A spectral transformation for finding complex eigenvalues of large sparse nonsymmetric matrices

Karl Meerbergen* Alastair Spence†

November 1994

Abstract

Iterative methods like subspace iteration for computing a few important eigenvalues of real large sparse nonsymmetric matrices are often preconditioned with the shift-invert transformation with complex shift. This paper presents a real rational transformation that can be used as an alternative to the shift-invert transform in subspace iteration. This idea is applied to finding rightmost eigenvalues which are of interest in stability studies of nonlinear partial differential equations. Also, an algorithm is given to detect eigenvalues to the right of a given line. The theory and algorithms are illustrated with numerical examples.

1 Introduction

Consider the eigenvalue problem

$$Au - \mu u = 0 \tag{1}$$

where A is a large sparse nonsymmetric real $N \times N$-matrix with eigenvalues $\mu_1, \ldots, \mu_N$. To find the eigenvalue of A closest to a given value, σ say, the most common technique is to apply an iterative method, say subspace iteration, to the shift-invert transformation

$$T_{si} = (A - \sigma I)^{-1}$$

where σ is called the shift. Since the eigenvalues of T_{si} are $\theta_i = 1/(\mu_i - \sigma), i = 1, \ldots, N$, eigenvalues close to σ are mapped to dominant eigenvalues of T_{si}, and eigenvalues of A far from σ are mapped close to zero. Subspace iteration converges

*KU Leuven, Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee-Leuven, Belgium, E-Mail : `Karl.Meerbergen@cs.kuleuven.ac.be`

†University of Bath, School of Mathematical Sciences, Claverton Down, Bath, BA2 7AY, United Kingdom, E-Mail : `A.Spence@maths.bath.ac.uk`

to dominant eigenvalues, and so, the eigenvalues closest to σ will be found, assuming that the subspace of the starting vectors is not deficient in the eigenvector associated with the desired eigenvalue [1, p208–209]. In many applications the shift will be complex and this means working in complex arithmetic, or doubling the size of the system to work in real arithmetic, or using some other strategy (see [11]).

The goal of this paper is to present a real transformation for use when σ is complex which has, roughly speaking, the same desirable mapping properties as T_{si} and which has some advantages in stability applications, but which allows the use of real arithmetic and uses real systems of size N. There are two reasons for the development of real preconditioners. First, one could use existing software that allows only real arithmetic, for example, real linear system solvers. This could be the case when trying to extend the applicability of large software packages which have been developed over many years for specific engineering problems. A second, and perhaps a less important consideration, is that twice the amount of storage is needed for complex operations.

The application we have in mind is the desire to find 'rightmost' eigenvalues of A. This situation arises in the determination of the stability of a steady state solution of the nonlinear system of the form

$$\dot{x} = f(x, \Lambda) \quad f : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N\,,\ x \in \mathbb{R}^N\,,\ \Lambda \in \mathbb{R}\,, \tag{2}$$

where x represents a state variable, and Λ a parameter. A steady state solution (x^*, Λ^*) is stable if the eigenvalues of the Jacobian matrix $A = f_x(x^*, \Lambda^*)$ have negative real parts. Typically f arises from the spatial discretisation of a partial differential equation and so N is large and A is sparse and direct calculation of the eigenvalues of A by the QR-method will not be feasible. The use of T_{si} is common in stability determination calculations (see for example Natarajan [8]), but a good choice of σ is often not known a priori, and a poor choice of σ in T_{si} can lead to the 'rightmost' pair being missed, and in consequence, an incorrect assignment of the stability of a steady solution. An example of the failure of T_{si} in the Olmstead equation is given in Section 3. One advantage of the proposed alternative is that such missassignments are less likely to occur.

In Section 2, the proposed transformation is defined and compared with T_{si}. We discuss some of its properties and at the end of the section, we present a subspace iteration algorithm to find the eigenvalue nearest σ. In Section 3 this algorithm is applied to the problem of finding 'rightmost' eigenvalues. We also give a discussion on the choice of the parameters in this transformation. The section is closed with numerical examples. Section 4 presents an algorithm to check if there is any eigenvalue of A to the right of a given line. In Section 5, we make some comments on the use of the transformation with Arnoldi's method and the nonsymmetric Lanczos method.

2 The spectral transformation

The transformation we shall use as an alternative to T_{si} for σ complex (with nonzero imaginary part) is

$$T_\sigma = I - (A - \alpha I)^{-2}(A - \sigma I)(A - \bar{\sigma} I) . \tag{3}$$

Here $\alpha \in \mathbb{R}$ and is not an eigenvalue of A. Before going into the details of the mapping two general comments are in order.

First, the eigenvalues of T_σ are

$$\theta_i \;=\; 1 - (\mu_i - \sigma)(\mu_i - \bar{\sigma})(\mu_i - \alpha)^{-2} , \quad i = 1, \ldots, N . \tag{4}$$

and so (a) if $\mu_i \simeq \sigma$, then $\theta_i \simeq 1$, (b) if μ_i lies far from α, σ and $\bar{\sigma}$, then $\theta_i \simeq 0$. Thus we see that if α is chosen appropriately, namely, not near any eigenvalue of A, then we would expect T_σ to have a dominant eigenvalue near 1.

Second, T_σ can be written as

$$\begin{aligned} T_\sigma &= I - (A - \alpha I)^{-2}(A - \sigma I)(A - \bar{\sigma} I) \\ &= 2(\mathrm{Re}(\sigma) - \alpha)(A - \alpha I)^{-1} - |\alpha - \sigma|^2 (A - \alpha I)^{-2} , \end{aligned} \tag{5}$$

which ensures only real quantities. Clearly, a matrix vector operation with T_σ invokes the solution of two linear systems with the matrix $(A - \alpha I)$. The main computational effort is in the determination of LU-factors of this matrix.

In the following subsection, we give a theoretical treatment of the mapping properties of T_σ, and in Subsections 2.2 and 2.3, we discuss numerical considerations.

2.1 The mapping properties of T_σ

In order to understand the effect of $T_\sigma(A)$ on the eigenvalues of A we need to understand the mapping properties of $T_\sigma(\mu)$. Write $\sigma = \beta + i\gamma$ and consider

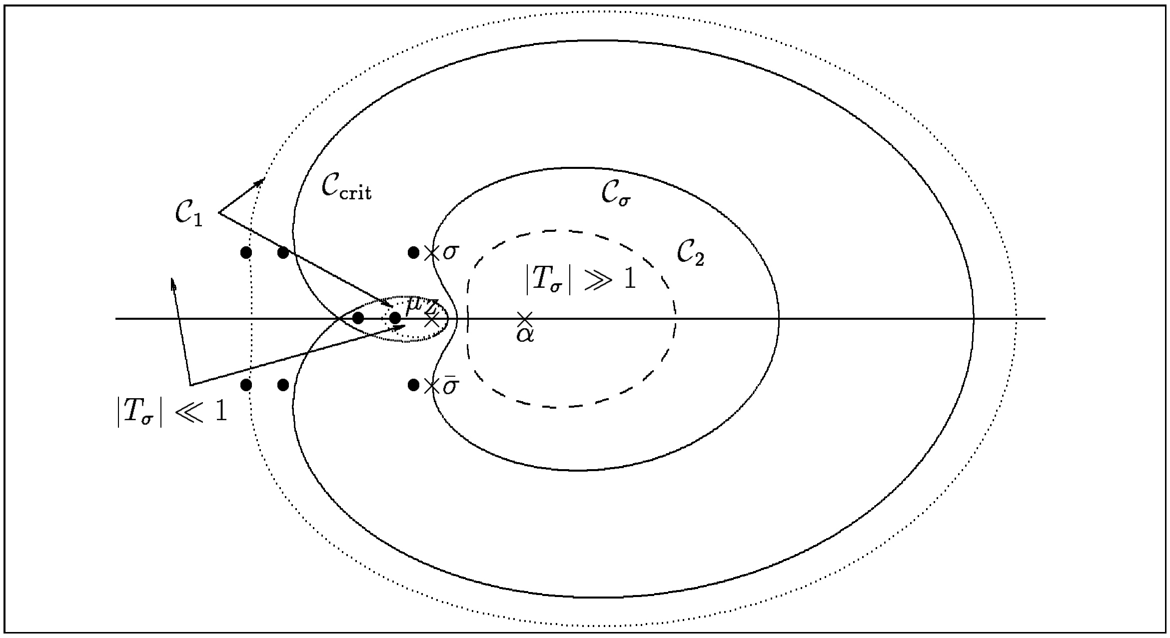
$$\theta = T_\sigma(\mu) = 1 - \frac{(\mu - \gamma)^2 + \beta^2}{(\mu - \alpha)^2} = \frac{2(\gamma - \alpha)\mu + \alpha^2 - \beta^2 - \gamma^2}{(\mu - \alpha)^2} = \frac{D(\mu)}{N(\mu)} . \tag{6}$$

To help understand the effect of $T_\sigma(\mu)$ we determine the contour lines of $|T_\sigma|$ (see Figure 1). These contour lines are also a convenient tool to aid the understanding of the convergence of subspace iteration.

