Shift-and-invert iteration for purely imaginary eigenvalues with application to the detection of Hopf Bifurcations in large scale problems

Karl Meerbergen *Alastair Spence*

Report TW 533, December 2008

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Shift-and-invert iteration for purely imaginary eigenvalues with application to the detection of Hopf Bifurcations in large scale problems

Karl Meerbergen *Alastair Spence*

Report TW 533, December 2008

Department of Computer Science, K.U.Leuven

Abstract

The detection of a Hopf bifurcation in a large scale dynamical system that depends on a physical parameter often consists of computing the right-most eigenvalues of a sequence of large sparse eigenvalue problems. This is not only an expensive operation, but the computation of right-most eigenvalues is often not reliable for the commonly used methods for large sparse matrices. In the literature a method has been proposed that computes a value of the parameter that corresponds to a Hopf point without actually computing right-most eigenvalues. This method utilises the Kronecker product and involves the solution of matrices of squared dimension, which is impractical for large scale applications.

However, if good starting guesses are available for the parameter and the purely imaginary eigenvalue at the Hopf point, then efficient algorithms are available. In this paper, we propose a method for obtaining such good starting guesses, based on finding purely imaginary eigenvalues of a two-parameter eigenvalue problem (possibly arising after a linearisation process). The problem is formulated as an inexact inverse iteration method that requires the solution of a sequence of Lyapunov equations with low rank right hand sides. It is this last fact that makes the method feasible for large systems. The power of our method is tested on three numerical examples, one of which is a discretised PDE with two space dimensions.

Keywords : Hopf bifurcations, dynamical systems, eigenvalue problem, Lyapunov equation.
MSC : Primary : 37M20, 65F15, Secondary: 65P30, 65P40

SHIFT-AND-INVERT ITERATION FOR PURELY IMAGINARY EIGENVALUES WITH APPLICATION TO THE DETECTION OF HOPF BIFURCATIONS IN LARGE SCALE PROBLEMS

KARL MEERBERGEN AND ALASTAIR SPENCE

Abstract. The detection of a Hopf bifurcation in a large scale dynamical system that depends on a physical parameter often consists of computing the right-most eigenvalues of a sequence of large sparse eigenvalue problems. This is not only an expensive operation, but the computation of right-most eigenvalues is often not reliable for the commonly used methods for large sparse matrices. In the literature a method has been proposed that computes a value of the parameter that corresponds to a Hopf point without actually computing right-most eigenvalues. This method utilises the Kronecker product and involves the solution of matrices of squared dimension, which is impractical for large scale applications.

However, if good starting guesses are available for the parameter and the purely imaginary eigenvalue at the Hopf point, then efficient algorithms are available. In this paper, we propose a method for obtaining such good starting guesses, based on finding purely imaginary eigenvalues of a two-parameter eigenvalue problem (possibly arising after a linearisation process). The problem is formulated as an inexact inverse iteration method that requires the solution of a sequence of Lyapunov equations with low rank right hand sides. It is this last fact that makes the method feasible for large systems. The power of our method is tested on three numerical examples, one of which is a discretised PDE with two space dimensions.

1. Introduction. This paper introduces a numerical procedure for the determination of the smallest λ for which the eigenvalue problem

$$(A + \lambda B)x = \mu M x \tag{1.1}$$

has a pair of purely imaginary μ's. Here λ is to be thought of as a physical parameter and μ denotes the eigenvalue of the generalised eigenvalue problem (1.1), and we are assuming the matrices are large and sparse.

This work is motivated by the bifurcation analysis of the non-linear dynamical system

$$\frac{du}{dt} = f(u, \lambda) \quad , \quad u(0) = u_0$$

where f is an operator in $(\mathbf{R}^n, \mathbf{R}) \mapsto \mathbf{R}^n$ with n large. Such analysis includes the computation of bifurcation diagrams, and more particularly, the stability analysis and detection of Hopf bifurcations, which lead to the birth of periodic solutions (see, for example, [33]. In many situations (e.g. nonlinear finite element computations), we have an equation of the form

$$M\frac{du}{dt} = f(u, \lambda)$$

where M is a large sparse symmetric positive definite mass matrix. In the case of steady state solutions, i.e. $du/dt = 0$, often the values of λ are sought for which the solution u changes from a stable to an unstable regime. In a linearized stability analysis, the steady state is said to be stable when the eigenvalues μ of

$$J(\lambda)x = \mu M x \tag{1.2}$$

have strictly negative real parts, with $J(\lambda)$ denoting the Jacobian matrix evaluated at the steady state $u(\lambda)$, namely, $J(\lambda) = \frac{\partial f}{\partial u}(u(\lambda), \lambda)$. Values of λ where eigenvalues of

