SIAM J. Matrix Anal. Appl.
Vol. 34, No. 4, pp. 1685–1707

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. An analysis that explains the fast convergence of this algorithm observed in numerical experiments is provided, based on which we propose a more efficient and robust algorithm. Furthermore, we generalize the same idea 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.

Key words. linear stability analysis, inverse iteration, Lyapunov solvers

AMS subject classifications. 65F15, 65F18, 65M22, 37M20

DOI. 10.1137/120897468

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 $\frac{\partial f}{\partial u}(\overline{u}(\alpha), \alpha) = \frac{\partial f}{\partial u}(\alpha)$, where $\overline{u}(\alpha)$ is the steady-state solution to (1.2) at α, i.e., $f(\overline{u}, \alpha) = 0$. Let the solution path be the following set: $\mathcal{S} = \{(\overline{u}, \alpha) | f(\overline{u}, \alpha) = 0\}$. We seek the critical point $(\overline{u}_c, \alpha_c)$ 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 systems arising from spatial discretization of two- or three-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 $(\overline{u}, \alpha)$ approaches the critical point, the rightmost eigenvalue

[*] Received by the editors November 1, 2012; accepted for publication (in revised form) by V. Simoncini September 27, 2013; published electronically December 17, 2013. This work was supported in part by the U.S. Department of Energy under grants DEFG0204ER25619 and DE-SC0009301 and by the U.S. National Science Foundation under grant DMS1115317.

http://www.siam.org/journals/simax/34-4/89746.html

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

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

of (1.1) moves toward the imaginary axis; at $(\overline{u}_c, \alpha_c)$, 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 $(\overline{u}_0, \alpha_0)$ 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 solvers such as the QR and QZ algorithms (see [23]) compute all the eigenvalues of (1.1), but they are too expensive for large n. Existing iterative eigenvalue solvers [23] 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 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 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 [17] proposed the *Lyapunov inverse iteration* method, which estimates the critical parameter α_c without computing the rightmost eigenvalues of (1.1). Assume $(\overline{u}_0, \alpha_0)$ is in the stable regime of $\mathcal{S}$ and is also in the neighborhood of the critical point $(\overline{u}_c, \alpha_c)$. 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 [17] that λ_c is the eigenvalue with smallest modulus of the eigenvalue problem

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

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 [17] 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, 17], 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 $(\overline{u}_c, \alpha_c)$. 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 $(\overline{u}_0, \alpha_0)$ close to $(\overline{u}_c, \alpha_c)$ and apply Lyapunov inverse iteration with confidence. In reality, α_c is unknown and we start from a point $(\overline{u}_0, \alpha_0)$ in the stable regime of $\mathcal{S}$ that may be distant from the critical point. In this scenario, the method of [8, 17] 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 $(\overline{u}_0, \alpha_0)$ is from $(\overline{u}_c, \alpha_c)$ 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 from the imaginary axis, then it is reasonable to assume that $(\overline{u}_0, \alpha_0)$ is far from the critical point as well, and therefore we should march along $\mathcal{S}$ using numerical continuation until we are close enough to $(\overline{u}_c, \alpha_c)$; otherwise, we can assume that $(\overline{u}, \alpha_0)$ is already in the neighborhood of the critical point and the method of [8, 17] 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, which demonstrate the fast convergence of this method. In section 4, we give an analysis that provides insight into the fast convergence. In addition, based on this analysis, we propose a more efficient version of Lyapunov inverse iteration and a way of validating its results. In section 5, we show that the analysis in sections 2 and 4 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). Finally, we make some concluding remarks in section 6.

2. Computing the distance between the rightmost eigenvalue(s) and the imaginary axis.

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

$$\Delta_1 z = \lambda \Delta_0 z, \tag{2.1}$$

where $\Delta_1 = S \otimes I_n + I_n \otimes S$ and $\Delta_0 = -2S \otimes S$. (I_n is the identity matrix of order n.) This result has been proved in [18] and we include a proof here for completeness. We proceed in two steps to prove this assertion. First, we show that $-\mathrm{Re}(\mu_1)$ is an eigenvalue of (2.1).

THEOREM 2.1. *Assume* $\mathbf{M}$ *is nonsingular. The eigenvalues of* (2.1) *are* $\lambda_{i,j} = -\frac{1}{2}(\mu_i + \mu_j)$, $i, j = 1, 2, \ldots, n$. *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. 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 $S^{-1} = \mathbf{M}^{-1}\mathbf{A}$ and P be an invertible matrix such that $S^{-1} = PJP^{-1}$. Then

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

This implies that (2.1) and $-\frac{1}{2}\left(J \otimes I_n + I_n \otimes J\right)$ have the same eigenvalues. Due to the special structure of the Jordan normal form J, $J \otimes I_n + I_n \otimes J$ is an upper triangular matrix whose diagonal entries are $\{\mu_i + \mu_j\}_{i,j=1}^n$. Consequently, the eigenvalues of $-\frac{1}{2}\left(J \otimes I_n + I_n \otimes J\right)$ are $\left\{-\frac{1}{2}(\mu_i + \mu_j)\right\}_{i,j=1}^n = \{\lambda_{i,j}\}_{i,j=1}^n$. 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 associated with the eigenvalue $\lambda_{i,j}$ of (2.1). For any pair (i,j) $(i,j = 1,2,\ldots,n)$,

$$\begin{aligned}\left(\Delta_0^{-1}\Delta_1\right) z_{i,j} &= -\frac{1}{2}\left(S^{-1} \otimes I_n + I_n \otimes S^{-1}\right)(x_i \otimes x_j)\\ &= -\frac{1}{2}\left(S^{-1}x_i \otimes x_j + x_i \otimes S^{-1}x_j\right)\\ &= -\frac{1}{2}\left(\mu_i x_i \otimes x_j + x_i \otimes \mu_j x_j\right) = \lambda_{i,j} z_{i,j}.\end{aligned}$$

Similarly, we can show that $\left(\Delta_0^{-1}\Delta_1\right) z_{j,i} = \lambda_{i,j} z_{j,i}$. □

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

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

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 with smallest modulus of* (2.1) *is* $-\mathrm{Re}(\mu_1)$.

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

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

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 $a_1 = a_2$, $b_1 = -b_2$, $-\mathrm{Re}(\mu_1) = \lambda_{1,2} = \lambda_{2,1}$, and similarly,

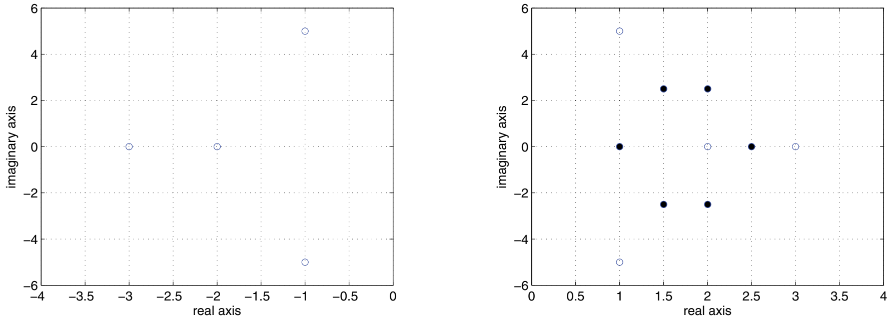
$$|\lambda_{1,2}|^2 = |\lambda_{2,1}|^2 = \frac{1}{4}\left[(a_1 + a_1)^2 + (b_1 - b_1)^2\right] \leq \frac{1}{4}\left[(a_i + a_j)^2 + (b_i + b_j)^2\right] = |\lambda_{i,j}|^2$$

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

In order to visualize Theorems 2.1 and 2.2, consider a 4 × 4 eigenvalue problem $\mathbf{A}x = \mu\mathbf{M}x$ whose eigenvalues are $\mu_{1,2} = -1 \pm 5i$, $\mu_3 = -2$ and $\mu_4 = -3$ (see Figure 2.1(a)). The eigenvalues of the corresponding 16 × 16 eigenvalue problem (2.1) are plotted in Figure 2.1(b), from which we can see that $\lambda_{1,2} = \lambda_{2,1} = 1$ is the eigenvalue with smallest modulus of (2.1).

