

LYAPUNOV INVERSE ITERATION FOR COMPUTING A FEW RIGHTMOST EIGENVALUES OF LARGE GENERALIZED EIGENVALUE PROBLEMS*

HOWARD C. ELMAN[†] AND MINGHAO WU[‡]

Abstract. In linear stability analysis of a large-scale dynamical system, we need to compute the rightmost eigenvalue(s) for a series of large generalized eigenvalue problems. Existing iterative eigenvalue solvers are not robust when no estimate of the rightmost eigenvalue(s) is available. In this study, we show that such an estimate can be obtained from Lyapunov inverse iteration applied to a special eigenvalue problem of Lyapunov structure. We also show that Lyapunov inverse iteration will always converge in only two steps if the Lyapunov equation in the first step is solved accurately enough. Furthermore, we generalize the analysis to a deflated version of this Lyapunov eigenvalue problem and propose an algorithm that computes a few rightmost eigenvalues for the eigenvalue problems arising from linear stability analysis. Numerical experiments demonstrate the robustness of the algorithm.

1. Introduction. This paper introduces an efficient algorithm for computing a few rightmost eigenvalues of generalized eigenvalue problems. We are concerned with problems of the form

$$\mathcal{J}(\alpha)x = \mu \mathbf{M}x \tag{1.1}$$

arising from linear stability analysis (see [11]) of the dynamical system

$$\mathbf{M}u_t = f(u, \alpha). \tag{1.2}$$

$\mathbf{M} \in \mathbb{R}^{n \times n}$ is called the mass matrix, and the parameter-dependent matrix $\mathcal{J}(\alpha) \in \mathbb{R}^{n \times n}$ is the Jacobian matrix $\frac{\partial f}{\partial u}(\bar{u}(\alpha), \alpha) = \frac{\partial f}{\partial u}(\alpha)$, where $\bar{u}(\alpha)$ is the steady-state solution to (1.2) at α , *i.e.*, $f(\bar{u}, \alpha) = 0$. Let the solution path be the following set: $\mathcal{S} = \{(\bar{u}, \alpha) | f(\bar{u}, \alpha) = 0\}$. We seek the critical point (\bar{u}_c, α_c) associated with transition to instability on \mathcal{S} . While the method developed in this study works for any dynamical system of the form (1.2), our primary interest is the ones arising from spatial discretization of 2- or 3-dimensional time-dependent partial differential equations (PDEs). Therefore, we assume n to be large and $\mathcal{J}(\alpha), \mathbf{M}$ to be sparse throughout this paper.

The conventional method of locating the critical parameter α_c is to monitor the rightmost eigenvalue(s) of (1.1) while marching along \mathcal{S} using numerical continuation (see [11]). In the stable regime of \mathcal{S} , the eigenvalues μ of (1.1) all lie to the left of the imaginary axis. As (\bar{u}, α) approaches the critical point, the rightmost eigenvalue of (1.1) moves towards the imaginary axis; at (\bar{u}_c, α_c) , the rightmost eigenvalue of (1.1) has real part zero, and finally, in the unstable regime, some eigenvalues of (1.1) have positive real parts. The continuation usually starts from a point (\bar{u}_0, α_0) in the stable regime of \mathcal{S} and the critical point is detected when the real part of the rightmost eigenvalue of (1.1) becomes nonnegative. Consequently, robustness and efficiency of the eigenvalue solver for the rightmost eigenvalue(s) of (1.1) are crucial for the performance of this method. Direct eigenvalue

*This work was supported in part by the U. S. Department of Energy under grant DEFG0204ER25619 and the U. S. National Science Foundation under grant DMS1115317.

[†]Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (elman@cs.umd.edu).

[‡]Applied Mathematics & Statistics, and Scientific Computation Program, Department of Mathematics, University of Maryland, College Park, MD 20742 (mwu@math.umd.edu).

solvers such as the QR and QZ algorithms (see [20]) compute all the eigenvalues of (1.1), but they are too expensive for large n . Existing iterative eigenvalue solvers [20] are able to compute a small set ($k \ll n$) of eigenvalues of (1.1) near a given shift (or target) $\sigma \in \mathbb{C}$ efficiently. For example, they work well when k eigenvalues of (1.1) with smallest modulus are sought, in which case $\sigma = 0$. One issue with such methods is that there is no robust way to determine a good choice of σ when we have no idea where the target eigenvalues may be. In the computation of the rightmost eigenvalue(s), the most commonly used heuristic choice for σ is zero, *i.e.*, we compute k eigenvalues of (1.1) with smallest modulus and hope that the rightmost one is one of them. When the rightmost eigenvalue is real, zero is a good choice. However, such an approach is not robust when the rightmost eigenvalues consist of a complex conjugate pair: the rightmost pair can be far away from zero and it is not clear how big k should be to ensure that they are found. Such examples can be found in the numerical experiments of this study.

Meerbergen and Spence [15] proposed the *Lyapunov inverse iteration* method, which estimates the critical parameter α_c without computing the rightmost eigenvalues of (1.1). Assume (\bar{u}_0, α_0) is in the stable regime of \mathcal{S} and is also in the neighborhood of the critical point (\bar{u}_c, α_c) . Let $\lambda_c = \alpha_c - \alpha_0$ and $\mathbf{A} = \mathcal{J}(\alpha_0)$. Then the Jacobian matrix $\mathcal{J}(\alpha_c)$ at the critical point can be approximated by $\mathbf{A} + \lambda_c \mathbf{B}$ where $\mathbf{B} = \frac{d\mathcal{J}}{d\alpha}(\alpha_0)$. It is shown in [15] that λ_c is the eigenvalue with smallest modulus of the eigenvalue problem

$$\mathbf{A}\mathbf{Z}\mathbf{M}^T + \mathbf{M}\mathbf{Z}\mathbf{A}^T + \lambda(\mathbf{B}\mathbf{Z}\mathbf{M}^T + \mathbf{M}\mathbf{Z}\mathbf{B}^T) = 0 \quad (1.3)$$

of Lyapunov structure and that λ_c can be computed by a matrix version of inverse iteration. Estimates of the rightmost eigenvalue(s) of (1.1) at α_c can be obtained as by-products. Elman *et al.* [8] refined the Lyapunov inverse iteration proposed in [15] to make it more robust and efficient and examined its performance on challenging test problems arising from fluid dynamics. Various implementation issues were discussed, including the use of inexact inner iterations, the impact of the choice of iterative method used to solve the Lyapunov equations, and the effect of eigenvalue distribution on performance. Numerical experiments demonstrated the robustness of their algorithm.

The method proposed in [8, 15], although it allows us to estimate the critical value of the parameter without computing the rightmost eigenvalue(s) of (1.1), only works in the neighborhood of the critical point (\bar{u}_c, α_c) . In [8], for instance, the critical parameter value α_c of all numerical examples is known *a priori*, so that we can pick a point (\bar{u}_0, α_0) close to (\bar{u}_c, α_c) and apply Lyapunov inverse iteration with confidence. In reality, α_c is unknown and we start from a point (\bar{u}_0, α_0) in the stable regime of \mathcal{S} that may be distant from the critical point. In this scenario, the method of [8, 15] cannot be used to estimate α_c , since $\mathcal{J}(\alpha_c)$ cannot be approximated by $\mathbf{A} + \lambda_c \mathbf{B}$. However, quantitative information about how far away (\bar{u}_0, α_0) is from (\bar{u}_c, α_c) can still be obtained by estimating the distance between the rightmost eigenvalue of (1.1) at α_0 and the imaginary axis: if the rightmost eigenvalue is far away from the imaginary axis, then it is reasonable to assume that (\bar{u}_0, α_0) is far away from the critical point as well, and therefore we should march along \mathcal{S} using numerical continuation until we are close enough to (\bar{u}_c, α_c) ; otherwise, we can assume that (\bar{u}_0, α_0) is already in the neighborhood of the critical point and the method of [8, 15] can be applied to estimate α_c .

The goal of this paper is to develop a robust method to compute a few rightmost eigenvalues of (1.1) in the stable regime of \mathcal{S} . The plan of the paper is as follows. In section 2, we show that the distance between the imaginary axis and the rightmost eigenvalue of (1.1) is the eigenvalue with smallest modulus of an eigenvalue problem similar in structure to (1.3). As a result, this eigenvalue

can be computed efficiently by Lyapunov inverse iteration. In section 3, we present numerical results for several examples arising from fluid dynamics, and provide an analysis of the fast convergence of Lyapunov inverse iteration observed in our experiments. Based on the analysis, we also propose a modified algorithm that guarantees convergence in only two iterations. In section 4, we show that the analysis in sections 2 and 3 can be generalized to a deflated version of the Lyapunov eigenvalue problem, which leads to an algorithm for computing k ($1 \leq k \ll n$) rightmost eigenvalues of (1.1). Details of the implementation of this algorithm are discussed in section 5. Finally, we make some concluding remarks in section 6.

2. Computing the distance between the rightmost eigenvalue(s) and the imaginary axis. Let (\bar{u}_0, α_0) be any point in the stable regime of \mathcal{S} and assume \mathbf{M} is nonsingular in (1.1). Let (μ_j, x_j) ($\|x_j\|_2 = 1$, $j = 1, 2, \dots, n$) be the eigenpairs of (1.1) at α_0 , where the real parts of μ_j , $Re(\mu_j)$, are in decreasing order, *i.e.*, $0 > Re(\mu_1) \geq Re(\mu_2) \geq \dots \geq Re(\mu_n)$. Then the distance between the rightmost eigenvalue(s) and the imaginary axis is $-Re(\mu_1)$. Let $\mathbf{A} = \mathcal{J}(\alpha_0)$ and $S = \mathbf{A}^{-1}\mathbf{M}$. To compute this distance, we first observe that $-Re(\mu_1)$ is the eigenvalue with smallest modulus of the $n^2 \times n^2$ generalized eigenvalue problem

$$\Delta_1 z = \lambda(-\Delta_0)z \quad (2.1)$$

where $\Delta_1 = S \otimes I + I \otimes S$ and $\Delta_0 = 2S \otimes S$ (I is the identity matrix of order n). We proceed in two steps to prove this assertion. First, we show that $-Re(\mu_1)$ is an eigenvalue of (2.1).

THEOREM 2.1. *The eigenvalues of (2.1) are $\lambda_{i,j} = -\frac{1}{2}(\mu_i + \mu_j)$, $i, j = 1, 2, \dots, n$. For any pair (i, j) , there are eigenvectors associated with $\lambda_{i,j}$ given by $z_{i,j} = x_i \otimes x_j$ and $z_{j,i} = x_j \otimes x_i$.*

Proof. Since (μ_j, x_j) ($j = 1, 2, \dots, n$) are the eigenpairs of $\mathbf{A}x = \mu\mathbf{M}x$, $(\frac{1}{\mu_j}, x_j)$ are the eigenpairs of S . We first prove that the eigenvalues of (2.1) are $\{\lambda_{i,j}\}_{i,j=1}^n$. Let J be the Jordan normal form of S and P be an invertible matrix such that $S = PJP^{-1}$. Then

$$\begin{aligned} (-\Delta_0)^{-1} \Delta_1 (P \otimes P) &= -\frac{1}{2} (PJP^{-1} \otimes PJP^{-1})^{-1} (PJP^{-1} \otimes I + I \otimes PJP^{-1}) (P \otimes P) \\ &= -\frac{1}{2} (PJ^{-1}P^{-1} \otimes PJ^{-1}P^{-1}) (PJ \otimes P + P \otimes PJ) \\ &= -\frac{1}{2} (P \otimes P) (J^{-1}P^{-1} \otimes J^{-1}P^{-1}) (PJ \otimes P + P \otimes PJ) \\ &= (P \otimes P) \left[-\frac{1}{2} (I \otimes J^{-1} + J^{-1} \otimes I) \right]. \end{aligned}$$

This implies that (2.1) and $-\frac{1}{2} (I \otimes J^{-1} + J^{-1} \otimes I)$ have the same eigenvalues. Due to the special structure of the Jordan normal form J , $I \otimes J^{-1} + J^{-1} \otimes I$ is an upper triangular matrix whose diagonal entries are $\{\mu_i + \mu_j\}_{i,j=1}^n$. Consequently, the eigenvalues of $-\frac{1}{2} (I \otimes J^{-1} + J^{-1} \otimes I)$ are $\{-\frac{1}{2}(\mu_i + \mu_j)\}_{i,j=1}^n = \{\lambda_{i,j}\}_{i,j=1}^n$. Therefore, the eigenvalues of (2.1) are $\{\lambda_{i,j}\}_{i,j=1}^n$ as well.