(1.2) cross the imaginary axis indicate a transition from a stable to unstable regime. When stability is lost due to a real eigenvalue μ passing through zero there are many techniques available to determine the critical value of λ, see, for example [13]. In contrast, at a Hopf bifurcation on a path of stable steady states, (1.2) has two purely imaginary eigenvalues, with the other eigenvalues having negative real parts. The detection of Hopf bifurcations is a particularly difficult task for large scale dynamical systems. However, if good starting values for λ and μ are known then there are good methods, usually based on Newton's Method, for their accurate determination, see, for example, [13]. The contribution of this paper is the determination of good starting values with which to start the Hopf calculation.

Perhaps the most straightforward method to detect Hopf points is to monitor the right-most eigenvalues of (1.2) for a discrete set of λ's. This requires the solution of an eigenvalue problem for each selected λ, which can be quite expensive, especially when the system size is large. The solution of large scale eigenvalue problems in this context has been studied intensively the last fifteen years. We refer to [2] for an overview of eigenvalue solvers, and [25] for an overview on methods for computing right-most eigenvalues. The shift-invert Arnoldi method [28] [32] with zero shift seems to be the most attractive approach since a matrix factorization of $J(\lambda)$ is available anyway. If a matrix factorization is not feasible, the JDQZ method [7], inexact rational Krylov [22] or Arnoldi's method with inexact shift-invert can be used. All these methods are quite reliable for computing eigenvalues near a point or target, but sometimes fail to compute the right-most eigenvalue. Therefore, an expensive validation phase can be employed to ensure that the right-most eigenvalue is indeed computed [26] [24].

A novel technique for the detection of Hopf bifurcation points in small scale dynamical systems was first introduced by Guckenheimer and co-workers, [16], [17], who introduced the bialternate product of $J(\lambda)$, defined as $(J(\lambda) \otimes I + I \otimes J(\lambda))/2$, which is an $n^2 \times n^2$ matrix, but which has a pair of zero eigenvalues when $J(\lambda)$ has a pair of purely imaginary eigenvalues. This approach was also used in [21], and [14], and expounded further in [13, §§4.4–4.5]. This construction forms the first theoretical step in our method, but we emphasise that we do not compute with the Kronecker product forms.

In this paper, we consider the situation where

$$J(u, \lambda) = (A + \lambda B) \ .$$

In a general setting, A and $B \in \mathbf{R}^{n \times n}$ could arise from a linearization of f around a specific value of u and λ. Here, A and B are usually nonsymmetric matrices and B can be singular. In this paper, we assume the following situation: $\lambda = 0$ corresponds to a stable steady state solution, i.e. all eigenvalues of $Ax = \mu Mx$ lie in the stable half plane. The goal is to compute the smallest λ for which the eigenvalue problem (1.1) has purely imaginary μ's. Generically, the μ's will be continuous functions of λ and the first λ for which there are μ's on the imaginary axis (including the case when $\mu = 0$), must correspond to a transition from a stable to unstable steady state, or to a Hopf point (or at least, to approximations of such points for the full Jacobian). A validation phase as in [26] [24] would no longer be required in this case.

First, note that if (1.1) has purely imaginary eigenvalues then

$$(A + \lambda B) \otimes M + M \otimes (A + \lambda B) \tag{1.3}$$

has a double eigenvalue zero. Mathematically, this translates into the following order

n^2 linear eigenvalue problem

$$(A \otimes M + M \otimes A)z + \lambda(B \otimes M + M \otimes B)z = 0 \ , \tag{1.4}$$

whose solution gives the values of λ for which (1.1) has purely imaginary eigenvalues μ. Although this is a nice mathematical property, it should only be used in this form for problems of small size. This paper is about the characterisation of the solutions of (1.4) and its efficient solution when A, B and M are large and sparse.