In the following, we suppose that $\alpha > \gamma$, which is always the case in our applications. Denote by $\mathcal{C}(c)$ the contour line $\{\mu : |T_\sigma(\mu)| = c\}$. This curve is described by the equation $|T_\sigma(x + iy)|^2 - c^2 = 0$ with $\mu = x + iy \in \mathcal{C}(c)$. Note that $|T_\sigma(x + iy)|^2 = |D(x + iy)|^2 / |N(x + iy)|^2$. Since $D(x + iy) \neq 0$ for $x + iy \neq \alpha$, $\mathcal{C}(c)$ is also described by $F(x, y; c) = |N|^2 - c^2|D|^2 \equiv 0$. A full expression for $F(x, y; c)$ derived directly from (6) is

$$\begin{aligned} F(x, y; c) &= (2(\gamma - \alpha)x + \alpha^2 - \beta^2 - \gamma^2)^2 + 4(\gamma - \alpha)^2 y^2 \\ &\quad - c^2((x - \alpha)^2 + y^2)^2 \qquad (7) \\ &= (2(\alpha - \gamma)(x - \alpha) + (\alpha - \gamma)^2 + \beta^2)^2 + 4(\gamma - \alpha)^2 y^2 \\ &\quad - c^2((x - \alpha)^2 + y^2)^2 \qquad (8) \\ &\equiv 0 . \end{aligned}$$

Figure 1: Contour lines of $|T_\sigma|$



The transformation is rather complicated, and we therefore give only properties of $\mathcal{C}(c)$ that we use later. Figure 1 shows several contour lines. The curve $\mathcal{C}_{\rm crit}$ has three real points. The curves $\mathcal{C}_\sigma$ and $\mathcal{C}_2 = \mathcal{C}(c_2)$ have two real points. The contour $\mathcal{C}_1 = \mathcal{C}(c_1)$ consists of 2 separate curves (as dots) each with two real points, so that $\mathcal{C}_1$ has 4 real points. In the following theorem an expression for the real points of $\mathcal{C}(c)$ is derived.

Theorem 1 *Define $c_{\rm crit}$ by*

$$c_{\rm crit} = (\alpha-\gamma)^2/((\alpha-\gamma)^2+\beta^2)\ .$$

Then the contour line $\mathcal{C}(c)$ crosses the real axis at 2 points if $c > c_{\rm crit}$, at 3 points if $c = c_{\rm crit}$ and at 4 points otherwise. Moreover the real points are

$$x = \alpha + \frac{\alpha-\gamma \pm \sqrt{(1+p)(\alpha-\gamma)^2 + p\beta^2}}{p} \tag{9}$$

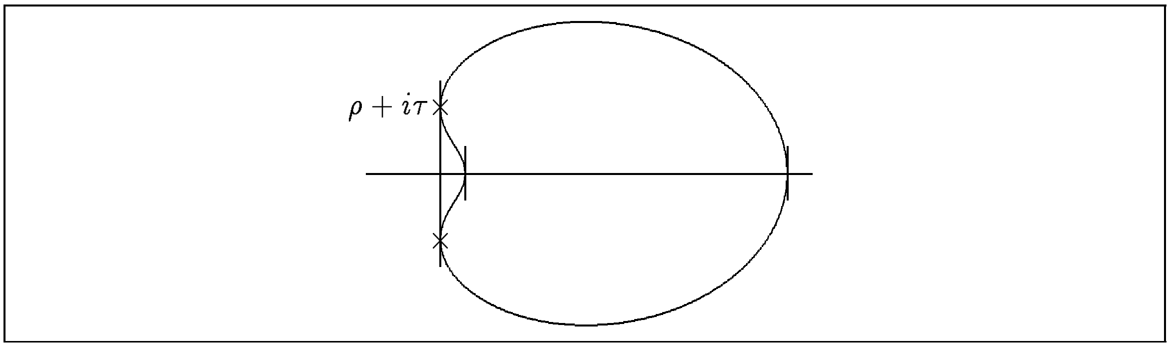
for $p = \pm c$ if $c \le c_{\rm crit}$ and for $p = c$ if $c > c_{\rm crit}$.

The proof is rather technical and the details are found in Appendix A.

Hence $\mathcal{C}(c)$ has 2 real points if $c > c_{\rm crit}$ and 3 real points if $c = c_{\rm crit}$. For $c < c_{\rm crit}$ there are 2 separate contours with each two real points, one lying inside the small loop of $\mathcal{C}_{\rm crit}$ and one enclosing $\mathcal{C}_{\rm crit}$ (see Figure 1).

In our analysis, attention is paid to the contour line $\mathcal{C}_\sigma = \mathcal{C}(1)$ that passes through σ and $\bar\sigma$. Figure 1 shows $\mathcal{C}_{\rm crit} = \mathcal{C}(c_{\rm crit})$, $\mathcal{C}_\sigma$, $\mathcal{C}_2 = \mathcal{C}(2)$ and $\mathcal{C}_1 = \mathcal{C}(0.45)$ for $c_{\rm crit} = 0.5$.

Figure 2: Tangent lines to $\mathcal{C}_\sigma$



It can be derived from (6) that T_σ has one (real) zero

$$\mu_Z = \frac{\alpha+\gamma}{2} - \frac{\beta^2}{2(\alpha-\gamma)} \ . \tag{10}$$

This zero always lies in the small loop of $\mathcal{C}_{\text{crit}}$.

$\mathcal{C}_\sigma$ has 2 real points and always encloses α. It is also tangent to either two or three vertical lines in the complex plane. Two of them are the vertical lines going through the real points of $\mathcal{C}_\sigma$. Depending on α and σ, there can be a third vertical line tangent to $\mathcal{C}_\sigma$ in a complex conjugate pair of points $\rho \pm i\tau$ (see Figure 2). We call the points of $\mathcal{C}_\sigma$ at which the contour line has vertical tangent lines, *tangent points*. The following theorem presents an expression for these points that we will use later.

Theorem 2 *For $\alpha > \gamma$ a complex pair of tangent points $\rho \pm i\tau$ of $\mathcal{C}_\sigma$ exists when*

$$\frac{\beta^2}{(\alpha-\gamma)^2} \le \sqrt{8} + 1 \ . \tag{11}$$

The tangent points $\rho \pm i\tau$ satisfy the equations

$$\rho = \alpha - \frac{\beta^2 + (\alpha-\gamma)^2}{4(\alpha-\gamma)} - \frac{(\alpha-\gamma)^3}{\beta^2+(\alpha-\gamma)^2} \tag{12a}$$

$$\tau^2 = 2(\alpha-\gamma)^2 - (\rho-\alpha)^2. \tag{12b}$$

The proof is rather technical and the details are given in Appendix B.

2.2 Practical mapping considerations

As discussed at the beginning of this section, if σ is given, we hope to choose α, such that the eigenvalues of A nearest σ correspond to dominant eigenvalues of T_σ. To explain this more clearly, we consider a model example. Let A have rightmost eigenvalues μ_1, $\mu_2 = \bar{\mu}_1$, μ_3 with $\text{Re}(\mu_3) < \text{Re}(\mu_1)$ (see the dots in Figure 1). Let σ be an approximation to μ_1. Then maps θ_1 and θ_2 given by (4) lie close to 1. Suppose also that α is chosen so that the other eigenvalues of A lie inside the small loop of

Table 1: Eigenvalues and maps for Example 1

i	μ_i	$\theta_i(T_{si})$	$\lvert\theta_i(T_{si})\rvert/\lvert\theta_1(T_{si})\rvert$	$\theta_i(T_\sigma)$	$\lvert\theta_i(T_\sigma)\rvert/\lvert\theta_1(T_\sigma)\rvert$
1	$20i$	1.4286	1	$1.0362 + 0.0007i$	1
2	$-20i$	$0.0004 + 0.0250i$	0.02	$1.0362 - 0.0007i$	1
3	-10	$-0.0191 + 0.0411i$	0.03	0.4333	0.4
4	-20	$-0.0250 + 0.0259i$	0.03	0.4998	0.5
5	-30	$-0.0233 + 0.0159i$	0.02	0.4822	0.5
6	-40	$-0.0202 + 0.0103i$	0.02	0.4470	0.4
7	$-1 + 20i$	-3.3333	2.3	$0.9852 + 0.0001i$	0.95
8	$-1 - 20i$	$-0.0002 + 0.0250i$	0.02	$0.9852 - 0.0001i$	0.95
9	$-2 + 20i$	-0.7692	0.5	$0.9391 + 0.0019i$	0.91
10	$-2 - 20i$	$-0.0008 + 0.0250i$	0.02	$0.9391 - 0.0019i$	0.91

$\mathcal{C}_{\text{crit}}$ or to the left of $\mathcal{C}_{\text{crit}}$ (see Figure 1), then the other eigenvalues are mapped close to zero. Hence, subspace iteration will converge to θ_1 and θ_2.

This technique works for σ with arbitrary imaginary part. However, there is a disadvantage when many eigenvalues of A lie rather close to μ_1 or μ_2. In this case their transformations lie close to θ_1 and θ_2, which results in slow convergence of subspace iteration. In our experience T_{si} is less sensitive to the clustering of eigenvalues around σ. The following example helps illustrate these points.

Example 1 *A small test example.*

Consider a 10×10-matrix with eigenvalues given in Table 1. Let $\sigma = -0.7 + 20i$ be a rough approximation to μ_1. Clearly, $\mu_3, \ldots, \mu_6$ lie far from σ, while $\mu_7, \ldots, \mu_{10}$ lie clustered near σ or $\bar{\sigma}$. For $\alpha = 19.3$, the maps of $T_{si} = (A - \sigma I)^{-1}$ and T_σ are shown in Table 1. For both transformations $\theta_1 = T_\sigma(\mu_1)$ belongs to the dominant part of the spectrum. Since $|\theta_i(T_{si})|/|\theta_1(T_{si})| \leq 0.5$, $i \neq 1$ and $i \neq 7$, the separation between θ_1 and most eigenvalues of T_{si} is good, so fast convergence with T_{si} is expected. For T_σ, $|\theta_i|/|\theta_1| \simeq 0.5$ for $i = 3, \ldots, 6$; this provides reasonable separation between θ_1 and the maps of eigenvalues far from σ, which is nice to provide fast convergence to μ_1. However, $|\theta_i|/|\theta_1| \simeq 1$ for $i = 7, \ldots, 10$, and this fact can slow down the convergence to μ_1 significantly.

