The calculation of the distance to instability by the computation of a Jordan block

M. A. Freitag*,[a], A. Spence[a]

[a] *University of Bath, Department of Mathematical Sciences*
Claverton Down, Bath BA2 7AY, UK

Abstract

In this paper a new fast algorithm for the computation of the distance of a stable matrix to the unstable matrices is provided. The method is based on finding a two-dimensional Jordan block corresponding to a pure imaginary eigenvalue in a certain two-parameter Hamiltonian eigenvalue problem introduced by Byers (SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881). Numerical results are presented for several examples and comparison is made with the methods of Boyd & Balakrishnan (Systems Control Lett., 15 (1990), pp. 1–7) and He & Watson (SIAM J. Matrix Anal. Appl., 20 (1998), pp. 101–116).

Key words: stable matrices, distance to instability

1. Introduction

This paper introduces a method for computing the distance of a stable matrix to the set of unstable matrices. Let A be a complex $n \times n$ matrix with all its eigenvalues in the open left half plane. In this case A is called a stable matrix. Stability is a very important property for many physical and engineering applications (see, for example, [11, 12] for a collection of examples). However, a perturbation E to the matrix A may lead to eigenvalues of $A + E$ crossing the imaginary axis and hence the matrix $A + E$ being unstable. Two important papers that deal with the problem of finding the smallest perturbation E which makes $A + E$ unstable are those of van Loan [14] and Byers [4] which we now discuss.

The smallest singular value of $A \in \mathbb{C}^{n\times n}$ satisfies

$$\sigma_{\min}(A) = \min\{\|E\| \,|\, \det(A + E) = 0,\, E \in \mathbb{C}^{n\times n}\}, \tag{1}$$

where $\|\cdot\|$ denotes either the 2-norm or the Frobenius norm (see [4]). The

*Corresponding author.
Email addresses: m.freitag@maths.bath.ac.uk (M. A. Freitag), as@maths.bath.ac.uk (A. Spence)

distance of a matrix A to instability can be described as

$$\beta(A) = \min\{\|E\| \,|\, \eta(A+E) = 0,\, E \in \mathbb{C}^{n\times n}\},$$

where $\eta(A) = \max\{\mathrm{Re}(\lambda) \,|\, \lambda \in \Lambda(A)\}$. If $\eta(A)$ is negative, A is stable and if $A + E$ has an eigenvalue on the imaginary axis then E is a destabilising perturbation. In this case $(A + E - \omega iI)z = 0$ for some $\omega \in \mathbb{R}$ and $z \in \mathbb{C}^n$. Using (1) this leads to the following definition of the measure of the distance to instability of a stable matrix A as defined in [14],

$$\beta(A) = \min_{\omega\in\mathbb{R}} \sigma_{\min}(A - \omega iI),$$

where $\sigma_{\min}(A - \omega iI)$ is the smallest singular value of $A - \omega iI$. Clearly for any $\omega \in \mathbb{R}$ an upper bound on $\beta(A)$ is

$$\beta(A) \le \sigma_{min}(A - \omega iI).$$

It was shown in [14] that a lower bound $\beta(A)$ is given by

$$\frac{1}{2}\mathrm{sep}(A) \le \beta(A),$$

where

$$\mathrm{sep}(A) = \min\{\|AY + YA^H\|,\, Y \in \mathbb{C}^{n\times n},\, \|Y\| = 1\}$$

is the separation of A and $-A^H$, where A^H denotes the complex conjugate transpose of A. Clearly, $\mathrm{sep}(A) = 0$ if A has an eigenvalue on the imaginary axis.

In [4] a bisection method for computing $\beta(A)$ was introduced. The method provides lower and upper bounds on $\beta(A)$ but requires the solution of a sequence of eigenvalue problems for the $2n \times 2n$ Hamiltonian matrix

$$H(\alpha) = \begin{bmatrix} A & -\alpha I \\ \alpha I & -A^H \end{bmatrix} \tag{2}$$

for a positive real α. In [4, Theorem 1] it has been shown that $H(\alpha)$ has a pure imaginary eigenvalue if and only if $\alpha \ge \beta(A)$. It is clear that the eigenvalues of $H(0)$ are the union of the eigenvalues of A with the eigenvalues of $-A^H$, where the latter are the eigenvalues of A mirrored in the imaginary axis. If α is increased from zero some eigenvalues of $H(\alpha)$ approach the imaginary axis. Hence, in order to find the distance to instability one needs to find the minimum value of α such that $H(\alpha)$ has two identical imaginary eigenvalues. This is the basis of the numerical methods in [4, 7, 2, 3]. The theoretical discussion and consequent numerical method in this paper also exploit this observation. Under the key assumption that $H(\alpha)$ has a Jordan block of dimension 2 at the critical value of α, we derive a stable numerical algorithm to calculate the desired α and hence the distance to instability.

He & Watson [7] built on the ideas in [4] and used a method based on inverse iteration for singular values in order to find a stationary point of $f(\omega) =$

$\sigma_{min}(A-\omega iI)$ and then solved an eigenvalue problem for $H(\alpha)$ in order to check if this point is a global minimum. Boyd & Balakrishnan [2] and Bruinsma & Steinbuch [3] proposed a quadratically convergent method for the more general task of finding the H_∞-norm of a transfer function matrix which reduces to the problem discussed here in the simplest case. This algorithm requires the computation of all eigenvalues of $H(\alpha)$ at each step.

In this paper we introduce a new algorithm to find the minimum value of α such that $H(\alpha)$ has a pure imaginary eigenvalue. Our method is based on the implicit determinant method of [13] but is extended to find the value α such that $H(\alpha)-\omega iI$ has a zero eigenvalue corresponding to a 2-dimensional Jordan block. Numerical experiments presented here indicate that this method proves to be significantly faster than the methods discussed in [4, 7, 2, 3].

In Section 2 we present some background theory of Hamiltonian matrices and describe the extension of the implicit determinant method [13] to the case when $H(\alpha)-\omega iI$ has a 2-dimensional Jordan block. In Section 3 we give a theoretical analysis of the solution structure of the path $\det(H(\alpha)-\omega iI)=0$ in the (ω,α)-plane near the critical value of α. In particular we prove that a certain pair of real nonlinear equations in the two unknowns (ω,α) has an isolated solution at the critical value of α, which may be computed in a stable way by Newton's method. Section 4 contains the details of the numerical implementation of our method, including an alternative approach using a symmetric system and a checking step as used in [7]. We also provide a description of and a theoretical comparison to the methods by Boyd & Balakrishnan [2] and He & Watson [7]. In Section 5 five numerical examples are given to illustrate the theory in this paper and allow comparison with the methods in [7] and [2].