Second, we show that $z_{i,j}$ is an eigenvector of (2.1) associated with the eigenvalue $\lambda_{i,j}$. For any pair (i, j) ($i, j = 1, 2, \dots, n$),

$$\Delta_1(x_i \otimes x_j) = (S \otimes I + I \otimes S)(x_i \otimes x_j) = (Sx_i) \otimes x_j + x_i \otimes (Sx_j) = \left(\frac{1}{\mu_i} + \frac{1}{\mu_j} \right) (x_i \otimes x_j),$$

and

$$\Delta_0(x_i \otimes x_j) = 2(S \otimes S)(x_i \otimes x_j) = 2(Sx_i) \otimes (Sx_j) = \frac{2}{\mu_i \mu_j} (x_i \otimes x_j).$$

Therefore, $\Delta_1 z_{i,j} = \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) \frac{\mu_i \mu_j}{2} \Delta_0 z_{i,j} = \lambda_{i,j} (-\Delta_0) z_{i,j}$. Similarly, we can show that $\Delta_1 z_{j,i} = \lambda_{i,j} (-\Delta_0) z_{j,i}$ \square

If μ_1 is real, then $-Re(\mu_1) = -\mu_1 = -\frac{1}{2}(\mu_1 + \mu_1) = \lambda_{1,1}$; if μ_1 is not real (*i.e.*, $\mu_1 = \overline{\mu_2}$ and $x_1 = \overline{x_2}$), then $-Re(\mu_1) = -\frac{1}{2}(\mu_1 + \overline{\mu_1}) = -\frac{1}{2}(\mu_1 + \mu_2) = \lambda_{1,2} = \lambda_{2,1}$. In both cases, by Theorem 2.1, $-Re(\mu_1)$ is an eigenvalue of (2.1).

We next show that $-Re(\mu_1)$ is the eigenvalue of (2.1) with smallest modulus.

THEOREM 2.2. *Assume all the eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$ lie in the left half of the complex plane. Then the eigenvalue of (2.1) with smallest modulus is $-Re(\mu_1)$.*

Proof. Let $\mu_j = a_j + ib_j$. Then $0 > a_1 \geq a_2 \geq \dots \geq a_n$. If the rightmost eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$ is real, then $-Re(\mu_1) = \lambda_{1,1}$, and since $0 > a_1 \geq a_2 \geq \dots \geq a_n$, it follows that

$$|\lambda_{1,1}|^2 = \frac{1}{4}(a_1 + a_1)^2 \leq \frac{1}{4} [(a_i + a_j)^2 + (b_i + b_j)^2] = |\lambda_{i,j}|^2$$

for any pair (i, j) . Alternatively, if the rightmost eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$ consist of a complex conjugate pair, then $a_1 = a_2$, $b_1 = -b_2$, $-Re(\mu_1) = \lambda_{1,2} = \lambda_{2,1}$, and similarly,

$$|\lambda_{1,2}|^2 = |\lambda_{2,1}|^2 = \frac{1}{4} [(a_1 + a_1)^2 + (b_1 - b_1)^2] \leq \frac{1}{4} [(a_i + a_j)^2 + (b_i + b_j)^2] = |\lambda_{i,j}|^2$$

for any pair (i, j) . In both cases, $-Re(\mu_1)$ is the eigenvalue of (2.1) with smallest modulus. \square

Simple example: Consider a 4×4 example of $\mathbf{A}x = \mu\mathbf{M}x$ whose eigenvalues are $\mu_{1,2} = -1 \pm 5i$, $\mu_3 = -2$ and $\mu_4 = -3$ (see Figure 2.1a). The eigenvalues of the corresponding 16×16 eigenvalue problem (2.1) are plotted in Figure 2.1b and listed in Table 2.1. As seen in Figure 2.1b, $\lambda_{1,1} = 1 - 5i$, $\lambda_{1,2} = \lambda_{2,1} = 1$ and $\lambda_{2,2} = 1 + 5i$ are the leftmost eigenvalues of (2.1), and $\lambda_{1,2} = \lambda_{2,1}$ are the eigenvalues of (2.1) with smallest modulus.

Table 2.1: The eigenvalues of (2.1) corresponding to the 4×4 example

$\lambda_{1,1} = 1 - 5i$			
$\lambda_{2,1} = \lambda_{1,2} = 1$		$\lambda_{2,2} = 1 + 5i$	
$\lambda_{3,1} = \lambda_{1,3} = 1.5 - 2.5i$	$\lambda_{3,2} = \lambda_{2,3} = 1.5 + 2.5i$	$\lambda_{3,3} = 2$	
$\lambda_{4,1} = \lambda_{1,4} = 2 - 2.5i$	$\lambda_{4,2} = \lambda_{2,4} = 2 + 2.5i$	$\lambda_{4,3} = \lambda_{3,4} = 2.5$	$\lambda_{4,4} = 3$

Assume $\mathbf{A}x = \mu\mathbf{M}x$ has a complete set of eigenvectors $\{x_j\}_{j=1}^n$. Then (2.1) also has a complete set of eigenvectors $\{z_{i,j}\}_{i,j=1}^n$. By Theorem 2.2, the distance between the imaginary axis and the rightmost eigenvalue(s), $-Re(\mu_1)$, can be found by inverse iteration applied to (2.1). Unfortunately, this approach is not suitable for large n because it involves solving linear systems of order n^2 . In [8, 15], an $n^2 \times n^2$ eigenvalue problem similar in structure to (2.1) is dealt with by rewriting an equation of Kronecker sums into an equation of Lyapunov form, *i.e.*, (1.3). Here, similarly, we can rewrite (2.1) into

$$SZ + ZS^T + \lambda(2SZS^T) = 0. \quad (2.2)$$

Any eigenpair (λ, z) of (2.1) is related to a solution (λ, Z) of (2.2), which we also refer to as an *eigenpair* of (2.2), by $z = \text{vec}(Z)$. By Theorem 2.1 and the relation between (2.1) and (2.2), $(\lambda_{i,j}, Z_{i,j})$ ($i, j = 1, 2, \dots, n$) are the eigenpairs of (2.2) where $Z_{i,j} = x_j x_i^T$; in addition, by Theorem 2.2,

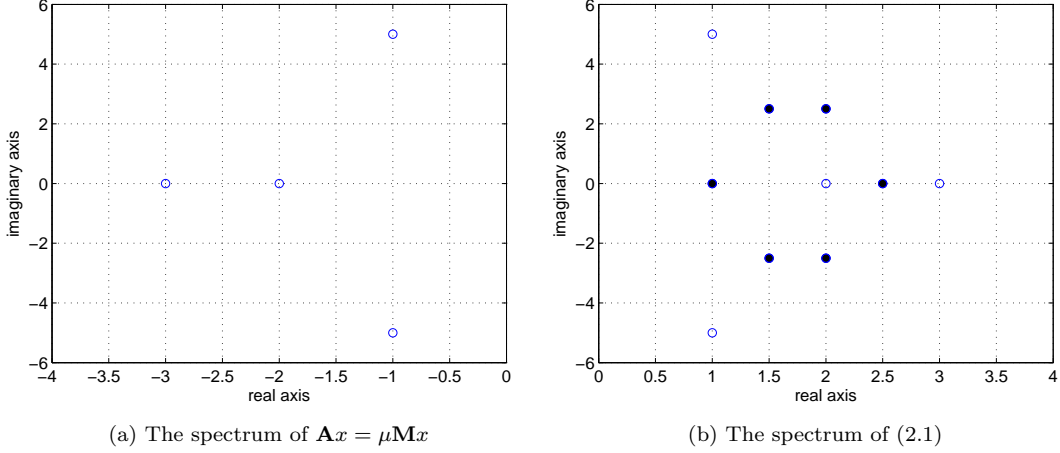


Fig. 2.1: The spectrum of $\mathbf{A}x = \mu\mathbf{M}x$ and (2.1) for the 4×4 example (\bullet : double eigenvalues).

$-Re(\mu_1)$ is the eigenvalue of (2.2) with smallest modulus. Furthermore, under certain conditions, $-Re(\mu_1)$ is an eigenvalue of (2.2) whose associated eigenvector is real, symmetric and of low rank. Assume the following: **(a1)** for any $1 < i \leq n$, if $Re(\mu_i) = Re(\mu_1)$, then $\mu_i = \bar{\mu}_1$; **(a2)** μ_1 is a simple eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$. Consequently, if μ_1 is real, $-Re(\mu_1)$ is a simple eigenvalue of (2.1) with the eigenvector $z_{1,1} = x_1 \otimes x_1$; otherwise, $-Re(\mu_1)$ is a double eigenvalue of (2.1) with the eigenvectors $z_{1,2} = x_1 \otimes \bar{x}_1$ and $z_{2,1} = \bar{x}_1 \otimes x_1$. When the eigenvectors of (2.2) are restricted to the subspace of $\mathbb{C}^{n \times n}$ consisting of symmetric matrices Z , then by Theorem 2.3 from [15], $-Re(\mu_1)$ has a unique (up to a scalar multiplier), real and symmetric eigenvector $x_1 x_1^T$ (if μ_1 is real), or $x_1 x_1^* + \bar{x}_1 \bar{x}_1^T$ (if μ_1 is not real) where x_1^* denotes the conjugate transpose of x_1 . Therefore, we can apply Lyapunov inverse iteration (see [8, 15]) to (2.2) to find $-Re(\mu_1)$, the eigenvalue of (2.2) with smallest modulus:

Algorithm 1 (Lyapunov inverse iteration for (2.2))

1. Given $V_0 \in \mathbb{R}^n$ with $\|V_0\|_2 = 1$ and $d_0 = 1$.
2. For $\ell = 1, 2, \dots$
 - 2.1. Rank reduction¹: compute $\tilde{S} = V_{\ell-1}^T S V_{\ell-1}$ and solve for the eigenvalue $\tilde{\lambda}_1$ of

$$\tilde{S}\tilde{Z} + \tilde{Z}\tilde{S}^T + \tilde{\lambda} \left(2\tilde{S}\tilde{Z}\tilde{S}^T \right) = 0 \quad (2.3)$$

with smallest modulus and its eigenvector $\tilde{Z}_1 = \tilde{V}\tilde{D}\tilde{V}^T$, where $\tilde{V} \in \mathbb{R}^{d_{\ell-1} \times r}$, $\tilde{D} \in \mathbb{R}^{r \times r}$ with $\|\tilde{D}\|_F = 1$, and $r = 1$ ($\ell = 1$) or 2 ($\ell \geq 2$).

- 2.2. Set $\lambda^{(\ell)} = \tilde{\lambda}_1$ and $Z^{(\ell)} = \mathcal{V}_\ell \tilde{D} \mathcal{V}_\ell^T$, where $\mathcal{V}_\ell = V_{\ell-1} \tilde{V}$.
- 2.3. If $(\lambda^{(\ell)}, Z^{(\ell)})$ is accurate enough, then stop.
- 2.4. Else, solve for Y_ℓ from

$$S Y_\ell + Y_\ell S^T = -2S Z^{(\ell)} S^T \quad (2.4)$$

¹When $\ell = 1$, (2.3) is a scalar equation and its eigenpair is $(\tilde{\lambda}_1, \tilde{Z}_1) = (-\tilde{S}^{-1}, 1)$.

in factored form: $Y_\ell = V_\ell D_\ell V_\ell^T$, where $V_\ell \in \mathbb{R}^{n \times d_\ell}$ is orthonormal and $D_\ell \in \mathbb{R}^{d_\ell \times d_\ell}$.

As the iteration proceeds, the iterate $(\lambda^{(\ell)}, Z^{(\ell)} = \mathcal{V}_\ell \tilde{D} \mathcal{V}_\ell^T)$ will converge to $(-Re(\mu_1), \mathcal{V} \mathcal{D} \mathcal{V}^T)$ where $\|\mathcal{D}\|_F = 1$ and $\mathcal{V} = x_1$ (if μ_1 is real) or $\mathcal{V} \in \mathbb{R}^{n \times 2}$ is an orthonormal matrix whose columns span $\{x_1, \bar{x}_1\}$ (if μ_1 is not real). Besides estimates of $-Re(\mu_1)$, we can also obtain from Algorithm 1 estimates of (μ_1, x_1) by solving the small 1×1 or 2×2 eigenvalue problem

$$(\mathcal{V}_\ell^T S \mathcal{V}_\ell) y = \theta y \quad (2.5)$$

and taking $\mu^{(\ell)} = \frac{1}{\theta}$ and $x^{(\ell)} = \mathcal{V}_\ell y$. As \mathcal{V}_ℓ converges to \mathcal{V} , $(\mu^{(\ell)}, x^{(\ell)})$ will converge to (μ_1, x_1) .

At each iteration of Algorithm 1, a large-scale *Lyapunov equation* (2.4) needs to be solved. We can rewrite (2.4) as

$$S Y_\ell + Y_\ell S^T = P_\ell C_\ell P_\ell^T \quad (2.6)$$

(see [8] for details) where P_ℓ is orthonormal and is of rank 1 ($\ell = 1$) or 2 ($\ell > 1$). The solution to (2.6), Y_ℓ , is real and symmetric and frequently has low rank (see [1, 16]). Since S is large, direct methods such as [2, 12] are not suitable. An iterative method that solves Lyapunov equations with large coefficient matrix and low-rank right-hand side is needed. Krylov-type methods for (2.6), such as the ‘‘standard’’ Krylov subspace method [13, 17], the Extended Krylov Subspace Method (EKSM) [18] and the Rational Krylov Subspace Method (RKSM) [6, 7], construct approximate solutions of the form $Y_\ell^{approx} = W X W^T$ where W is an orthonormal matrix whose columns span the Krylov subspace and X is the solution to the small, projected Lyapunov equation $(W^T S W) X + X (W^T S W)^T = (W^T P_\ell) C_\ell (W^T P_\ell)^T$, which can be obtained using direct methods. For example, the standard Krylov subspace method [13, 17] builds the mp -dimensional Krylov subspace

$$\mathcal{K}_m(S, P_\ell) = span \{P_\ell, S P_\ell, \dots, S^{m-1} P_\ell\} \quad (2.7)$$

where m is the number of block Arnoldi steps and p is the block size (*i.e.*, rank of P_ℓ). The main cost of solving (2.6) using Krylov-type methods is $(m-1)p$ linear solves with coefficient matrix $a\mathbf{A} + b\mathbf{M}$, where values of the scalars a, b depend on the Krylov method used.

