is something of a surprising result, and suggests that the size of the linearization used (Picard: quadratic, Newton: cubic) is more significant than the convergence speed. The rational approximation solver scales the poorest – as would perhaps be expected, given the large linearization it requires. The linear trend on the log-log plot permits an estimated of its computational complexity as order 4.4.

The current implementations of these methods are not suitable for larger problems (coefficient sizes > 10). There are, however, several immediate avenues for improvement. Firstly, significant improvements in the cost of the operator determinant method itself (and particularly the staircase algorithm) have very recently been made [29], and this is an active area of research. Secondly, more advanced polynomial linearization techniques are being developed which have the potential to generate non-singular polynomial linearizations (enabling the use of fast non-singular solvers) [30–32]. Thirdly, at least one solver for linear singular problems is available: the Tensor Rayleigh Quotient Iteration (TRQI) [33]. The TRQI is iterative and very fast, however its convergence is very erratic, only becoming reliable very near eigenvalues – hence its common use as a form of eigenvalue refinement [21]. Implementations of our Picard algorithm with TRQI yielded a fast but unreliable solver. Nevertheless there are methods by which its rapidity could be harnessed – e.g. with an outer loop process which took smaller steps (suggesting the use of under-relaxation), or with a predictor-corrector type method. The methods in this paper provide a base from which such improvements can be realized. Moreover, there is also much further work to be done on unstructured iterative methods [1,12] which can also be applied to the problems we are considering.

C. Applications

In this work we considered a section model with Theodorsen aerodynamics: tailored approximations were used for this system (Section III) and the structures of our methods were based around it (e.g. in Section V). However, the techniques involved are general and apply to a wide variety of systems. Further applications include other parametric stability problems in aeroelasticity – including nonlinear problems, as the analysis of nonlinear instability often involves the solution of linearized problems. Multiparameter methods have indeed previously been referenced in the detection of Hopf bifurcations [34]. Wider areas of application may include other disciplines involving structural stability (e.g. aircraft flight dynamics) as well as other forms of stability (e.g. hydrodynamic). While the solution of large problems – e.g. those arising from finite-element discretization – is currently not feasible with the methods presented, they are ideal for use in optimization and parametric studies. This applies particularly to the approximate direct solvers, which (with a sufficiently accurate approximation) are reliably able to compute all the flutter points of the system without any manual input or tuning. Potential applications in this area include preliminary and / or multi-disciplinary aeronautical design optimization; and real-time flutter or flight dynamics prediction. Multiparameter methods may find application in parametric stability problems across aeroelasticity and many other branches of engineering.

VII. Conclusions

In this paper we have presented two strategies and four specific solution methods for the solution of semi-structured flutter problems involving Theodorsen's function. Both strategies were based on approximating Theodorsen's function: either by a fixed global approximation or a continually-updated local approximation. The first two solution methods were direct, and were based on a rational and a fractional-order approximation to Theodorsen's function. The method based on the rational approximation is significantly more effective, being over two orders of magnitude faster, with little decrease in accuracy. These methods are direct, and so the accuracy of the final solution is fixed by the nature of the approximation used – however, these methods may be easily modified to apply to different approximations, and the approximate results can be refined via iterative methods. The other two solution methods were iterative, and were based on first- and second-order Taylor series approximations to Theodorsen's function (Picard and Newton iterations, respectively). The first-flutter point convergence basins for the methods, when applied to the τ-λ form of the system, are excellent, and occupy most of the upper-right quarter plane. However, the presence of the divergence point in the Υ-χ form does detract from the first-flutter point convergence basin in these variables. Different solution methods for the Picard- or

Newton-linearized problem are compared and discussed. The computation times and scalability of our methods are tested and discussed. While our current methods do not yet scale to very large system sizes, they are ideal for the optimization and parametric study of smaller models such as may arise in multi-disciplinary aeronautical design optimization. The methods presented in this paper serve both as a base from which faster algorithms may be devised, and a useful introduction to multiparameter methods for parametric stability analysis in aeroelasticity and in other fields.

References

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

[2] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., *Aeroelasticity*, Addison-Wesley, Reading, MA, 1957.