2. Background theory and the implicit determinant method

In this section we provide some background results on the spectral properties of $H(\alpha)$ defined by (2), present the main assumption and describe the mathematical approach that is used for both the theoretical development (Sections 3 and 4) and as a numerical tool (Section 5).

In the analysis of Hamiltonian matrices it is standard to introduce the matrix J, defined by

$$J=\begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \tag{3}$$

with I_n the identity matrix of size n, so that

$$J^H=J^{-1}=-J, \quad J^HJ=I, \quad J^2=-I, \quad \text{and} \quad (JH(\alpha))^H=JH(\alpha). \tag{4}$$

Using these properties it is easily shown that, if (λ,x) is a right eigenpair of $H(\alpha)$, then $(-\bar{\lambda},(Jx)^H)$ is a left eigenpair of $H(\alpha)$. An immediate consequence of this last result is that if $\lambda=i\omega$ is an imaginary eigenvalue of $H(\alpha)$ defined by (2) with eigenvector x, then the corresponding left eigenvector is $(Jx)^H$. If

$H(\alpha)$ has a pure imaginary eigenvalue $\lambda = i\omega$, $\omega \in \mathbb{R}$ and the corresponding eigenvector $x \in \mathbb{C}^{2n}$ is partitioned as $x = \begin{bmatrix} v \\ u \end{bmatrix}$, with $v, u \in \mathbb{C}^n$ then

$$(H(\alpha) - \omega i I)x = 0 \quad \Longleftrightarrow \quad \begin{bmatrix} A & -\alpha I \\ \alpha I & -A^H \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = \omega i \begin{bmatrix} v \\ u \end{bmatrix} \tag{5}$$

$$\Longleftrightarrow (A - \omega i I)v = \alpha u \quad \text{and} \quad (A - \omega i I)^H u = \alpha v. \tag{6}$$

Hence, α is a singular value of $(A - \omega i I)$ with right and left singular vectors v and u respectively.

Let α^* denote the minimum value of α at which $H(\alpha)$ has a pure imaginary eigenvalue, say $\omega^* i$ with corresponding $x^* = \begin{bmatrix} v^* \\ u^* \end{bmatrix}$. Hence $\alpha^* = \beta(A)$. It is easy to show from (6) that $\|u\| = \|v\|$. Also, $E := -\alpha^* \dfrac{u^* v^{*H}}{v^{*H} v^*}$ is the desired perturbation with $\|E\| = \alpha^*$. Our key assumption is:

Assumption 1. *$(\omega^* i, x^*)$ is a defective eigenpair of $H(\alpha^*)$ of algebraic multiplicity 2.*

Thus

$$(H(\alpha^*) - \omega^* i I)x^* = 0, \quad x^* \neq 0, \quad \text{and} \quad \dim\ker(H(\alpha^*) - \omega^* i I) = 1, \tag{7}$$

and if we denote the left null vector of $(H(\alpha^*) - \omega^* i I)$ by y^* then

$$y^{*H} x^* = 0. \tag{8}$$

Also, if $\hat{x}^*$ denotes a generalised eigenvector of $\omega^* i$ satisfying

$$(H(\alpha^*) - \omega^* i I)\hat{x}^* = x^*, \tag{9}$$

then Assumption 1 implies that

$$y^{*H} \hat{x}^* \neq 0. \tag{10}$$

Our theoretical analysis and numerical method rely on an approach called the implicit determinant method introduced in [13] which has its roots in an algorithm due to Griewank and Reddien [6] for bifurcation analysis of nonlinear parameter dependent problems. We start with the following Theorem about the nonsingularity of a certain bordered matrix.

Theorem 2. *Let Assumption 1 be satisfied and for some $c \in \mathbb{C}^n$ assume*

$$c^H x^* \neq 0. \tag{11}$$

Then the $(2n+1) \times (2n+1)$ complex matrix

$$M(\omega, \alpha) = \begin{bmatrix} H(\alpha) - \omega i I & Jc \\ c^H & 0 \end{bmatrix} \tag{12}$$

is nonsingular at $\omega = \omega^*$, $\alpha = \alpha^*$.

Proof. Using (7), Lemma 2.8 of [10] proves that $M(\alpha^*, \omega^*)$ is nonsingular if ${c^H}x^* \neq 0$ and ${y^*}^H Jc \neq 0$. However, $y^* = Jx^*$ and $J^H J = I$, so the second inequality reduces to the first one which is true by assumption (11). □

Since $M(\alpha^*, \omega^*)$ is nonsingular then so is $M(\omega, \alpha)$ for (ω, α) near (ω^*, α^*).

Let us now consider the following linear system where $c \in \mathbb{C}^n$ satisfies (11) and for (ω, α) near (ω^*, α^*),

$$\begin{bmatrix} H(\alpha) - \omega i I & Jc \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x(\omega, \alpha) \\ f(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{13}$$

where x and f are smooth functions of ω and α via the Implicit Function Theorem (cf. [6], [13, equation (26)]).

Using Cramer's rule we get

$$f(\omega, \alpha) = \frac{\det(H(\alpha) - \omega i I)}{\det M(\omega, \alpha)}, \tag{14}$$

and hence, as $M(\omega, \alpha)$ is nonsingular in a neighbourhood of (ω^*, α^*), we have

$$f(\omega, \alpha) = 0 \Longleftrightarrow \det(H(\alpha) - \omega i I) = 0. \tag{15}$$

Note also

$$f(\omega, \alpha) = 0 \Longleftrightarrow x(\omega, \alpha) \in \ker(H(\alpha) - \omega i I). \tag{16}$$

Here we see the main idea of the method, namely to seek solutions of

$$f(\omega, \alpha) = 0 \tag{17}$$

and hence recover values of α and ω such that $H(\alpha)$ has a pure imaginary eigenvalue ωi and find the corresponding eigenvector as a byproduct.