In step 2.1 (rank reduction) of Algorithm 1, although it may look like computing the reduced-rank matrix $\tilde{S} = V_{\ell-1}^T S V_{\ell-1}$ requires another $d_{\ell-1}$ linear solves with coefficient matrix \mathbf{A} , in fact, if a Krylov-type method is used to solve (2.6), \tilde{S} can be obtained from the *Arnoldi decomposition* computed by the Krylov-type method for no additional cost when $\ell > 1$. Assume the standard Krylov method is used to solve (2.6). In the $(\ell-1)^{st}$ iteration of Algorithm 1, it computes the Arnoldi decomposition

$$S V_{\ell-1} = V_{\ell-1} H_m + W_{m+1} H_{m+1,m} E_m^T \quad (2.8)$$

and the approximate solution $Y_1^{approx} = V_{\ell-1} D_{\ell-1} V_{\ell-1}^T$, where the columns of $V_{\ell-1}$ form an orthonormal basis for the Krylov subspace $\mathcal{K}_m(S, P_{\ell-1})$ and are orthogonal to W_{m+1} . (In addition, $H_m \in \mathbb{R}^{mr \times mr}$ is block upper Hessenberg, $W_{m+1} \in \mathbb{R}^{n \times r}$ is orthonormal, $H_{m+1,m} \in \mathbb{R}^{r \times r}$ and E_m holds the last r columns of the identity matrix of order mr , where $r = 1$ or 2 .) This implies that in the ℓ^{th} iteration of Algorithm 1, \tilde{S} is simply H_m , which has been computed already. The Arnoldi decomposition computed by EKSM or RKSM has a form that is more complicated than (2.8); nonetheless, it produces the matrix \tilde{S} needed for the next iteration as well.

3. Numerical experiments. In this section, we test Algorithm 1 on several problems arising from fluid dynamics. Note that when (1.2) comes from a standard (*e.g.*, finite element) discretization of the incompressible Navier-Stokes equations, the mass matrix \mathbf{M} is singular, leading to infinite eigenvalues of (1.1) and singular $S = \mathbf{A}^{-1}\mathbf{M}$. As in [8], we use the shifted, nonsingular mass matrix proposed in [4], which maps the infinite eigenvalues of (1.1) to finite ones away from the imaginary axis and leaves the finite eigenvalues of (1.1) unchanged. From here on, \mathbf{M} refers to this shifted mass matrix.

3.1. Example 1: driven-cavity flow. Linear stability analysis of this flow is studied in many papers, for example, [9]. The Q_2 - Q_1 mixed finite element discretization (with a 64×64 mesh) of the Navier-Stokes equations gives rise to a generalized eigenvalue problem (1.1) of order $n = 9539$, where the parameter α is the *Reynolds number* (denoted by Re) of the flow. (The Reynolds number of this flow is defined to be $Re = \frac{1}{\nu}$, where ν is the *kinematic viscosity*). Figure 3.1a depicts the path traced out by the eight rightmost eigenvalues of (1.1) for $Re = 2000, 4000, 6000, 7800$, at which the steady-state solution to (1.2) is stable. As the Reynolds number increases, the following trend can be observed: the eight rightmost eigenvalues all move towards the imaginary axis, and they become more clustered as they approach the imaginary axis. In addition, although the rightmost eigenvalue starts off being real, one conjugate pair of complex eigenvalues (whose imaginary parts are about $\pm 3i$) move faster towards the imaginary axis than the other eigenvalues and eventually they become the rightmost. They first cross the imaginary axis at $Re \approx 7929$, causing instability in the steady-state solution of (1.2) (see [8]).

Finding the conjugate pair of rightmost eigenvalues of (1.1) at a high Reynolds number (for example, at $Re = 7800$) can be difficult. Suppose we are trying to find the rightmost eigenvalues at $Re = 7800$ by conventional methods, such as computing k eigenvalues of (1.1) with smallest modulus using the *Implicitly Restarted Arnoldi* (IRA) method [19]. If we use the Matlab function ‘eigs’ (which implements the IRA method) with its default setting, then k has to be as large as 250, since there are many eigenvalues that have smaller modulus than the rightmost pair. This leads to at least 500 linear solves (with coefficient matrix \mathbf{A}), and in practice, many more. More importantly, note that the decision $k = 250$ is made based on *a priori* knowledge of where the rightmost eigenvalues lie. In general, we cannot identify a good value for k that guarantees that the rightmost eigenvalues will be found.

For four various Reynolds numbers between 2000 and 7800, we apply Algorithm 1 (with RKSM as the Lyapunov solver) to calculate the distance between the rightmost eigenvalue(s) of (1.1) and the imaginary axis. The results are reported in Table 3.2 (see Table 3.1 for notation). The initial guess V_0 is chosen to be a random vector of unit norm in \mathbb{R}^n , the stopping criterion for the eigenvalue residual is $\|\mathfrak{R}_\ell\|_F < 10^{-8}$, and the stopping criterion for the Lyapunov solve is

$$\|R_\ell\|_F < 10^{-9} \cdot \|P_\ell C_\ell P_\ell^T\|_F = 10^{-9} \cdot \|C_\ell\|_F.$$

Note that both residual norms $\|\mathfrak{R}_\ell\|_F$ and $\|R_\ell\|_F$ are cheap to compute (see [8] for details). Therefore, the main cost of each iteration is about d_ℓ linear solves of order n . All linear systems are solved using direct methods. As shown in Table 3.2, the distances between the rightmost eigenvalue(s) of (1.1) and the imaginary axis at $Re = 2000, 4000, 6000, 7800$ are 0.03264, 0.01608, 0.01084, 0.00514, respectively. We also obtain estimates of the rightmost eigenvalue of (1.1) at the four Reynolds numbers: -0.03264, -0.01608, -0.01084, and -0.00514+2.69845i.

We note two trends seen in these results. First, surprisingly, for all the Reynolds numbers considered, Algorithm 1 converges to the desired tolerance ($\|\mathfrak{R}_\ell\|_F < 10^{-8}$) in only 2 iterations.

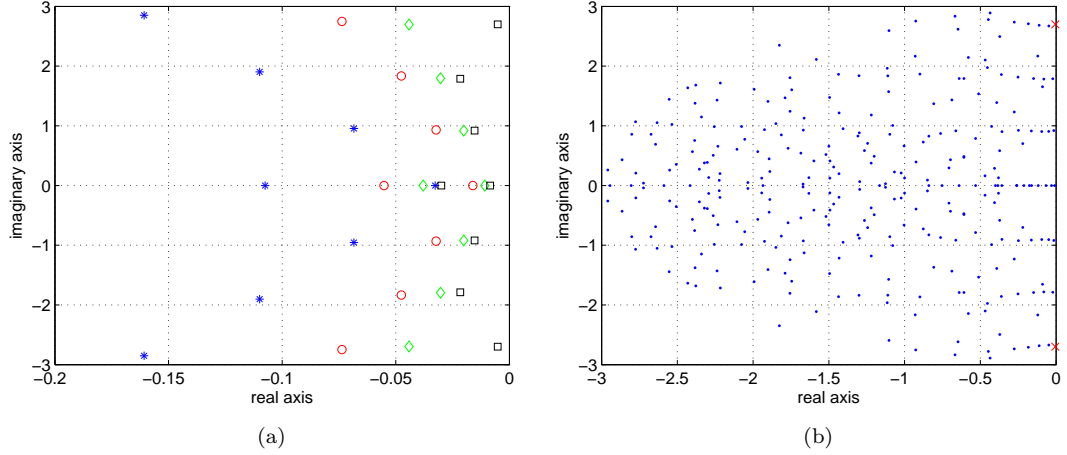


Fig. 3.1: (a) The eight rightmost eigenvalues for driven-cavity flow at different Reynolds numbers (* : $Re = 2000$, \circ : $Re = 4000$, \diamond : $Re = 6000$, \square : $Re = 7800$). (b) The 300 eigenvalues with smallest modulus at $Re = 7800$ (\times : the rightmost eigenvalues).

Table 3.1: Notation for Algorithm 1

Symbol	Definition
$\lambda^{(\ell)}$	the estimate of $-Re(\mu_1)$, <i>i.e.</i> , the eigenvalue of (2.2) with smallest modulus
$Z^{(\ell)}$	the estimated eigenvector of (2.2) associated with $-Re(\mu_1)$
$\mu^{(\ell)}$	the estimated rightmost eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$ computed from (2.5)
Y_ℓ^{approx}	the approximate solution to the Lyapunov solution (2.6)
\mathfrak{R}_ℓ	$SZ^{(\ell)} + Z^{(\ell)}S^T + \lambda^{(\ell)}(2SZ^{(\ell)}S^T)$, the residual of the Lyapunov eigenvalue problem (2.2)
R_ℓ	$SY_\ell^{approx} + Y_\ell^{approx}S^T - P_\ell C_\ell P_\ell^T$, the residual of the Lyapunov equation (2.6)
d_ℓ	dimension of the Krylov subspace, <i>i.e.</i> , rank of Y_ℓ^{approx}

That is, only the first Lyapunov equation

$$SY_1 + Y_1S^T = P_1C_1P_1^T \quad (3.1)$$

needs to be solved, where $P_1 \in \mathbb{R}^n$ and $C_1 \in \mathbb{R}$. Second, as the Reynolds number increases, it becomes more expensive to solve the Lyapunov equation to the same order of accuracy ($\|R_\ell\|_F < 10^{-9} \cdot \|C_\ell\|_F$), since Krylov subspaces of increasing dimension are needed (156, 241, 307 and 366 for the four Reynolds numbers). We also tested Algorithm 1 using the standard Krylov method [17] to solve the Lyapunov systems. To solve (3.1) to the same accuracy, this method requires subspaces of dimension 525, 614, 770 and 896 for the four Reynolds numbers, which are much larger than those required by RKSM (see Figure 3.2 for comparison). As a result, the standard method requires many more linear solves.

Table 3.2: Algorithm 1 applied to Example 1 (Lyapunov solver: RKSM)

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\ \mathfrak{R}_\ell\ _F$	$\ R_\ell\ _F$	d_ℓ
Re=2000					
1	884.383	-884.383	1.32049e+02	1.40794e-10	156
2	0.03264	-0.03264	2.56263e-11	—	—
Re=4000					
1	-17765.8	17765.8	6.58651e+03	3.52618e-10	241
2	0.01608	-0.01608	4.25055e-10	—	—
Re=6000					
1	1301.24	-1301.24	8.55652e+02	6.52387e-10	307
2	0.01084	-0.01084	7.11628e-10	—	—
Re=7800					
1	695.951	-695.951	6.58622e+02	9.02875e-10	366
2	0.00514	-0.00514+2.69845i	3.62567e-11	—	—

3.2. Example 2: flow over an obstacle. For linear stability analysis of this flow, see [8]. The Q_2 - Q_1 mixed finite element discretization (with a 32×128 mesh) of the Navier-Stokes equations gives rise to a generalized eigenvalue problem (1.1) of order $n = 9512$. Figure 3.3a depicts the path traced out by the six rightmost eigenvalues of (1.1) for $Re = 100, 200, 300, 350$ in the stable regime, and Figure 3.3b shows the 300 eigenvalues of (1.1) with smallest modulus at $Re = 350$. (In this example, the Reynolds number $Re = \frac{2}{\nu}$.) As for the previous example, as the Reynolds number increases, the six rightmost eigenvalues all move towards the imaginary axis, and the rightmost eigenvalue changes from being real (at $Re = 100$) to complex (at $Re = 200, 300, 350$). The rightmost pair of eigenvalues of (1.1) cross the imaginary axis and the steady-state solution to (1.2) loses its stability at $Re \approx 373$.

We again apply Algorithm 1 to estimate the distance between the rightmost eigenvalue(s) of (1.1) and the imaginary axis for the four Reynolds numbers mentioned above. The results are reported in Table 3.3. The stopping criteria for both Algorithm 1 and the Lyapunov solve (2.6) remain unchanged, *i.e.*, $\|\mathfrak{R}_\ell\|_F < 10^{-8}$ and $\|R_\ell\|_F < 10^{-9} \cdot \|C_\ell\|_F$. For all four Reynolds numbers, Algorithm 1 converges rapidly. In fact, we will show in section 3.4 that if the Lyapunov equation (3.1) is solved more accurately, Algorithm 1 will converge in two iterations in all four cases as observed in the previous example. Again we compare the performance of the standard Krylov method and RKSM in solving (3.1). As for the cavity flow, the Krylov method needs a significantly larger subspace than RKSM to compute a solution of the same accuracy (see Figure 3.4).