In spite of the negative features when σ is near a cluster of eigenvalues of A, T_σ has some significant advantages to recommend its use when clustering does not occur. In particular :

- If σ is a poor choice of shift, so that the rightmost eigenvalues lie some-way to the right of σ, then they can be easily missed by T_{si} but they can be found by T_σ because, roughly speaking, eigenvalues of A to the right of σ are mapped to the dominant eigenvalues of T_σ. Clearly, in Example 1, T_{si} maps μ_7 to the dominant eigenvalue of T_{si} and so this eigenvalue converges faster than μ_1. T_σ, on the other hand, maps μ_1 to the dominant eigenvalue of T_σ. In this sense, T_σ is more forgiving of a poor choice of σ. An example of the superiority of T_σ over T_{si} occurs in the Olmstead model in Section 3.

- T_σ is a real transformation, so only real linear systems of size N have to be solved instead of complex ones in T_{si}, which facilitates the use of existing software that allows only real arithmetic. The use of real systems only can be an important gain in storage and computation time.

2.3 The application of subspace iteration on T_σ

In this section, a subspace iteration algorithm with T_σ is proposed. The basic algorithm is given, assuming that α and σ are known. The speed of convergence towards an eigenvalue of A is analysed.

Each loop of Algorithm 2.1 consists of the power method with T_σ on a block of m linearly independent vectors simultaneously. This is used to obtain a subspace V_k rich in the desired eigendirections. After these iterations, we use the Galerkin projection with the matrix A onto the subspace spanned by the vectors V_k. One strategy might be to form $H_k = V_k^T T_\sigma V_k$ and try to find μ by calculating the inverse of $T_\sigma(\mu)$. However, since a double eigenvalue at 1 is expected, inverting $T_\sigma(\mu)$ is not possible. Rather, we compute the eigenvalues of $V_k^T A V_k$ as is proposed by Saad [12, Algorithm 7.4, p.237].

Algorithm 2.1: *Subspace_iteration with* T_σ

```
procedure {V_k, μ̂_i, i = 1,...,m}=Subspace_Iteration(α, σ, V_0)
    Given α and σ.
    Given V_0 = [v_1,...,v_m] a set of m orthonormal vectors.
    for j = 1 to k do
        Form V_j ← T_σ V_{j-1}.
        Orthonormalise V_j.
    end for
    Form H_k = V_k^T A V_k.
    Compute the eigenvalues μ̂, i = 1,...,m of H_k by the QR-method.
    Order the eigenvalues by decreasing real part.
end procedure
```

The subspace span(V_k) in Algorithm 2.1 converges to the dominant subspace of dimension m of T_σ for increasing k. The following theorem shows an expression for the rate of convergence of a simple eigenvalue of A, i.e. an expression for $|\mu_i - \hat{\mu}_i|$ where $\hat{\mu}_i$ is an approximation of μ_i computed by Algorithm 2.1.

Theorem 3 *Suppose that the eigenvalues of A are ordered by decreasing $|T_\sigma(\mu_i)|$, $i = 1, \ldots, N$, and*

$$|T_\sigma(\mu_1)| \geq |T_\sigma(\mu_2)| \geq \ldots \geq |T_\sigma(\mu_m)| > |T_\sigma(\mu_{m+1})| \geq \ldots \geq |T_\sigma(\mu_N)| \tag{13}$$

and that $S_0 = \mathrm{span}(V_0)$ *is not deficient in the dominant subspace* $M = \mathrm{span}(u_1, \ldots, u_m)$ *of* T_σ. *If* μ_i, $i = 1, \ldots, m$ *are simple eigenvalues of* A, *then for each* $i = 1, \ldots, m$ *there is an approximate eigenvalue* $\hat{\mu}_i$ *generated by Algorithm 2.1 that converges to* μ_i *with*

$$|\mu_i - \hat{\mu}_i| = O\left(\left|\frac{T_\sigma(\mu_{m+1})}{T_\sigma(\mu_i)}\right|^k\right). \tag{14}$$

Proof. The proof relies on the fact that the eigenvectors of A are eigenvectors of T_σ. One can show that

$$\|(I - V_k V_k^H)u_i\| = O\left(\left|\frac{T_\sigma(\mu_{m+1})}{T_\sigma(\mu_i)}\right|^k\right) \tag{15}$$

where $V_k V_k^H u_i$ is the orthogonal projection of u_i on $\mathrm{span}(V_k)$. The proof is a modification of the proof of Lemma 6.2.1 in [1, p.253-254] that proposes a bound on $\|(I - V_k V_k^H)u_i\|$ for subspace iteration applied to A^k and Lemma 7.6.1 in [1, p.309-310] that proposes a similar bound for subspace iteration with Chebyshev polynomials. Lemma 6.2.4 in [1, p.255] shows that $|\mu_i - \hat{\mu}_i| = O(\|(I - V_k V_k^H)u_i\|)$. From this lemma and (15), (14) immediately follows. □

From this theorem it can be seen that the eigenvalues μ_i of A with largest map θ_i converge with a rate of convergence given by $\rho_i = (|T_\sigma(\mu_i)|/|T_\sigma(\mu_{m+1})|)^k$. Theorem 3 justifies the use of contour lines to analyse the convergence of eigenvalues of A in Algorithm 2.1.

In a practical procedure, we use Algorithm 2.2 consisting of several iterations of Algorithm 2.1.

Algorithm 2.2: *Practical algorithm*

procedure Iterative Procedure
  Given α and σ.
  Let $V = [v_1, \ldots, v_m]$ be a random set of m orthonormal vectors.
  for $j = 1$ **to** convergence of $\hat{\mu}_1$ **do**
    Do subspace iteration : $V = \text{Subspace\_Iteration}(\alpha, \sigma, V)$
  end for
end procedure

3 Computing rightmost eigenvalues of A

The algorithm described in last section will be used to compute rightmost eigenvalues close to σ and $\bar{\sigma}$. In this section, we describe two strategies to set the parameters α and σ, supposing that a small number of approximate rightmost eigenvalues are known. The shift σ is chosen as the rightmost approximate eigenvalue $\hat{\mu}_1$ (and $\bar{\sigma} = \hat{\mu}_2$). Only the free parameter α needs to be chosen, and to do so it is important to understand well the mapping properties of T_σ. It is difficult to manipulate the formulae of contour lines of T_σ, which makes it hard to obtain precise numerical information from the theoretical analysis. First, we propose two strategies for the choice of α, that will be compared by some examples at the end of this section.

- *Strategy 1.*

  Theorem 4 *For the choice of* $\alpha = \mathrm{Re}(\sigma) + \mathrm{Im}(\sigma)$, *the line* $\{\mu : \mathrm{Re}(\mu) = \mathrm{Re}(\sigma)\}$ *is tangent to* C_σ *at* σ *and* $\bar{\sigma}$. *This situation is shown in Figure 1.*

  Proof. Let $\sigma = \gamma + i\beta$. The real part of the tangent points $\rho \pm i\tau$ is given by (12a). If $\alpha = \gamma + \beta$ then $\beta = \alpha - \gamma$. Hence ρ in (12a) becomes $\rho = \gamma$, and from (12b), it follows that $\tau = \alpha - \gamma = \beta$. □

  This parameter setting is simple and can be carried out without knowledge of any eigenvalues of A. The property that σ and $\bar{\sigma}$ are tangent points, provides that eigenvalues to the right of σ and $\bar{\sigma}$ will converge faster than other eigenvalues.

- *Strategy 2.* Suppose that $\sigma = \hat{\mu}_1$ and $\bar{\sigma} = \hat{\mu}_2$. Choose α such that $|T_\sigma(\hat{\mu}_3)|$ is minimal for $\alpha \geq \gamma$. Due to the complicated nature of the transform, a closed formula for α is not available. Note that eigenvalues to the right of σ and $\bar{\sigma}$ do not necessarily converge faster than the other eigenvalues since σ and $\bar{\sigma}$ are no longer tangent points.