It is straightforward to show that $f(\omega, \alpha)$ is real: left multiply the first row of (13) by $x(\omega, \alpha)^H J$ to get

$$f(\omega, \alpha) = x(\omega, \alpha)^H J(H(\alpha) - \omega i I)x(\omega, \alpha), \tag{18}$$

where we have used $J^2 = -I$ and $x(\omega, \alpha)^H c = 1$, from the second row of (13), and the fact that $J(H(\alpha) - \omega i I)$ is Hermitian shows that $f(\omega, \alpha)$ is real. Equation (15) describes an important theoretical equivalence, namely, that under the assumptions of Theorem 2, the zero set of $\det(H(\alpha) - \omega i I)$ near (ω^*, α^*) is precisely the zero set of $f(\omega, \alpha)$ near (ω^*, α^*), where f is a real function of two real variables. We shall exploit this equivalence in more detail in the next section.

We end this section by noting that if Assumption 1 holds then α^* is a simple singular value of $A - \omega^* i I$.

3. Theoretical analysis of the path $\det(H(\alpha) - \omega i I) = 0$ in the (ω, α)-plane

In this section we use the equivalence given by (15) to give a theoretical analysis of the structure of the solutions of $\det(H(\alpha) - i\omega I) = 0$ in the (ω, α)-plane near (ω^*, α^*). We do this by analysing the path $f(\omega, \alpha) = 0$. We start with a simple Lemma which has significant consequences.

Lemma 3. *Let Assumption 1 be satisfied and assume (11) holds. Consider the real curve $f(\omega, \alpha) = 0$. Then, near (ω^*, α^*)*

$$f_\alpha(\omega, \alpha) = \|x(\omega, \alpha)\|^2 > 0. \tag{19}$$

Proof. Differentiate the linear system (13) with respect to α to obtain

$$\begin{bmatrix} H(\alpha) - \omega i I & Jc \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_\alpha(\omega, \alpha) \\ f_\alpha(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} Jx(\omega, \alpha) \\ 0 \end{bmatrix}, \tag{20}$$

where we have used $H_\alpha(\alpha) = -J$ from (2). Multiply the first equation from the left by $(Jx(\omega, \alpha))^H$, the left null vector of $H(\alpha) - \omega i I$, to give

$$f_\alpha(\omega, \alpha) = (Jx(\omega, \alpha))^H Jx(\omega, \alpha) = \|Jx(\omega, \alpha)\|^2 = \|x(\omega, \alpha)\|^2 > 0,$$

where we have used $J^H J = I$ and $x(\omega, \alpha)^H c = 1$. □

Hence, under Assumption 1 and (11), and using the result of Lemma 3, the Implicit Function Theorem shows that near (ω^*, α^*), $\alpha = \alpha(\omega)$ and $f(\omega, \alpha(\omega)) = 0$. Thus, there is a smooth path of solutions to $f(\omega, \alpha) = 0$ parameterised by ω in the (ω, α)-plane near (ω^*, α^*).

Next we focus attention on values f_ω and $f_{\omega\omega}$ at the point (ω^*, α^*).

Lemma 4. *Let Assumption 1 be satisfied, assume (11) holds and let A in (2) be a stable matrix. Then*

$$(a) \qquad f^*_\omega := f_\omega(\omega^*, \alpha^*) = 0 \tag{21}$$

$$(b) \qquad f^*_{\omega\omega} := f_{\omega\omega}(\omega^*, \alpha^*) < 0. \tag{22}$$

Proof. (a) Similar to the proof of Lemma 3 we start by differentiating (13) with respect to ω to obtain

$$\begin{bmatrix} H(\alpha) - \omega i I & Jc \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_\omega(\omega, \alpha) \\ f_\omega(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} ix(\omega, \alpha) \\ 0 \end{bmatrix}. \tag{23}$$

Evaluate at (ω^*, α^*), multiply the first row from the left by ${y^*}^H = (Jx^*)^H$ to get

$$f_\omega(\omega^*, \alpha^*) = i{y^*}^H x(\omega^*, \alpha^*) = 0 \tag{24}$$

using (8). Hence, using (9), the first row of (23) evaluated at (ω^*, α^*) with (24) gives

$$x^*_\omega := x_\omega(\omega^*, \alpha^*) = i\hat{x}^*, \tag{25}$$

so $x_\omega(\omega^*, \alpha^*)$ is a generalised eigenvector belonging to $\omega^* i$.

(b) Differentiate the linear system (23) with respect to ω to obtain

$$\begin{bmatrix} H(\alpha) - \omega i I & Jc \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_{\omega\omega}(\omega, \alpha) \\ f_{\omega\omega}(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} 2i x_\omega(\omega, \alpha) \\ 0 \end{bmatrix}. \tag{26}$$

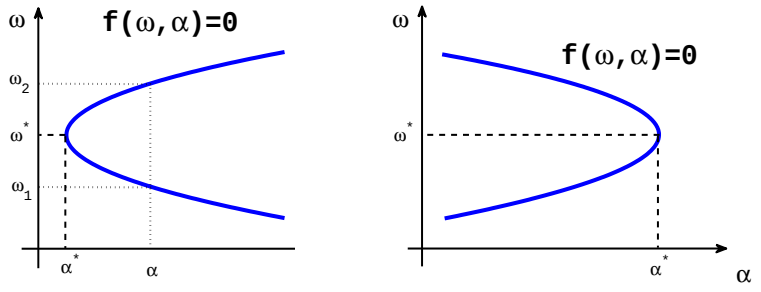
Evaluate at (ω^*, α^*), multiply the first equation on the left by ${y^*}^H$ to get

$$f_{\omega\omega}(\omega^*, \alpha^*) = 2i {y^*}^H x_\omega(\omega^*, \alpha^*) = -2{y^*}^H \hat{x}^* \neq 0,$$

using (25) and (10). Hence, using simple calculus it is easy to show that near (ω^*, α^*), with $\alpha^* = \alpha(\omega^*)$, the Taylor series expansion of $\alpha(\omega)$ has the form

$$\alpha(\omega) = \alpha^* - (\omega - \omega^*)^2 \frac{f^*_{\omega\omega}}{2 f^*_\alpha} + \text{h.o.t.},$$

so that, recalling (19), there are two possibilities for the solution structure, depending on the sign of $f^*_{\omega\omega}$ (see Figure 1). To prove (22) and hence case