3.3. Example 3: double-diffusive convection problem. This is a model of the effects of convection and diffusion on two solutions in a box heated at one boundary (see Chapter 8 of [21]). The governing equations use Boussinesq approximation and are given in [3] and [5]. Linear stability analysis of this problem is considered in [10]. The imaginary parts of the rightmost eigenvalues of (1.1) near the critical point (\bar{u}_c, α_c) have fairly large magnitude, and as a result, the rightmost eigenvalues are further away from zero than many of the real eigenvalues close to the imaginary axis. Conventional methods, such as IRA with a zero shift, tend to converge to the real eigenvalues close to the imaginary axis instead of the rightmost pair.

We consider an artificial version $\mathbf{A}x = \mu x$ of this problem, where \mathbf{A} is tridiagonal of order

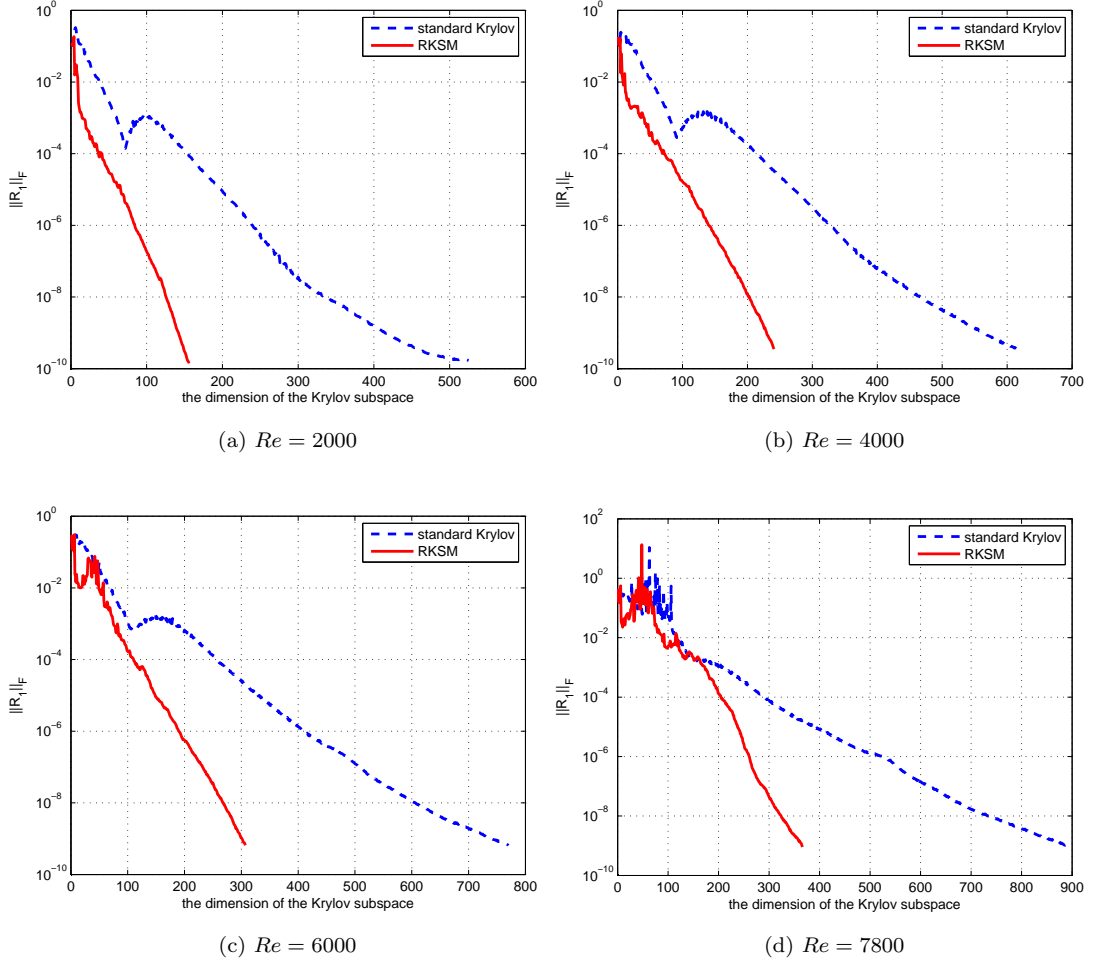


Fig. 3.2: Comparison of the standard Krylov method and RKSM for solving (3.1) in Example 1

$n = 10,000$ with eigenvalues $\mu_{1,2} = -0.05 \pm 25i$ and $\mu_j = -(j-1) \cdot 0.1$ for all $3 \leq j \leq n$. The 300 eigenvalues of \mathbf{A} with smallest modulus are plotted in Figure 3.5 (left). A similar problem is studied in [14]. If we use the Matlab function ‘eigs’ with zero shift to compute its rightmost eigenvalues, at least 251 eigenvalues of \mathbf{A} have to be computed to ensure that $\mu_{1,2}$ will be found. This approach requires a minimum 502 linear solves under the default setting of ‘eigs’, and again in practice many more will be needed. We apply Algorithm 1 to this problem (with the same stopping criteria for the inner and outer iterations as in the previous two examples) and the results are reported in Table 3.4. It converges in just 3 iterations, requiring 90 linear solves to solve the two Lyapunov equations to desired accuracy. As in the previous examples, RKSM needs a Krylov subspace of significantly smaller dimension than the standard Krylov method (see Figure 3.5 (right)).

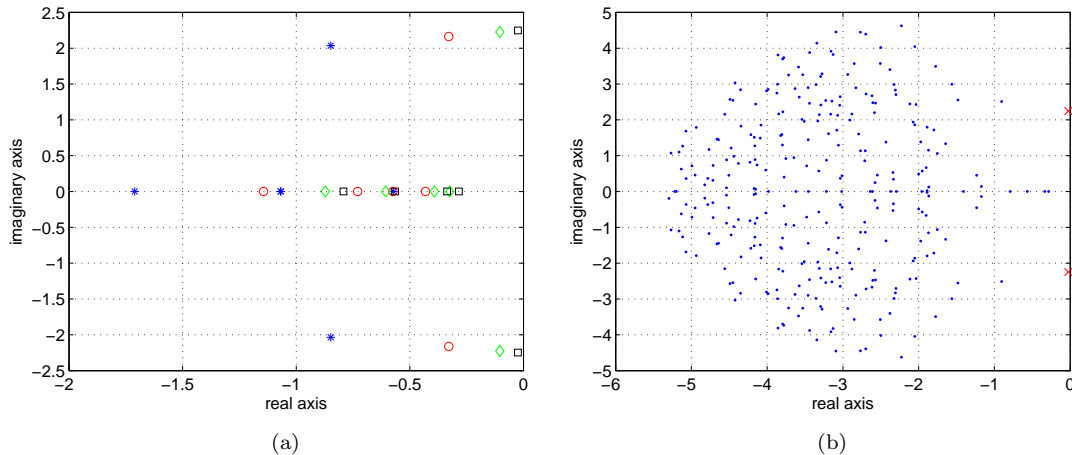


Fig. 3.3: (a) The six rightmost eigenvalues for flow over an obstacle at different Reynolds numbers (* : $Re = 100$, \circ : $Re = 200$, \diamond : $Re = 300$, \square : $Re = 350$). (b) The 300 eigenvalues with smallest modulus at $Re = 350$ (\times : the rightmost eigenvalues).

3.4. Analysis of the convergence of Algorithm 1. In the numerical experiments, we have shown that Algorithm 1 converges rapidly. In particular, it converges in just two iterations in many cases, for example, the driven-cavity flow at $Re = 2000, 4000, 6000, 7800$. In other words, only one Lyapunov solve (3.1) is needed to obtain an eigenvalue estimate of desired accuracy in these cases. The analysis below provides some insight into this fast convergence.

We introduce some notation to be used in the analysis. Assuming (1.1) and therefore (2.1) have complete sets of eigenvectors, let $\{(\lambda_k, z_k)\}_{k=1}^{n^2}$ denote the eigenpairs of (2.1), where $\|z_k\|_2 = 1$ and $\{\lambda_k\}$ have increasing moduli, *i.e.*, $|\lambda_{k_1}| \leq |\lambda_{k_2}|$ if $k_1 < k_2$. By Theorem 2.1, for each $1 \leq k \leq n^2$, there exist $1 \leq i, j \leq n$ such that $\lambda_k = \lambda_{i,j} = -\frac{1}{2}(\mu_i + \mu_j)$ and $z_k = z_{i,j} = x_i \otimes x_j$.² Let $Z_k = x_j x_i^T$, *i.e.*, $\text{vec}(Z_k) = z_k$, so that $\{(\lambda_k, Z_k)\}_{k=1}^{n^2}$ are the eigenpairs of (2.2). In addition, let $L_p = \{\lambda_1, \lambda_2, \dots, \lambda_p\}$ contain the p eigenvalues of (2.1) with smallest modulus, and let $E_p = \{\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_d}\}$ be the smallest subset of eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$ that satisfies the following: for any $\lambda_k \in L_p$, there exist $\mu_{i_s}, \mu_{i_t} \in E_p$ such that $\lambda_k = \lambda_{i_s, i_t}$. Let $X_p = \{x_{i_1}, x_{i_2}, \dots, x_{i_d}\}$ hold the eigenvectors of $\mathbf{A}x = \mu \mathbf{M}x$ associated with E_p . For a concrete example, consider again the 4×4 example in section 2. From Table 2.1, $L_7 = \{\lambda_{1,2}, \lambda_{2,1}, \lambda_{3,3}, \lambda_{1,3}, \lambda_{3,1}, \lambda_{2,3}, \lambda_{3,2}\}$, $E_7 = \{\mu_1, \mu_2, \mu_3\}$, and $X_7 = \{x_1, x_2, x_3\}$.

We first look at standard inverse iteration applied to (2.1). Let the starting guess be $z^{(1)} = v \otimes v \in \mathbb{R}^{n^2}$, where $v \in \mathbb{R}^n$ is a random vector of unit norm. Since $\{z_k\}_{k=1}^{n^2}$ are linearly independent, $z^{(1)}$ can be written as $\sum_{k=1}^{n^2} \xi_k z_k$ with $\xi_k \in \mathbb{C}$ (assume $\xi_k \neq 0$ for any k). Note that the coefficients $\{\xi_k\}_{k=1}^{n^2}$ have the following properties: if $z_{k_1} = \overline{z_{k_2}}$, then $\xi_{k_1} = \overline{\xi_{k_2}}$; moreover, if $z_{k_1} = x_i \otimes x_j$ and

²Both sets of symbols $\{\lambda_{i,j}, z_{i,j}\}_{i,j=1}^n$ and $\{\lambda_k, z_k\}_{k=1}^{n^2}$ denote the eigenpairs of (2.1). The double subscripts indicate the special structure of the eigenpairs, whereas the single subscripts arrange the eigenvalues in ascending order of their moduli. Our choice between the two notations depends on the context.

Table 3.3: Algorithm 1 applied to Example 2 (Lyapunov solver: RKSM)

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\ \mathfrak{R}_\ell\ _F$	$\ R_\ell\ _F$	d_ℓ
Re=100					
1	-2.42460	2.42460	1.15123e+1	2.44638e-09	45
2	0.57285	-0.57285	1.28322e-4	1.15950e-11	22
3	0.57285	-0.57285	4.86146e-6	1.64039e-09	18
4	0.57285	-0.57285	9.49881e-7	5.11488e-09	10
5	0.57285	-0.57285	1.35238e-7	5.65183e-09	4
6	0.57285	-0.57285	2.30716e-8	8.18398e-10	4
7	0.57285	-0.57285	8.43416e-9	—	—
Re=200					
1	-2.45074	2.45074	1.16834e+1	3.00582e-09	63
2	0.32884	-0.32884+2.16396i	3.86737e-5	2.20976e-10	86
3	0.32884	-0.32884+2.16393i	1.30869e-8	2.04006e-10	46
4	0.32884	-0.32884+2.16393i	1.47390e-9	—	—
Re=300					
1	-2.47804	2.47804	1.18371e+01	4.49864e-09	75
2	0.10405	-0.10405+2.22643i	7.59831e-07	3.60446e-10	86
3	0.10405	-0.10405+2.22643i	2.18881e-10	—	—
Re=350					
1	-2.49317	2.49317	1.19385e+01	3.40780e-09	85
2	0.02411	-0.02411+2.24736i	2.80626e-08	3.84715e-10	90
3	0.02411	-0.02411+2.24736i	1.46747e-11	—	—

Table 3.4: Algorithm 1 applied to Example 3 (Lyapunov solver: RKSM)

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\ \mathfrak{R}_\ell\ _F$	$\ R_\ell\ _F$	d_ℓ
1	109.973	-109.973	4.33472e+00	3.28342e-11	40
2	0.05000	-0.05000+25.0000i	4.76830e-08	3.01965e-12	50
3	0.05000	-0.05000+25.0000i	1.56010e-13	—	—

$z_{k_2} = x_j \otimes x_i$ for some pair (i, j) , then $\xi_{k_1} = \xi_{k_2}$. The first property is due to the fact that $z^{(1)}$ is real, and the second one is a result of the special tensor structure of $z^{(1)}$. In the first step of inverse iteration, we solve the linear system

$$\Delta_1 y_1 = (-\Delta_0)z^{(1)}. \quad (3.2)$$

The solution to (3.2) is $y_1 = \Delta_1^{-1}(-\Delta_0)z^{(1)} = \sum_{k=1}^{n^2} \frac{\xi_k}{\lambda_k} z_k$. Let y_1^p be a truncated approximation of y_1 consisting of its p dominant components, *i.e.*, $y_1^p = \sum_{k=1}^p \frac{\xi_k}{\lambda_k} z_k$ for some $p \ll n^2$.