  Lemma 1 *Given* $\sigma = \gamma + i\beta$ *and* $\zeta = x + iy \in \mathbb{C}$. *Then*

  $$|T_\sigma(\zeta)| \textit{ is minimal }, \tag{16}$$

  for α *satisfying the third degree equation in* $(\alpha - \gamma)$

  $$\begin{aligned}&[(x-\gamma)^2 - y^2 + \beta^2]\ (\alpha-\gamma)^3\\ &-12(x-\gamma)[(x-\gamma)^2 + y^2 + \beta^2]\ (\alpha-\gamma)^2\\ &+[2((x-\gamma)^2 + y^2)^2 + \beta^2((x-\gamma)^2 - y^2 - \beta^2)]\ (\alpha-\gamma)\\ &+(x-\gamma)\beta^2[(x-\gamma)^2 + y^2 + \beta^2] = 0\ .\end{aligned} \tag{17}$$

  Proof. From (7), we find that $|T_\sigma(x+iy)|^2 = c^2$ is

  $$c^2 = \frac{(2(\gamma-\alpha)x + \alpha^2 - \beta^2 - \gamma^2)^2 + 4(\gamma-\alpha)^2 y^2}{((x-\alpha)^2 + y^2)^2}\ .$$

  For $\zeta = x + iy$, α makes $|T_\sigma(\zeta)|$ minimal when $\mathrm{d}c^2/\mathrm{d}\alpha = 0$. To simplify the calculations, we introduce the variables $x' = x - \gamma$ and $\alpha' = \alpha - \gamma$. The solution of (16) can also be derived from $\mathrm{d}c'^2/\mathrm{d}\alpha' = 0$ with $\alpha = \alpha' + \gamma$ and

  $$c'^2 = \frac{(-2\alpha' x' + \alpha'^2 - \beta^2)^2 + 4\alpha'^2 y^2}{((x'-\alpha')^2 + y^2)^2}\ .$$

The manipulations to obtain (17) are quite extensive, and we do not present them here. □

The roots of (17) are easily computed from the 3×3 companion matrix according to the polynomial. Note that (17) has 1 or 3 real roots, denoted by ζ_i, $i = 1, \nu$, with $\nu = 1$ or 3. As a result ζ_i, $i = 1, \ldots, \nu$ make $|T_\sigma(\hat{\mu}_3)|$ minimal or maximal. Since we wish $\alpha \geq \gamma$, we add the constraint $\alpha \geq \gamma$ to (16). Therefore α is selected from the values $\{\zeta_1, \ldots, \zeta_\nu, \gamma\}$ with smallest $|T_\sigma(\hat{\mu}_3)|$.

We illustrate both strategies with two examples.

Example 2 *Brusselator model ($N = 2048$) [5]*

The equations

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \frac{D_u}{L^2}\left[\frac{\partial^2 u}{\partial X^2} + \frac{\partial^2 u}{\partial Y^2}\right] - (B+1)u + u^2 v + C \\
\frac{\partial v}{\partial t} &= \frac{D_v}{L^2}\left[\frac{\partial^2 v}{\partial X^2} + \frac{\partial^2 v}{\partial Y^2}\right] - u^2 v + Bu
\end{aligned}$$

for u and $v \in (0,1) \times (0,1)$ with homogeneous Dirichlet boundary conditions form a 2D reaction-diffusion model where u and v represent the concentrations of two reactants. The equations are discretised with central differences with grid-sizes $h_u = h_v = 1/(n+1)$ with $n = (N/2)^{1/2}$. For $x^T = [u_{1,1}, v_{1,1}, u_{1,2}, v_{1,2}, \ldots, u_{n,n}, v_{n,n}]$, the discretised equations can be written as $\dot{x} = f(x)$. We wish to compute the rightmost eigenvalues μ_1 and μ_2 of the Jacobian matrix $A = \partial f / \partial x$ for $B = 5.45$, $C = 2$, $D_u = 0.004$, $D_v = 0.008$ and $L = 1$ for $u(X, Y) = A$ and $v(X, Y) = B/A$.

Suppose that the eigenvalues $\xi_{1,2} = -0.2455 \pm 1.6123i$ and $\xi_3 = \xi_4 = -0.3069$, of the Jacobian for $L = 0.5$, are known. In a continuation algorithm, we might step from $L = 0.5$ to $L = 1$ and would hope that ξ_1 and ξ_2 were good approximations to μ_1 and μ_2. Therefore in Algorithm 2.1 set $\sigma = \xi_1$. The other parameters of the algorithm are $m = 10$ and $k = 5$. The experiments were run on an IBM RS6000 workstation. To measure the accuracy of the eigenpair $(\hat{\mu}_i, \hat{u}_i)$ we use the residual norm

$$\|A\hat{u}_i - \hat{\mu}_i \hat{u}_i\|_2 / \|\hat{u}_i\|_2 \ . \tag{18}$$

The residual norms, the execution times and number of iterations of both strategies for choosing α are shown in Table 2. We also considered complex subspace iteration using T_{si} with complex shift. Note that the storage of 5 complex vectors is as expensive as storing 10 real vectors, therefore, we use $m = 5$ in complex subspace iteration with T_{si}. Figure 3 shows the rightmost eigenvalues of A, plotted as bullets, and a few contour lines of T_σ. The eigenvalues ξ_i that determine the choice of α are plotted as circles. Note that both strategies produce roughly the same α. This can be explained as follows. With Strategy 1, $T_\sigma(\xi_3) = 0.0707$ is small. This is not surprising, because $T_\sigma(\mathrm{Re}(\sigma)) = 0$ and $\xi_3 \simeq \mathrm{Re}(\sigma)$. With Strategy 2, $T_\sigma(\hat{\mu}_3) = 0$. In fact, both strategies map ξ_3 to a small value, so one can expect Strategy 1 and Strategy 2 to produce roughly the same α.

Table 2: Results for the Brusselator

strategy	T_σ strat. 1	T_σ strat. 2	T_{si}
iteration	residuals		
1	$7.3\,10^{1}$	$3.2\,10^{1}$	$1.8\,10^{-2}$
2	$1.9\,10^{0}$	$1.8\,10^{0}$	$4.7\,10^{-4}$
3	$1.4\,10^{0}$	$1.3\,10^{0}$	$2.3\,10^{-5}$
4	$8.4\,10^{-2}$	$4.9\,10^{-2}$	$1.8\,10^{-5}$
5	$1.0\,10^{-2}$	$1.7\,10^{-2}$	
6	$3.1\,10^{-3}$	$3.8\,10^{-3}$	
7	$7.5\,10^{-4}$	$8.3\,10^{-4}$	
8	$2.0\,10^{-4}$	$1.7\,10^{-4}$	
9	$4.1\,10^{-5}$	$7.5\,10^{-5}$	
10	$2.9\,10^{-5}$	$3.2\,10^{-5}$	
11	$6.6\,10^{-6}$	$4.9\,10^{-6}$	
σ	$-0.2455 + 1.6123i$		
α	1.3668	1.30657	
time (sec.)	309	309	140
converged eigenvalues	$0.106654 \pm 1.901178i$ $-0.070329 \pm 1.762906i$		$0.106654 + 1.901178$

Figure 3: Spectrum and contour lines for the Brusselator and Olmstead models

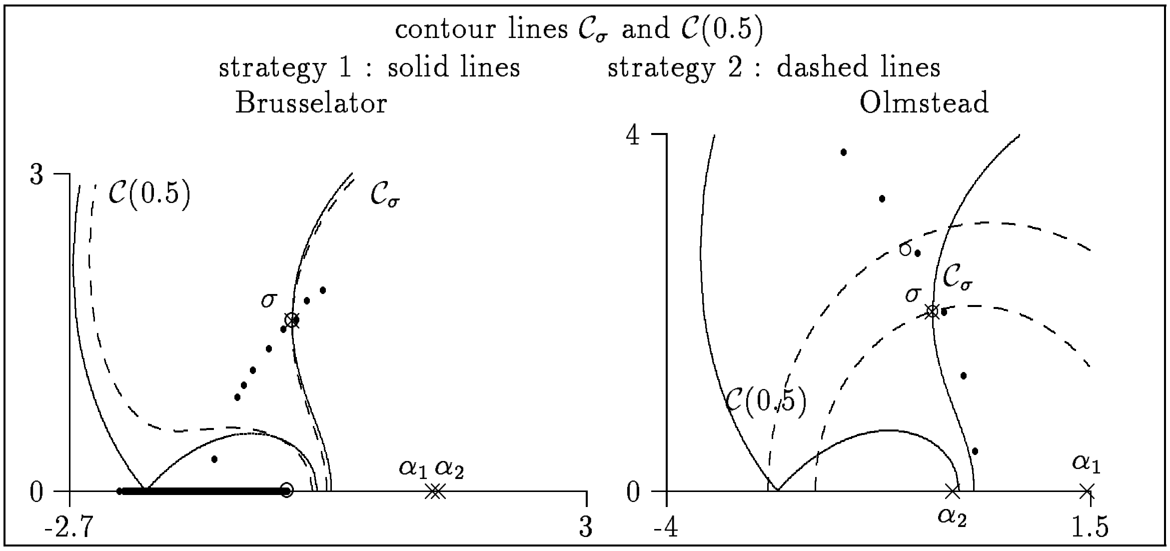


Table 3: Results for the Olmstead model

strategy	T_σ strat. 1	T_σ strat. 2	T_{si}
iteration		residuals	
1	$2.2\,10^{0}$	$1.7\,10^{1}$	$3.2\,10^{-4}$
2	$7.0\,10^{-2}$	$8.2\,10^{-5}$	$4.8\,10^{-5}$
3	$9.8\,10^{-3}$	$7.5\,10^{-12}$	$2.2\,10^{-7}$
4	$5.2\,10^{-4}$		
5	$3.1\,10^{-5}$		
6	$2.7\,10^{-6}$		
σ		$-0.549994 + 2.01185i$	
α	1.461856	-0.290487	
time (sec.)	10.5	3.6	3.0
converged eigenvalues	0.0 ± 0.447212 -0.149997 ± 1.295174	$0.0 \pm 0.447212i$ -0.149997 ± 1.295174	$-0.149997 + 1.295174i$ $-0.4000 + 2.000i$

Example 3 *Olmstead model* ($N = 1000$) *[10].*

The model represents the flow of a layer of viscoelastic fluid heated from below. The equations are

$$\begin{cases} \dfrac{\partial u}{\partial t} &= (1-C)\dfrac{\partial^2 v}{\partial X^2} + C\dfrac{\partial^2 u}{\partial X^2} + Ru - u^3 \\ B\dfrac{\partial v}{\partial t} &= u - v \end{cases}$$

with boundary conditions $u(0) = u(1) = 0$ and $v(0) = v(1) = 0$. u represents the speed of the fluid and v is related to viscoelastic forces. The equation was discretised with central differences with gridsize $h = 1/(N/2)$. After discretisation the equation may be written as $\dot{x} = f(x)$ with $x^T = [u_1, v_1, u_2, v_2, \ldots, u_{N/2}, v_{N/2}]$. The size of the Jacobian matrix $A = \partial f / \partial x$ is 1000. We consider the Jacobian for the parameter values $B = 2$, $C = 0.1$ and $R = 0.6$ for the trivial steady state $[u, v] = 0$.