(a) Case $f^*_{\omega\omega} < 0$ (b) Case $f^*_{\omega\omega} > 0$

Figure 1: Curve $f(\omega, \alpha) = 0$ in the (ω, α)-plane for $f^*_{\omega\omega} < 0$ and $f^*_{\omega\omega} > 0$

(a) in Figure 1 applies, we use contradiction. Assume $f^*_{\omega\omega} > 0$, so case (b) in Figure 1 applies. Now, from (19), the curve of $f(\omega, \alpha) = 0$ has no intersections. Also, since α^* is the minimum value that $f^*_\omega = 0$ (that is where $H(\alpha)$ has a Jordan block with pure imaginary eigenvalues) the curve $f(\omega, \alpha) = 0$ cannot turn back on itself for $\alpha < \alpha^*$. The curve is bounded using the Implicit Function Theorem applied to (13). Hence, the path $f(\omega, \alpha) = 0$ meets the imaginary axis at $\alpha = 0$ and so $H(0)$ has pure imaginary eigenvalues. This contradicts the fact that A is stable. Hence our assumption that $f^*_{\omega\omega} > 0$ is false.

□

Lemmata 3 and 4 prove that the solution structure of $f(\omega, \alpha) = 0$, and hence of $\det(H(\alpha) - i\omega I) = 0$, is as in Figure 1a. Thus, for $\alpha > \alpha^*$, there are two

real values of ω, say ω_1 and ω_2 as in Figure 1a, that correspond to algebraically simple eigenvalues of $H(\alpha)$, since $f_\omega \neq 0$ at these points. For $\alpha < \alpha^*$ there are no real solutions. Not surprisingly, this is in complete agreement with the treatment in [4]. Also, we have shown that, under Assumption 1, when two algebraically simple pure imaginary eigenvalues of $H(\alpha)$ coalesce at α^* as α varies, then they split to form two complex eigenvalues off the imaginary axis. They do not simply pass through each other and remain on the imaginary axis. In the terminology of Hamiltonian systems, the two pure imaginary eigenvalues have opposite signature. However, most useful for the numerical method in this paper is the discussion in the following paragraph.

In the language of bifurcation theory, if $H(\alpha^*) - i\omega^* I$ has a 2-dimensional Jordan block (Assumption 1), then (ω^*, α^*) is a structurally stable quadratic turning point of $f(\omega, \alpha) = 0$ as in Figure 1a (see also [9, Figure 3.1]). The point (ω^*, α^*) may be calculated in a stable manner by solving

$$g(\omega, \alpha) = \begin{bmatrix} f(\omega, \alpha) \\ f_\omega(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{27}$$

since the Jacobian of $g(\omega, \alpha)$ at the root (ω^*, α^*) is the 2×2 matrix

$$G(\omega^*, \alpha^*) = \begin{bmatrix} f_\omega(\omega^*, \alpha^*) & f_\alpha(\omega^*, \alpha^*) \\ f_{\omega\omega}(\omega^*, \alpha^*) & f_{\omega\alpha}(\omega^*, \alpha^*) \end{bmatrix} = \begin{bmatrix} 0 & f_\alpha^* \\ f_{\omega\omega}^* & f_{\omega\alpha}^* \end{bmatrix}, \tag{28}$$

which is nonsingular by (19) and (22). Note that when solving $g(\omega, \alpha) = 0$ the variables ω and α are independent variables since they are not restricted to lie on the curve $f(\omega, \alpha) = 0$. We describe how to solve $g(\omega, \alpha) = 0$ in the next section.

The equivalence of solutions of $f(\omega, \alpha) = 0$ and solutions of $\det(H(\alpha) - i\omega I) = 0$ shows that the calculation of (ω^*, α^*) such that $H(\alpha^*) - i\omega^* I$ has a 2-dimensional Jordan block is a stable numerical process. The key point here is that α is allowed to vary, so this result does not contradict the fact that the computation of a Jordan block of a *fixed* matrix is an unstable process.

The analysis in this section is in the spirit of that in [8] but is completely different and leads to a considerably simpler numerical method.

4. The calculation of $\alpha^* = \beta(A)$

The analysis in the previous section shows that the critical values (ω^*, α^*) such that $H(\alpha^*) - i\omega^* I$ has a 2-dimensional Jordan block may be calculated numerically by finding a zero of the two real nonlinear equations in two real variables given by (27). The analysis also shows that under Assumptions 1 and (11) (ω^*, α^*) is an isolated zero of $g(\omega, \alpha)$. It is natural to compute (ω^*, α^*) using Newton's method, and this is what we now explain. One nice feature of the numerical method is that it mirrors directly the theory of the previous section.

4.1. Newton's method

We now describe Newton's method to solve $g(\omega, \alpha) = 0$. Newton's method with a starting guess $(\omega^{(0)}, \alpha^{(0)})$ gives the sequence of linear systems

$$\begin{aligned} G(\omega^{(i)}, \alpha^{(i)}) \begin{bmatrix} \Delta\omega^{(i)} \\ \Delta\alpha^{(i)} \end{bmatrix} &= -g(\omega^{(i)}, \alpha^{(i)}), \\ \begin{bmatrix} \omega^{(i+1)} \\ \alpha^{(i+1)} \end{bmatrix} &= \begin{bmatrix} \omega^{(i)} \\ \alpha^{(i)} \end{bmatrix} + \begin{bmatrix} \Delta\omega^{(i)} \\ \Delta\alpha^{(i)} \end{bmatrix}, \end{aligned} \tag{29}$$

for $i = 0, 1, 2 \ldots$ until convergence, where the Jacobian is

$$G(\omega^{(i)}, \alpha^{(i)}) = \begin{bmatrix} f_\omega(\omega^{(i)}, \alpha^{(i)}) & f_\alpha(\omega^{(i)}, \alpha^{(i)}) \\ f_{\omega\omega}(\omega^{(i)}, \alpha^{(i)}) & f_{\omega\alpha}(\omega^{(i)}, \alpha^{(i)}) \end{bmatrix}. \tag{30}$$

The values of $f^{(i)}$, $f_\alpha^{(i)}$, $f_\omega^{(i)}$ and $f_{\omega\omega}^{(i)}$ are calculated from (13), (20), (23) and (26). The value for $f_{\omega\alpha}^{(i)}$ can be calculated by differentiating (23) with respect to α, that is

$$\begin{bmatrix} H(\alpha) - \omega i I & Jc \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_{\omega\alpha}(\omega, \alpha) \\ f_{\omega\alpha}(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} Jx_\omega(\omega, \alpha) + ix_\alpha(\omega, \alpha) \\ 0 \end{bmatrix}. \tag{31}$$

Hence, in order to calculate $g(\omega, \alpha)$ and $G(\omega, \alpha)$ given by (27) and (30) we need to solve the systems (13), (23), (26), (20) and (31) which all use the same nonsingular system matrix $M(\omega, \alpha)$ from (12) and hence, only one LU factorisation is needed per iteration.