Next, we consider Algorithm 1 applied to (2.2). Let the starting guess be $Z^{(1)} = vv^T$, where v is the vector that determines the starting vector for standard inverse iteration. At step 2.4 of the first iteration, we solve the Lyapunov equation (3.1) where $\text{vec}(P_1 C_1 P_1^T) = \text{vec}(-2SZ^{(1)}S^T) =$

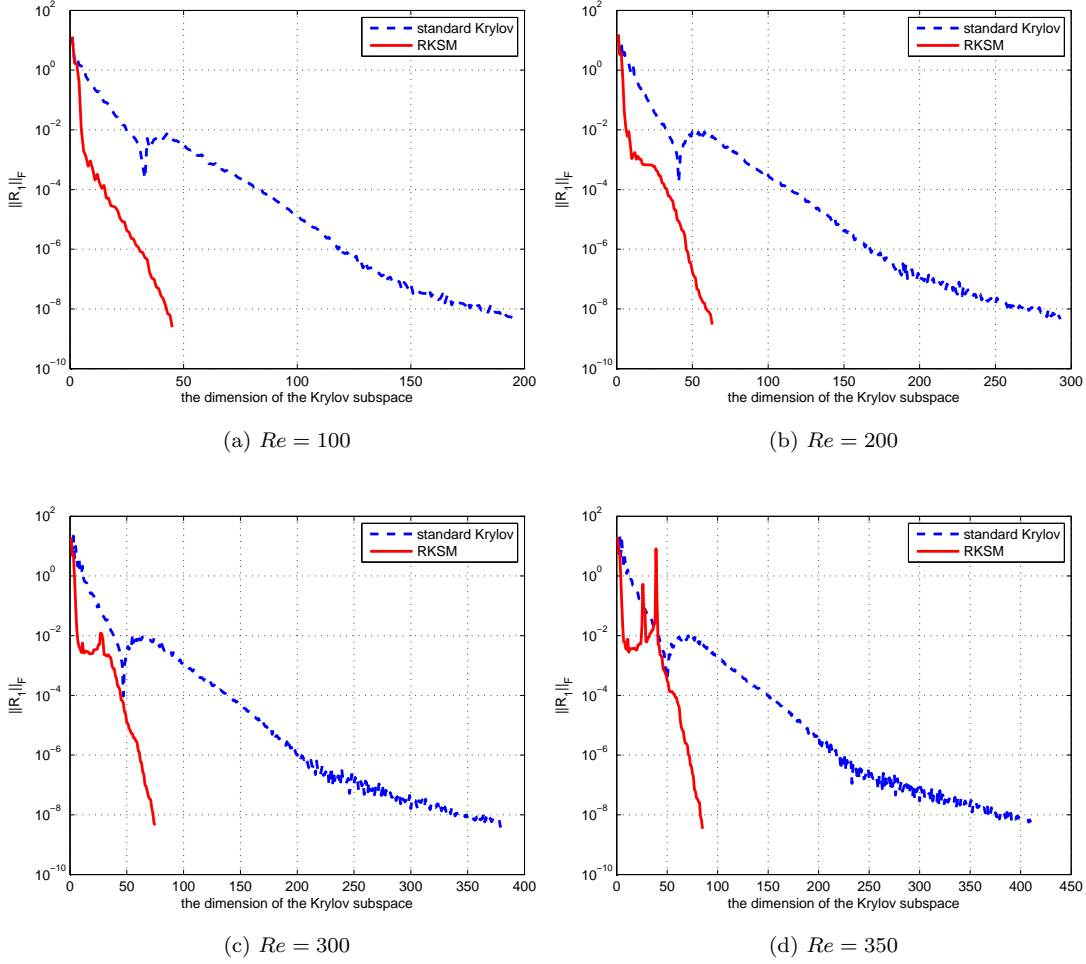


Fig. 3.4: Comparison of the standard Krylov method and RKSM for solving (3.1) in Example 2

$(-\Delta_0)z^{(1)}$ (see (2.4)) and $\text{vec}(Y_1) = y_1$, *i.e.*,

$$Y_1 = \sum_{k=1}^{n^2} \frac{\xi_k}{\lambda_k} Z_k. \quad (3.3)$$

Let Y_1^p denote the truncation of Y_1 that satisfies $\text{vec}(Y_1^p) = y_1^p$, *i.e.*, $Y_1^p = \sum_{k=1}^p \frac{\xi_k}{\lambda_k} Z_k$. Assume p is chosen such that if $\lambda_{i,j} \in L_p$, then $\overline{\lambda_{i,j}}, \lambda_{j,i} \in L_p$ as well. Under this assumption and by properties of the coefficients $\{\xi_k\}_{k=1}^{n^2}$, Y_1^p is real and symmetric and can be written as $Y_1^p = U\mathcal{G}U^T$ where $U \in \mathbb{R}^{n \times d}$ is an orthonormal matrix whose columns span X_p and $\mathcal{G} \in \mathbb{R}^{d \times d}$ is symmetric. Recall from section 2 that the target eigenpair of (2.2) sought by Algorithm 1 is $(\lambda_1, \mathcal{V}\mathcal{D}\mathcal{V}^T)$, where $\lambda_1 = -\text{Re}(\mu_1)$, and $\mathcal{V} \in \mathbb{R}^{n \times r}$ (with $r = 1$ or 2) is x_1 (if μ_1 is real) or an orthonormal matrix

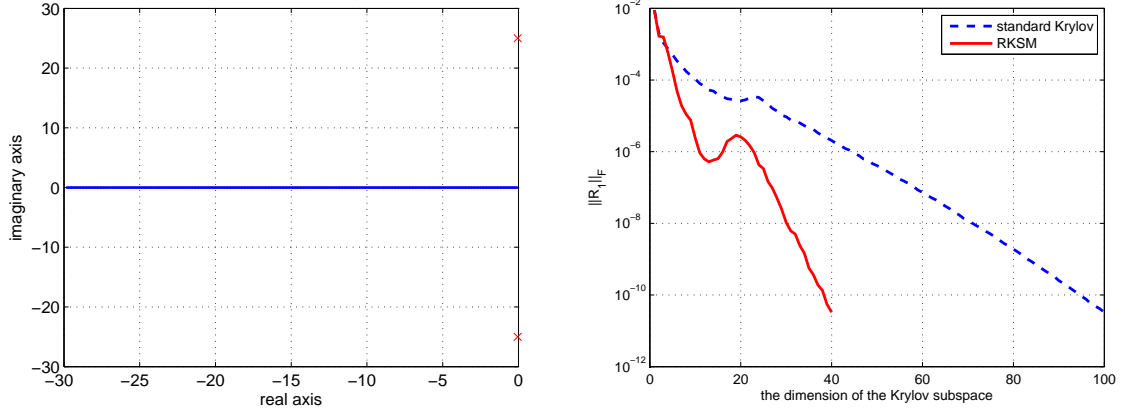


Fig. 3.5: Left: the 300 eigenvalues with smallest modulus (\times : the rightmost eigenvalues). Right: Comparison of the standard Krylov method and RKSM for solving (3.1) in Example 3

whose columns span $\{x_1, \bar{x}_1\}$ (if μ_1 is not real). Since $(\lambda_1, \mathcal{V}\mathcal{D}\mathcal{V}^T)$ is an eigenpair of (2.2),

$$S(\mathcal{V}\mathcal{D}\mathcal{V}^T) + (\mathcal{V}\mathcal{D}\mathcal{V}^T)S^T + \lambda_1(2S(\mathcal{V}\mathcal{D}\mathcal{V}^T)S^T) = 0. \quad (3.4)$$

For any $p \geq 1$, $x_1, \bar{x}_1 \in X_p$. Thus, \mathcal{U} can be taken to have the form $\mathcal{U} = [\mathcal{V}, \mathcal{V}^\perp]$ where $\mathcal{V}^T\mathcal{V}^\perp = \mathbf{0}$.

Assume that step 2.4 of the first iteration of Algorithm 2 produces an approximate solution to (3.1) of the form Y_1^p by some means (that is, the approximate solution consists of the p dominant terms of (3.3) where p satisfies the assumption above). Then at step 2.1 (rank reduction) of the second iteration, we solve the $d \times d$ projected eigenvalue problem

$$(\mathcal{U}^T S \mathcal{U}) \tilde{Z} + \tilde{Z} (\mathcal{U}^T S \mathcal{U})^T + \tilde{\lambda} (2(\mathcal{U}^T S \mathcal{U}) \tilde{Z} (\mathcal{U}^T S \mathcal{U})^T) = 0 \quad (3.5)$$

for the eigenvalue with smallest modulus, $\tilde{\lambda}_1$, and its associated real, symmetric and rank- r eigenvector \tilde{Z}_1 ($r = 1$ or 2). We then obtain an estimate of the target eigenpair of (2.2), namely $(\tilde{\lambda}_1, \mathcal{U}\tilde{Z}_1\mathcal{U}^T)$. It can be shown that this estimate is in fact exact, that is, $(\tilde{\lambda}_1, \mathcal{U}\tilde{Z}_1\mathcal{U}^T) = (\lambda_1, \mathcal{V}\mathcal{D}\mathcal{V}^T)$.

THEOREM 3.1. *If*

$$D_1 = \begin{bmatrix} \mathcal{D} & \\ & \mathbf{0} \end{bmatrix}_{d \times d},$$

then (λ_1, D_1) is an eigenpair of (3.5).

Proof. Left-multiply (3.4) by \mathcal{U}^T and right-multiply by \mathcal{U} :

$$\mathcal{U}^T S(\mathcal{V}\mathcal{D}\mathcal{V}^T)\mathcal{U} + \mathcal{U}^T(\mathcal{V}\mathcal{D}\mathcal{V}^T)S^T\mathcal{U} + \lambda_1 2\mathcal{U}^T S(\mathcal{V}\mathcal{D}\mathcal{V}^T)S^T\mathcal{U} = 0.$$

Since $\mathcal{U} = [\mathcal{V}, \mathcal{V}^\perp]$, it follows that $\mathcal{V}\mathcal{D}\mathcal{V}^T = \mathcal{U}D_1\mathcal{U}^T$. Therefore,

$$(\mathcal{U}^T S \mathcal{U}) D_1 + D_1 (\mathcal{U}^T S^T \mathcal{U}) + \lambda_1 (2(\mathcal{U}^T S \mathcal{U}) D_1 (\mathcal{U}^T S^T \mathcal{U})) = 0.$$

□

PROPOSITION 3.2. *The eigenvalue of (3.5) with smallest modulus is $\tilde{\lambda}_1 = \lambda_1$.*

Proof. Let $\tilde{S} = \mathcal{U}^T S \mathcal{U}$. Since the columns of \mathcal{U} span X_p , the eigenvalues of \tilde{S} are $\mu_{i_1}^{-1}, \mu_{i_2}^{-1}, \dots, \mu_{i_d}^{-1}$. Let $\tilde{\Delta}_1 = \tilde{S} \otimes I + I \otimes \tilde{S}$ and $\tilde{\Delta}_0 = 2\tilde{S} \otimes \tilde{S}$. By Theorem 2.1, the eigenvalues of the $d^2 \times d^2$ eigenvalue problem

$$\tilde{\Delta}_1 \tilde{z} = \tilde{\lambda} \left(-\tilde{\Delta}_0 \right) \tilde{z} \quad (3.6)$$

are $\tilde{\lambda}_{s,t} = -\frac{1}{2}(\mu_{i_s} + \mu_{i_t})$ for any $1 \leq s, t \leq d$. Since $\mu_1, \bar{\mu}_1 \in E_p$, Theorem 2.2 implies that the eigenvalue of (3.6) with smallest modulus is $\tilde{\lambda}_1 = \lambda_1$, the eigenvalue of (2.1) with smallest modulus. Since (3.5) and (3.6) have the same eigenvalues, the eigenvalue of (3.5) with smallest modulus is λ_1 as well. □

Recall that we solve the Lyapunov equation (3.1) using an iterative solver such as RKSM, which produces a real, symmetric approximate solution Y_1^{approx} . By Theorem 3.1 and Proposition 3.2, if $Y_1^{approx} = Y_1^p$, then after the rank-reduction step in the second iteration of Algorithm 1, we obtain the exact eigenpair $(\lambda_1, \mathcal{V} \mathcal{D} \mathcal{V}^T)$ of (2.2) that we are looking for. In reality, it is unlikely that the approximate solution Y_1^{approx} we compute will be exactly Y_1^p . However, since Y_1^p consists of the p dominant terms of the exact solution (3.3), if Y_1^{approx} is accurate enough, then $Y_1^{approx} \approx Y_1^p$ for some p .