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), $-\mathrm{Re}(\mu_1)$, 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, 17], 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

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



(a)

(b)

FIG. 2.1. *The spectra of* $\mathbf{A}x = \mu\mathbf{M}x$ *and* (2.1) *for the* 4×4 *example* (○: *simple eigenvalues;* ●: *double eigenvalues).*

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 $z = vec(Z)$. By Theorem 2.1 and the relation between (2.1) and (2.2), $(\lambda_{i,j}, Z_{i,j})$ $(i, j = 1, 2, \ldots, n)$ are the eigenpairs of (2.2), where $Z_{i,j} = x_j x_i^T$; in addition, by Theorem 2.2, $-\mathrm{Re}(\mu_1)$ is the eigenvalue of (2.2) with smallest modulus. Furthermore, under certain conditions, $-\mathrm{Re}(\mu_1)$ 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 $\mathrm{Re}(\mu_i) = \mathrm{Re}(\mu_1)$, then $\mu_i = \overline{\mu_1}$; (a2) μ_1 is a simple eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$. Consequently, if μ_1 is real, $-\mathrm{Re}(\mu_1)$ is a simple eigenvalue of (2.1) with the eigenvector $z_{1,1} = x_1 \otimes x_1$; otherwise, $-\mathrm{Re}(\mu_1)$ is a double eigenvalue of (2.1) with the eigenvectors $z_{1,2} = x_1 \otimes \overline{x_1}$ and $z_{2,1} = \overline{x_1} \otimes x_1$. 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 [17], $-\mathrm{Re}(\mu_1)$ has a unique (up to a scalar multiplier), real, and symmetric eigenvector $x_1 x_1^* + \overline{x_1} x_1^T$, where x_1^* denotes the conjugate transpose of x_1. Therefore, we can apply Lyapunov inverse iteration (see [8, 17]) to (2.2) to find $-\mathrm{Re}(\mu_1)$, the eigenvalue of (2.2) with smallest modulus:

```
ALGORITHM 1. Lyapunov inverse iteration for (2.2).
  1. Given $\mathcal{V}_0 \in \mathbb{R}^n$ with $\|\mathcal{V}_0\|_2 = 1$. Let $Z^{(0)} = \mathcal{V}_0 \mathcal{V}_0^T$.
  2. For $\ell = 1, 2, \ldots$
       2.1. Solve for $Y_\ell$ from
  (2.3)                 $SY_\ell + Y_\ell S^T = -2SZ^{(\ell-1)}S^T$
            in factored form: $Y_\ell = V_\ell D_\ell V_\ell^T$, where $V_\ell \in \mathbb{R}^{n \times d_\ell}$.
       2.2. Rank reduction: let $\widetilde{S} = V_\ell^T S V_\ell$ and solve
  (2.4)                 $\widetilde{S}\widetilde{Z} + \widetilde{Z}\widetilde{S}^T + \widetilde{\lambda}\left(2\widetilde{S}\widetilde{Z}\widetilde{S}^T\right) = 0$
            for the eigenvalue with smallest modulus, $\widetilde{\lambda}_1$, and its eigenvector
            $\widetilde{Z}_1 = \widetilde{\mathcal{V}}\widetilde{\mathcal{D}}\widetilde{\mathcal{V}}^T$.
       2.3. Set $\lambda^{(\ell)} = \widetilde{\lambda}_1$ and $Z^{(\ell)} = \mathcal{V}_\ell \widetilde{\mathcal{D}} \mathcal{V}_\ell^T$, where $\mathcal{V}_\ell = V_\ell \widetilde{\mathcal{V}}$.
       2.4. If $\left(\lambda^{(\ell)}, Z^{(\ell)}\right)$ has converged, then stop.
```

Compared to standard inverse iteration, Algorithm 1 has an extra step consisting of rank reduction. In this step, we project the eigenvalue problem (2.2) onto the subspace spanned by the columns of V_ℓ. This step guarantees that the eigenvector iterate $Z^{(\ell)}$ is always of rank 1 or 2, which is essential for the efficient solution of (2.3).

Our numerical experiments (see section 3) show that as long as (2.3) is solved accurately enough, the iterate $\left(\lambda^{(\ell)}, Z^{(\ell)} = \mathcal{V}_\ell \widetilde{D} \mathcal{V}_\ell^T\right)$ will converge quickly to the target eigenpair $\left(-\text{Re}(\mu_1), \mathcal{V}\mathcal{D}\mathcal{V}^T\right)$, 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 $\{x_1, \overline{x_1}\}$ (if μ_1 is not real). Besides estimates of $-\text{Re}(\mu_1)$, we can also obtain from Algorithm 1 estimates of (μ_1, x_1) by solving the small 1×1 or 2×2 eigenvalue problem

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

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

At each iteration of Algorithm 1, a large-scale *Lyapunov equation* (2.3) needs to be solved, where $Z^{(\ell-1)}$ is chosen to be of rank 1 when $\ell = 1$ and is of rank 1 or 2 when $\ell > 1$ due to the rank-reduction step. We can rewrite (2.3) as

$$SY_\ell + Y_\ell S^T = P_\ell C_\ell P_\ell^T \tag{2.6}$$

(see [8] for details), where P_ℓ is orthonormal and of rank 1 or 2. The solution to (2.6), Y_ℓ, is real and symmetric and frequently has low-rank approximation (see [1, 12, 15, 19]), i.e., $d_\ell \ll n$. Since S is large, direct methods such as [2, 13] 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 [14, 20], the extended Krylov subspace method (EKSM) [21], and the rational Krylov subspace method (RKSM) [6, 7], construct approximate solutions of the form $Y_\ell^{\text{approx}} = WXW^T$, where W is an orthonormal matrix whose columns span the Krylov subspace and X is the solution to the small, projected Lyapunov equation $(W^TSW)X + X(W^TSW)^T = (W^TP_\ell)C_\ell(W^TP_\ell)^T$, which can be obtained using direct methods. For example, the standard Krylov subspace method [14, 20] builds the mp-dimensional Krylov subspace

$$\mathcal{K}_m(S, P_\ell) = span\left\{P_\ell, SP_\ell, \ldots, S^{m-1}P_\ell\right\}, \tag{2.7}$$

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 $\widetilde{S} = V_\ell^T S V_\ell$ requires another d_ℓ linear solves with coefficient matrix $\mathbf{A}$, in fact, if a Krylov-type method is used to solve (2.6), $\widetilde{S}$ can be obtained from the *Arnoldi decomposition* computed by the Krylov-type method for no additional cost. Assume the standard Krylov method is used to solve (2.6). It computes the Arnoldi decomposition

$$SV_\ell = V_\ell H_m + W_{m+1} H_{m+1,m} E_m^T \tag{2.8}$$

and the approximate solution $Y_\ell^{\text{approx}} = V_\ell D_\ell V_\ell^T$, where the columns of $[V_\ell, W_{m+1}]$ form an orthonormal basis for the Krylov subspace $\mathcal{K}_{m+1}(S, P_\ell)$. (In addition,

$H_m \in \mathbb{R}^{mr\times mr}$ is block upper Hessenberg, $H_{m+1,m} \in \mathbb{R}^{r\times r}$, and E_m holds the last r columns of the identity matrix of order mr, where $r = 1$ or 2.) This implies that $\widetilde{S}$ is simply H_m in (2.8). The Arnoldi decomposition computed by EKSM or RKSM has a form that is more complicated than (2.8); nonetheless, it produces the matrix $\widetilde{S}$ as well. Moreover, since $d_\ell \ll n$, the small eigenvalue problem (2.4) can be solved by standard inverse iteration with a direct Lyapunov solver and no rank-reduction step.

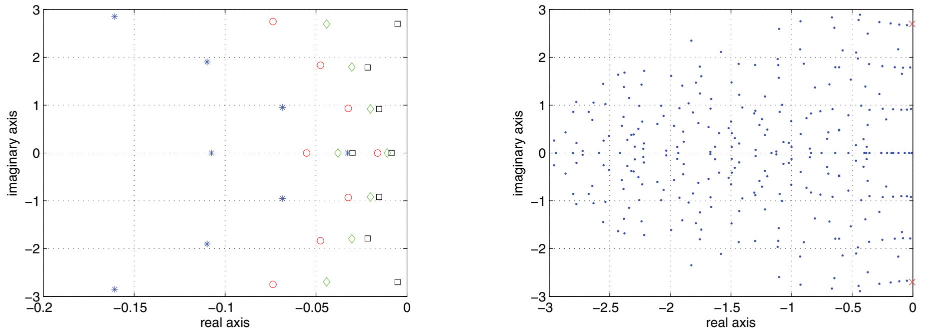
3. Numerical experiments.