Note that Newton's method is only carried out in the two-dimensional (ω, α)-plane, which is intuitively natural since once the two scalars (ω^*, α^*) are calculated the problem is essentially solved.

Algorithm 1 (Newton's method)**.** Given $(\omega^{(0)}, \alpha^{(0)})$ and $c \in \mathbb{C}^n$ such that $M(\omega^{(0)}, \alpha^{(0)})$ is nonsingular; set $i = 0$:

(i) Solve (13) and (23) (using the $x(\omega^{(i)}, \alpha^{(i)})$ obtained in (13) for the right hand side of (23)) in order to find

$$g(\omega^{(i)}, \alpha^{(i)}) = \begin{bmatrix} f(\omega^{(i)}, \alpha^{(i)}) \\ f_\omega(\omega^{(i)}, \alpha^{(i)}) \end{bmatrix},$$

and $x(\omega^{(i)}, \alpha^{(i)})$, $x_\omega(\omega^{(i)}, \alpha^{(i)})$.

(ii) Solve (20), (26) and (31) (using $x_\alpha(\omega^{(i)}, \alpha^{(i)})$ obtained in (20)) in order to find the Jacobian $G(\omega^{(i)}, \alpha^{(i)})$ given by (30).

(iii) Newton update: Solve (29) in order to get $(\omega^{(i+1)}, \alpha^{(i+1)})$

(iv) Repeat until convergence.

The method is well-defined, that is, the Jacobian $G(\omega^{(i)}, \alpha^{(i)})$ is nonsingular for a starting guess that is close enough to the solution, since the matrix $G(\omega^*, \alpha^*)$ given by (28) is nonsingular under Assumption 1 and (11). Convergence is quadratic as seen in the examples in Section 5.

Finally, note that a good choice for c in matrix $M(\omega^{(i)}, \alpha^{(i)})$ from (12) which satisfies assumption (11) is

$$c \approx x(\omega^*, \alpha^*). \tag{32}$$

In the numerical examples we take $\alpha^{(0)} = 0$ and for $\omega^{(0)}$ we choose the imaginary part of the eigenvalue of A which is closest to the imaginary axis. Then we take

$$c = x^{(0)} = \begin{bmatrix} v(\omega^{(0)}, \alpha^{(0)}) \\ u(\omega^{(0)}, \alpha^{(0)}) \end{bmatrix},$$

where $v(\omega^{(0)}, \alpha^{(0)})$ and $u(\omega^{(0)}, \alpha^{(0)})$ are right and left singular vectors of $A - \omega^{(0)} iI$.

4.2. Symmetric method

Instead of working with the non-Hermitian system (13),

$$\begin{bmatrix} H(\alpha) - \omega iI & Jc \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x(\omega, \alpha) \\ f(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{33}$$

we could work with a Hermitian system. Multiplying (33) by $\begin{bmatrix} -J & 0 \\ 0^H & 1 \end{bmatrix}$ leads to the Hermitian system

$$\begin{bmatrix} -JH(\alpha) + \omega iJ & c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x(\omega, \alpha) \\ f(\omega, \alpha) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

as $(JH(\alpha))^H = (JH(\alpha))$ and $J^H = -J$. Hence, instead of an LU factorisation, an LDL^T transformation could be used when solving the systems within Newton's method. However for our relatively small-scale examples we have not seen a significant benefit; similar timings are obtained as for the non-Hermitian formulation. Taking the Hermitian form may be beneficial for large-scale problems, however.

4.3. Testing step

It is possible that the computed α, say α^{comp}, is actually too large, that is, there exists a smaller value of $\alpha^* < \alpha^{\text{comp}}$ such that $H(\alpha^*)$ has two pure imaginary eigenvalues and for any small value θ the matrix $H(\alpha^* - \theta)$ does not have any pure imaginary eigenvalue. Therefore we include a checking step into our algorithm, in order to ensure that the computed α^{comp} is the smallest possible α such that $H(\alpha)$ has two imaginary eigenvalues. We use the checking idea of He & Watson [7].

After convergence of Algorithm 1, that is

$$|\alpha^{(i+1)} - \alpha^{(i)}| \leq \tau \quad \text{and} \quad \|g(\omega^{(i)}, \alpha^{(i)})\| \leq \tau, \tag{34}$$

for some tolerance τ, we set $\alpha^{\text{comp}} = \alpha^{(i+1)}$ and check the eigenvalues of $H(\alpha^{\text{comp}} - \theta)$ for some small tolerance θ. If $H(\alpha^{\text{comp}} - \theta)$ does not have a pure imaginary eigenvalue we stop the algorithm and set $\alpha^* = \alpha^{\text{comp}}$.

Otherwise, we reduce the value of α^{comp} using

$$\alpha^{\text{comp}} := \delta \cdot \alpha^{\text{comp}}, \quad \delta \in (0, 1)$$

until we find a value of α^{comp} such that $H(\alpha^{\text{comp}})$ does not have a pure imaginary eigenvalue. Then we set $\alpha^{(0)} = \alpha^{\text{comp}}$ and restart Algorithm 1 with $(\omega^{(0)}, \alpha^{(0)})$, where for $\omega^{(0)}$ we choose the imaginary part of the eigenvalue of $H(\alpha^{(0)})$ which is closest to the imaginary axis.

Note that this checking step is carried out in a similar fashion as in He & Watson [7] (that is, we use the QR method in order to solve this eigenvalue problem) and tests if the smallest value of α^* is found. Our method is very fast but is not guaranteed to find the minimum value for α at first since it is based on Newton's method. However, in all our test problems it did, in fact, find the desired α^*. An implementation of a hybrid scheme which uses our new method to provide starting guesses for the algorithm in [2, 3] is a possible extension that we do not explore here.