This analysis suggests that the eigenvalue residual norm $\|\mathfrak{R}_2\|_F$ can be made arbitrarily small as long as the residual norm of the Lyapunov system $\|R_1\|_F$ is small enough. Therefore, we propose the following modified version of Algorithm 1 (given $\tau_{lyap}, \tau_{eig} > 0$):

Algorithm 2 (Modified Algorithm 1)

1. Given $V_0 \in \mathbb{R}^n$ with $\|V_0\|_2 = 1$. Set $\ell = 1$ and *firsttry* = **true**.
2. Rank reduction: compute $\tilde{S} = V_{\ell-1}^T S V_{\ell-1}$ and solve for the eigenvalue $\tilde{\lambda}_1$ of (2.3) with smallest modulus and its eigenvector $\tilde{Z}_1 = \tilde{V} \tilde{D} \tilde{V}^T$.
3. Set $(\lambda^{(\ell)}, Z^{(\ell)}) = (\tilde{\lambda}_1, \mathcal{V}_\ell \tilde{D} \mathcal{V}_\ell^T)$ where $\mathcal{V}_\ell = V_{\ell-1} \tilde{V}$, and compute $\|\mathfrak{R}_\ell\|_F$.
4. While $\|\mathfrak{R}_\ell\|_F > \tau_{eig}$:
 - 4.1 if *firsttry*
 - compute an approximate solution $Y_1^{approx} = V_1 D_1 V_1^T$ to (3.1) such that $\|R_1\|_F < \tau_{lyap} \cdot \|C_1\|_F$; set $\ell = 2$ and *firsttry* = **false**;
 - 4.2 else
 - solve (3.1) more accurately and update V_1 ;
 - 4.3 repeat steps 2 and 3 to compute $\lambda^{(\ell)}, Z^{(\ell)}$ and $\|\mathfrak{R}_\ell\|_F$.

In this algorithm, if $(\lambda^{(2)}, Z^{(2)})$ is not an accurate enough eigenpair ($\|\mathfrak{R}_2\|_F \geq \tau_{eig}$), this is fixed by improving the accuracy of the Lyapunov system (3.1). The discussion above shows that this will be enough to produce an accurate eigenpair. Moreover, it is possible to get an improved solution to (3.1) by augmenting the solution we have in hand. Assume that at step 4.1 of Algorithm 2, we compute an approximate solution $Y_1^{approx} = V_1 D_1 V_1^T$ to (3.1) where the columns of V_1 span the Krylov subspace $\mathcal{K}_m(S, P_1)$ (see (2.7) for definition), and then obtain an iterate $(\lambda^{(2)}, Z^{(2)})$ in steps 2 and 3. If $\|\mathfrak{R}_2\|_F \geq \tau_{eig}$, we perform one more block Arnoldi step to extend the existing Krylov subspace to $\mathcal{K}_{m+1}(S, P_1)$, obtain a new approximate solution $Y_1^{approx} = V_1 D_1 V_1^T$ to (3.1) where the columns of V_1 now span the augmented Krylov subspace $\mathcal{K}_{m+1}(S, P_1)$, and check convergence

in steps 2 and 3 again. We keep extending the Krylov subspace at our disposal until the outer iteration converges to the desired tolerance τ_{eig} .

We test Algorithm 2 on Example 2 and Example 3, where Algorithm 1 converged in more than two iterations (see Tables 3.3 and 3.4). As in the previous experiments, we choose $\tau_{lyap} = 10^{-9}$ and $\tau_{eig} = 10^{-8}$. The results are reported in Tables 3.5 and 3.6, from which it can be seen that if (3.1) is solved accurately enough, Lyapunov inverse iteration converges to the desired tolerance in only two iterations, as observed for the driven-cavity flow (see Table 3.2). In other words, it requires only one Lyapunov solve (3.1). By comparing Tables 3.4 and 3.6, for example, we can see that in order to compute an accurate enough approximate solution Y_1^{approx} to (3.1), the dimension of the Krylov subspace used must be increased from 40 to 43.

Table 3.5: Algorithm 2 applied to Example 2 (Lyapunov solver: RKSM)

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\ \mathfrak{R}_\ell\ _F$	$\ R_\ell\ _F$	d_ℓ
Re=100					
1	-2.42460	2.42460	1.15123e+1	4.45806e-12	65
2	0.57285	-0.57285	7.98509e-9	—	—
Re=200					
1	-2.45074	2.45074	1.16834e+1	2.79438e-12	83
2	0.32884	-0.32884+2.16393i	7.67144e-9	—	—
Re=300					
1	-2.47804	2.47804	1.18371e+1	1.35045e-10	86
2	0.10405	-0.10405+2.22643i	6.10287e-9	—	—
Re=350					
1	-2.49317	2.49317	1.19385e+1	1.07068e-09	88
2	0.02411	-0.02411+2.24736i	7.16343e-9	—	—

Table 3.6: Algorithm 2 applied to Example 3 (Lyapunov solver: RKSM)

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\ \mathfrak{R}_\ell\ _F$	$\ R_\ell\ _F$	d_ℓ
1	109.973	-109.973	4.33472e+0	4.71359e-12	43
2	0.05000	-0.05000+25.0000i	6.88230e-9	—	—

4. Computing k rightmost eigenvalues. In section 2, we showed that when all the eigenvalues of (1.1) lie in the left half of the complex plane, the distance between the rightmost eigenvalue(s) and the imaginary axis, $-Re(\mu_1)$, is the eigenvalue of (2.2) with smallest modulus. As a result, this eigenvalue can be computed by Lyapunov inverse iteration, which also gives us estimates of the rightmost eigenvalue(s) of (1.1). In section 3, various numerical experiments demonstrate the robustness and efficiency of the Lyapunov inverse iteration applied to (2.2). In particular, we showed in section 3.4 that if the first Lyapunov equation (3.1) is solved accurately enough, then Lyapunov inverse iteration will converge in only two steps. As seen in sections 3.1 and 3.2, when we march along the solution path \mathcal{S} , it may be the case that an eigenvalue that is not the rightmost moves towards the imaginary rapidly, becomes the rightmost eigenvalue at some point and even-

tually crosses the imaginary axis first, causing instability in the steady-state solution. Therefore, besides the rightmost eigenvalue(s), it is helpful to monitor a few other eigenvalues in the rightmost part of the spectrum as well. In this section, we show how Lyapunov inverse iteration can be applied repeatedly in combination with deflation to compute k rightmost eigenvalues of (1.1), where $1 < k \ll n$.

We continue to assume that we are at a point (\bar{u}_0, α_0) in the stable regime of the solution path \mathcal{S} and that the eigenvalue problem $\mathbf{A}x = \mu\mathbf{M}x$ with $\mathbf{A} = \mathcal{J}(\alpha_0)$ has a complete set of eigenvectors $\{x_i\}_{i=1}^n$. For any $i \leq k$, we also assume the following (as in assumptions **(a1)** and **(a2)** in section 2): **(a1')** if $Re(\mu_j) = Re(\mu_i)$ and $j \neq i$, then $\mu_j = \bar{\mu}_i$; **(a2')** μ_i is a simple eigenvalue. Let $\mathcal{E}_t = \{\mu_1, \mu_2, \dots, \mu_t\}$ be the set containing t rightmost eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$ and $\mathcal{X}_t = [x_1, x_2, \dots, x_t] \in \mathbb{C}^{n \times t}$ be the matrix that holds the t corresponding eigenvectors. Here t is chosen such that $t < k$ and if $\mu_i \in \mathcal{E}_t$, then $\bar{\mu}_i \in \mathcal{E}_t$ as well. We will show that given \mathcal{X}_t , we can use the methodology described in section 2 to find $-Re(\mu_{t+1})$, that is, that $-Re(\mu_{t+1})$ is the eigenvalue with smallest modulus of a certain $n^2 \times n^2$ eigenvalue problem with a Kronecker structure like that of (2.1), and it can be computed using Lyapunov inverse iteration.

LEMMA 4.1. *Assume all the eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$ lie in the left half of the complex plane. Then in the subset $\{\lambda_{i,j}\}_{i,j>t}$ of all the eigenvalues of (2.1), the one with smallest modulus is $-Re(\mu_{t+1})$.*

Proof. If μ_{t+1} is real, then $-Re(\mu_{t+1}) = \lambda_{t+1,t+1}$. If μ_{t+1} is not real, by assumptions **(a1')**, **(a2')** and the choice of t , $\mu_{t+2} = \bar{\mu}_{t+1}$, which implies that $-Re(\mu_{t+1}) = \lambda_{t+1,t+2} = \lambda_{t+2,t+1}$. The rest of the proof is very similar to that of Theorem 2.2. \square

Consequently, if we can formulate a problem with a Kronecker structure like that of (2.1) whose eigenvalues are $\{\lambda_{i,j}\}_{i,j>t}$, then $-Re(\mu_{t+1})$ can be computed by Lyapunov inverse iteration applied to this problem. We will show how such a problem can be concocted and establish some of its properties that are similar to those of (2.1).

Let Θ_t be the diagonal matrix whose diagonal elements are $\mu_1^{-1}, \mu_2^{-1}, \dots, \mu_t^{-1}$, so that $S\mathcal{X}_t = \mathcal{X}_t\Theta_t$. Since $\mathbf{A}x = \mu\mathbf{M}x$ has a complete set of eigenvectors, there exists an orthonormal matrix $Q_t \in \mathbb{R}^{n \times t}$ such that $\mathcal{X}_t = Q_t G_t$, where $G_t \in \mathbb{C}^{t \times t}$ is nonsingular. Let

$$\widehat{S} = (I - Q_t Q_t^T) S, \quad \widehat{\Delta}_1 = \widehat{S} \otimes I + I \otimes \widehat{S}, \quad \text{and} \quad \widehat{\Delta}_0 = 2\widehat{S} \otimes \widehat{S}.$$

We claim that the distance between μ_{t+1} and the imaginary axis, $-Re(\mu_{t+1})$, is the eigenvalue of

$$\widehat{\Delta}_1 z = \lambda \left(-\widehat{\Delta}_0 \right) z, \quad z \in \text{Range} \left(\widehat{\Delta}_0 \right) \tag{4.1}$$

with smallest modulus. To prove this claim, we first study the eigenpairs of \widehat{S} .

LEMMA 4.2. *The matrix $I - Q_t Q_t^T$ where Q_t is defined above and $I \in \mathbb{R}^{n \times n}$ is the identity matrix has the following properties:*

1. $(I - Q_t Q_t^T) Q_t = \mathbf{0}$;
2. $(I - Q_t Q_t^T)^i = (I - Q_t Q_t^T)$ for any integer $i \geq 1$;
3. $(I - Q_t Q_t^T)^i S (I - Q_t Q_t^T)^j = (I - Q_t Q_t^T) S$ for any integers $i, j \geq 1$.

Proof. The first two properties hold for any orthonormal matrix and the proof is omitted here. To prove the third property, we first show that $(I - Q_t Q_t^T) S (I - Q_t Q_t^T) = (I - Q_t Q_t^T) S$. Since

$S\mathcal{X}_t = \mathcal{X}_t\Theta_t$ and $\mathcal{X}_t = Q_tG_t$, $SQ_tQ_t^T = Q_tG_t\Theta_tG_t^{-1}Q_t^T$ (G_t is invertible). Thus,

$$\begin{aligned} (I - Q_tQ_t^T)S(I - Q_tQ_t^T) &= (I - Q_tQ_t^T)S - (I - Q_tQ_t^T)SQ_tQ_t^T \\ &= (I - Q_tQ_t^T)S - (I - Q_tQ_t^T)Q_tG_t\Theta_tG_t^{-1}Q_t^T \\ &= (I - Q_tQ_t^T)S \end{aligned}$$

by the first property. This together with the second property establishes the third property. \square

LEMMA 4.3. Let $\hat{\theta}_i = 0$ for $i \leq t$ and $\hat{\theta}_i = \frac{1}{\mu_i}$ for $i > t$. Let $\hat{x}_i = x_i$ for $i \leq t$ and $\hat{x}_i = (I - Q_tQ_t^T)x_i$ for $i > t$. Then $(\hat{\theta}_i, \hat{x}_i)$ ($i = 1, 2, \dots, n$) are the eigenpairs of \hat{S} .

Proof. Let g_i be the i^{th} column of G_t . If $i \leq t$, $x_i = Q_tg_i$, thus

$$\hat{S}x_i = (I - Q_tQ_t^T)SQ_tg_i = (I - Q_tQ_t^T)Q_tG_t\Theta_tG_t^{-1}g_i = \mathbf{0}$$

by the first property in Lemma 4.2. If $i > t$,

$$\hat{S}(I - Q_tQ_t^T)x_i = (I - Q_tQ_t^T)S(I - Q_tQ_t^T)x_i = (I - Q_tQ_t^T)Sx_i = \frac{1}{\mu_i}(I - Q_tQ_t^T)x_i$$

by the third property in Lemma 4.2. \square

Knowing the eigenpairs of \hat{S} , we can find the eigenpairs of $\hat{\Delta}_0$ and $\hat{\Delta}_1$ with no difficulty.

LEMMA 4.4. The eigenvalues of $\hat{\Delta}_1$ are

1. 0, if $i, j \leq t$;
2. $\frac{1}{\mu_i}$, if $i > t$ and $j \leq t$;
3. $\frac{1}{\mu_j}$, if $i \leq t$ and $j > t$;
4. $\frac{1}{\mu_i} + \frac{1}{\mu_j}$, if $i, j > t$.