It is known that the rightmost eigenvalues for the Jacobian at $R = 0.3$ are $\xi_{1,2} = -0.549994 \pm 2.01185i$ and $\xi_{3,4} = -0.900614 \pm 2.65528i$. The rightmost eigenvalues for $R = 0.6$ are wanted, and we use as shift the known rightmost eigenvalue at $R = 0.3$, i.e. $\sigma = \xi_1$. We used Algorithm 2.1 with $m = 10$ and $k = 5$. The results of both strategies for choosing α are shown in Table 3. We also used subspace iteration using T_{si} with $m = 5$. The rightmost eigenvalues of the Jacobian and a few contour lines of T_σ are shown in Figure 3. As is seen T_{si} does not locate the rightmost eigenvalue (which is precisely on the boundary of stability for the application).

In Section 2.2, we mentioned that failure to compute the rightmost eigenvalues is less likely with T_σ than with T_{si}, because there is an eigenvalue near α that will be magnified by T_σ. In Table 3, one observes that subspace iteration with T_{si} converges to $\mu_3 = -0.15 + 1.30i$ instead of $\mu_1 = 0.45i$. This can be explained as follows. The rate of convergence of an eigenvalue μ_i of A in subspace iteration with

T_{si} is proportional to $|\theta_i|$ with $\theta_i = T_{si}(\mu_i)$. The 4 eigenvalues of A nearest σ and their maps are $(\mu_5, |\theta_5|) = (-0.450 + 2.01i, 6.7)$, $(\mu_7, |\theta_7|) = (-0.75 + 2.67i, 1.5)$, $(\mu_3, |\theta_3|) = (-0.15 + 1.30i, 1.2)$ and $(\mu_1, |\theta_1|) = (0.45i, 0.6)$. It is clear that μ_3, μ_5 and μ_7 converge (much) faster in subspace iteration than μ_1 and this explains why μ_1 is missed. Note that in Figure 3 the contour lines of T_σ are all to the left of μ_1, μ_3 and μ_5, which favours convergence to μ_1.

4 Validation of a candidate rightmost eigenvalue

Subspace iteration with T_{si} fails in Example 3 because the shift σ lies too far from μ_1. In general, subspace iteration with T_{si} and a badly chosen shift will sometimes not converge to the rightmost eigenvalue [2, 7, 6]. In general, it is hard to construct preconditioners that guarantee the computation of the rightmost eigenvalue(s). One situation that might arise is that $\hat{\mu}_c$ is a candidate rightmost eigenvalue, say formed by any iterative eigenvalue solver, e.g. Arnoldi with shift-invert, or may indeed be an arbitrary guess. The problem discussed in this section is the validation of $\hat{\mu}_c$, i.e. checking if there are any eigenvalues of A to the right of $\hat{\mu}_c$.

Before presenting the validation procedure we introduce the concept of domain of confidence of subspace iteration (see also Meerbergen, Spence and Roose [7]). Recall that the rate of convergence of an eigenvalue μ_i of A in Algorithm 2.2 is given by $\rho_i = |\theta_i|/|\theta_{m+1}|$ where the $\theta_i = T_\sigma(\mu_i)$, $i = 1, \ldots, N$ are ordered by decreasing modulus. Since $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_m$, it is expected that μ_1 converges first, then μ_2 and so on. This is usually the case when k is large enough. More generally, if an eigenvalue μ_i has been computed to an acceptable accuracy, we may expect that all eigenvalues μ_j with $|T_\sigma(\mu_j)| > |T_\sigma(\mu_i)|$ have been computed as well, which is confirmed in practice. We therefore define the *domain of confidence* of μ_i by $\mathcal{D}(\mu_i) = \{\mu : |T_\sigma(\mu)| \geq |T_\sigma(\mu_i)|\}$, with the following property : if μ_i has converged, then all eigenvalues in $\mathcal{D}(\mu_i)$ are expected to have converged as well. Note that the boundaries of the domains of confidence of eigenvalues of A coincide with contour lines of $|T_\sigma|$.

Meerbergen, Spence and Roose [7] developed a two step algorithm based on subspace iteration with the Cayley transform and shift-invert with complex shift. In the first step, the real Cayley transform searches eigenvalues close to the real axis, while in the second step, shift-invert looks for eigenvalues with large imaginary part. If there are no eigenvalues with large imaginary parts, the second step takes a long time to converge to an eigenvalue close to the real axis and one has to stop the algorithm before convergence of any eigenvalue of A. In this paper a one step technique with T_σ is proposed that does not suffer from this problem. The problem of validation is equivalent to checking if there is an eigenvalue of A in the region $\mathcal{R} = \{\mu : \mathrm{Re}(\mu) \geq \mathrm{Re}(\hat{\mu}_c)\}$. In a practical situation the region $\mathcal{R}$ can be reduced by some bound on the eigenvalues of A, e.g. by Gerschgorin's bounds. Sometimes $\mathcal{R}$ can be determined based on sharp mathematical or physical bounds on the eigenvalues as is the case for sectorial operators. An operator is called sectorial [4, p.45] when there exist constants $a \in \mathbb{R}$ and $\phi \in (0, \pi/2)$ such that the eigenvalues of the operator are contained in

$$\mathcal{S} = \{\mu : 0 \leq |\arg(a - \mu)| \leq \phi\} \; . \tag{19}$$

The theorem in [4, p.32] gives useful bounds on the spectrum for a class of operators. If a and ϕ are known, the validation can be carried out in the region $\mathcal{R} = \mathcal{S} \cap \mathcal{G}$ with $\mathcal{S}$ the sector given by (19) and $\mathcal{G}$ given by Gerschgorin's theorem. In the following, we suppose that $\mathcal{R}$ is reduced to a disk segment with centre on the real axis, see Figure 4. In our example, Gerschgorin's theorem is used to bound the eigenvalues. For validation we use Algorithm 2.2 with α and σ chosen like in Figure 4 : denote by μ_R the upper left 'corner' of $\mathcal{R}$; set $\sigma = \mu_R$ and determine α such that $\mathrm{Re}(\sigma)$ lies on $\mathcal{C}_\sigma$. Since the rate of convergence of μ_i only depends on the modulus of θ_i, eigenvalues close to σ, $\bar{\sigma}$ or $\mathrm{Re}(\sigma)$ converge with the same rate. This guarantees convergence towards eigenvalues with small and large imaginary part at the same time. Following lemma shows that this is the case for $\alpha = \mathrm{Re}(\sigma) + \mathrm{Im}(\sigma)/\sqrt{2}$.

Lemma 2 (Strategy 3). *For* $\alpha = \mathrm{Re}(\sigma) + \mathrm{Im}(\sigma)/\sqrt{2}$, $\mathrm{Re}(\sigma)$ *is the leftmost real point and is a tangent point of* $\mathcal{C}_\sigma$*, see Figure 4.*

Proof. Let $\sigma = \gamma + i\beta$. We will prove that if γ is the left real point of $\mathcal{C}_\sigma$, that then $\alpha = \gamma + \beta/\sqrt{2}$. The real points of $\mathcal{C}_\sigma$ are given by (9) for $p = 1$ (not for $p = -1$, because $c = 1 > c_{\mathrm{crit}}$, see Theorem 2). Suppose that the left real point of $\mathcal{C}_\sigma$ is $\gamma = \mathrm{Re}(\sigma)$. Thus for the left real point,

$$\gamma = \alpha + (\alpha - \gamma - \sqrt{2(\alpha - \gamma)^2 + \beta^2}) \; ,$$

which is equivalent to

$$2(\alpha - \gamma) = \sqrt{2(\alpha - \gamma)^2 + \beta^2} \; .$$

The solution of this equation is $\alpha - \gamma = \pm\beta/\sqrt{2}$ or $\alpha = \gamma \pm \beta/\sqrt{2}$. This includes $\alpha = \mathrm{Re}(\sigma) + \mathrm{Im}(\sigma)/\sqrt{2}$. The fact that γ is a real point of $\mathcal{C}_\sigma$, implies that it is a tangent point (see the proof of Theorem 2). □

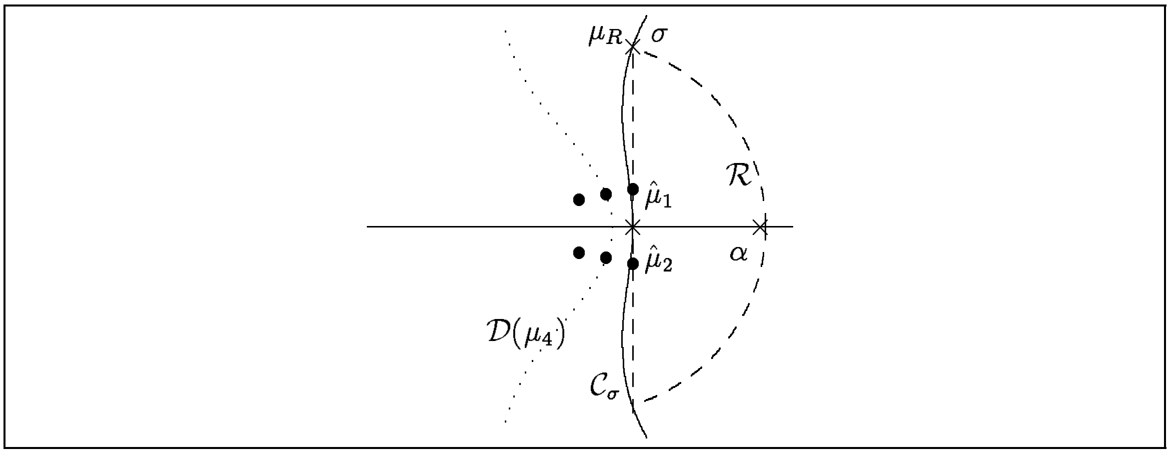
When validating $\hat{\mu}_c$, the following situations may occur.