[3] Lind, R., "Match-Point Solutions for Robust Flutter Analysis," *Journal of Aircraft*, vol. 39, 2002, pp. 91–99.
doi: 10.2514/2.2900

[4] Borglund, D., and Ringertz, U., "Efficient computation of robust flutter boundaries using the μ-k method," *Journal of Aircraft*, vol. 43, 2006, pp. 1763–1769.
doi: 10.2514/1.20190

[5] Borglund, D., "Robust aeroelastic stability analysis considering frequency-domain aerodynamic uncertainty," *Journal of Aircraft*, vol. 40, 2003, pp. 189–193.
doi: 10.2514/2.3074

[6] Namini, A., Albrecht, P., and Bosch, H., "Finite element-based flutter analysis of cable-suspended bridges," *Journal of Structural Engineering*, vol. 118, 1992, pp. 1509–1526.
doi: 10.1061/(ASCE)0733-9445(1992)118:6(1509)

[7] Chen, P. C., "Damping perturbation method for flutter solution: the g-method," *AIAA Journal*, vol. 38, 2000, pp. 1519–1524.
doi: 10.2514/2.1171

[8] Hochstenbach, M. E., Muhič, A., and Plestenjak, B., "On linearizations of the quadratic two-parameter eigenvalue problem," *Linear Algebra and its Applications*, vol. 436, Apr. 2012, pp. 2725–2743.
doi: 10.1016/j.laa.2009.12.022

[9] Muhič, A., and Plestenjak, B., "On the quadratic two-parameter eigenvalue problem and its linearization," *Linear Algebra and its Applications*, vol. 432, 2010, pp. 2529–2542.
doi: 10.1016/j.laa.2009.12.022

[10] Muhič, A., and Plestenjak, B., "On the singular two-parameter eigenvalue problem," *Electronic Journal of Linear Algebra*, vol. 18, 2009, pp. 420–437.

doi: 10.13001/1081-3810.1322

[11] Hodges, D. H., and Pierce, G. A., *Introduction to Structural Dynamics and Aeroelasticity*, Cambridge University Press, Cambridge, UK, 2011.

[12] Plestenjak, B., 2016, "Numerical methods for nonlinear two-parameter eigenvalue problems," *BIT Numerical Mathematics*, vol. 56, pp. 241–262.

doi: 10.1007/s10543-015-0566-9

[13] White, F. M., *Fluid mechanics*, McGraw-Hill, New York, 2009.

[14] Abramowitz, M., and Stegun, I., *Handbook of Mathematical Functions*, National Bureau of Standards, Washington D.C., 1972

[15] Brunton, S. L., and Rowley, C. W., "Empirical state-space representations for Theodorsen's lift model," *Journal of Fluids and Structures*, vol. 38, 2013, pp. 174–186.

doi: 10.1016/j.jfluidstructs.2012.10.005

[16] Jones, R., "Operational treatment of the nonuniform-lift theory in airplane dynamics," NACA-TN-667, 1938

[17] Swinney, D., "A fractional calculus model of aeroelasticity," Master's Thesis, Air Force Institute of Technology, OH, 1989.

[18] Johansson, R., *Numerical python: a practical techniques approach for industry*, Apress, New York, 2015.

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

[20] Pons, A., and Gutschmidt, S., 2016, "Aeroelastic Flutter of Continuous Systems: A Generalized Laplace Transform Method," *Journal of Applied Mechanics*, vol. 83, p. 081005.

doi: 10.1115/1.4033597

[21] Plestenjak, B., and Muhič, A., 2014, *MultiParEig 1.0*, University of Ljubljana, Ljubljana, Slovenia.

[22] Almezel, S., 2013, Topics in fixed point theory, Springer, New York.

[23] Hochstenbach, M. E., Muhič, A., and Plestenjak, B., 2015, "Jacobi–Davidson methods for polynomial two-parameter eigenvalue problems," *Journal of Computational and Applied Mathematics*, vol. 288, pp. 251–263.

doi: 10.1016/j.cam.2015.04.019

[24] Hochstenbach, M. E., Kosir, T., and Plestenjak, B., 2004, "A Jacobi-Davidson Type Method for the Two-Parameter Eigenvalue Problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, pp. 477–497.

doi: 10.1137/S0895479802418318

[25] Hochstenbach, M. E., and Plestenjak, B., "A Jacobi-Davidson Type Method for a Right Definite Two-Parameter Eigenvalue Problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, Jan. 2002, pp. 392–410.

doi: 10.1137/S0895479801395264

[26] Berinde, V., *Iterative Approximation of Fixed Points*, Springer, Berlin, 2007.

[27] Voss, H., "Nonlinear Eigenvalue Problems," *Handbook of Linear Algebra*, 2nd ed., Chapman & Hall/CRC, Boca Raton, FL, 2013, pp. 1063-1086.

[28] Quarteroni, A., Sacco, R., and Saleri, F., *Numerical mathematics*, Springer, Berlin, 2007.

[29] Plestenjak, B., and Muhič, A., 2016, *MultiParEig 2.2*, University of Ljubljana, Ljubljana, Slovenia.