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 imcompressible 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 driven-cavity 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 $\mathcal{R}$) of the flow. (The Reynolds number of this flow is defined to be $\mathcal{R} = \frac{1}{\nu}$, where ν is the *kinematic viscosity*.) Figure 3.1(a) depicts the path traced out by the eight rightmost eigenvalues of (1.1) for $\mathcal{R} = 2000, 4000, 6000, 7800$, 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 toward 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 toward the imaginary axis than the other eigenvalues and eventually they become the rightmost. They first cross the imaginary axis at $\mathcal{R} \approx 7929$, 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 $\mathcal{R} = 7800$) can be difficult. Suppose we are trying to find



(a)

(b)

FIG. 3.1. (a) *The eight rightmost eigenvalues for driven-cavity flow at different Reynolds numbers (*: $\mathcal{R} = 2000$; ○: $\mathcal{R} = 4000$; ◇: $\mathcal{R} = 6000$; □: $\mathcal{R} = 7800$).* (b) *The 300 eigenvalues with smallest modulus at $\mathcal{R} = 7800$ (×: the rightmost eigenvalues).*

the rightmost eigenvalues at $\mathcal{R} = 7800$ by conventional methods, such as computing k eigenvalues of (1.1) with smallest modulus using the *implicitly restarted Arnoldi* (IRA) method [22]. 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 3.1. (See Table 3.2 for notation.) The initial guess $\mathcal{V}_0$ is chosen to be a random vector of unit norm in $\mathbb{R}^n$, the stopping criterion for the eigenvalue residual is

$$\|R_\ell^{\text{eig}}\|_F < 10^{-8},$$

and the stopping criterion for the Lyapunov solve is

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

Note that both residual norms $\|R_\ell^{\text{eig}}\|_F$ and $\|R_\ell^{\text{lyap}}\|_F$ are cheap to compute (see [8] for details). Therefore, the main cost of each iteration is about d_ℓ linear solves of order n (see Chapter 5 of [25]). All linear systems are solved using direct methods. As shown in Table 3.1, the distances between the rightmost eigenvalue(s) of (1.1) and the imaginary axis at $\mathcal{R} = 2000, 4000, 6000, 7800$ are 0.03264, 0.01608, 0.01084,

TABLE 3.1
Algorithm 1 applied to Example 1 (Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
		$\mathcal{R} = 2000$			
1	0.03264	−0.03264	2.56263e-11	1.40794e-10	156
		$\mathcal{R} = 4000$			
1	0.01608	−0.01608	4.25055e-10	3.52618e-10	241
		$\mathcal{R} = 6000$			
1	0.01084	−0.01084	7.11628e-10	6.52387e-10	307
		$\mathcal{R} = 7800$			
1	0.00514	−0.00514+2.69845i	3.62567e-11	9.02875e-10	366

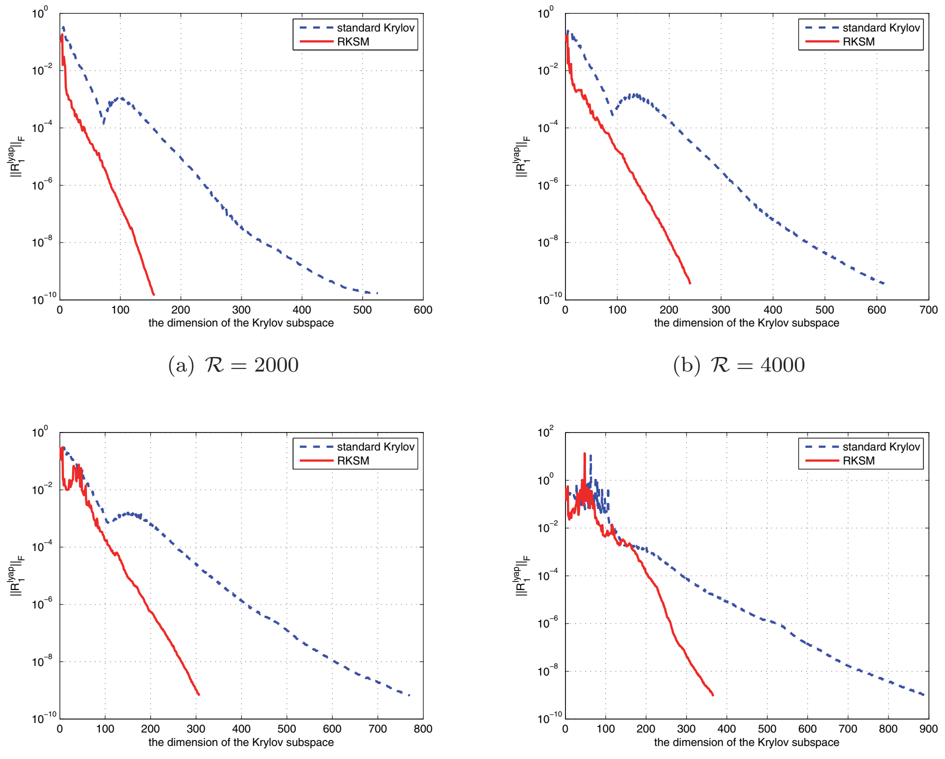
TABLE 3.2
Notation for Algorithm 1.

Symbol	Definition
$\lambda^{(\ell)}$	the estimate of $-\text{Re}(\mu_1)$, i.e., the eigenvalue of (2.2) with smallest modulus
$Z^{(\ell)}$	the estimated eigenvector of (2.2) associated with $-\text{Re}(\mu_1)$
$\mu^{(\ell)}$	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)
R_ℓ^{eig}	$SZ^{(\ell)} + Z^{(\ell)}S^T + \lambda^{(\ell)}(2SZ^{(\ell)}S^T)$, the residual of the Lyapunov eigenvalue problem (2.2)
R_ℓ^{lyap}	$SY_\ell^{\text{approx}} + Y_\ell^{\text{approx}}S^T - P_\ell C_\ell P_\ell^T$, the residual of the Lyapunov equation (2.6)
d_ℓ	dimension of the Krylov subspace, i.e., rank of Y_ℓ^{approx}

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 $-0.00514{+}2.69845\mathrm{i}$.

We note two trends in these results. First, surprisingly, for all the Reynolds numbers considered, Algorithm 1 converges to the desired tolerance ($\|R_\ell^{\mathrm{eig}}\|_F < 10^{-8}$) in only one iteration. That is, only the first Lyapunov equation (2.6) with $\ell = 1$ 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 ($\|R_\ell^{\mathrm{lyap}}\|_F < 10^{-9} \cdot \|C_\ell\|_F$), 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 [20] to solve the Lyapunov systems. To solve (2.6) 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.

3.2. Example 2: Flow over an obstacle. For linear stability analysis of flow over an obstacle, 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 3.3(a) depicts the path traced out by the six rightmost eigenvalues of (1.1) for $\mathcal{R} = 100, 200, 300, 350$ in the stable regime, and Figure 3.3(b) shows the 300 eigenvalues of (1.1) with smallest modulus at $\mathcal{R} = 350$. (In this



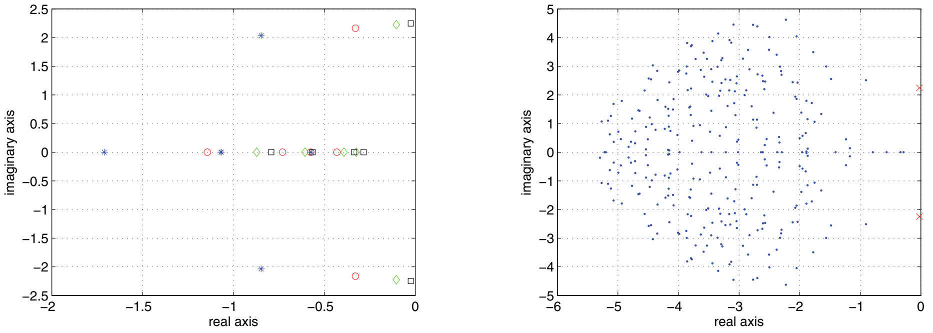
(a) $\mathcal{R} = 2000$

(b) $\mathcal{R} = 4000$

(c) $\mathcal{R} = 6000$

(d) $\mathcal{R} = 7800$

FIG. 3.2. *Comparison of the standard Krylov method and RKSM for solving* (2.6) *in Example* 1.



(a) (b)

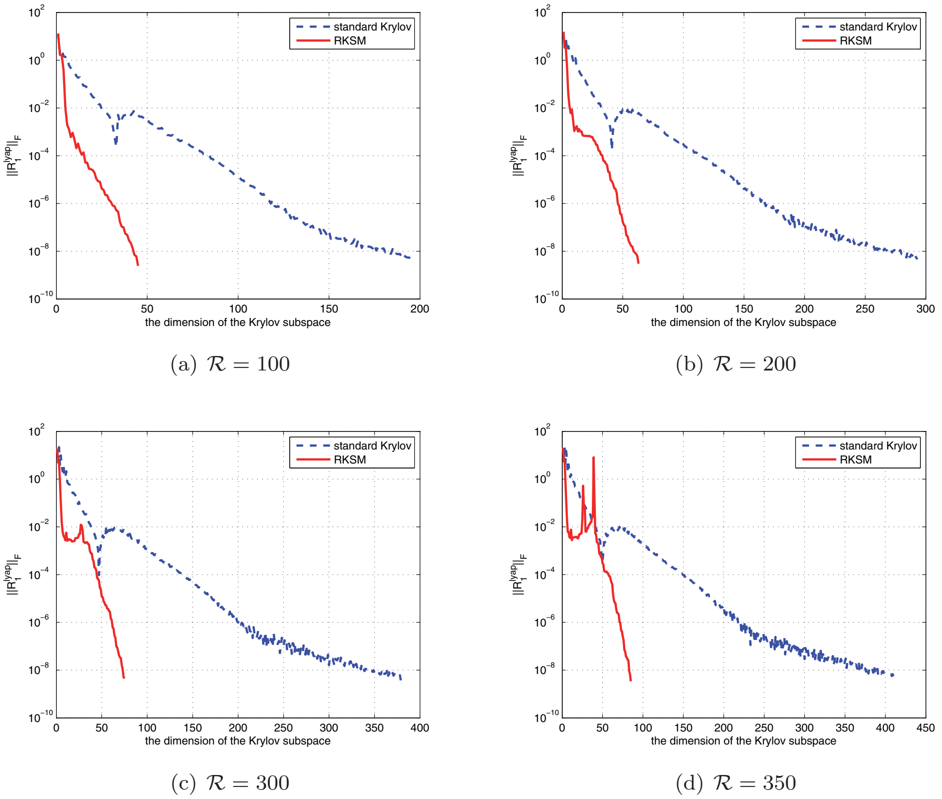
FIG. 3.3. (a) *The six rightmost eigenvalues for flow over an obstacle at different Reynolds numbers* (∗: $\mathcal{R} = 100$; ○: $\mathcal{R} = 200$; ◇: $\mathcal{R} = 300$; □: $\mathcal{R} = 350$). (b) *The* 300 *eigenvalues with smallest modulus at* $\mathcal{R} = 350$ (×: *the rightmost eigenvalues*).