1. There is an eigenvalue μ_l to the right of $\hat{\mu}_c$ in $\mathcal{R}$. This eigenvalue is mapped to the dominant eigenvalue of T_σ and will be found by Algorithm 2.2.

2. There is no eigenvalue to the right of $\hat{\mu}_c = \mu_1$ (Figure 4). Of course, subspace iteration still converges to the dominant eigenvalues of T_σ. Suppose that $\theta_1, \ldots, \theta_4$ are the dominant eigenvalues of T_σ and that μ_1, ..., μ_4 have converged. The domain of confidence of μ_4 is shown in Figure 4 and encloses $\mathcal{C}_\sigma$ and $\mathcal{R}$ totally. By its definition $\mathcal{D} = \mathcal{D}(\mu_4)$ only contains eigenvalues that are computed, namely $\mu_1, \ldots, \mu_4$. As a conclusion, no other eigenvalues in $\mathcal{R}$ than μ_1 and μ_2 are expected.

In the second situation, we hope that $\mathcal{D}$ encloses $\mathcal{R}$ to ensure that all eigenvalues in $\mathcal{R}$ have been computed. Of course, this depends on the geometry of both $\mathcal{D}$ and $\mathcal{R}$. In the following lemma, we prove that $\mathcal{C}_\sigma$ encloses $\mathcal{R}$ if $\mathcal{R}$ is a segment with centre to the left of α. So, in the case of Figure 4, $\mathcal{C}_\sigma \subset \mathcal{D}$ ensures $\mathcal{R} \subset \mathcal{D}$.

Figure 4: Validation strategy with T_σ



Lemma 3 *Let $\sigma = \gamma + i\beta$. If $\alpha = \gamma + \beta/\sqrt{2}$, the contour line $\mathcal{C}_\sigma$ encloses the segment $\mathcal{R} = \{\mu : \mathrm{Re}(\mu) \geq \gamma$ and $|\mu - \zeta| \leq ((\gamma - \zeta)^2 + \beta^2)^{1/2}\}$ with centre $\zeta \leq \alpha$ and radius $((\gamma - \zeta)^2 + \beta^2)^{1/2}$.*

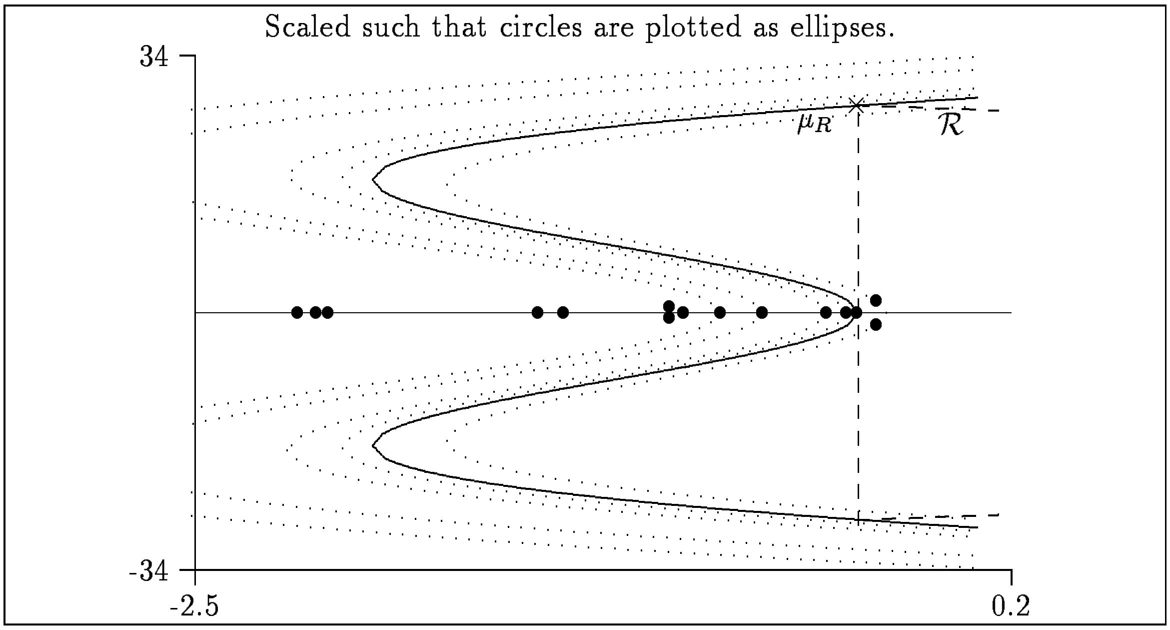
A proof for this lemma is presented in Appendix C.

Note that the choice of α and σ guarantees convergence towards the eigenvalues in $\mathcal{C}_\sigma$, but does not ensure fast convergence. In general, the rate of convergence becomes smaller when $\mathrm{Im}(\sigma)$ becomes larger. We make this point clearer for the case that many eigenvalues of A lie close to $\mathrm{Re}(\sigma)$. Let $\mu_1 = \mathrm{Re}(\sigma)$ then $\theta_1 = -1$, and let μ_2 lie between μ_1 and the zero of T_σ, μ_Z. From (10) we derive that $\mu_Z = \mathrm{Re}(\sigma) - \mathrm{Im}(\sigma)/\sqrt{8}$ for $\alpha = \mathrm{Re}(\sigma) + \mathrm{Im}(\sigma)/\sqrt{2}$. So, if $\mathrm{Im}(\sigma)$ becomes larger, μ_Z moves to the left. Hence μ_2 lies relatively closer to μ_1 and this fact 'pulls' θ_2 towards θ_1. This diminishes the separation between θ_1 and θ_2 and decreases the convergence rate.

Example 4 *Brusselator model (see Example 2 for N =1250).*

We wish to compute the rightmost eigenvalue of the Jacobian A for $L = 0.5$. The rightmost eigenvalues of A are $\mu_{1,2} = -0.2455 \pm 1.6123i$ and $\mu_3 = \mu_4 = -0.312609$, shown in Figure 5. The rightmost eigenvalues for $L = 0.3$ are $\xi_1 = \xi_2 = -0.4890$ and $\xi_{3,4} = -1.8820 \pm 0.2795i$. When we use this information to compute μ_1 at $L = 0.5$, namely, we make the choice $\sigma = \xi_1 = -0.4890$ in subspace iteration with T_{si} (see the examples in Section 3), we compute μ_3 and μ_4 rather than μ_1. So, in fact, μ_1 is missed. To be more sure about this result we can validate with $\hat{\mu}_c = \mu_3 = -0.312609$. From Gerschgorin's bounds, one can derive the segment $\mathcal{R}$ with $\mu_R = -0.312609 + 27.414022$. The validation procedure uses T_σ with $\sigma = \mu_R$ and $\alpha = \mathrm{Re}(\sigma) + \mathrm{Im}(\sigma)/\sqrt{2} = 19.072032$. Figure 5 shows $\mathcal{C}_\sigma$, $\mathcal{R}$ and the domains of confidence of the rightmost eigenvalues, i.e. the contour lines of T_σ going through the eigenvalues. The rates of convergence are respectively $\rho_{1,2} = 1.15$, $\rho_{3,4} = 1.13$, $\rho_5 = 1.12$, $\rho_{6,7} = 1.10$, $\rho_{8,9} = 1.06$ and $\rho_{10} = 1.03$. Subspace iteration with $m = 10$ and $k = 10$ computed $\hat{\mu}_{1,2} = -0.248383 \pm 1.609693i$ with residual norm $3.6\,10^{-4}$, given by (18), in 7 iterations and 129 seconds on the IBM RS6000. The domain of

Figure 5: Spectrum and contour lines of the Brusselator model



confidence does not enclose $\mathcal{R}$ totally, so eventually, one could restart the procedure with $\hat{\mu}_c = \hat{\mu}_1$.

5 Extension to Krylov methods

One could also try to apply Arnoldi's method or Lanczos' method to T_σ. However, as we now explain, there are some difficulties. An important property of these methods is that they are translation invariant, i.e. if the application of Arnoldi's method on T_σ produces m approximate eigenvalues $\hat{\theta}_i, i = 1, \ldots, m$, then Arnoldi's method with $\delta_1 T_\sigma - \delta_2 I$ gives approximate eigenvalues $\delta_1 \hat{\theta}_i - \delta_2, i = 1, \ldots, m$. This translation invariance means that the contour lines of $|T_\sigma|$ are not helpful in understanding the performance of Arnoldi's method on T_σ. Rather, the convergence of Krylov methods like Arnoldi has to be explained in terms of separation of eigenvalues of $T_\sigma(A)$ rather than contour lines of $|T_\sigma(\mu)|$. It is more difficult to analyse separation of eigenvalues for Arnoldi than analysing contour lines for subspace iteration, so it is hard to pick α and σ to ensure convergence to the wanted eigenvalue [7, 6]. On the other hand, if Arnoldi's method works, it often works faster than subspace iteration, see [9] for the symmetric eigenvalue problem. The main argument to support the use of subspace iteration for nonsymmetric problems is one of reliability rather than speed, and that is why it is the favoured choice in many applications.