[30] Plestenjak, B., and Hochstenbach, M. E., 2016, "Roots of Bivariate Polynomial Systems via Determinantal Representations," *SIAM Journal on Scientific Computing*, vol. 38, pp. A765–A788.

doi: 10.1137/140983847

[31] Plestenjak, B., 2016, "Minimal determinantal representations of bivariate polynomials," *arXiv preprint*, arXiv:1607.03969.

[32] Boralevi, A., van Doornmalen, J., Draisma, J., Hochstenbach, M. E., and Plestenjak, B., 2016, "Uniform determinantal representations," *arXiv preprint*, arXiv:1607.04873.

[33] Plestenjak, B., 2000, "A Continuation Method for a Right Definite Two-Parameter Eigenvalue Problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, pp. 1163–1184.

doi: 10.1137/S0895479898346193

[34] Meerbergen, K., and Spence, A., 2010, "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*, vol. 31, pp. 1982–1999.

doi: 10.1137/080742890

Tables

Table 1: Parameter values for the section model

Parameter	Value
mass ratio – μ	20
radius of gyration – r	0.4899
bending nat. freq. – ω_h	$0.5642\ \mathrm{rad/s}$
torsional nat. freq. – ω_θ	$1.4105\ \mathrm{rad/s}$
bending damping – ζ_h	$1.4105\ \%$
torsional damping – ζ_θ	$2.3508\ \%$
static imbalance – r_θ	-0.1
pivot point location – a	-0.2

Figures

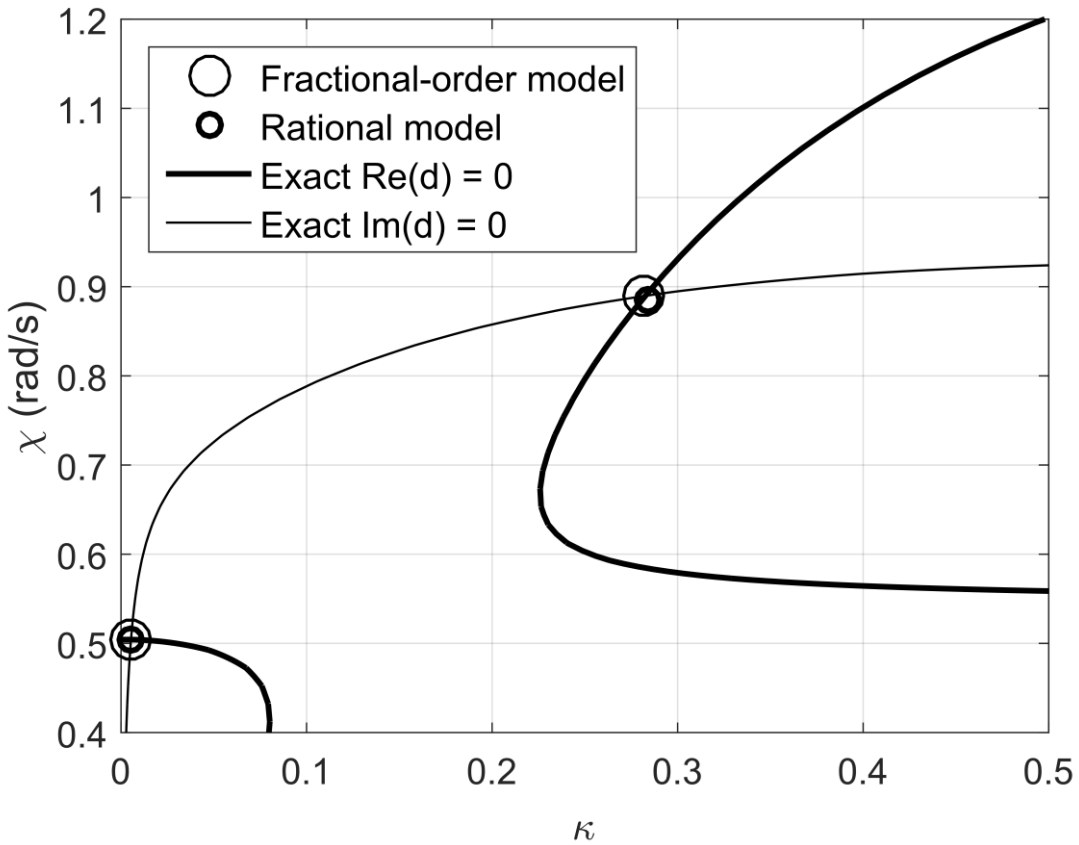

Figure 2: Direct flutter-point solutions for the two approximate models superimposed over a contour plot of the exact system.