For the sake of completeness, we should mention the connection with the two-parameter eigenvalue problem. The problem that we discuss in this paper can also be written in the form

$$\begin{aligned} Ax + \lambda Bx - \mu Mx &= 0 \\ A\bar{x} + \lambda B\bar{x} + \mu M\bar{x} &= 0 \end{aligned}$$

where the second equation is the complex conjugate of the first and we have assumed that μ is purely imaginary. Such problems can be solved with the Jacobi-Davidson method [20] [19], which seeks (λ, μ) pairs near a target point (σ, τ), often with $\tau = 0$, since a good starting guess for μ typically won't be known. However, for the detection of a Hopf bifurcation this is a dangerous strategy and a situation that we want to avoid. This paper shows an alternative approach, where a starting guess for μ isn't required, but where the computation of μ is hidden in the method.

The plan of the paper is as follows. We first show properties of the eigenvalue problem (1.4) in §2. We show connections between (1.1) and (1.4) and prove that the λ of interest is a simple eigenvalue of (1.4) restricted to an appropriate subspace. In §3, we present an inverse iteration method for solving (1.4). Inverse iteration is an obvious choice, since we want to compute λ nearest zero. A starting guess of μ is not required, but the final value of μ can be computed as a by-product once λ and an eigenvector are known. Equation (1.4) is written as a Lyapunov-like equation, where the eigenvectors are matrices of low rank. The fact that we work with low rank matrices utilizes the efficiency of traditional solvers for large scale Lyapunov equations with low rank right-hand sides. We chose a Lyapunov solver using Arnoldi's method in our numerical tests. This method can be viewed as an inexact inverse iteration method and we can use convergence results for simple eigenvalues in the case that the solution λ corresponds to a Hopf bifurcation. The details of the algorithm that we used in our numerical experiments are explained in §4. Numerical examples, including two physical applications, are given in §5 and support the theory in this paper.

2. Two eigenvalue problems. In this section, we discuss the properties of the $n^2 \times n^2$ generalised eigenvalue problem (1.4) and describe the relationship between its solutions and the solutions of (1.1). We also discuss a reformulation to an equivalent $n \times n$ problem which is more suitable for computations.

We first define the $n^2 \times n^2$ matrices:

$$\begin{aligned} \Delta_0 &= B \otimes M + M \otimes B \\ \Delta_1 &= A \otimes M + M \otimes A \end{aligned}$$

and rewrite (1.4) as

$$(\Delta_1 + \lambda\Delta_0)z = 0 \ . \tag{2.1}$$

The following theorem relates the solutions of (1.4) with those of (1.1).

THEOREM 2.1. *For a given λ, let (μ_j, x_j) be an eigenpair of (1.1). Then*

1. *if $\mu_1 = 0$ is a simple eigenvalue and there are no other eigenpairs of the form $\pm\beta i$ or $\pm\alpha$, then λ is a simple eigenvalue of (2.1) with eigenvector $z = x_1 \otimes x_1$;*
2. *if $\mu_{1,2} = \pm\beta i \in \mathbf{I}$ are two simple purely imaginary eigenvalues and there are no other eigenpairs of the form $\pm\gamma i$ or $\pm\alpha$, and no zero eigenvalue, then λ is a double eigenvalue with eigenvector $z = \xi_1 x_1 \otimes \bar{x}_1 + \xi_2 \bar{x}_1 \otimes x_1$ for any ξ_1, $\xi_2 \in \mathbf{C}$;*
3. *if $\mu_{1,2} = \pm\alpha \in \mathbf{R}$ are two simple real eigenvalues and there are no other eigenpairs of the form $\pm\beta i$ or $\pm\gamma$, and no zero eigenvalue, then λ is a double eigenvalue with eigenvector $z = \xi_1 x_1 \otimes x_2 + \xi_2 x_1 \otimes x_2$ for any ξ_1, $\xi_2 \in \mathbf{C}$.*

Proof. If $\mu = 0$, then

$$(\Delta_1 + \lambda\Delta_0)x_1 \otimes x_1 = (A + \lambda B)x_1 \otimes Mx_1 + Mx_1 \otimes (A + \lambda B)x_1 = 0 \ ,$$

which proves item 1. Items 2 and 3 follow from that $(A + \lambda B)x_1 = \mu M x_1$ and $(A + \lambda B)x_2 = -\mu M x_2$, where $\mu = \mu_1 = -\mu_2$. We can write

$$\begin{aligned}(\Delta_1 + \lambda\Delta_0)x_1 \otimes x_2 &= (A + \lambda B)x_1 \otimes Mx_2 + Mx_1 \otimes (A + \lambda B)x_2 \\ &= (A + \lambda B + \mu M)x_1 \otimes Mx_2 + Mx_1 \otimes (A + \lambda B - \mu M)x_2 \\ &= 0 \otimes Mx_2 + Mx_1 \otimes 0 = 0 \ .\end{aligned}$$

□

We now show that the converse of this theorem holds if M is nonsingular.