6 Conclusions

In this paper the real transformation

$$T_\sigma = (A - \alpha I)^{-2}(A - \sigma I)(A - \bar{\sigma} I)$$

is proposed for use with subspace iteration. This transformation is useful for various reasons :

1. T_σ is a good alternative to the complex shift-invert transformation $T_{si} = (A - \sigma I)^{-1}$ if factorising the complex matrix $(A - \sigma I)$ is not feasible due to software or hardware limitations (see Introduction). If the matrix can be factorised easily, the use of T_{si} is preferred for reasons of speed. If robustness is more important than speed, then one should consider the use of T_σ. Because of the complicated nature of the transform a good choice for α is not straightforward, but in this paper we propose three strategies. Strategy 1 is interesting for finding eigenvalues to the right of a given value σ. Strategy 3 can be used for finding eigenvalues to the right of the line segment going through σ and $\bar{\sigma}$. The effect of Strategy 2 depends on the particular case, but in a sense it can be seen as an optimal choice for finding the eigenvalues closest to σ.

2. The validation of a candidate rightmost eigenvalue $\hat{\mu}_c$ is an important and challenging problem. In various problems T_σ can be used for validation purposes as explained in Section 4. Convergence towards the rightmost eigenvalue is very often guaranteed. The rate of convergence on the other hand may be very small. Note that here the choice of the region $\mathcal{R}$ that encloses all eigenvalues to the right of $\hat{\mu}_c$, is crucial. If $\mathcal{R}$ is very large, validation can be very slow.

   One can think of variations on this validation technique by using other strategies for choosing α or to combine it with the Cayley transform [3, 7].

In both applications, subspace iteration with T_σ is relatively slow compared to Arnoldi with $(A - \alpha I)^{-1}$ or $(A - \sigma I)^{-1}$, but it is often preferred for reasons of robustness.

7 Acknowledgement

The authors are grateful to the British Council and the Belgian National Fund for Scientific Research (NFWO) for supporting the collaboration between the Katholieke Universiteit Leuven and the University of Bath.

References

[1] F. Chatelin. *Eigenvalues of matrices.* Pure and applied mathematics. John Wiley and Sons, 1993.

[2] T.J. Garratt. *The Numerical Detection of Hopf Bifurcations in Large Systems Arising in Fluid Mechanics.* PhD thesis, University of Bath, UK, 1991.

[3] T.J. Garratt, G. Moore, and A. Spence. Two methods for the numerical detection of Hopf bifurcations. In R. Seydel, F.W. Schneider, and H. Troger, editors, *Bifurcation and Chaos : Analysis, Algorithms, Applications*, pages 119–123. Birkhäuser, 1991.

[4] P. Grindrod. *Patterns and waves : the theory and applications of reaction-diffusion equations.* Oxford Applied Mathematics and Computing Science Series. Clarendon Press, Oxford, 1991.

[5] B.D. Hassard, N. Kazarinoff, and Y.H. Wan. *Theory and Applications of Hopf Bifurcation.* Cambridge University Press, Cambridge, 1981.

[6] K. Meerbergen. Preconditioners for computing right-most eigenvalues. Report TW206, Department of Computer Science, KU Leuven, 1994. Presented in the Course on Recent Advances in Iterative methods for solving Algebraic Systems and Eigenvalue Problems, Leuven, March 1994.

[7] K. Meerbergen, A. Spence, and D. Roose. Shift-invert and Cayley transforms for detection of rightmost eigenvalues of nonsymmetric matrices. *BIT*, 34:409–423, 1994.

[8] R. Natarajan. An Arnoldi-based iterative scheme for nonsymmetric matrix pencils arising in finite element stability problems. *J. Comput. Phys.*, 100:128–142, 1992.

[9] B. Nour-Omid, B.N. Parlett, and R.L. Taylor. Lanczos versus subspace iteration for solution of eigenvalue problems. *Int. J. Numer. Meth. Eng.*, 19:859–871, 1983.

[10] W.E. Olmstead, W.E. Davis, S.H. Rosenblat, and W.L. Kath. Bifurcation with memory. *SIAM J. Appl. Math.*, 46:171–188, 1986.

[11] B.N. Parlett and Y. Saad. Complex Shift and Invert Strategies for Real Matrices. *Linear Algebra Appl.*, 88/89:575–595, 1987.

[12] Y. Saad. *Numerical methods for large eigenvalue problems.* Algorithms and Architectures for Advanced Scientific Computing. Manchester University Press, Manchester, 1992.

A Proof for Theorem 1

Theorem 5 *Define c_{crit} by*

$$c_{\text{crit}} = (\alpha-\gamma)^2/((\alpha-\gamma)^2+\beta^2)\ .$$

Then the contour line $\mathcal{C}(c)$ crosses the real axis at 2 points if $c > c_{\text{crit}}$, at 3 points if $c = c_{\text{crit}}$ and at 4 points otherwise. Moreover the real points are

$$x = \alpha + \frac{\alpha-\gamma \pm \sqrt{(1+p)(\alpha-\gamma)^2 + p\beta^2}}{p} \tag{20}$$

for $p = \pm c$ if $c \leq c_{\text{crit}}$ and for $p = c$ if $c > c_{\text{crit}}$.

Proof. First, we derive (20). The real points of $\mathcal{C}(c)$ are recovered from (7) for $y = 0$:

$$(2(\gamma-\alpha)x + \alpha^2 - \beta^2 - \gamma^2)^2 - c^2(x-\alpha)^4 = 0\ .$$

Splitting this expression in two factors, gives

$$(2(\gamma-\alpha)x + \alpha^2 - \beta^2 - \gamma^2) + p(x-\alpha)^2 = 0 \tag{21}$$

with $p = \pm c$. Note that (21) can also be derived from (6) for $\mu = x \in I\!R$ and $T_\sigma(\mu) = p$.

Equation (21) can be written as

$$p(x-\alpha)^2 + 2(\gamma-\alpha)(x-\alpha) - \beta^2 - (\alpha-\gamma)^2 = 0\ .$$

The solution of this quadratic equation in $(x-\alpha)$ is (20). These solutions are real iff $\Delta = (1+p)(\alpha-\gamma)^2 + p\beta^2 \geq 0$, i.e.

$$p \geq -\frac{(\alpha-\gamma)^2}{(\alpha-\gamma)^2+\beta^2} = -c_{\text{crit}}\ .$$

Note that $0 \leq c_{\text{crit}} \leq 1$. We always have that $c \geq -c_{\text{crit}}$ because $c \geq 0$. Hence for $p = c$, (21) always has a solution. For $p = -c$, (21) has a solution when $p = -c \geq -c_{\text{crit}}$ or $c \leq c_{\text{crit}}$. If $c = c_{\text{crit}}$, then $\Delta = 0$ for $p = -c$ and then we have a double real point, that is counted only once. This shows our statement on the number of real points. $\square$

B Proof for Theorem 2

Theorem 6 *For $\alpha > \gamma$ a complex pair of tangent points $\rho \pm i\tau$ of $\mathcal{C}_\sigma$ exists when*

$$\frac{\beta^2}{(\alpha-\gamma)^2} \leq \sqrt{8} + 1\ . \tag{22}$$

The tangent points $\rho \pm i\tau$ satisfy the equations

$$\rho = \alpha - \frac{\beta^2 + (\alpha-\gamma)^2}{4(\alpha-\gamma)} - \frac{(\alpha-\gamma)^3}{\beta^2+(\alpha-\gamma)^2} \tag{23a}$$

$$\tau^2 = 2(\alpha-\gamma)^2 - (\rho-\alpha)^2. \tag{23b}$$

Proof. The points of $\mathcal{C}_\sigma$ in the neighbourhood of $\rho+i\tau$ can be expressed by $r(s)+is$ where s is a continuous parameter. For a tangent point $\rho+i\tau$, $r'(\tau)=0$. Since $F(r,s;1)\equiv 0$,

$$\frac{\mathrm{d}F}{\mathrm{d}s}\equiv 0 \qquad \text{so} \qquad F'_r(r,s;1)r'(s)+F'_s(r,s;1)\equiv 0\ . \tag{24}$$

For the tangent point $\rho+i\tau$, $r'=0$, $r=\rho$ and $s=\tau$. Hence (24) is reduced to $F'_s(\rho,\tau,1)=0$ or

$$8(\gamma-\alpha)^2\tau-4((\rho-\alpha)^2+\tau^2)\tau=0\ .$$

This equation has two solutions, namely

$$1.\qquad \tau=0 \tag{25a}$$

$$2.\qquad 2(\gamma-\alpha)^2-((\rho-\alpha)^2+\tau^2)=0\,. \tag{25b}$$

Since we are not interested in real tangent points in this proof, (25a) can be dropped. The remaining part (25b) is equivalent to (23b). Plugging (23b) in $F(\rho,\tau;1)=0$ with F given by (7), results in

$$(2(\gamma-\alpha)(\rho-\alpha)-\beta^2-(\alpha-\gamma)^2)^2+8(\gamma-\alpha)^4-4(\gamma-\alpha)^2(\rho-\alpha)^2-(2(\alpha-\gamma)^2)^2=0\ .$$