The eigenvalues of $\hat{\Delta}_0$ are

1. 0, if $i \leq t$ or $j \leq t$;
2. $\frac{2}{\mu_i\mu_j}$, if $i, j > t$.

Moreover, for each eigenvalue of $\hat{\Delta}_0$ or $\hat{\Delta}_1$, there are eigenvectors associated with it given by $\hat{z}_{i,j} = \hat{x}_i \otimes \hat{x}_j$ and $\hat{z}_{j,i} = \hat{x}_j \otimes \hat{x}_i$.

Proof. See the proof of Theorem 2.1. \square

Under the assumption that $\mathbf{A}x = \mu\mathbf{M}x$ has a complete set of eigenvectors, $\hat{\Delta}_0$ also has a complete set of eigenvectors $\{\hat{z}_{i,j}\}_{i,j=1}^n$. By Lemma 4.4, $\text{Range}(\hat{\Delta}_0) = \text{span}\{\hat{z}_{i,j}\}_{i,j>t}$.

THEOREM 4.5. The eigenvalues of (4.1) are $\{\lambda_{i,j}\}_{i,j>t}$. For any $\lambda_{i,j}$ with $i, j > t$, there are eigenvectors $\hat{z}_{i,j}$ and $\hat{z}_{j,i}$ associated with it.

Proof. The proof follows immediately from Lemma 4.4 and the proof of Theorem 2.1. \square

THEOREM 4.6. Assume all the eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$ lie in the left half of the complex plane. Then the eigenvalue of (4.1) with smallest modulus is $-Re(\mu_{t+1})$.

Proof. By Theorem 4.5, it suffices to show that $|Re(\mu_{t+1})| \leq |\lambda_{i,j}|$ for any $i, j > t$, which is true by Lemma 4.1. \square

If we can restrict the search space of eigenvectors to $\text{Range}(\hat{\Delta}_0)$, we can apply inverse iteration to $\hat{\Delta}_1 z = \lambda(-\hat{\Delta}_0)z$ to compute $-Re(\mu_{t+1})$. Let

$$\mathbf{P}_t = \{Z \in \mathbb{C}^{n \times n} | Z = (I - Q_tQ_t^T)X(I - Q_tQ_t^T) \text{ where } X \in \mathbb{C}^{n \times n}\}.$$

Since

$$\text{Range}(\widehat{\Delta}_0) = \text{span} \{\widehat{x}_i \otimes \widehat{x}_j\}_{i,j>t} = \text{span} \{(I - Q_t Q_t^T) x_i \otimes (I - Q_t Q_t^T) x_j\}_{i,j>t},$$

if $Z \in \mathbb{P}_t$, then $z = \text{vec}(Z) \in \text{Range}(\widehat{\Delta}_0)$, and *vice versa*. Therefore, (4.1) can be rewritten in the form of a matrix equation,

$$\widehat{S}Z + Z\widehat{S}^T + \lambda(2\widehat{S}Z\widehat{S}^T) = 0, \quad Z \in \mathbb{P}_t. \quad (4.2)$$

By Theorem 4.6, $-Re(\mu_{t+1})$ is the eigenvalue of (4.2) with smallest modulus. As in section 2, under certain conditions, we can show that $-Re(\mu_{t+1})$ is an eigenvalue of (4.2) with a unique, real, symmetric and low-rank eigenvector. Let $\mathbb{P}_t^s = \{Z \in \mathbb{P}_t | Z = Z^T\}$ be the subspace of \mathbb{P}_t consisting of symmetric matrices. As a result of assumptions **(a1')** and **(a2')**, when the eigenspace of (4.2) is restricted to \mathbb{P}_t^s , $-Re(\mu_{t+1})$ is an eigenvalue of (4.2) that has the unique (up to a scalar multiplier), real and symmetric eigenvector

$$(I - Q_t Q_t^T) x_{t+1} x_{t+1}^T (I - Q_t Q_t^T) \text{ or } (I - Q_t Q_t^T) (x_{t+1} x_{t+1}^* + \overline{x_{t+1}} x_{t+1}^T) (I - Q_t Q_t^T).$$

Therefore, if we can restrict the search space for the target eigenvector of (4.2) to \mathbb{P}_t^s , Lyapunov inverse iteration can be applied to (4.2) to compute $-Re(\mu_{t+1})$. Moreover, the analysis of section 3.4 (which applies to (2.2)) can be generalized directly to (4.2). This means that to compute $-Re(\mu_{t+1})$, it suffices to find an accurate solution to

$$\widehat{S}\mathcal{Y}_1 + \mathcal{Y}_1\widehat{S}^T = -2\widehat{S}Z_1\widehat{S}^T \quad (4.3)$$

in \mathbb{P}_t^s . In general, solutions to (4.3) are not unique: any matrix of the form $\mathcal{Y}_1 + Q_t X Q_t^T$ where $X \in \mathbb{C}^{n \times n}$ is also a solution, since $\widehat{S}Q_t = \mathbf{0}$ by Lemma 4.3. However, in the designated search space \mathbb{P}_t^s , the solution to (4.3) is indeed unique. In addition, we can obtain estimates for the eigenpair $(\mu_{t+1}, \widehat{x}_{t+1})$ of \widehat{S} in the same way we compute estimates for (μ_1, x_1) in section 2 (see (2.5)).

The analysis above leads to the following algorithm for computing k rightmost eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$:

Algorithm 3 (compute k rightmost eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$)

1. Initialization: $t = 0$, $\mathcal{E}_t = \emptyset$, $\mathcal{X}_t = \emptyset$, $Q_t = \mathbf{0}$, and $\widehat{S} = (I - Q_t Q_t^T) S$.
2. While $t < k$:
 - 2.1. Solve (4.2) for the eigenvalue with smallest modulus, $-Re(\mu_{t+1})$, and its corresponding eigenvector in \mathbb{P}_t^s .
 - 2.2. Compute an estimate $(\mu_{t+1}^{approx}, \widehat{x}_{t+1}^{approx})$ for $(\mu_{t+1}, \widehat{x}_{t+1})$.
 - 2.3. Update:
 - if μ_{t+1}^{approx} is real:
$$\mathcal{E}_{t+1} \leftarrow \{\mathcal{E}_t, \mu_{t+1}^{approx}\}, \quad \widehat{\mathcal{X}}_{t+1} \leftarrow [\widehat{\mathcal{X}}_t, \widehat{x}_{t+1}^{approx}], \quad t \leftarrow t + 1;$$
 - else:
$$\mathcal{E}_{t+2} \leftarrow \{\mathcal{E}_t, \mu_{t+1}^{approx}, conj(\mu_{t+1}^{approx})\}, \quad \widehat{\mathcal{X}}_{t+2} \leftarrow [\widehat{\mathcal{X}}_t, \widehat{x}_{t+1}^{approx}, conj(\widehat{x}_{t+1}^{approx})],$$

$$t \leftarrow t + 2.$$
 - 2.4. Compute the thin QR factorization of $\widehat{\mathcal{X}}_t$: $[Q, R] = qr(\widehat{\mathcal{X}}_t, 0)$, and let $Q_t = Q$, $\widehat{S} = (I - Q_t Q_t^T) S$.

At each iteration of Algorithm 3, we compute the $(t+1)^{st}$ rightmost eigenvalue μ_{t+1} , or the $(t+1)^{st}$ and $(t+2)^{nd}$ rightmost eigenvalues $(\mu_{t+1}, \overline{\mu_{t+1}})$. The iteration terminates when k rightmost eigenvalues have been found. In this algorithm, we need to compute the eigenvalue with smallest modulus for several Lyapunov eigenvalue problems (4.2) corresponding to different values of t . One way to do this is to simply apply Algorithm 2 to each of these problems. In the next section, we will discuss the way step 2.1 of Algorithm 3 is implemented, which is much more efficient. Note that the technique for computing $-Re(\mu_{t+1})$ ($t > 0$) introduced in this section is based on the assumption that Q_t , whose columns form an orthonormal basis for $\{x_j\}_{j=1}^t$, is given. In Algorithm 3, Q_t is taken to be a matrix whose columns form an orthonormal basis for the columns of $\widehat{\mathcal{X}}_t$. Such an approach is justified in the next section as well.

We apply this algorithm to compute a few rightmost eigenvalues for some cases of the examples considered in section 3. The results for step 2.1 in each iteration of Algorithm 3 are reported in Table 4.2 (see Table 4.1 for notation). For example, consider the driven-cavity flow at $Re = 7800$. From Table 4.2, we can find the eight rightmost eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$: $\mu_{1,2} = -0.00514 \pm 2.69845i$, $\mu_3 = -0.00845$, $\mu_{4,5} = -0.01531 \pm 0.91937i$, $\mu_{6,7} = -0.02163 \pm 1.78863i$, and $\mu_8 = -0.02996$ (see Figure 3.1a).

Table 4.1: Notation for Algorithm 3

Symbol	Definition
$\lambda^{(t)}$	the estimate of $-Re(\mu_{t+1})$, <i>i.e.</i> , the eigenvalue of (4.2) with smallest modulus
$Z^{(t)}$	the estimated eigenvector of (4.2) associated with $-Re(\mu_{t+1})$
μ_{t+1}^{approx}	the estimated $(t+1)^{st}$ rightmost eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$, <i>i.e.</i> , μ_{t+1}
\mathfrak{R}_t	$\widehat{S}Z^{(t)} + Z^{(t)}\widehat{S}^T + \lambda^{(t)}2\left(\widehat{S}Z^{(t)}\widehat{S}^T\right)$, the residual of the Lyapunov eigenvalue problem (4.3)

5. Implementation details of Algorithm 3. In the previous section, we proposed an algorithm that finds k rightmost eigenvalues of (1.1) by computing the eigenvalue with smallest modulus for a series of Lyapunov eigenvalue problems (4.2) corresponding to different values of t . In this section, more details of how to implement this algorithm efficiently will be discussed.

5.1. Efficient solution of the Lyapunov eigenvalue problems. We first make a preliminary observation.

PROPOSITION 5.1. *The unique solution to (4.3) in \mathbb{P}_t^s is $\mathcal{Y}_1 = (I - Q_t Q_t^T) Y_1 (I - Q_t Q_t^T)$, where Y_1 is the solution to (3.1).*

Proof. The proof is straightforward with the help of Lemma 4.2. \square

By Proposition 5.1, if we know the solution Y_1 to (3.1), we can formally write down the unique solution \mathcal{Y}_1 to (4.3) in \mathbb{P}_t^s . In practice, we do not know Y_1 ; instead, we solve (3.1) using an iterative solver (such as RKSM), which produces an approximate solution Y_1^{approx} . The following proposition shows that we can obtain from Y_1^{approx} an approximate solution to (4.3) that is essentially as accurate a solution to (4.3) as Y_1^{approx} is to (3.1).

PROPOSITION 5.2. *Let $\mathcal{Y}_1^{approx} = (I - Q_t Q_t^T) Y_1^{approx} (I - Q_t Q_t^T)$ and $\mathcal{R}_1 = \widehat{S}\mathcal{Y}_1^{approx} + \mathcal{Y}_1^{approx}\widehat{S}^T + 2\widehat{S}Z_1\widehat{S}^T$. Then $\|\mathcal{R}_1\|_F \leq 4\|R_1\|_F$, where $R_1 = SY_1^{approx} + Y_1^{approx}S^T + 2SZ_1S^T$.*

Table 4.2: Algorithm 3 applied to Examples 1, 2 and 3

t	$\lambda^{(t)}$	μ_{t+1}^{approx}	$\ \mathfrak{R}_t\ _F$	t	$\lambda^{(t)}$	μ_{t+1}^{approx}	$\ \mathfrak{R}_t\ _F$
Example 1 (Re=6000), $k = 8$				Example 1 (Re=7800), $k = 8$			
0	0.01084	-0.01084	7.11628e-10	0	0.00514	-0.00514+2.69845i	3.62567e-11
1	0.02006	-0.02006+0.91945i	5.31308e-11	2	0.00845	-0.00845	1.92675e-09
3	0.03033	-0.03033+1.79660i	1.57820e-11	3	0.01531	-0.01531+0.91937i	1.06201e-10
5	0.03794	-0.03794	2.27041e-10	5	0.02163	-0.02163+1.78863i	6.81321e-11
6	0.04418	-0.04418+2.69609i	4.50346e-11	7	0.02996	-0.02996	1.86935e-10
Example 2 (Re=300), $k = 6$				Example 2 (Re=350), $k = 6$			
0	0.10405	-0.10405+2.22643i	6.10287e-09	0	0.02411	-0.02411+2.24736i	7.16343e-09
2	0.32397	-0.32397	2.83185e-10	2	0.28408	-0.28408	1.46365e-10
3	0.39197	-0.39197	3.34178e-11	3	0.33571	-0.33571	3.81057e-11
4	0.60628	-0.60628	1.31394e-07	4	0.56485	-0.56485	6.03926e-08
5	0.87203	-0.87203	1.77364e-06	5	0.79196	-0.79196	9.20079e-07
Example 3, $k = 6$							
0	0.05000	-0.05000+25.0000i	6.88230e-09				
2	0.10000	-0.10000	1.50946e-12				
3	0.20000	-0.20000	8.60934e-09				
4	0.30000	-0.30000	1.31349e-07				
5	0.40000	-0.40000	4.08853e-06				

Proof. Using Lemma 4.2, we can show easily that $\mathcal{R}_1 = (I - Q_t Q_t^T) R_1 (I - Q_t Q_t^T)$. Therefore,

$$\|\mathcal{R}_1\|_F \leq \|I - Q_t Q_t^T\|_F^2 \|R_1\|_F \leq (1 + \|Q_t\|_F^2)^2 \|R_1\|_F = 4\|R_1\|_F.$$

□

This analysis suggests the following strategy for step 2.1 of Algorithm 3: when $t = 0$, we compute $-Re(\mu_1)$ by applying Algorithm 2 to (2.2), in which a good approximate solution \mathcal{Y}_1^{approx} to (3.1) is computed; in any subsequent iteration where $0 < t \leq k-1$, instead of applying Lyapunov inverse iteration again to (4.2), we simply get the approximate solution \mathcal{Y}_1^{approx} to (4.3) specified in Proposition 5.2, from which $-Re(\mu_{t+1})$ can be computed. Details of this approach are described below.