example, the Reynolds number $\mathcal{R} = \frac{2}{\nu}$.) As for the previous example, as the Reynolds number increases, the six rightmost eigenvalues all move toward the imaginary axis, and the rightmost eigenvalue changes from being real (at $\mathcal{R} = 100$) to complex (at $\mathcal{R} = 200, 300, 350$). The rightmost pair of eigenvalues of (1.1) crosses the imaginary axis and the steady-state solution to (1.2) loses its stability at $\mathcal{R} \approx 373$.

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., $\|R_\ell^{\text{eig}}\|_F < 10^{-8}$ and $\|R_\ell^{\text{lyap}}\|_F < 10^{-9} \cdot \|C_\ell\|_F$. For all four Reynolds numbers, Algorithm 1 converges rapidly. In fact, we will show in section 4.2 that if the Lyapunov equation (2.6) is solved more accurately, Algorithm 1 will converge in one iteration in all four cases as observed in the previous example. Again we compare the performance of the standard

TABLE 3.3
Algorithm 1 *applied to Example* 2 *(Lyapunov solver: RKSM).*

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



(a) $\mathcal{R} = 100$

(b) $\mathcal{R} = 200$

(c) $\mathcal{R} = 300$

(d) $\mathcal{R} = 350$

FIG. 3.4. *Comparison of the standard Krylov method and RKSM for solving* (2.6) *in Example* 2.

Krylov method and RKSM in solving (2.6). 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 [24]). 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 $(\overline{u}_c, \alpha_c)$ have fairly large magnitude, and as a result, the rightmost eigenvalues are further 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 $n = 10,000$ 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 [16]. 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

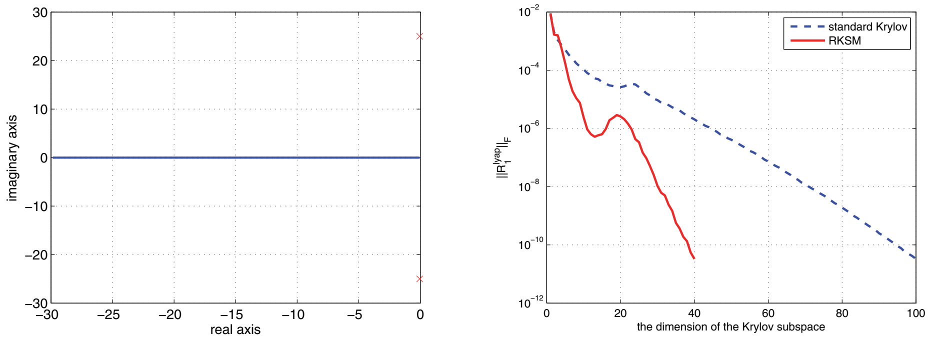


FIG. 3.5. *Left: the* 300 *eigenvalues with smallest modulus (×: the rightmost eigenvalues). Right: Comparison of the standard Krylov method and RKSM for solving* (2.6) *in Example* 3.

TABLE 3.4
Algorithm 1 *applied to Example* 3 *(Lyapunov solver: RKSM).*

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\mathrm{eig}}\|_F$	$\|R_\ell^{\mathrm{lyap}}\|_F$	d_ℓ
1	0.05000	−0.05000+25.0000i	4.76830e-08	3.28342e-11	40
2	0.05000	−0.05000+25.0000i	1.56010e-13	3.01965e-12	50

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 three 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)).

4. Analysis and improvements of Algorithm 1.

In section 3, our numerical experiments show that Algorithm 1 converges rapidly. In particular, it converges in just one iteration in several cases, i.e, the driven-cavity problem at $\mathcal{R} = 2000, 4000, 6000, 7800$. In other words, only one Lyapunov solve (2.6) is needed to obtain an eigenvalue estimate of desired accuracy in these cases. Such behavior is usually not seen in standard inverse iteration, whose convergence rate is known to be linear (see [23], for example). In this section, we present an analysis that provides insight into the fast convergence of Algorithm 1. In essence, the behavior of this algorithm depends on how well the eigenvector associated with the rightmost eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$ can be approximated by the Krylov subspace built to solve the Lyapunov equation (2.6). We also propose improvements to Algorithm 1 that make it more efficient and robust.

4.1. Analysis and modification of Algorithm 1.

Recall that at each iteration of Algorithm 1, we first apply an iterative method such as RKSM to compute an approximate solution $Y_\ell^{\mathrm{approx}} = V_\ell D_\ell V_\ell^T$ to (2.6), where $V_\ell \in \mathbb{R}^{n\times d_\ell}$ is orthonormal; then we project the eigenvalue problem (2.2) onto the subspace spanned by the columns of V_ℓ and solve (2.4) for its eigenvalue with smallest modulus. Also recall that the eigenvector of (2.2) associated with $-\mathrm{Re}(\mu_1)$ is $\mathcal{V}\mathcal{D}\mathcal{V}^T$, which is of rank 1 or 2. We first give a sufficient condition for Algorithm 1 to converge to the target eigenpair $\left(-\mathrm{Re}(\mu_1), \mathcal{V}\mathcal{D}\mathcal{V}^T\right)$ in just one step.

Let $U \in \mathbb{R}^{n\times d}$ be an orthonormal matrix whose column space includes $\{x_1, \overline{x_1}\}$. Then $\mathcal{V} = U\mathcal{G}$ for some $\mathcal{G} \in \mathbb{R}^{d\times 1}$ or $\mathbb{R}^{d\times 2}$. It is not difficult to show that μ_1 is an eigenvalue of $(U^TSU)^{-1}$.

THEOREM 4.1. *Let U be as described above. Then $\left(-\mathrm{Re}(\mu_1), \mathcal{G}\mathcal{D}\mathcal{G}^T\right)$ is an eigenpair of*

$$(U^TSU)\widetilde{Z} + \widetilde{Z}(U^TSU)^T + \widetilde{\lambda}\left(2(U^TSU)\widetilde{Z}(U^TSU)^T\right) = 0, \tag{4.1}$$

which is of the same structure as (2.4).

Proof. Since $\left(-\mathrm{Re}(\mu_1), \mathcal{V}\mathcal{D}\mathcal{V}^T\right)$ is an eigenpair of (2.2),

$$S\left(\mathcal{V}\mathcal{D}\mathcal{V}^T\right) + \left(\mathcal{V}\mathcal{D}\mathcal{V}^T\right)S^T - \mathrm{Re}(\mu_1)\left(2S\left(\mathcal{V}\mathcal{D}\mathcal{V}^T\right)S^T\right) = 0. \tag{4.2}$$

Left-multiply (4.2) by U^T and right-multiply by U:

$$U^TS(\mathcal{V}\mathcal{D}\mathcal{V}^T)U + U^T(\mathcal{V}\mathcal{D}\mathcal{V}^T)S^TU - \mathrm{Re}(\mu_1)2U^TS(\mathcal{V}\mathcal{D}\mathcal{V}^T)S^TU = 0.$$

Since $\mathcal{V} = U\mathcal{G}$ and U is orthonormal,

$$\begin{aligned}&\left(U^TSU\right)\left(\mathcal{G}\mathcal{D}\mathcal{G}^T\right) + \left(\mathcal{G}\mathcal{D}\mathcal{G}^T\right)\left(U^TS^TU\right)\\ &\quad - \mathrm{Re}(\mu_1)\left(2\left(U^TSU\right)\left(\mathcal{G}\mathcal{D}\mathcal{G}^T\right)\left(U^TS^TU\right)\right) = 0. \qquad \square\end{aligned}$$