4.4. Algorithm complexity

In [2] and [3], a quadratically convergent method for the more general task of finding the H_∞-norm of a transfer function matrix has been proposed. This can be translated into our framework to give the following algorithm:

Algorithm 2 (Boyd&Balakrishnan)**.** Given A, $\alpha \geq \beta(A)$ and a tolerance:

(i) Compute all pure imaginary eigenvalues $iw_1, \ldots, iw_l$ of $H(\alpha)$ given by (2), ordered so that $w_1 \leq w_2 \leq \ldots \leq w_l$.

(ii) Set $s_k = \dfrac{w_k + w_{k+1}}{2}$, $k = 1, \ldots, l-1$ and update $\alpha = \min_k \sigma_{\min}(A - s_k iI)$

(iii) Repeat until convergence.

This algorithm converges quadratically, however, requires the solution of a $2n \times 2n$ eigenvalue problem at each step. Now, the cost of calculating all the eigenvalues using the QR method is $10(2n)^3 = 80n^3$ flops (see [5, p. 359]). In this discussion we ignore the cost of step (ii), though if l is large this may be significant also.

The Newton method we propose also converges quadratically, however, requires a LU factorisation of a $(2n+1) \times (2n+1)$ matrix at each step, costing $\dfrac{2}{3}(2n+1)^3 \approx \dfrac{16}{3}n^3$ flops (this is a conservative estimate for a full matrix, for sparse matrices this can be reduced). As a precaution we also implement a test step as in [7]. This test step which uses an eigenvalue computation requires $80n^3$ flops.

Let N_1 be the number of Newton steps for our new method, N_2 be the number of test steps and N_3 be the number of eigenvalue solves required by the Boyd&Balakrishnan algorithm. Then our new method is more efficient if

$$\left(\frac{16}{3}N_1 + 80\right) N_2 < 80N_3.$$

Due to the quadratic convergence $N_3 \approx 6-7$. Our algorithm usually takes only one test step $N_2 = 1$ with a quadratically convergent Newton method, N_1 is always less than 6, and hence is likely to be faster than method of [2]. If we take $N_3 = 6$ and $N_1 = 6$ (as both algorithms are quadratically convergent), then this rough heuristic complexity argument suggests the approach in this paper is likely to outperform the method in [2] unless $N_2 = 5$ test steps are required.

The method in this paper updates estimates of (ω, α) in one step (using Newton's method applied to two real nonlinear equations), whereas the methods in [2, 3] and [7] involve a two stage process: first there is an update in the ω-direction, then an update in the α-direction. Thus, the method in this paper has the potential to be faster.

In the next section we present several numerical examples that show the numerical performance of our method. We also compare iteration numbers and CPU times of our proposed method with the algorithms in [2, 3] and [7].

5. Numerical examples

We first describe three examples that were used in [7]. In all problems we apply both the Newton method in order to find the distance to instability $\beta(A)$. We use $\alpha^{(0)} = 0$ and for $\omega^{(0)}$ we choose the imaginary part of the eigenvalue of A which is closest to the imaginary axis. For c in (13) we take $c = x^{(0)} = \begin{bmatrix} v(\omega^{(0)}, \alpha^{(0)}) \\ u(\omega^{(0)}, \alpha^{(0)}) \end{bmatrix}$, where $v(\omega^{(0)}, \alpha^{(0)})$ and $u(\omega^{(0)}, \alpha^{(0)})$ are right and left singular vectors of $A - \omega^{(0)} iI$. These are obvious choices to make as we are looking for the value of α closest to zero and it is expected that the value of ω closest to the imaginary axis is most likely to be a good starting guess for ω^*. We will see that with those initial guesses for Newton's method we obtain very good results for all our examples. All computations were performed in MATLAB Version 7.8.0.347 (R2009a).

Example 5. Consider

$$A = \begin{bmatrix} -0.4+6i & 1 & & & \\ 1 & -0.1+i & 1 & & \\ & 1 & -1-3i & 1 & \\ & & 1 & -5+i & \end{bmatrix}$$

which has eigenvalues (rounded to 3 significant digits)

$$\Lambda(A) = \{-0.41 + 5.80i, -0.04 + 0.95i, -0.92 - 2.62i, -5.13 + 0.87i\}$$

so that A is stable. The imaginary part of the eigenvalue of A which is closest to the imaginary axis is $\omega^{(0)} = 0.953057740164838$. We stop the computation once (34) is satisfied, where $\tau = 10^{-11}$. In order to check the eigenvalues of $H(\alpha^{\text{comp}} - \theta)$ we use $\theta = 10^{-10}$.

Table 1: Results for Example 5.

	Newton method		
i	$\omega^{(i)}$	$\alpha^{(i)}$	$\|g(\omega^{(i)}, \alpha^{(i)})\|$
0	0.953057740164838	0	-
1	0.953036248966042	0.031887014318100	1.594990002001362e-02
2	0.953014724735990	0.031887009443620	2.257727998248458e-04
3	0.953014724704841	0.031887014303200	2.447309320759901e-09
4	0.953014724704841	0.031887014303200	6.161287176187464e-16

The results for Example 5 are shown in Table 1 and we indeed see very fast (quadratic) convergence of Newton's method.

For this simple 4×4 example the computation times for Newton method, the Boyd & Balakrishnan algorithm and the method proposed by He & Watson are very small and not reported here.

Example 6. Let A be the matrix `bwm200.mtx` from the matrix market library [1]. It is the discretised Jacobian of the Brusselator wave mode for a chemical reaction. The dimension of this matrix is 200, with 796 nonzero elements. The 32 rightmost eigenvalues, which are all in the left half plane are shown in Figure 2. The initial guess $(\omega^{(0)}, \alpha^{(0)})$ and the value for c is as discussed

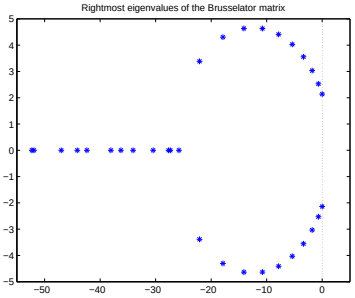


Figure 2: Rightmost eigenvalues of the Brusselator matrix in Example 6.

in the first paragraph of this section. The eigenvalue of A which is closest to the imaginary axis is $-0.000018199876628 \pm 2.139497522076030i$ and hence $\omega^{(0)} = 2.139497522076030$. We stop the computation once (34) is satisfied, where $\tau = 10^{-10}$. In order to check the eigenvalues of $H(\alpha^{\text{comp}} - \theta)$ we use $\theta = 10^{-14}$.