Recall that \mathcal{Y}_1^{approx} computed by an iterative solver such as RKSM is of the form $V_1 D_1 V_1^T$, where $V_1 \in \mathbb{R}^{n \times d_1}$ is orthonormal and $d_1 = \text{rank}(V_1) \ll n$. We first rewrite \mathcal{Y}_1^{approx} in a similar form $U(\Sigma W^T D_1 W \Sigma) U^T$, where $U \Sigma W^T$ is the ‘thin’ singular value decomposition (SVD) of $(I - Q_t Q_t^T) V_1$. Then we can compute an estimate for $-Re(\mu_{t+1})$ in the same way we compute estimates for $-Re(\mu_1)$ in Algorithm 1 or 2. That is, we solve the small, projected Lyapunov eigenvalue problem

$$\tilde{S}_h \tilde{Z} + \tilde{Z} \tilde{S}_h^T + \tilde{\lambda} (2\tilde{S}_h \tilde{Z} \tilde{S}_h^T) = 0 \quad (5.1)$$

for its eigenvalue with smallest modulus, where $\tilde{S}_h = U^T \hat{S} U$. Recall that the matrix $\tilde{S} = V_1^T S V_1$ in (2.3) can be obtained with no additional cost from the Arnoldi decomposition (for instance, (2.8)).

Here, \tilde{S}_h can be computed cheaply as well since

$$U^T \hat{S} U = \Sigma^{-1} W^T V_1^T (I - Q_t Q_t^T) S V_1 W \Sigma^{-1}$$

by Lemma 4.2, and $S V_1$ is given by the same Arnoldi decomposition. Let $\tilde{\lambda}_1$ be the eigenvalue with smallest modulus of (5.1), and $\tilde{Z}_1 = \tilde{V} \tilde{D} \tilde{V}^T$ be the eigenvector associated with $\tilde{\lambda}_1$. Then the estimated $-Re(\mu_{t+1})$ and eigenvector of (4.2) associated with it are $\lambda^{(t)} = \tilde{\lambda}_1$ and $Z^{(t)} = \mathcal{V}_t \tilde{D} \mathcal{V}_t^T$, where $\mathcal{V}_t = U \tilde{V}$. In addition, by solving the small eigenvalue problem $(\mathcal{V}_t^T \hat{S} \mathcal{V}_t) y = \theta y$, we get an estimate $\mu_{t+1}^{approx} = \frac{1}{\theta}$ for μ_{t+1} and $\hat{x}_{t+1}^{approx} = \mathcal{V}_t y$ for \hat{x}_{t+1} .

5.2. Efficient computation of the matrix Q_t . At each iteration of Algorithm 3, we need an orthonormal basis for $\{x_j\}_{j=1}^t$, the eigenvectors associated with the t rightmost eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$. When $t = 0$, we can get estimates for x_1 from Lyapunov inverse iteration applied to (4.2); however, when $t > 0$, we are only able to get estimates for the eigenvectors of the deflated matrix \hat{S} . We will discuss how an orthonormal basis for $\{x_j\}_{j=1}^t$ can be computed efficiently from these estimates.

We first consider the simplest case where all k rightmost eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$ are real. In this case, we have the following result.

PROPOSITION 5.3. *For any t such that $1 \leq t \leq k$,*

$$\text{span} \left\{ (I - Q_{j-1} Q_{j-1}^T) x_j \right\}_{j=1}^t = \text{span} \{x_j\}_{j=1}^t. \quad (5.2)$$

Proof. We argue by induction.

1. When $t = 1$, since $Q_0 = \mathbf{0}$,

$$\text{span} \left\{ (I - Q_0 Q_0^T) x_1 \right\} = \text{span} \{x_1\}.$$

The claim is trivially true.

2. When $t = 2$, since $x_2 = (I - Q_1 Q_1^T) x_2 + Q_1 \alpha_1$ where $\alpha_1 \in \mathbb{C}$,

$$\text{span} \left\{ (I - Q_0 Q_0^T) x_1, (I - Q_1 Q_1^T) x_2 \right\} = \text{span} \{x_1, x_2 - Q_1 \alpha_1\} = \text{span} \{x_1, x_2\}.$$

3. Assume the claim is true for any t that satisfies $3 \leq t \leq k - 1$. Now we want to show that it is true for $t + 1$. Note that $x_{t+1} = (I - Q_t Q_t^T) x_{t+1} + Q_t \alpha_t$, where $\alpha_t \in \mathbb{C}^t$. Then by the induction hypothesis,

$$\text{span} \left\{ (I - Q_{j-1} Q_{j-1}^T) x_j \right\}_{j=1}^{t+1} = \text{span} \{x_1, x_2, \dots, x_t, x_{t+1} - Q_t \alpha_t\} = \text{span} \{x_j\}_{j=1}^{t+1}.$$

□

Consequently, if we can find an orthonormal basis for $\{(I - Q_{j-1} Q_{j-1}^T) x_j\}_{j=1}^t$, then this is an orthonormal basis for $\{x_j\}_{j=1}^t$ as well. In Algorithm 3, the j^{th} column of the matrix $\hat{\mathcal{X}}_t$ is approximately $(I - Q_{j-1} Q_{j-1}^T) x_j$; therefore, Q_t can be approximated by computing the thin QR factorization of $\hat{\mathcal{X}}_t$.

As seen in Table 4.2, some of the k rightmost eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$ are not real. In this case, t may increase by 2 instead of 1 from one iteration to the next. Let $\mathcal{T} = \{t_i\}_{i=0}^s$ ($t_i < t_{i+1}$) be the collection of every value of t for which we need to solve the Lyapunov eigenvalue problem

(4.2). Then $t_0 = 0$, $t_s = k - 1$ or $k - 2$, and $t_{i+1} - t_i = 1$ or 2 . For example, in the case of the cavity flow at $Re = 7800$, if $k = 8$, then $\mathcal{T} = \{0, 2, 3, 5, 7\}$. In the same way we prove Proposition 5.2, we can show that for any $1 \leq i \leq s$,

$$\text{span} \left\{ \left(I - Q_{t_{j-1}} Q_{t_{j-1}}^T \right) x_{t_j}, \left(I - Q_{t_j} Q_{t_j}^T \right) \overline{x_{t_j}} \right\}_{j=1}^i = \text{span} \{ x_{t_j}, \overline{x_{t_j}} \}_{j=1}^i. \quad (5.3)$$

In Algorithm 3, if μ_{t_j} is real, then the t_j^{th} column of $\widehat{\mathcal{X}}_{t_i}$ is approximately $\left(I - Q_{t_{j-1}} Q_{t_{j-1}}^T \right) x_{t_j}$; otherwise, the t_j^{th} and $(t_j - 1)^{\text{st}}$ columns of $\widehat{\mathcal{X}}_{t_i}$ hold estimates for $\left(I - Q_{t_{j-1}} Q_{t_{j-1}}^T \right) x_{t_j}$ and $\left(I - Q_{t_j} Q_{t_j}^T \right) \overline{x_{t_j}}$. By (5.3), Q_{t_i} can be approximated by computing the thin QR factorization of $\widehat{\mathcal{X}}_{t_i}$.

6. Conclusion. In this paper, we have developed a robust and efficient method of computing a few rightmost eigenvalues of (1.1) at any point (\bar{u}_0, α_0) in the stable regime. We have shown that the distance between the rightmost eigenvalue of (1.1) and the imaginary axis is the eigenvalue with smallest modulus of an $n^2 \times n^2$ eigenvalue problem (2.1). Since (2.1) has the same Kronecker structure as the one considered in previous work [8, 15], this distance can be computed by the Lyapunov inverse iteration developed and studied in these references, which also produces estimates of the rightmost eigenvalue(s) as by-products. An analysis of the fast convergence of Lyapunov inverse iteration in this particular application is given, which indicates that the algorithm will converge in two steps as long as the first Lyapunov equation is solved accurately enough. Furthermore, assuming t rightmost eigenpairs are known, we show that all the main theoretical results proven for (2.1) can be generalized to the deflated problem (4.1), whose eigenvalue with smallest modulus is the distance between the $(t + 1)^{\text{st}}$ rightmost eigenvalue and the imaginary axis. Finally, an algorithm that computes a few rightmost eigenvalues of (1.1) is proposed, followed by a discussion on how to implement it efficiently. The method developed in this study together with the method proposed in [8, 15] constitute a robust way of detecting the transition to instability in the steady-state solution of a large-scale dynamical system.

References.

- [1] A. C. Antoulas, D. C. Sorensen, and Y. Zhou, *On the decay of Hankel singular values and related issues*, Technical Report 01-09, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2001. Available from http://www.caam.rice.edu/~sorensen/Tech_Reports.html.
- [2] R. H. Bartels and G. W. Stewart, *Algorithm 432: solution of the matrix equation $AX + XB = C$* , Comm. of the ACM, **15** (1972), pp. 820–826.
- [3] K. A. Cliffe, T. J. Garratt, and A. Spence, *Calculation of eigenvalues of the discretised Navier-Stokes and related equations*, The Mathematics of Finite Elements and Applications VII MAFELAP, 1990, pp. 479–486.
- [4] ———, *Eigenvalues of block matrices arising from problems in fluid mechanics*, SIAM J. Matrix Anal. Appl., **15** (1994), pp. 1310–1318.
- [5] K. A. Cliffe and K. H. Winters, *Convergence properties of the finite-element method for Benard convection in an infinite layer*, J. Comput. Phys, **60** (1985), pp. 346–351.
- [6] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., **49** (2011), pp. 1875–1898.
- [7] V. Druskin and V. Simoncini, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems & Control Letters, **60** (2011), pp. 546–560.
- [8] H. Elman, K. Meerbergen, A. Spence, and M. Wu, *Lyapunov inverse iteration for identifying Hopf bifurcations in models of incompressible flow*, University of Maryland Department of Computer Science TR-4975 and University of Maryland Institute for Advanced Computer Studies TR-2011-04, 2011. Available from <http://www.cs.umd.edu/~elman/papers/lyap-inverse-tr.pdf>. To appear in SIAM J. Sci. Comput.
- [9] A. Fortin, M. Jardak, and J. Gervais, *Localization of Hopf bifurcation in fluid flow problems*, Int. J. Numer. Methods Fluids, **24** (1997), pp. 1185–1210.

- [10] T. J. Garratt, *The numerical detection of Hopf bifurcations in large systems arising in fluid mechanics*, Ph.D. Thesis, University of Bath, UK, 1991.
- [11] W. J. F. Govaerts, *Numerical Methods for Bifurcations of Dynamical Equilibria*, SIAM, Philadelphia, 2000.
- [12] S. J. Hammarling, *Numerical solution of the stable, non-negative definite Lyapunov equation*, IMA J. Numerical Anal., **2** (1982), pp. 303–323.
- [13] I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., **31** (1994), pp. 227–251.
- [14] K. Meerbergen and D. Roose, *Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems*, IMA J. Numer. Anal., **16** (1996), pp. 297–346.
- [15] K. Meerbergen and A. Spence, *Inverse iteration for purely imaginary eigenvalues with application to the detection of Hopf bifurcation in large scale problems*, SIAM J. Matrix Anal. Appl., **31** (2010), pp. 1982–1999.
- [16] T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett., **40** (2000), pp. 139C144.
- [17] Y. Saad, *Numerical solution of large Lyapunov equations*, Signal Processing, Scattering, Operator Theory, and Numerical Methods. Proceedings of the International Symposium MTN-89, vol III, 1990, pp. 503–511.
- [18] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., **29** (2007), pp. 1268–1288.
- [19] D. C. Sorensen, *Implicit application of polynomial filters in a k -step Arnoldi method*, SIAM J. Matrix Anal. Appl., **13** (1992), pp. 357–385.
- [20] G. W. Stewart, *Matrix Algorithms Volume II: Eigensystems*, SIAM, Philadelphia, 2001.
- [21] J. S. Turner, *Buoyancy Effects in Fluids*, Cambridge University Press, Cambridge, 1973.