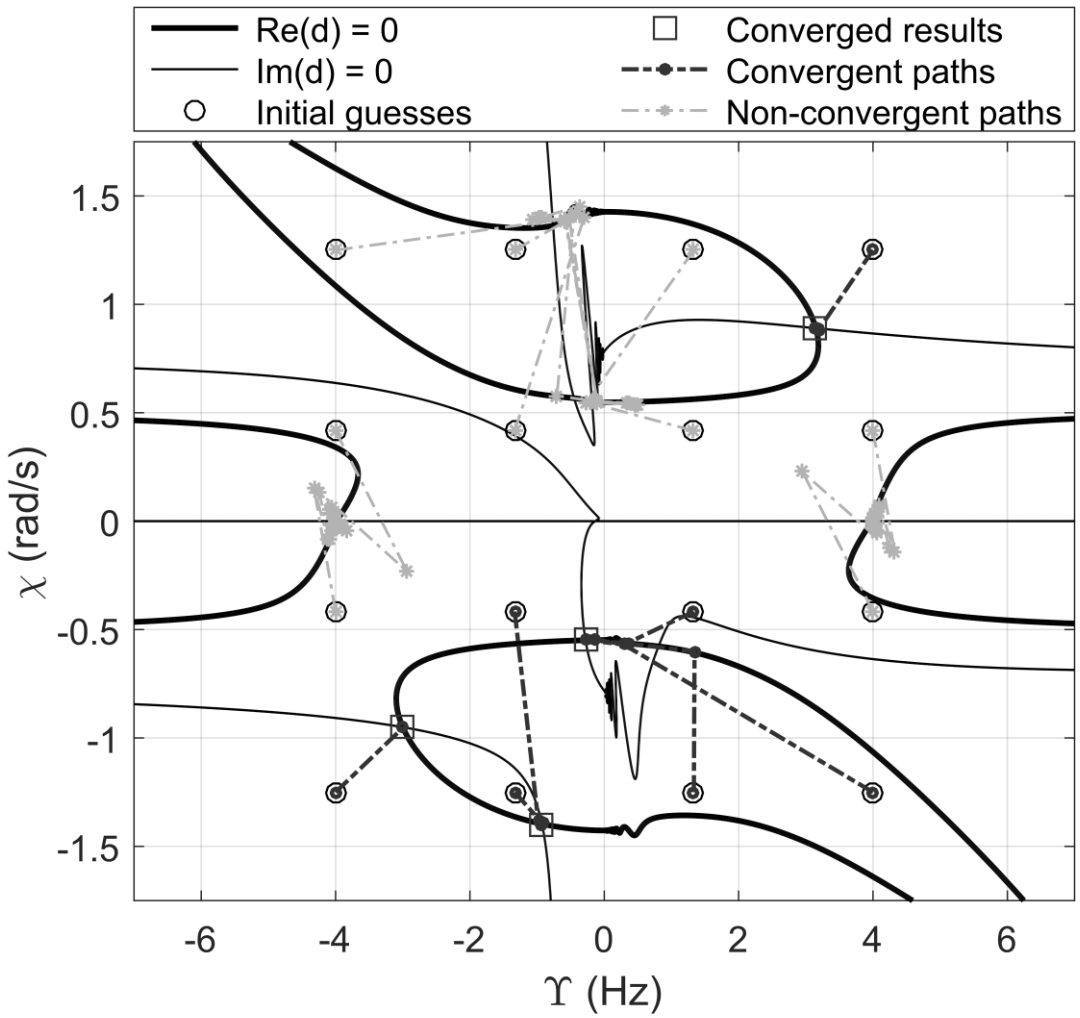

Figure 3: Sixteen iteration paths of the Picard iteration applied to the ϒ-χ form of the section model.