COROLLARY 4.2. *Assume that the rightmost eigenvalue of $\left(U^TSU\right)^{-1}$ is μ_1. Then the eigenvalue with smallest modulus of* (4.1) *is $-\mathrm{Re}(\mu_1)$.*

Proof. Let $\widetilde{\Delta}_1 = (U^TSU) \otimes I_d + I_d \otimes (U^TSU)$ and $\widetilde{\Delta}_0 = -2(U^TSU) \otimes (U^TSU)$. Since the rightmost eigenvalue of $\left(U^TSU\right)^{-1}$ is μ_1, by Theorem 2.2, the eigenvalue with smallest modulus of the $d^2 \times d^2$ eigenvalue problem

$$\widetilde{\Delta}_1\widetilde{z} = \widetilde{\lambda}\widetilde{\Delta}_0\widetilde{z} \tag{4.3}$$

is $-\mathrm{Re}(\mu_1)$. The eigenvalue problems (4.1) and (4.3) have the same eigenvalues. Therefore, the eigenvalue with smallest modulus of (4.1) is $-\mathrm{Re}(\mu_1)$ as well. $\square$

Theorem 4.1 together with Corollary 4.2 shows that if the column space of V_1 in $Y_1^{\mathrm{approx}} = V_1D_1V_1^T$ contains $\{x_1, \overline{x_1}\}$ and the rightmost eigenvalue of $\left(V_1^TSV_1\right)^{-1}$ is μ_1, then the estimated eigenpair we obtain after one iteration of Algorithm 1 will be

$$\left(-\mathrm{Re}(\mu_1), V_1(\mathcal{G}\mathcal{D}\mathcal{G}^T)V_1^T\right) = \left(-\mathrm{Re}(\mu_1), \mathcal{V}\mathcal{D}\mathcal{V}^T\right),$$

which is already exact.

In the ideal case where Y_1^{approx} is exact (i.e., $Y_1^{\mathrm{approx}} = Y_1$), we indeed have the desired V_1. Recall that we use $Z^{(0)} = \mathcal{V}_0\mathcal{V}_0^T$ as the starting guess of Algorithm 1, where $\mathcal{V}_0 \in \mathbb{R}^n$ has unit norm. Assume $\mathcal{V}_0 = \sum_{i=1}^n \xi_i x_i$ where $\xi_i \neq 0$ for any i (i.e., $\mathcal{V}_0$ is not deficient in any eigenvector of $\mathbf{A}x = \mu\mathbf{M}x$). Since the solution to the $n^2 \times n^2$ linear system $\Delta_1 y_1 = \Delta_0(\mathcal{V}_0 \otimes \mathcal{V}_0)$ is given by $y_1 = \sum_{i,j=1}^n \frac{\xi_i\xi_j}{\lambda_{i,j}} z_{i,j}$, the solution to the corresponding Lyapunov equation (2.6) with $\ell = 1$ is

$$Y_1 = \sum_{i,j=1}^n \frac{\xi_i\xi_j}{\lambda_{i,j}} Z_{i,j},$$

the column space of which is the invariant subspace of $\mathbf{A}x = \mu\mathbf{M}x$ that contains every eigenvector. Thus, the analysis above shows that in this case Algorithm 1 converges to the target eigenpair in just one iteration. This stands in contrast to standard inverse iteration, which exhibits linear convergence even if the analogue of (2.6) is solved exactly.

This observation does not correspond to a practical computation. When (2.6) is solved exactly, the "projected" eigenvalue problem (2.4) is of the same order as the original one (2.2). Indeed, the two problems are equivalent: (2.4) is the Lyapunov form of the similarity-transformed version of (2.1) given by

$$\widetilde{\Delta}_1 \tilde{z} = \lambda \widetilde{\Delta}_0 \tilde{z}, \qquad \widetilde{\Delta}_i = \left(V_1^T \otimes V_1^T\right) \Delta_i (V_1 \otimes V_1), \quad \tilde{z} = \left(V_1^T \otimes V_1^T\right) z.$$

(Note in particular that V_1 and $V_1 \otimes V_1$ are square and orthogonal.) Thus, in this discussion, we have simply replaced one large Lyapunov eigenvalue problem, (2.2), with another one, (2.4).

Algorithm 1 is practical because we compute an approximate solution $Y_1^{\text{approx}} = V_1 D_1 V_1^T$ to (2.6) using an iterative method, where V_1 and Y_1^{approx} are of low rank. This makes the projected problem (2.4) inexpensive to solve. Although there is no guarantee that the column space of V_1 includes $\{x_1, \overline{x_1}\}$ or that μ_1 is the rightmost eigenvalue of $(V_1^T S V_1)^{-1}$, in our experience, when a good Lyapunov solver such as RKSM is used, we can always compute a low-rank Y_1^{approx} that is accurate enough for Algorithm 1 to converge in one iteration. For instance, consider again Examples 2 and 3, for which Algorithm 1 with $\tau_{\text{lyap}} = 10^{-9}$ converges in more than one iteration (see Tables 3.3 and 3.4). If we solve (2.6) more accurately, as seen in Tables 4.1 and 4.2, Algorithm 1 converges in one iteration as well. (The stopping criterion for the eigenvalue computation is unchanged, i.e., $\|R_\ell^{\text{eig}}\|_F < 10^{-8}$.) Moreover, comparison of the two sets of tables shows that this new strategy, where we force Algorithm 1 to converge in one iteration by solving (2.6) more accurately, significantly increases the efficiency of Algorithm 1.

The results in Tables 4.1 and 4.2 were obtained by augmenting the stopping criterion for the Lyapunov problem (2.6) with a condition associated with the eigenvalue problem (2.2). That is, we require the approximate solution Y_1^{approx} to satisfy both $\|R_1^{\text{lyap}}\|_F < \tau_{\text{lyap}} \cdot \|C_1\|_F$ and $\|R_1^{\text{eig}}\|_F < \tau_{\text{eig}}$, where $\tau_{\text{lyap}} = 10^{-9}$ and $\tau_{\text{eig}} = 10^{-8}$. An algorithmic form of this strategy is given by Algorithm 2. In this algorithm, if $\|R_1^{\text{eig}}\| < \tau_{\text{eig}}$ is not met, then we keep improving the accuracy of Y_1^{approx} until it is satisfied. This can be done by augmenting the solution we have in hand. Assume that at step 2.1 of Algorithm 2, we compute an approximate solution $Y_1^{\text{approx}} = V_1 D_1 V_1^T$ where the column space of V_1 is the Krylov subspace $\mathcal{K}_m(S, P_1)$. (See (2.7) for a

Table 4.1
Modified Algorithm 1 applied to Example 2 (Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
		$\mathcal{R} = 100$			
1	0.57285	−0.57285	7.98509e-9	4.45806e-12	65
		$\mathcal{R} = 200$			
1	0.32884	−0.32884+2.16393i	7.67144e-9	2.79438e-12	83
		$\mathcal{R} = 300$			
1	0.10405	−0.10405+2.22643i	6.10287e-9	1.35045e-10	86
		$\mathcal{R} = 350$			
1	0.02411	−0.02411+2.24736i	7.16343e-9	1.07068e-09	88

Table 4.2
Modified Algorithm 1 applied to Example 3 (Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
1	0.05000	−0.05000+25.0000i	6.88230e-9	4.71359e-12	43

Algorithm 2. Modified Algorithm 1.

```
1. Given $\mathcal{V}_0 \in \mathbb{R}^n$ with $\|\mathcal{V}_0\|_2 = 1$. Let $Z^{(0)} = \mathcal{V}_0\mathcal{V}_0^T$, $\|R_1^{\text{eig}}\|_F = \infty$ and
   lyap-cvg = False.
2. While $\|R_1^{\text{eig}}\|_F \geq \tau_{\text{eig}}$
    2.1. If lyap-cvg = False
           Compute an approximate solution $Y_1^{\text{approx}} = V_1 D_1 V_1^T$ to (2.6)
           with $\ell = 1$ that satisfies $\|R_1^{\text{lyap}}\|_F < \tau_{\text{lyap}} \cdot \|C_1\|_F$, and set
           lyap-cvg = True.
    2.2. Else
           Compute a more accurate $Y_1^{\text{approx}} = V_1 D_1 V_1^T$.
    2.3. Rank reduction: let $\widetilde{S} = V_1^T S V_1$ and solve for the eigenvalue
         $\widetilde{\lambda}_1$ of (2.4) with smallest modulus and its eigenvector $\widetilde{Z}_1 = \widetilde{\mathcal{V}}\widetilde{\mathcal{D}}\widetilde{\mathcal{V}}^T$,
         where $\|\widetilde{\mathcal{D}}\|_F$=1.
    2.4. Set $\lambda^{(1)} = \widetilde{\lambda}_1$ and $Z^{(1)} = \mathcal{V}_1\widetilde{\mathcal{D}}\mathcal{V}_1^T$, where $\mathcal{V}_1 = V_1\widetilde{\mathcal{V}}$.
```

definition.) If $\|R_1^{\text{eig}}\|_F \geq \tau_{\text{eig}}$, then we perform one more Arnoldi step to extend the existing Krylov subspace to $\mathcal{K}_{m+1}(S, P_1)$, obtain a new approximate solution $Y_1^{\text{approx}} = V_1 D_1 V_1^T$ where the column space of V_1 is now the augmented Krylov subspace $\mathcal{K}_{m+1}(S, P_1)$, and check convergence again.