Table 2: Results for Example 6.

	Newton method		
i	$\omega^{(i)}$	$\alpha^{(i)}$	$\|g(\omega^{(i)}, \alpha^{(i)})\|$
0	2.139497522076343	0	-
1	2.139497522045502	0.000008240971700	4.191183100020208e-06
2	2.139497522014727	0.000008240971687	3.828424651364583e-07
3	2.139497522014739	0.000008240971689	1.624908978281682e-10
4	2.139497522014746	0.000008240971691	8.163859421299612e-11

Table 2 shows the results for Example 6 and we observe fast convergence of Newton's method.

Table 3 shows the computation times for this example. The iterations and CPU times in the first row refer to the ones of the algorithm used by Boyd & Balakrishnan [2] (see also Bruinsma & Steinbuch [3]) and the ones in the second row refer to the ones of the algorithm used by He & Watson [7]. We have reimplemented the methods of He & Watson and Boyd & Balakrishnan in a newer version of MATLAB in order to make the results comparable to our computation times. The algorithm in He & Watson delivers a lower and upper

Table 3: CPU times for Example 6.

Algorithm	Iterations	CPU times	Test steps	Test steps times	CPU times without test steps
Boyd/Balakrishnan	6	31.85 s	–	–	–
He/Watson	92	2.37 s	$1-2$	0.63 s	1.73 s
Newton	4	1.43 s	1	0.47 s	0.96 s

bound on α. We have carried out several tests for each of the three methods in order to obtain average CPU times. Based on the times in Table 3, for this example, Newton's method is faster than the method proposed in [2] and the performance of our algorithm is comparable to the performance of the method in [7], although, in general, slightly faster.

Example 7. Consider the Orr-Sommerfeld operator

$$\frac{1}{\gamma R}L^2 v - i(UL - U'')v = \lambda L v, \quad \text{where} \quad L = \frac{d^2}{dx^2} - \gamma^2 \quad \text{and} \quad U = 1 - x^2.$$

Discretising the operator on $v \in [-1, 1]$ using finite differences (see [7] for details) yields a generalised eigenproblem

$$B_n u_n = \lambda_n L_n u_n$$

and with $\gamma = 1$, $R = 1000$ and $n = 1000$ we obtain a standard eigenvalue problem $A_n u_n = L_n^{-1} B_n u_n = \lambda_n u_n$ and the spectrum of A_{1000} is plotted in Figure 3.

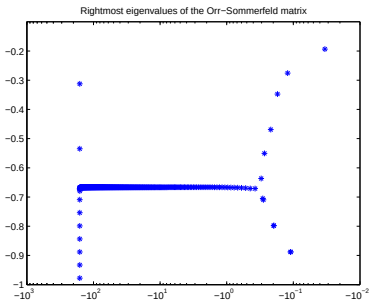


Figure 3: Eigenvalues of the Orr-Sommerfeld matrix in Example 7.

The eigenvalue of A_{1000} closest to the imaginary axis is $-0.033552884928942 - 0.193436725413768i$, and so we choose $\omega^{(0)} = -0.193436725413768$. We stop the computation once (34) is satisfied, where $\tau = 10^{-10}$ and in order to check the eigenvalues of $H(\alpha^{\text{comp}} - \theta)$ we use $\theta = 10^{-6}$.

Table 4: Results for Example 7.

	Newton method		
i	$\omega^{(i)}$	$\alpha^{(i)}$	$\|g(\omega^{(i)}, \alpha^{(i)})\|$
0	-0.193436725409075	0	-
1	-0.195968902440330	0.001984096006025	5.512849722200156e-03
2	-0.199587128435624	0.001967128516845	3.151734196608377e-03
3	-0.199755034853323	0.001978156503348	1.142239611083313e-04
4	-0.199755999429793	0.001978172281285	7.372625327573070e-07
5	-0.199755999447160	0.001978172281961	1.253426205668754e-11

Table 5: CPU times for Example 7.

Algorithm	Iterations	CPU times	Test steps	Test steps times	CPU times without test steps
Boyd/Balakrishnan	5	406.92 s	–	–	–
He/Watson	1785	249.00 s	1	10.51 s	238.49 s
Newton	5	15.92 s	1	10.44 s	5.48 s

Table 4 shows the convergence and Table 5 the computation times for Example 7. Clearly, in terms of iteration numbers and computing time our algorithm outperforms the method by Boyd & Balakrishnan and He & Watson. The method by Boyd & Balakrishnan does not require a test step, however, Newton's method only requires one test step and is overall much faster than the other two methods.

The final two examples are different from the ones in [7], nevertheless we still compare the performance of our method with the ones in [2, 3] and [7].

Example 8. Consider the matrix `tols340.mtx` from the matrix market library [1]. It arises in aeroelasticity and is a highly nonnormal matrix of dimension 340, with 2196 nonzero entries. The eigenvalues which are all on the left half plane are shown in Figure 4. The eigenvalue closest to the imaginary axis is $-0.156000000000045 + 155.9999219999807i$, so we choose $\omega^{(0)} = 155.9999219999807$. We stop the computation once (34) is satisfied, where $\tau = 10^{-12}$ and in order to check the eigenvalues of $H(\alpha^{\text{comp}} - \theta)$ we use $\theta = 10^{-6}$.

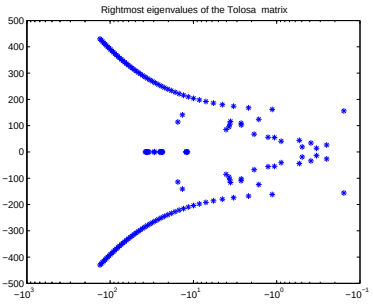


Figure 4: Eigenvalues of the Tolosa matrix in Example 8.

Table 6: Results for Example 8.