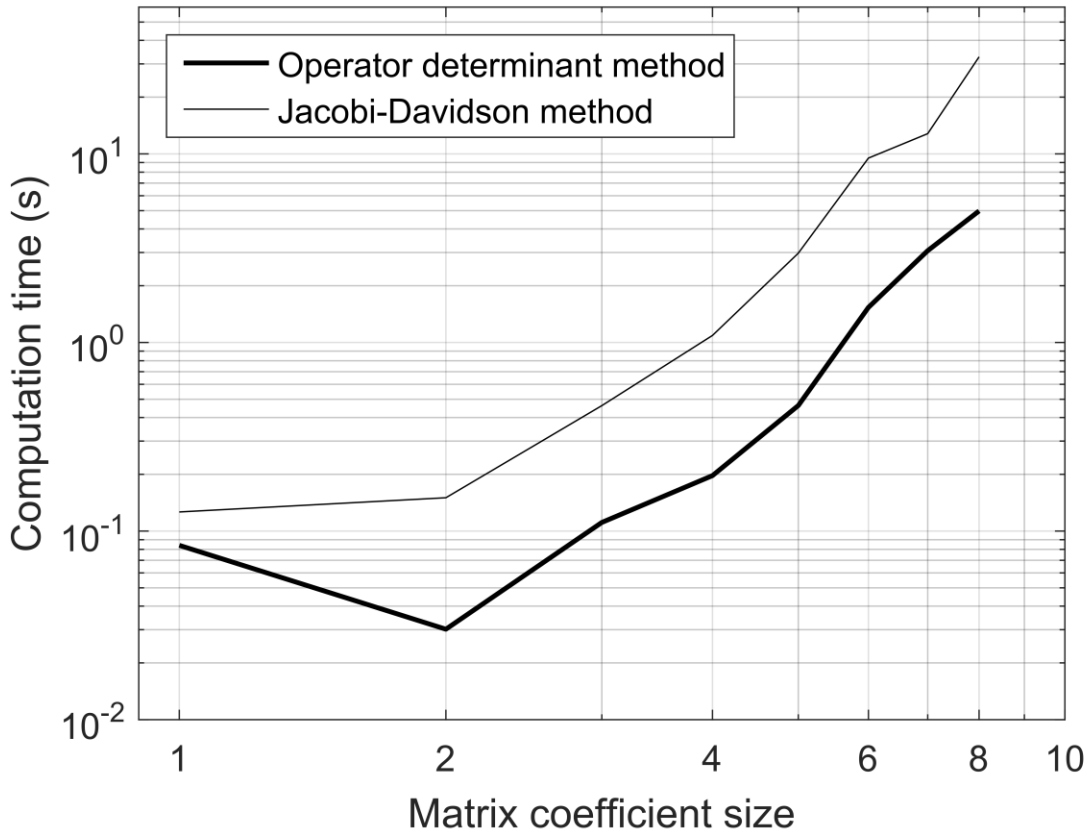

Figure 4: Convergence times for the Picard iteration with the Jacobi-Davidson and operator determinant linear solvers, applied to the τ-λ form of the section model.