4.2. Restarting Algorithm 2: A validation step. In section 4.1, we have shown that if the column space of V_1 contains $\{x_1, \overline{x}_1\}$ and the rightmost eigenvalue of $(V_1^T S V_1)^{-1}$ is μ_1, then Algorithm 1 will converge to $\left(-\text{Re}(\mu_1), \mathcal{V}\mathcal{D}\mathcal{V}^T\right)$ in just one step. In particular, this is true if Y_1^{approx} is the exact solution. By a similar analysis, we can show that regardless of how accurate Y_1^{approx} might be, if the column space of V_1 contains $\{x_j, \overline{x_j}\}$ ($x_j \neq x_1$ or $\overline{x_1}$) but no $\{x_1, \overline{x}_1\}$ and the rightmost eigenvalue of $(V_1^T S V_1)^{-1}$ is μ_j, Algorithm 1 converges to a *wrong* eigenpair, also in one step.

In Algorithm 1, we solve (2.6) using a Krylov-type method until $\|R_\ell^{\text{lyap}}\|_F < \tau_{\text{lyap}} \cdot \|C_\ell\|_F$ is satisfied. The columns of V_ℓ therefore span a Krylov subspace. For any given $\tau_{\text{lyap}} > 0$, there is no guarantee that this Krylov subspace contains $\{x_1, \overline{x}_1\}$. In Algorithm 2, we keep extending the Krylov subspace until both $\|R_1^{\text{lyap}}\|_F < \tau_{\text{lyap}} \cdot \|C_1\|_F$ and $\|R_1^{\text{eig}}\|_F < \tau_{\text{eig}}$ are satisfied. This ensures that *some* $\{x_j, \overline{x_j}\}$ is found, but there is no guarantee that these are the ones we want, corresponding to $j = 1$. Thus, although Algorithm 2 works well in the experiments described above, which include some challenging problems, it is possible that this algorithm will produce an eigenvalue that is not the rightmost one. In this section, we study how the performance of Algorithm 2 depends on the choice of τ_{lyap} and propose a method of validating its computational result.

We first study the robustness of Algorithm 2 by testing it with $\tau_{\text{eig}} = 10^{-8}$ fixed and three different choices of τ_{lyap}: 10^{-3}, 10^{-6}, and 10^{-9}. In addition to the three examples considered in section 3, we also consider a much more difficult version of the double-diffusive convection problem where the rightmost eigenvalues are $\mu_{1,2} = -0.05 \pm 2500i$ instead of $-0.05 \pm 25i$ and the other eigenvalues remain unchanged. This problem will be referred to as Example 4.

For all cases of Examples 1, 2, and 3, Algorithm 2 converges to the correct rightmost eigenvalue for all three choices of τ_{lyap}. However, for Example 4, when $\tau_{\text{lyap}} = 10^{-3}$ or 10^{-6}, Algorithm 2 converges to the third rightmost eigenvalue $\mu_3 = -0.1$. The computational results for this example are reported in Table 4.3.

Table 4.3

Algorithm 2 with different choices of τ_{lyap} applied to Example 4 (Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
$\tau_{\text{lyap}} = 10^{-3}$					
1	0.10000	−0.10000	9.77100e-9	3.58087e-9	33
$\tau_{\text{lyap}} = 10^{-6}$					
1	0.10000	−0.10000	9.77100e-9	3.58087e-9	33
$\tau_{\text{lyap}} = 10^{-9}$					
1	0.05000	−0.05000+2500.00i	9.54205e-9	6.90965e-12	67

These results show that Algorithm 2 can be sensitive to the choice of τ_{lyap}. We need some way to determine whether the computed eigenvalue $\mu^{(1)}$ is truly the rightmost. Recall that we use a random vector $\mathcal{V}_0 \in \mathbb{R}^n$ with unit norm as the starting guess of Algorithm 2. Let $\sigma = \frac{1}{\mu^{(1)}}$ and p be a natural number. To enhance the robustness of Algorithm 2, we can run it again with the same τ_{lyap}, τ_{eig} and a starting guess in which the previously computed eigenvector is filtered out:

$$\mathcal{V}_0^{\text{filter}} = (S - \sigma I_n)^p \, \mathcal{V}_0 \tag{4.4}$$

if $\mu^{(1)}$ is real or

$$\mathcal{V}_0^{\text{filter}} = [(S - \sigma I_n)\,(S - \overline{\sigma} I_n)]^p \, \mathcal{V}_0 \tag{4.5}$$

if $\mu^{(1)}$ is not real. (In the actual computation, a normalized $\mathcal{V}_0^{\text{filter}}$, with Euclidean norm 1, is used.) The filtration damps the eigenvectors corresponding to eigenvalues of S close to $\frac{1}{\mu^{(1)}}$ and emphasizes those associated with eigenvalues of S far from $\frac{1}{\mu^{(1)}}$. As a result, the latter ones are more likely to be found when we restart Algorithm 2 with $\mathcal{V}_0^{\text{filter}}$. For example, consider the case where $\mu_{1,2} = -\epsilon_1 \pm \beta i$, $\mu_3 = -\epsilon_2$, and $\mu_4 = -\epsilon_3$ with $0 < \epsilon_1 < \epsilon_2 < \epsilon_3 \ll 1 < \beta$. That is, the rightmost eigenvalue has large imaginary part but there are also real eigenvalues close to zero. Suppose we apply Algorithm 2 with a bad choice of τ_{lyap} and get $\mu^{(1)} = \mu_3$ as the estimate for the rightmost eigenvalue, as observed for Example 4 when $\tau_{\text{lyap}} = 10^{-3}$ or 10^{-6}. Then there will be no component of x_3 in $\mathcal{V}_0^{\text{filter}}$. In addition, it can be shown that for any $\beta > \epsilon_3$,

$$\left|\frac{1}{\mu_2} - \frac{1}{\mu_3}\right| = \left|\frac{1}{\mu_1} - \frac{1}{\mu_3}\right| = \left|\frac{1}{-\epsilon_1 + \beta i} - \frac{1}{-\epsilon_2}\right| > \left|\frac{1}{-\epsilon_3} - \frac{1}{-\epsilon_2}\right| = \left|\frac{1}{\mu_4} - \frac{1}{\mu_3}\right|.$$

It follows that the components of x_1 and x_2 will typically be larger in $\mathcal{V}_0^{\text{filter}}$ than that of an eigenvector associated with real eigenvalues close to zero. That is, when μ_1 is one of a complex conjugate pair of eigenvalues that are further away from zero than many real eigenvalues (as seen in Examples 1 and 3), the filtering strategy makes it much easier to find μ_1 when Algorithm 2 is restarted.[1]

We now summarize our strategy for computing the rightmost eigenvalue of $\mathbf{A}x = \mu\mathbf{M}x$ as follows. It requires applying Algorithm 2 to (2.2) twice using fixed τ_{lyap}, τ_{eig} and different starting guesses. First, we apply Algorithm 2 with a random initial guess $\mathcal{V}_0$ to (2.1), which requires one approximate Lyapunov solve for (2.6). Then as a validation step, we restart Algorithm 2 with the the filtered starting vector (4.4) or

[1] We also remark that we found this variant of Algorithm 2 with filtering to be more robust for locating critical eigenvalues than the deflation strategy discussed in the next section.

Table 4.4

Restarted Algorithm 2 with different choices of τ_{lyap} applied to Example 4 (Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
		$\tau_{\text{lyap}} = 10^{-3}$			
1	0.05000	−0.05000+2500.00i	9.90794e-9	7.20542e-12	66
		$\tau_{\text{lyap}} = 10^{-6}$			
1	0.05000	−0.05000+2500.00i	9.90794e-9	7.20542e-12	66
		$\tau_{\text{lyap}} = 10^{-9}$			
1	0.10000	−0.10000	9.35439e-9	1.39151e-11	8

(4.5), where σ is chosen to be the estimated rightmost eigenvalue previously found. This step requires another approximate solve of (2.6) with a different right-hand side. It is our experience that a greater p in (4.4) or (4.5) enhances the robustness of this approach, yet it also makes computing $\mathcal{V}_0^{\text{filter}}$ more expensive. To compromise on these points, we choose $p = 3$, a small value greater than one, in all our experiments presented below.