THEOREM 2.2. *Let (λ, z) be an eigenpair of (2.1) and M be non-singular. Then one of the following situations is true.*

1. *If λ is simple, then $z = x \otimes x$ where $(\mu = 0, x)$ is an eigenpair of (1.1).*
2. *If λ is a double eigenvalue, then there are $\mu \in \mathbf{C}$ and x_1 and $x_2 \in \mathbf{C}^n$ so that (μ, x_1) and $(-\mu, x_2)$ are simple eigenpairs of (1.1) and there are ξ_1 and $\xi_2 \in \mathbf{C}$ so that $z = \xi_1 x_1 \otimes x_2 + \xi_2 x_2 \otimes x_1$.*

Proof. Let $A + \lambda B = MX\Gamma X^{-1}$ be a Jordan canonical form associated with (1.1), where $X = [x_1, \ldots, x_n]$ and $\mu_1, \ldots, \mu_n$ are the main diagonal elements of Γ, then (2.1) is equivalent to

$$(MX \otimes MX)(\Gamma \otimes I + I \otimes \Gamma)(X^{-1} \otimes X^{-1})z = 0 \ .$$

Since $MX \otimes MX$ and $X^{-1} \otimes X^{-1}$ have full rank,

$$\Gamma \otimes I + I \otimes \Gamma$$

has at least one zero main diagonal element. The main diagonal elements of $\Gamma \otimes I + I \otimes \Gamma$ are $\mu_j + \mu_i$ where μ_j are eigenvalues of (1.1). Since $\Gamma \otimes I + I \otimes \Gamma$ is upper triangular, $\mu_j + \mu_i$ are eigenvalues of $(A + \lambda B) \otimes M + M \otimes (A + \lambda B)$ with associated eigenvectors $z = x_i \otimes x_j$ and $z = x_j \otimes x_i$.

If λ is a double eigenvalue, the two eigenvalues correspond to $\mu_1 + \mu_2 = 0$ and $\mu_2 + \mu_1 = 0$. Since A, B and M are real, we must have that $\mu_{1,2} = \pm\alpha$ or $\mu_{1,2} = \pm\beta i$ (possibly zero). The eigenvectors of the double eigenvalue zero must have the form $\xi_1 x_1 \otimes x_2 + \xi_2 x_2 \otimes x_1$. If $\mu_{1,2} = \pm\beta i$, then $x_2 = \bar{x}_1$.

If λ is a simple eigenvalue, $\mu_1 + \mu_2$ can only be simple and zero when $\mu_1 = \mu_2 = 0$. The associated eigenvector is $z = x \otimes x$ where x is the corresponding eigenvector of (1.1).

This proves the theorem. □

Theorems 2.1 and 2.2 show the correspondence between purely imaginary eigenpairs of (1.1) and solutions of (2.1). By first writing the eigenvectors of (2.1) as $n \times n$ matrices so that $z = \text{vec}(Z)$, where vec is a function that puts all columns of Z in a vector, we can write (2.1) as

$$MZA^T + AZM^T + \lambda(MZB^T + BZM^T) = 0 \ . \tag{2.2}$$

Although we do not solve this equation directly, the introduction of the matrix notation of the eigenvector is going to be useful for the inverse iteration method that we introduce further. In the remainder of the text, we call both z and Z an eigenvector.

THEOREM 2.3. *Assume that λ is a real eigenvalue of (2.1), and the conditions of Theorem 2.1 are satified. Then:*

1. *if $\mu = 0$ is simple, Z is a symmetric matrix of rank 1, i.e. $Z = xx^T$;*
2. *if $\mu = \pm\beta i$, there is a real symmetric eigenvector of rank two $Z = xx^* + \bar{x}x^T$, which is unique up to a scalar factor.*
3. *if $\mu = \pm\alpha$ is real, there is a symmetric eigenvector of rank two $Z = x_1x_2^T + x_2x_1^T$, which is unique up to a scalar factor;*

If Z is rank 2, it is indefinite.