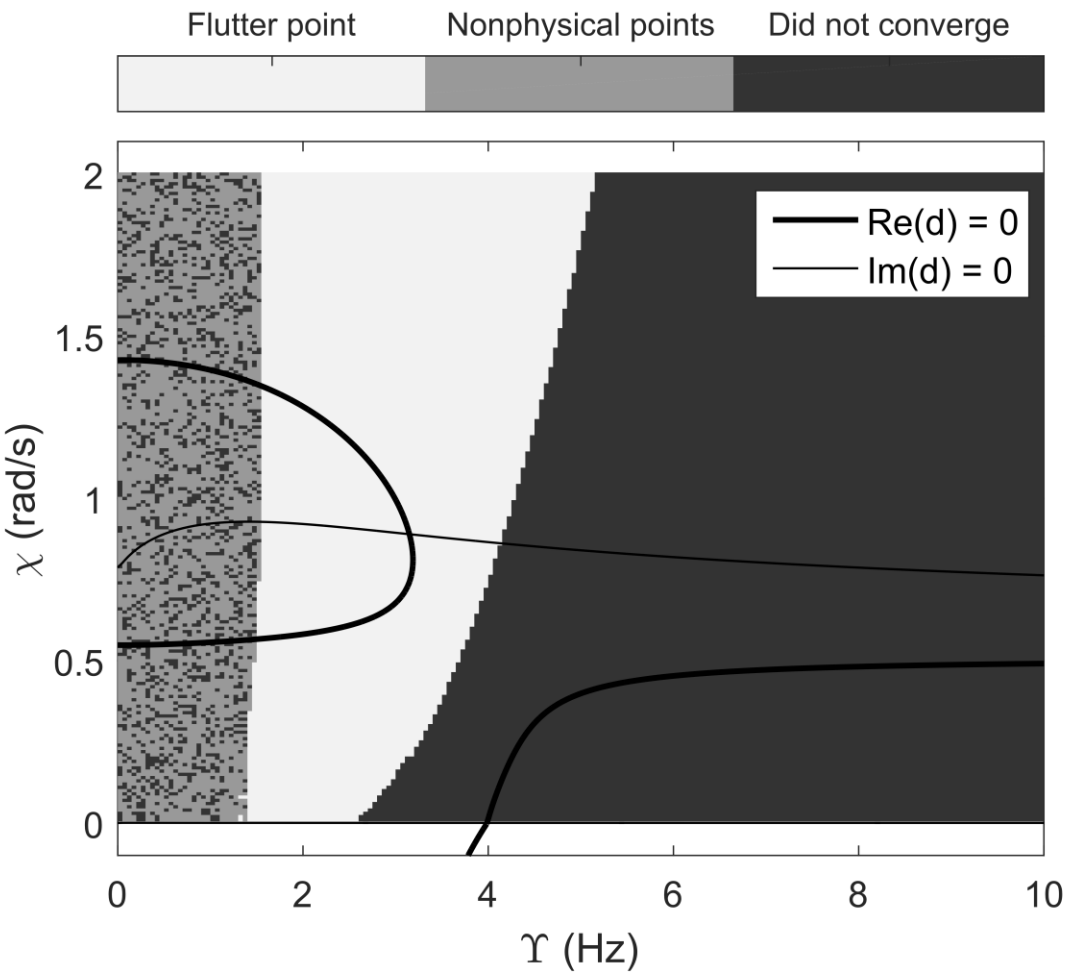

Figure 5: The convergence basins of the Picard iteration applied to the Υ-χ form of the section model.

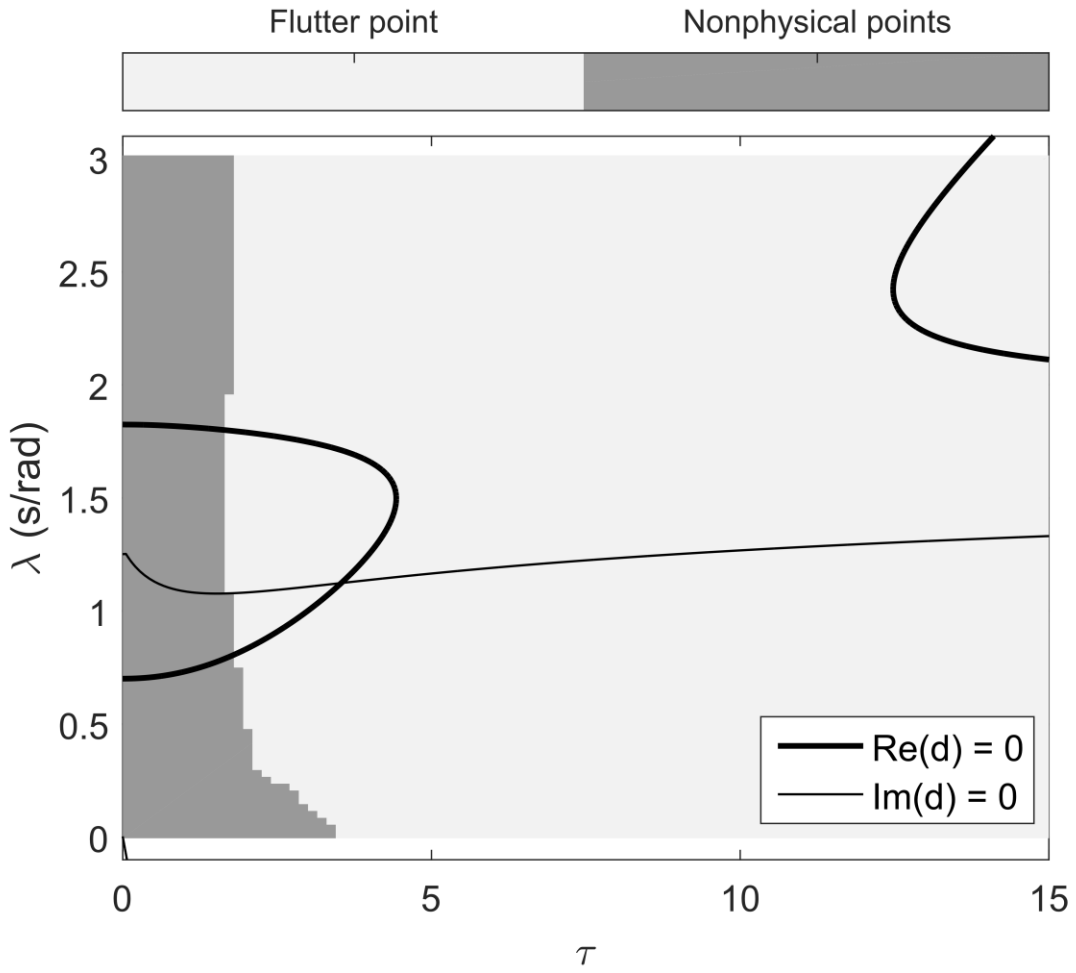

Figure 6: The convergence basins of the Picard iteration applied to the τ-λ form of the section model.