Consider again Example 4. The results of Algorithm 2 using a random initial guess and three different choices of τ_{lyap} have been presented in Table 4.3. We present the results of the restarted Algorithm 2 in Table 4.4. The shift σ in the filtration is chosen based on the results reported in Table 4.3: when $\tau_{\text{lyap}} = 10^{-3}$ or 10^{-6}, the filtered vector (4.4) with $\sigma = \frac{1}{-0.1}$ is used; and when $\tau_{\text{lyap}} = 10^{-9}$, the filtered vector (4.5) with $\sigma = \frac{1}{-0.05+2500i}$ is used. The results show that the restarted Algorithm 2 with either 10^{-3} or 10^{-6} is able to find the correct rightmost eigenvalue, which was missed the first time we applied Algorithm 2. When $\tau_{\text{lyap}} = 10^{-9}$, $\mu_3 = -0.1$ is found. Since this is to the left of $-0.05 \pm 2500i$ found by applying Algorithm 2 the first time, the restart provides additional evidence that $-0.05 \pm 2500i$ are indeed the rightmost eigenvalues.

It can be seen from this example that the restart serves two purposes: on one hand, if Algorithm 2 does find the rightmost eigenvalue, restarting it increases our confidence that we have found the correct one; on the other hand, if Algorithm 2 first finds a wrong eigenvalue due to a bad choice of τ_{lyap}, the restarted algorithm produces the correct rightmost eigenvalue and thus increases the robustness of our approach.

To further examine the performance of this method, we carry out two additional tests. In the first test, we apply both Algorithm 2 and restarted Algorithm 2 with $\tau_{\text{lyap}} = 10^{-2}$ and $\tau_{\text{eig}} = 10^{-8}$ to Example 1 at $\mathcal{R} = 7800$. This value of τ_{lyap} is deliberately chosen so that Algorithm 2 fails to find the rightmost eigenvalues $-0.00514 \pm 2.69845i$ and finds the third rightmost eigenvalue $\mu_3 = -0.00845$ instead (see Table 4.5). If we then restart Algorithm 2 with the same choice of τ_{lyap} and the filtered vector (4.4), where $\sigma = \frac{1}{-0.00845}$, the rightmost eigenvalue is found. In the second test, we consider yet another version of the double-diffusive convection problem with $-0.05 \pm 25000i$ being the rightmost eigenvalue and other eigenvalues unchanged (Example 5). For this example, Algorithm 2 fails to find the rightmost eigenvalue even when τ_{lyap} is as small as 10^{-9} (see Table 4.6). Again, restarting it with the same choice of τ_{lyap} and the filtered vector (4.4) with $\sigma = \frac{1}{-0.1}$ produces the correct rightmost eigenvalue.

The strategy proposed in this section (i.e., applying Algorithm 2 and then restarting it) is shown to be robust and efficient in our numerical experiments, in which we have explored quite extreme choices of τ_{lyap} and μ_1.

Table 4.5
Algorithm 2 and restarted Algorithm 2 applied to Example 1 at $\mathcal{R} = 7800$ ($\tau_{\text{lyap}} = 10^{-2}$; Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
		Algorithm 2			
1	0.00845	−0.00845	1.91350e-9	7.95474e-3	89
		Restarted Algorithm 2			
1	0.00514	−0.00514+2.69845i	9.49871e-9	9.92614e-8	289

Table 4.6
Algorithm 2 and restarted Algorithm 2 applied to Example 5 ($\tau_{\text{lyap}} = 10^{-9}$; Lyapunov solver: RKSM).

ℓ	$\lambda^{(\ell)}$	$\mu^{(\ell)}$	$\|R_\ell^{\text{eig}}\|_F$	$\|R_\ell^{\text{lyap}}\|_F$	d_ℓ
		Algorithm 2			
1	0.10000	−0.10000	5.57759e-9	2.80401e-11	33
		Restarted Algorithm 2			
1	0.05000	−0.05000+25000.0i	1.95395e-9	1.42125e-12	91

5. 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, $-\text{Re}(\mu_1)$, is the eigenvalue with smallest modulus of (2.2). 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). 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 toward the imaginary rapidly, becomes the rightmost eigenvalue at some point, and eventually 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 $(\overline{u}_0, \alpha_0)$ 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 $\text{Re}(\mu_j) = \text{Re}(\mu_i)$ and $j \neq i$, then $\mu_j = \overline{\mu_i}$; (a2′) μ_i is a simple eigenvalue. Let $\mathcal{E}_t = \{\mu_1, \mu_2, \ldots, \mu_t\}$ be the set containing t rightmost eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$ and let $\mathcal{X}_t = [x_1, x_2, \ldots, x_t] \in \mathbb{C}^{n\times t}$ 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 $\overline{\mu_i} \in \mathcal{E}_t$ as well. We will show that given $\mathcal{X}_t$, we can use the methodology described in section 2 to find $-\text{Re}(\mu_{t+1})$, that is, that $-\text{Re}(\mu_{t+1})$ 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 5.1. *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 $-\text{Re}(\mu_{t+1})$.*

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

Consider again the 4×4 example discussed in section 2. The set $\{\lambda_{i,j}\}_{i,j>2}$ is $\{\lambda_{3,3} = 2,\ \lambda_{3,4} = \lambda_{4,3} = 2.5,\ \lambda_{4,4} = 3\}$, in which $\lambda_{3,3} = -\mathrm{Re}(\mu_3)$ is the one with smallest modulus.

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 $-\mathrm{Re}(\mu_{t+1})$ 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 $\mu_1^{-1}, \mu_2^{-1}, \ldots, \mu_t^{-1}$, 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

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

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

$$\widehat{\Delta}_1 z = \lambda \widehat{\Delta}_0 z, \quad z \in \mathrm{Range}\left(\widehat{\Delta}_0\right). \tag{5.1}$$

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

LEMMA 5.2. *The matrix $I - Q_tQ_t^T$, where Q_t is defined above and $I \in \mathbb{R}^{n\times n}$ is the identity matrix, has the following properties:*

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

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

by the first property. This together with the second property establishes the third property. ▯

LEMMA 5.3. *Let $\widehat{\theta}_i = 0$ for $i \leq t$ and $\widehat{\theta}_i = \frac{1}{\mu_i}$ for $i > t$. Let $\widehat{x}_i = x_i$ for $i \leq t$ and $\widehat{x}_i = \left(I - Q_tQ_t^T\right)x_i$ for $i > t$. Then $\left(\widehat{\theta}_i, \widehat{x}_i\right)$ $(i = 1, 2, \ldots, n)$ are the eigenpairs of $\widehat{S}$.*

Proof. Let g_i be the *ith* column of G_t. If $i \leq t$, $x_i = Q_tg_i$, thus

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

by the first property in Lemma 5.2. If $i > t$,

$$\begin{aligned}\widehat{S}\left(I - Q_tQ_t^T\right) x_i &= \left(I - Q_tQ_t^T\right) S \left(I - Q_tQ_t^T\right) x_i \\ &= \left(I - Q_tQ_t^T\right) Sx_i = \frac{1}{\mu_i}\left(I - Q_tQ_t^T\right) x_i\end{aligned}$$

by the third property in Lemma 5.2. ▯

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

Lemma 5.4. *The eigenvalues of $\widehat{\Delta}_1$ 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 $\widehat{\Delta}_0$ 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 $\widehat{\Delta}_0$ or $\widehat{\Delta}_1$, there are eigenvectors associated with it given by $\widehat{z}_{i,j} = \widehat{x}_i \otimes \widehat{x}_j$ and $\widehat{z}_{j,i} = \widehat{x}_j \otimes \widehat{x}_i$.

Proof. See the proof of Theorem 2.1. □

Under the assumption that $\mathbf{A}x = \mu \mathbf{M}x$ has a complete set of eigenvectors, $\widehat{\Delta}_0$ also has a complete set of eigenvectors $\{\widehat{z}_{i,j}\}_{i,j=1}^n$. By Lemma 5.4, $\text{Range}\big(\widehat{\Delta}_0\big) = span\,\{\widehat{z}_{i,j}\}_{i,j>t}$.

Theorem 5.5. *The eigenvalues of* (5.1) *are $\{\lambda_{i,j}\}_{i,j>t}$. For any $\lambda_{i,j}$ with $i, j > t$, there are eigenvectors $\widehat{z}_{i,j}$ and $\widehat{z}_{j,i}$ associated with it.*

Proof. The proof follows immediately from Lemma 5.4 and the proof of Theorem 2.1. □

Theorem 5.6. *Assume all the eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$ lie in the left half of the complex plane. Then the eigenvalue with smallest modulus of* (5.1) *is* $-\text{Re}(\mu_{t+1})$.