Proof. The proof for $\mu = 0$ follows from $z = x \otimes x$ and so $Z = xx^T$. For μ real or purely imaginary, we know that the eigenvectors take the form

$$Z = \xi_1 x_1 x_2^T + \xi_2 x_2 x_1^T \ ,$$

where x_1 and x_2 are not parallel. If ξ_1 or ξ_2 is zero, Z has rank one. We now prove that for a real symmetric Z, the rank is two. From $Z - Z^T = 0$ we derive $(\xi_1 - \xi_2)x_1x_2^T + (\xi_2 - \xi_1)x_2x_1^T = 0$ so that $\xi_1 = \xi_2$. This implies that for a real symmetric Z, we always have a rank two matrix. Note that for case 2, $x_1 = x$ and $x_2 = \bar{x}$.

We can rewrite

$$Z = x_1x_2^T + x_2x_1^T \tag{2.3}$$

$$= (x_1 + x_2)(x_1 + x_2)^T - (x_1 - x_2)(x_1 - x_2)^T \ , \tag{2.4}$$

which has one positive semi-definite term and one negative semi-definite term, so Z has one positive and one negative eigenvalue (and $n - 2$ zero eigenvalues), which concludes the proof of the theorem. ∎

The observation of Theorem 2.3 is very significant for our work. From the theorem we may conclude that for the three cases considered in Theorems 2.1, 2.2 and 2.3, λ is a simple eigenvalue when (2.1) is restricted to the subspace of symmetric eigenvectors Z. This observation is important, since theory for eigenvalue solvers is significantly easier for simple than for double eigenvalues.

The solution of (2.1) restricted to the symmetric eigenvector space is related to, but distinct from, the method discussed in [17] and [13, §§4.4–4.5], where it is suggested to solve (2.1) restricted to the anti-symmetric eigenvector space. The advantage is that a simple zero μ cannot produce an eigenvalue λ in this space, since its corresponding eigenvector, Z, is always symmetric (Theorem 2.3, case 1). Hence, solving (2.1) restricted to the anti-symmetric eigenvector space avoids the computation of a simple zero μ. In contrast, we have chosen the restriction to symmetric eigenvectors, since the inverse iteration method is then related to the solution of a Lyapunov equation, which is a rather well-known problem.

A natural representation of a low rank symmetric matrix is its truncated eigendecomposition. In practice, we will write Z as $Z = VDV^T$ where D is a diagonal matrix and $V^TV = I$. For a rank two Z, D is a 2×2 matrix and V is $n \times 2$.

Once Z and λ are computed, the computation of μ readily follows from the solution of the two by two problem

$$V^T(A + \lambda B)Vy = \mu V^T MVy \ . \tag{2.5}$$

Indeed, since

$$Z = xx^* + \bar{x}x^T = VDV^T \ ,$$

x and $\bar{x}$ are spanned by the columns of V.

When Δ_0 is singular, $\lambda = \infty$ is an eigenvalue with the nullspace of Δ_0 as eigenspace. Note that in case of a singular B, μ can take any finite value where the eigenvectors lie in the common nullspace. Here is a simple example.

EXAMPLE 1. *Let*

$$A = \begin{bmatrix} 2 & -1 & \\ 1 & 2 & \\ & & 3 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad M = I \ .$$

The eigenvalues of the pencil $\Delta_0 + \lambda\Delta_1$ are -2 (double eigenvalue), $-2 \pm i$ (double eigenvalues), $-5 \pm i$ (double eigenvalues), and ∞. For $\lambda = -2$, we have $\mu = \pm i$ (which indicates a Hopf point), For $\lambda = -2+i$, we have $\mu = 2i, i$. Complex λ have no physical meaning and are discarded. With the infinite λ, we can associate any finite μ with eigenvector $e_3 \otimes e_3$.

Eigenvectors associated with $\lambda = -2$ are $(1, -i, 0)\otimes(1, i, 0)$ and $(1, i, 0)\otimes(1, -i, 0)$.

3. Inverse iteration. In this section we describe the inverse iteration method (Algorithm 3.2) [11, §7.6.1] to find the solution to (2.1) (equivalently (2.2)) nearest to zero. The algorithmic details are given in §4.