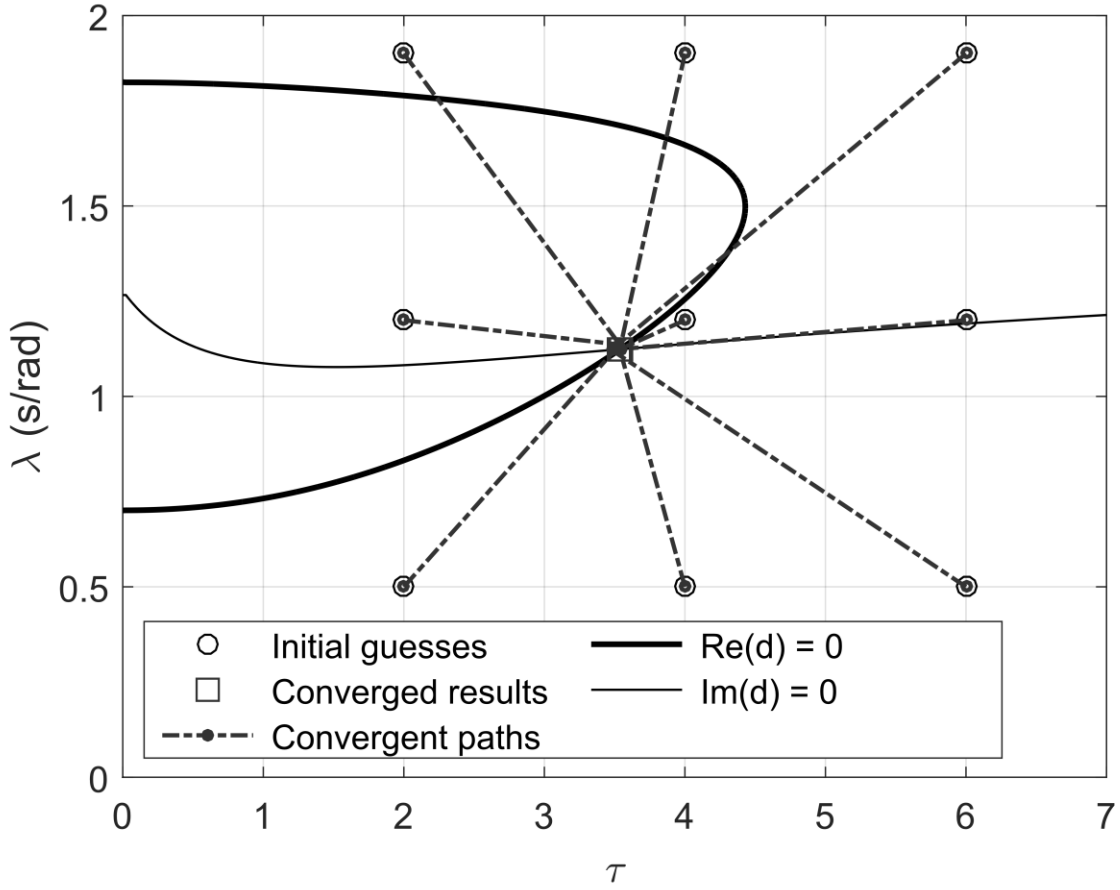

Figure 7: Nine iteration paths of the Newton iteration applied to the τ-λ form of the section model.