Proof. By Theorem 5.5, it suffices to show that $|\text{Re}(\mu_{t+1})| \leq |\lambda_{i,j}|$ for any $i, j > t$, which is true by Lemma 5.1. □

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

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

Since

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

if $Z \in \mathbb{P}_t$, then $z = vec(Z) \in \text{Range}\big(\widehat{\Delta}_0\big)$ and vice versa. Therefore, (5.1) can be rewritten in the form of a matrix equation,

$$\widehat{S}Z + Z\widehat{S}^T + \lambda\left(2\widehat{S}Z\widehat{S}^T\right) = 0, \ Z \in \mathbb{P}_t. \tag{5.2}$$

By Theorem 5.6, $-\text{Re}(\mu_{t+1})$ is the eigenvalue with smallest modulus of (5.2). As in section 2, under certain conditions, we can show that $-\text{Re}(\mu_{t+1})$ is an eigenvalue of (5.2) with a unique, real, symmetric, and low-rank eigenvector. Let $\mathbb{P}_t^s = \big\{Z \in \mathbb{P}_t | Z = Z^T\big\}$ be the subspace of $\mathbb{P}_t$ consisting of symmetric matrices. As a result of assumptions (a1′) and (a2′), when the eigenspace of (5.2) is restricted to $\mathbb{P}_t^s$, $-\text{Re}(\mu_{t+1})$ is an eigenvalue of (5.2) that has the unique (up to a scalar multiplier), real, and symmetric eigenvector $\left(I - Q_t Q_t^T\right)\left(x_{t+1}x_{t+1}^* + \overline{x_{t+1}}x_{t+1}^T\right)\left(I - Q_t Q_t^T\right)$. Therefore, if we restrict the search space for the target eigenvector of (5.2) to $\mathbb{P}_t^s$, Lyapunov inverse iteration can be applied to (5.2) to compute $-\text{Re}(\mu_{t+1})$.

The analysis above leads to a method for computing k rightmost eigenvalues of $\mathbf{A}x = \mu\mathbf{M}x$, which is outlined in Algorithm 3. At each iteration of Algorithm 3, we

Algorithm 3. Compute k rightmost eigenvalues of $\mathbf{A}x = \mu \mathbf{M}x$.

```
1. Initialization: $t = 0$, $\mathcal{E}_t = \emptyset$, $\widehat{\mathcal{X}}_t = \emptyset$, and $\widehat{S} = S$.
2. While $t < k$:
    2.1. Solve (5.2) for the eigenvalue with smallest modulus,
         $-\mathrm{Re}(\mu_{t+1})$, and its corresponding eigenvector in $\mathbb{P}_t^s$;
         compute $(\mu_{t+1}, \widehat{x}_{t+1})$.
    2.2. Update:
         if $\mu_{t+1}$ is real:
             $\mathcal{E}_{t+1} \leftarrow \{\mathcal{E}_t, \mu_{t+1}\}$, $\widehat{\mathcal{X}}_{t+1} \leftarrow \left[\widehat{\mathcal{X}}_t, \widehat{x}_{t+1}\right]$, $t \leftarrow t+1$;
         else:
             $\mathcal{E}_{t+2} \leftarrow \{\mathcal{E}_t, \mu_{t+1}, conj\,(\mu_{t+1})\}$,
             $\widehat{\mathcal{X}}_{t+2} \leftarrow \left[\widehat{\mathcal{X}}_t, \widehat{x}_{t+1}, conj\,(\widehat{x}_{t+1})\right]$, $t \leftarrow t+2$.
    2.3. Compute $Q_t \in \mathbb{R}^{n \times t}$ whose columns form an orthonormal
         basis of $\widehat{\mathcal{X}}_t$; let $\widehat{S} = \left(I - Q_t Q_t^T\right) S$.
```

Table 5.1
Algorithm 3 *applied to Examples* 1, 2, 3 *and* 4.

t	Estimated μ_{t+1}	t	Estimated μ_{t+1}
Example 1 (**$\mathcal{R} = 6000$**)**,** $k = 8$		**Example 1** (**$\mathcal{R} = 7800$**)**,** $k = 8$	
0	-0.01084	0	-0.00514+2.69845i
1	-0.02006+0.91945i	2	-0.00845
3	-0.03033+1.79660i	3	-0.01531+0.91937i
5	-0.03794	5	-0.02163+1.78863i
6	-0.04418+2.69609i	7	-0.02996
Example 2 (**$\mathcal{R} = 300$**)**,** $k = 6$		**Example 2** (**$\mathcal{R} = 350$**)**,** $k = 6$	
0	-0.10405+2.22643i	0	-0.02411+2.24736i
2	-0.32397	2	-0.28408
3	-0.39197	3	-0.33571
4	-0.60628	4	-0.56485
5	-0.87203	5	-0.79196
Example 3, $k = 6$		**Example 4,** $k = 6$	
0	-0.05000+25.0000i	0	-0.05 000+2500.00i
2	-0.10000	2	-0.10000
3	-0.20000	3	-0.20000
4	-0.30000	4	-0.30000
5	-0.40000	5	-0.40000

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 (5.2) corresponding to different values of t, and each of them can be solved using Algorithm 2. (A more efficient way of solving them is proposed in Chapter 4 of [25].) Note that the orthonormal matrix Q_t is computed from $\widehat{\mathcal{X}}_t$ instead of $\mathcal{X}_t$, which is unavailable to us. It can be shown easily that $\widehat{\mathcal{X}}_t$ and $\mathcal{X}_t$ actually have the same column space.

We apply this algorithm to compute a few rightmost eigenvalues for some cases of the examples considered in section 3. At each iteration of Algorithem 1, the eigenvalue problem (5.2) is solved using Algorithm 2 with $\tau_{\mathrm{lyap}} = 10^{-9}$ and $\tau_{\mathrm{eig}} = 10^{-8}$. The computed k rightmost eigenvalues for each case are reported in Table 5.1. For example, consider the driven-cavity flow at $\mathcal{R} = 7800$. From Table 5.1, 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 3.1(a)).

6. Conclusion. In this paper, we have developed a robust and efficient method for computing a few rightmost eigenvalues of (1.1) at any point $(\overline{u}_0, \alpha_0)$ 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, 17], 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 is given, which leads to a validation step that increases its robustness. 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 (5.1), whose eigenvalue with smallest modulus is the distance between the $(t+1)$st rightmost eigenvalue and the imaginary axis. Finally, an algorithm that computes a few rightmost eigenvalues of (1.1) is proposed. The method developed in this study together with the method proposed in [8, 17] constitutes a robust way of detecting the transition to instability in the steady-state solution of a large-scale dynamical system.

Acknowledgment. We thank the referees for their insightful comments and suggestions, which have helped us develop a deeper understanding of the algorithms presented here.

REFERENCES

[1] A. C. Antoulas, D. C. Sorensen, and Y. Zhou, *On the Decay of Hankel Singular Values and Related Issues*, Tech. report 01-09, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2001; also available online 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. 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*, in Proceedings of Mathematics of Finite Elements and Applications VII, J. R. Whiteman, ed., Academic Press, New York, 1990, pp. 479–486.

[4] K. A. Cliffe, T. J. Garratt, and A. Spence, *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 Lett., 60 (2011), pp. 546–560.

[8] H. C. Elman, K. Meerbergen, A. Spence, and M. Wu, *Lyapunov inverse iteration for identifying Hopf bifurcations in models of incompressible flow*, SIAM J. Sci. Comput., 34 (2012), pp. A1584–A1606.

[9] A. Fortin, M. Jardak, and J. Gervais, *Localization of Hopf bifurcation in fluid flow problems*, Internat. 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. Govaerts, *Numerical Methods for Bifurcations of Dynamical Equilibria*, SIAM, Philadelphia, 2000.

[12] L. Grasedyck, *Low-rank solutions of the Sylvester equation*, Numer. Linear Algebra Appl., 11 (2004), pp. 371–389.

[13] S. J. Hammarling, *Numerical solution of the stable, non-negative definite Lyapunov equation*, IMA J. Numer. Anal., 2 (1982), pp. 303–323.

[14] I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.

[15] D. Kressner and C. Tobler, *Low-Rank Tensor Krylov Subspace Methods for Parameterized Linear Systems*, Tech. report 2010-16, ETH, Zurich, 2010; also available online from http://www.math.ethz.ch/∼kressner.

[16] 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.

[17] 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.

[18] K. Meerbergen and R. Vandebril, *A reflection on the implicitly restarted Arnoldi method for computing eigenvalues near a vertical line*, Linear Algebra Appl., 436 (2012), pp. 2828–2844.

[19] T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: The symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.

[20] Y. Saad, *Numerical solution of large Lyapunov equations*, in Signal Processing, Scattering, Operator Theory, and Numerical Methods, M. A. Kaashoek, J. H. van Schuppen, and A. C. Ran, eds., Birkhäuser Boston, Cambridge, MA, pp. 503–511.

[21] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[22] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[23] G. W. Stewart, *Matrix Algorithms Volume 2: Eigensystems*, SIAM, Philadelphia, 2001.

[24] J. S. Turner, *Buoyancy Effects in Fluids*, Cambridge University Press, Cambridge, UK, 1973.

[25] M. Wu, *Linear Stability Analysis Using Lyapunov Inverse Iteration*, Ph.D. thesis, University of Maryland, College Park, 2012.