As is standard, the accuracy of an approximate eigenpair is measured by the residual norm. Let $r = \Delta_1 z + \lambda\Delta_0 z$ and

$$R = (A + \lambda B)ZM^T + MZ(A + \lambda B)^T \tag{3.1}$$

so that $r = \mathrm{vec}(R)$. The eigenvalue approximation λ is, as usual, computed by the Rayleigh quotient

$$\lambda = \frac{z^*\Delta_1 z}{z^*\Delta_0 z} \ . \tag{3.2}$$

ALGORITHM 3.1 (Inverse iteration).

1. Given $\mathbf{z}_0 \in \mathbf{C}^{n^2}$.
2. For $j = 1, 2, \ldots,$ do:
   2.1. Solve $\mathbf{y}_j$ from $\Delta_1\mathbf{y}_j = \Delta_0\mathbf{z}_{j-1}$.
   2.2. Normalize: $\mathbf{z}_j = \mathbf{y}_j/\|\mathbf{y}_j\|$.

However, applying inverse iteration directly to (2.1) is obviously a bad idea, since we work with n^2-vectors, which is not feasible when n is large.

To overcome this difficulty, we convert the inverse iteration method into one for n-dimensional problems using the Lyapunov notation that we introduced in last section,

see (2.2). We shall see that this allows the opportunity for significant computational savings. Using $z_j = \mathrm{vec}(Z_j)$. Step 2.1 in Algorithm 3.1 becomes

$$AY_jM^T + MY_jA^T = BZ_{j-1}M^T + MZ_{j-1}B^T \ . \tag{3.3}$$

By inverting M or A, this becomes a Lyapunov equation. Let us assume that we invert A, we then obtain

$$Y_jS^T + SY_j = TZ_{j-1}S^T + SZ_{j-1}T^T \tag{3.4}$$

where $S = A^{-1}M$ and $T = A^{-1}B$. Note that the right-hand side in (3.4) is symmetric. This leads to a recasting of Algorithm 3.1 as follows:

ALGORITHM 3.2 (Inverse iteration).

1. Given $x_0 \in \mathbf{C}^n$.
2. Let $Z_0 = x_0x_0^T$.
3. For $j = 1, 2, \ldots,$ do:
   3.1. Compute the right-hand side $F_{j-1} = TZ_{j-1}S^T + SZ_{j-1}T^T$
   3.2. Solve Y_j from $Y_jS^T + SY_j = F_{j-1}$
   2.2. Normalize: $Z_j = Y_j/\|Y_j\|_F$

In the first iteration of Algorithm 3.2, the right-hand side

$$F_0 = (Tx_0)(Sx_0)^T + (Sx_0)(Tx_0)^T$$

has rank two (assuming that $Bx_0 \neq Mx_0$) and is symmetric. Note that F_0 is indefinite, since it can be written as

$$F_0 = \frac{1}{2}(Tx_0 + Sx_0)(Tx_0 + Sx_0)^* - \frac{1}{2}(Tx_0 - Sx_0)(Tx_0 - Sx_0)^* \ .$$

Now, though F_0 has rank two, there is no reason why Y_1 should have rank two. However, we hope the eigenvalues of Y_1 quickly decay, so that a good low rank approximation of Y_j is feasible, and it is this feature that provides the computational savings. The solution of Lyapunov equations often can be approximated by low rank matrices for applications arising from dynamical systems [30] [1]. The solution of the Lyapunov equation is not unique when S has eigenvalues of opposite sign. These correspond indeed to zero eigenvalues of Δ_1. Since we assumed that the eigenvalues of $M^{-1}A$ lie in the negative half plane, the eigenvalues of $A^{-1}M$ have the same properties. As a result the solution of (3.4) is unique.

LEMMA 3.1. *Assume that the solution of (3.4) is unique. In each iteration of Algorithm 3.2, Z_j is real symmetric.*

Proof. For $j = 1$, the right-hand side F_0 is symmetric. By transposing (3.4), we see that Y_1^T is also a solution to (3.4). Since the solution is unique, $Y_1 = Y_1^T$.

By induction, we can now show that all iterates are real symmetric. This proves the lemma. □

If the sequence $\{Z_j\}_{j\geq 0}$ converges, it converges to a symmetric matrix. We thus compute an eigenvector of (2.1) that is symmetric. From Theorem 2.3, we know that this eigenvector is unique, i.e. the desired eigenvalue is simple. Note that if we have a random starting vector x_0, Algorithm 3.2 converges to the λ nearest zero and its associated eigenvector.