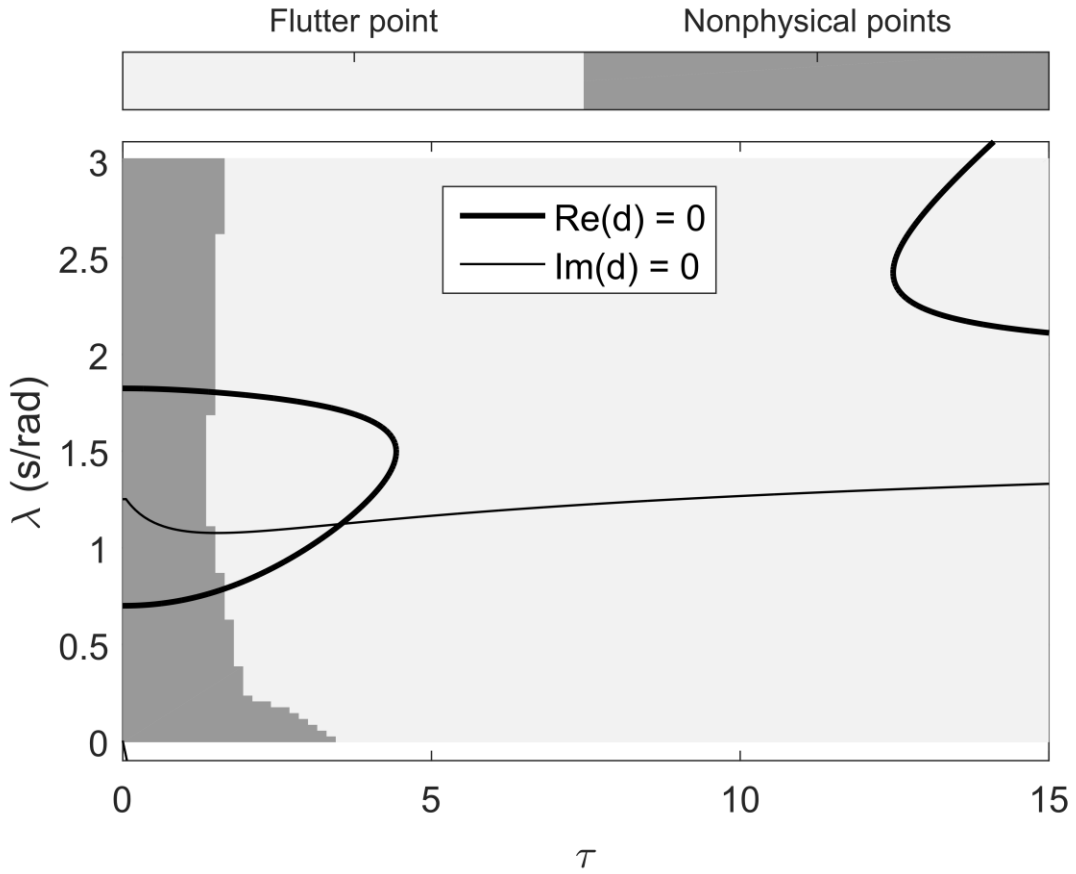

Figure 8: The convergence basins of the Newton iteration applied to the τ-λ form of the section model.

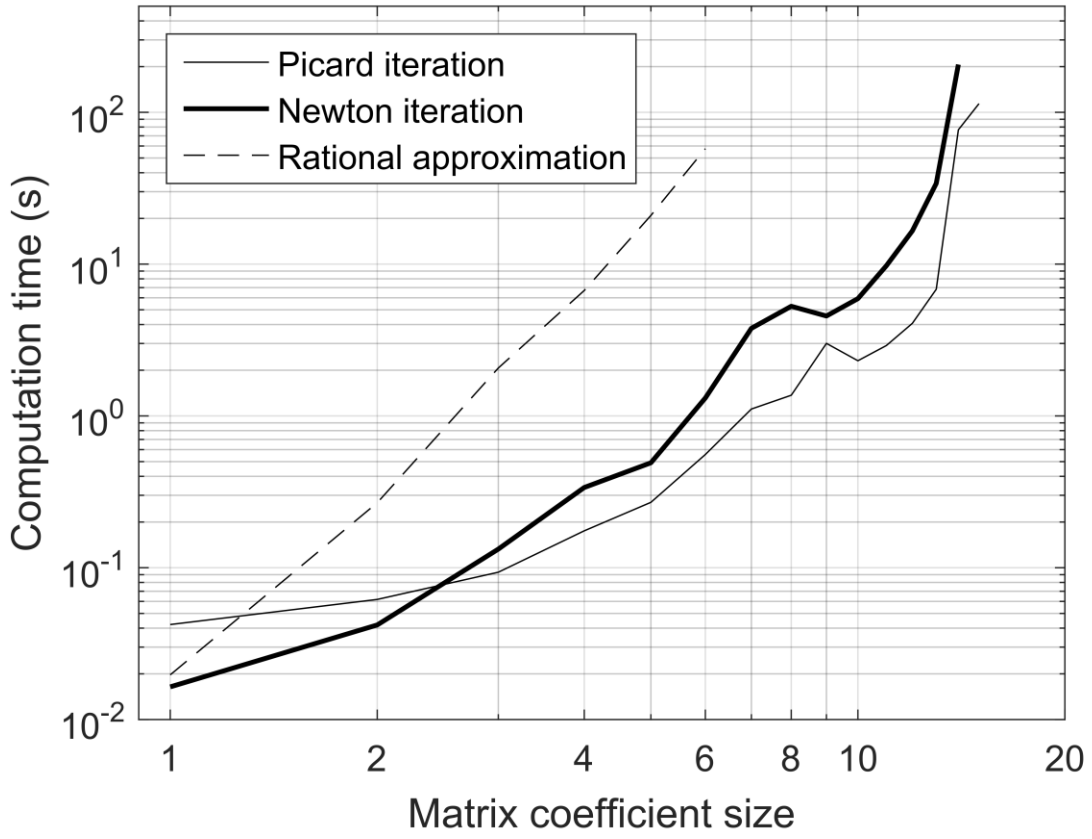

Figure 9: Convergence times for the Picard iteration, Newton iteration and rational approximation solver, applied to systems of varying matrix coefficient size.