By expanding the first quadratic form,

$$(\beta^2+(\alpha-\gamma)^2)^2+4(\gamma-\alpha)^4-4(\gamma-\alpha)(\beta^2+(\alpha-\gamma)^2)(\rho-\alpha)=0\ ,$$

(23a) follows. When using (23b), we implicitly suppose that $\tau^2\geq 0$. This constraint adds an extra condition to ρ and τ, namely, from (23b),

$$2(\alpha-\gamma)^2-(\rho-\alpha)^2\geq 0\ .$$

This constraint holds when

$$-\sqrt{2}(\alpha-\gamma)\leq\rho-\alpha\leq\sqrt{2}(\alpha-\gamma)\ . \tag{26}$$

So (23a) and (23b) give rise to a complex tangent point if (26) holds. Since in (23a) $\rho<\alpha$ when $\alpha>\gamma$, the right inequality of (26) always holds. The left inequality of (26) for ρ given by (23a) becomes

$$-\sqrt{2}(\alpha-\gamma)\leq\frac{\beta^2+(\alpha-\gamma)^2}{4(\alpha-\gamma)}-\frac{(\alpha-\gamma)^3}{\beta^2+(\alpha-\gamma)^2}\ ,$$

which is equivalent to

$$-4\sqrt{2}(\alpha-\gamma)^2(\beta^2+(\alpha-\gamma)^2)\leq-(\beta^2+(\alpha-\gamma)^2)^2-4(\alpha-\gamma)^4$$

or after reduction

$$\beta^4+2(1-\sqrt{8})(\alpha-\gamma)^2\beta^2+(5-2\sqrt{8})(\alpha-\gamma)^4\leq 0\ .$$

This is a polynomial inequality of degree 2 in $\beta^2/(\alpha-\gamma)^2$. The inequality holds when $\beta^2/(\alpha-\gamma)^2$ lies between $(\sqrt{8}-3)=-0.1716$ and $(\sqrt{8}+1)=3.8284$. Since $(\sqrt{8}-3)<0$ and $\beta^2/(\alpha-\gamma)^2>0$, the condition (22) is sufficient. $\square$

C Proof of Lemma 3

Lemma 4 *Let $\sigma = \gamma + i\beta$. If $\alpha = \gamma + \beta/\sqrt{2}$, the contour line $\mathcal{C}_\sigma$ encloses the segment $\mathcal{R} = \{\mu : \mathrm{Re}(\mu) \geq \gamma$ and $|\mu - \zeta| \leq ((\gamma - \zeta)^2 + \beta^2)^{1/2}\}$ with centre $\zeta \leq \alpha$ and radius $((\gamma-\zeta)^2 + \beta^2)^{1/2}$.*

Proof. First note that it is sufficient to prove the lemma for $\zeta = \alpha$ since $\mathcal{R}$ becomes smaller for ζ moving to the left (see Figure 6). The boundary of $\mathcal{R}$ consists of a straight line segment $\mathcal{L}$ and a circle segment $\mathcal{E}$. In the first part of the proof, we show that $\mathcal{C}_\sigma$ encloses the verticle line $\mathcal{L}$. In the second part, we show that $\mathcal{E}$ is enclosed by the contour.

1. We first prove that the cross points of $\mathcal{C}_\sigma$ and the vertical line $\mathcal{L} = \{\mu : \mathrm{Re}(\mu) = \gamma\}$ are σ and $\bar{\sigma}$. From (8) and for $x = \gamma$ it follows that

   $$((\alpha-\gamma)^2 - \beta^2)^2 + 4(\alpha-\gamma)^2 y^2 - ((\alpha-\gamma)^2 + y^2)^2 = 0 \ ,$$

   and since $(\alpha - \gamma)^2 = \beta^2/2$,

   $$\beta^4/4 + 2\beta^2 y^2 - \beta^4/4 - \beta^2 y^2 - y^4 \ .$$

   This is finally reduced to

   $$\beta^2 y^2 - y^4 = 0 \ ,$$

   with solutions $y = 0$ and $y = \pm\beta$. The solution $y = \pm\beta$ corresponds to σ and $\bar{\sigma}$.

   $\mathcal{C}_\sigma$ has a tangent point at γ (because γ is a real point of $\mathcal{C}_\sigma$) and tangent points at $(\rho, \pm\tau)$ given by (23a) and (23b) (because (22) is satisfied). From (23a) and $\beta = \sqrt{2}(\alpha - \gamma)$, we see that $\rho = \gamma - \frac{1}{12}(\alpha - \gamma) < \gamma$. Since $\mathcal{C}_\sigma$ has only σ, $\mathrm{Re}(\sigma)$ and $\bar{\sigma}$ common with $\mathcal{L}$, and its left tangent points lie to the left of $\mathcal{L}$, $\mathcal{C}_\sigma$ encloses $\mathcal{L}$. This property is independent of ζ.

2. In this part, we prove that $\mathcal{C}_\sigma$ encloses $\mathcal{E}$. First, we show that σ and $\bar{\sigma}$ are the only cross points of $\mathcal{C}_\sigma$ and $\mathcal{E}$. In the second part, we prove that $\mathcal{R}$ lies inside $\mathcal{C}_\sigma$.

   Let $x + iy \in \mathcal{C}_\sigma$. Let in (8) $\alpha = \gamma + \beta/\sqrt{2}$ and $c = 1$, then

   $$(\sqrt{2}\beta(x-\alpha) + \frac{3}{2}\beta^2)^2 + 2\beta^2 y^2 - ((x-\alpha)^2 + y^2)^2 = 0 \ . \qquad (27)$$

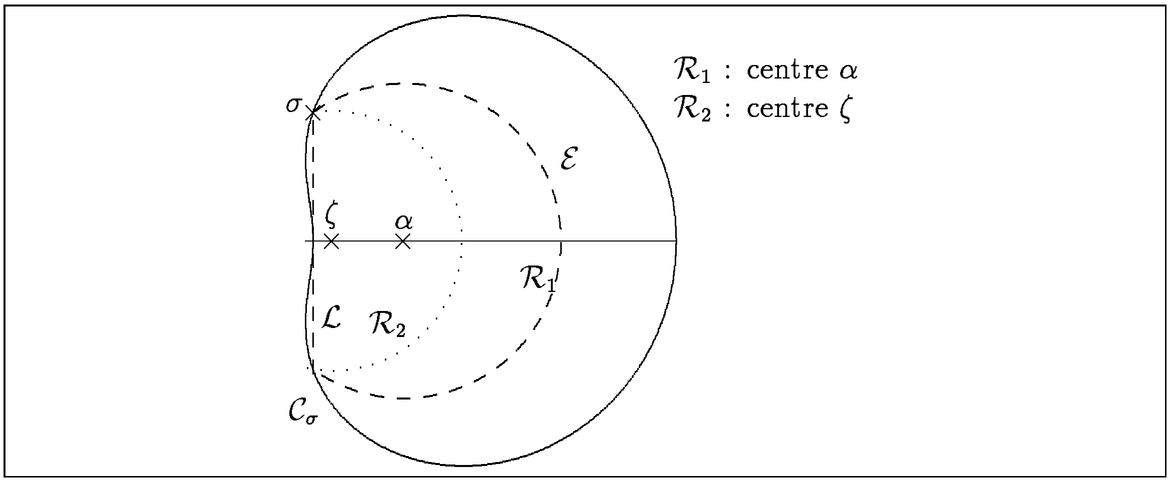
   The equation of the circle segment $\mathcal{E}$ is

   $$(x-\alpha)^2 + y^2 = (\gamma - \alpha)^2 + \beta^2, \quad \gamma \leq x \leq \alpha + ((\alpha-\gamma)^2 + \beta^2)^{1/2}$$

   or

   $$y^2 = \frac{3}{2}\beta^2 - (x-\alpha)^2, \quad \gamma \leq x \leq \alpha + \sqrt{3/2}\beta \ . \qquad (28)$$

Figure 6: $\mathcal{C}_\sigma$ enclosing $\mathcal{R}$



Substituting y^2 from (28) in (27) gives

$$\left(\sqrt{2}\beta(x-\alpha)+\frac{3}{2}\beta^2\right)^2+2\beta^2\left(\frac{3}{2}\beta^2-(x-\alpha)^2\right)-\left(\frac{3}{2}\beta^2\right)^2=0$$

or

$$3\sqrt{2}\beta^3(x-\alpha)+3\beta^4=0\ .$$

The solution of this equation is $x=\alpha-\beta/\sqrt{2}=\gamma$. By replacing x by γ in (28) we see that the points σ and $\bar{\sigma}$ are the only cross points.

Thus we conclude that $\mathcal{E}$ lies either inside $\mathcal{C}_\sigma$ or outside $\mathcal{C}_\sigma$.

Now it is sufficient to prove that one point of $\mathcal{E}$ is enclosed by $\mathcal{C}_\sigma$ to prove that $\mathcal{R}$ lies totally inside $\mathcal{C}_\sigma$. The real points of $\mathcal{C}_\sigma$ given by (20) are γ and $\alpha+3\beta/\sqrt{2}$. Hence all real points between γ and $\alpha+3\beta/\sqrt{2}$ are contained by $\mathcal{C}_\sigma$. So, also the real point $\alpha+\sqrt{3/2}\beta$ of $\mathcal{E}$ lies in $\mathcal{C}_\sigma$. As a result, $\mathcal{R}$ is enclosed by $\mathcal{C}_\sigma$.

$\square$