	Newton method		
i	$\omega^{(i)}$	$\alpha^{(i)}$	$\|g(\omega^{(i)}, \alpha^{(i)})\|$
0	155.9999219999809	0	-
1	155.9998829972845	0.0019997968879	9.999037055532748e-04
2	155.9998439945399	0.0019997968254	1.602513548769279e-06
3	155.9998439945282	0.0019997968879	3.125412339698120e-11
4	155.9998439945282	0.0019997968879	3.801367746908066e-16

Table 7: CPU times for Example 8.

Algorithm	Iterations	CPU times	Test steps	Test steps times	CPU times without test steps
Boyd/Balakrishnan	4	90.55 s	–	–	–
He/Watson	> 33000	> 2248 s	> 11	> 18 s	> 2230 s
Newton	4	3.62 s	1	1.70 s	1.92 s

Table 6 and 7 show the results for Example 8. The last column in Table 6 shows the quadratic convergence of Newton's method. The He & Watson algorithm is very slow for this example and we also found that it gives unreliable results (for the iteration number given in the table we obtained 0.002347235693050 for the lower bound and 0.002348235693050 for the upper bound). Moreover, the CPU times for our new algorithm are much smaller than the ones for the Boyd & Balakrishnan or the He & Watson method.

Example 9. In this example we consider the matrix `rdb450.mtx` from the matrix market library [1]. Similarly to Example 6 it comes from a reaction-diffusion Brusselator model. It is of dimension 450 and has 2580 nonzero entries. Its eigenvalues, which are all on the left half plane, are plotted in 5. We choose $\omega^{(0)} = 1.610747974050455$ as the eigenvalue closest to the imaginary axis is $-0.247220948810185+1.610747974050455i$. We stop the computation once (34) is satisfied, where $\tau = 10^{-10}$ and in order to check the eigenvalues of $H(\alpha^{\text{comp}} - \theta)$ we use $\theta = 10^{-12}$.

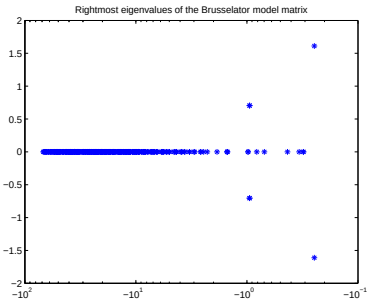


Figure 5: Eigenvalues of the Brusselator model matrix in Example 9.

Table 8: Results for Example 9.

	Newton method		
i	$\omega^{(i)}$	$\alpha^{(i)}$	$\|g(\omega^{(i)}, \alpha^{(i)})\|$
0	1.610747974050403	0	-
1	1.602450963999748	0.084279716628557	4.377644240662334e-02
2	1.593940375957182	0.084228092288813	5.815770068107961e-03
3	1.593892581595785	0.084277383410081	3.800213113694599e-05
4	1.593892567251320	0.084277384643143	9.629615041218073e-09
5	1.593892567251319	0.084277384643143	1.027150511281111e-15

The results for Example 9 are given in Tables 8 and 9. From Table 8 we see that Newton method converges quadratically. Table 9 shows that Newton's method outperforms both the Boyd & Balakrishnan and the He & Watson algorithm with respect to computation time. With respect to iteration numbers

Table 9: CPU times for Example 9.

Algorithm	Iterations	CPU times	Test steps	Test steps times	CPU times without test steps
Boyd/Balakrishnan	4	549.92 s	–	–	–
He/Watson	169	45.57 s	$4-6$	19.41 s	26.16 s
Newton	5	8.58 s	1	3.80 s	4.78 s

the Boyd & Balakrisnan and our method are comparable, however, our iterations are faster, as we essentially only carry out Newton's method for a real function in two real variables.

One iteration of the He & Watson method is often quicker than one step of our method, as the size of the matrices used in their algorithm is smaller. However, we need many fewer iterations as Newton's method in (ω, α) space converges quadratically.

6. Conclusion

We have given an new algorithm for computing the distance of a stable matrix to the nearest unstable one. Numerical results show that this algorithm is competitive with and in almost all cases outperforms earlier algorithms. Our method is very quick, however, is not guaranteed to find the minimum value for α at first (although it does so in all our test problems). For that reason we use it in conjuction with a checking step as in [7].

Acknowledgements

The authors would like to thank two anonymous referees for their valuable comments which improved the paper considerably.

References

[1] B. Boisvert, R. Pozo, K. Remington, B. Miller, and R. Lipman, *Matrix market*. available online at `http://math.nist.gov/MatrixMarket/`.

[2] S. Boyd and V. Balakrishnan, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its* $\mathbf{L}_\infty$*-norm*, Systems Control Lett., 15 (1990), pp. 1–7.

[3] N. A. Bruinsma and M. Steinbuch, *A fast algorithm to compute the* H_∞*-norm of a transfer function matrix*, Systems Control Lett., 14 (1990), pp. 287–293.

[4] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, 3rd ed., 1996.

[6] A. Griewank and G. W. Reddien, *Characterization and computation of generalized turning points*, SIAM J. Numer. Anal., 21 (1984), pp. 176–185.

[7] C. He and G. A. Watson, *An algorithm for computing the distance to instability*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 101–116.

[8] A. Ilchmann and B. Werner, *Characterization of the stability radius via bifurcation techniques*, Appl. Anal., 52 (1994), pp. 231–245.

[9] A. D. Jepson and A. Spence, *The numerical solution of nonlinear equations having several parameters. I. Scalar equations*, SIAM J. Numer. Anal., 22 (1985), pp. 736–759.

[10] H. B. Keller, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, P. H. Rabinowitz, ed., Academic Press, New York, 1977, pp. 359–384.

[11] W. Kerner, *Large-scale complex eigenvalue problems*, J. Comput. Phys., 85 (1989), pp. 1–85.

[12] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, New York, 1992.

[13] A. Spence and C. Poulton, *Photonic band structure calculations using nonlinear eigenvalue techniques*, J. Comput. Phys., 204 (2005), pp. 65–81.

[14] C. F. Van Loan, *How near is a stable matrix to an unstable matrix?*, in Linear algebra and its role in systems theory (Brunswick, Maine, 1984), vol. 47 of Contemp. Math., Amer. Math. Soc., Providence, RI, 1985, pp. 465–478.