Since Y_j is symmetric, we represent it by its eigendecomposition $Y_j = V_jD_jV_j^T$, where D_j is a diagonal matrix and $V_j^TV_j = I$. The fact that Y_j (and Z_j) converge to a rank two matrix and that the solution of Lyapunov equations often have quickly

decaying eigenvalues, means that we can truncate the eigendecomposition of Y_j and keep a low rank approximation without introducing large errors. This is the first approximation that we make in the method. Truncation introduces a residual in the Lypunov equation (3.4). Let $Y_j = V_p D_p V_p^T + V_{n-p} D_{n-p} V_{n-p}^T$, then (3.4) becomes

$$S(V_p D_p V_p^T) + (V_p D_p V_p^T) S^T = F_{j-1} + G_j \tag{3.5}$$

with

$$G_j = -\{S(V_{n-p} D_{n-p} V_{n-p}^T) + (V_{n-p} D_{n-p} V_{n-p}^T) S^T\} \ .$$

The term G_j becomes a residual term in

$$\Delta_1 y_j = \Delta_0 z_{j-1} + \mathrm{vec}(G_j) \ .$$

From inexact inverse iteration theory [22] [23] [12] [8] [9], we know that inexact inverse iteration converges to the simple eigenvalue nearest zero when $\|G_j\|_F \leq \tau \|r_{j-1}\|_2$ for all $j \geq 1$ where $r_j = \Delta_1 z_j + \lambda_j \Delta_0 z_j$ is the eigenvalue residual. We will not discuss in this paper how the truncation can be carried out to guarantee that $\|G_j\|_F$ is smaller than the prescribed tolerance. The important point is that the tolerance for $\|G_j\|$ decreases when eigenvalue estimates become more accurate.

Assuming that we have n^2 independent eigenvectors for (2.1), we can decompose Z_j into

$$Z_j = \sum_{l,m=1}^{n} \alpha_j^{(l,m)} x_l x_m^T \ ,$$

where $x_m \otimes y_l$ for $l, m = 1, \ldots, n$ are the eigenvectors of (2.1). At convergence, Z_j has rank two, i.e. with appropriate numbering of the eigenpairs of (2.1), we have

$$\lim_{j \to \infty} Z_j = x_1 x_2^T + x_2 x_1^T \ .$$

The error on Z_j,

$$\Xi_j = Z_j - x_1 x_2^T - x_2 x_1^T$$

is a linear combination of the 'other' eigenvectors. The matrix Z_j has thus two large eigenvalues and $n-2$ smaller eigenvalues depending on $\|\Xi_j\|$. Since $\|\Xi_j\|$ converges to zero, truncating small eigenvalues in Z_j becomes easier and easier.

Each iteration requires the solution of a Lyapunov equation. Methods for small dense problems such as the Bartels and Stewart [3] method are not feasible for large sparse problems of the type we consider here. Other methods are based on ADI and the Smith method [29]. We refer to [18] for an overview. In the context of this paper, Krylov methods are perhaps most appropriate since we can perform linear solves with A and matrix vector products with B and M [34] [35] [31]. Such a Lyapunov solver introduces an additional residual term in (3.5). The size of this residual term can be controlled by the Lyapunov solver.

4. Algorithmic details. Here we discuss some of the details of the implementation of Algorithm 3.2.

We first rewrite Algorithm 3.2 in terms of the eigendecomposition of Z_j. The method now looks as follows:

ALGORITHM 4.1 (Inverse iteration).

1. Given $V_0 = [x_0] \in \mathbf{C}^{n\times 1}$ and $D_0 = 1$, i.e. $(Z_0 = x_0 x_0^T)$
2. For $j = 1, 2, \ldots,$ do:
   2.1. Compute the eigendecomposition of the right-hand side

$$P_{j-1}C_{j-1}P_{j-1}^T = (TV_{j-1})D_{j-1}(SV_{j-1})^T + (SV_{j-1})D_{j-1}(TV_{j-1})^T$$

   2.2. Approximate $Y_j = V_j D_j V_j^T$ with $Y_j S^T + SY_j = P_{j-1}C_{j-1}P_{j-1}^T$
   2.3. Truncate to rank p: $Y_j = V_p D_p V_p^T$ using tolerance τ.
   2.4. Normalize: $D_j = D_j/\|D_j\|_F$.
   2.5. Compute λ_j by the Rayleigh quotient (3.2).

We now explain the different steps in detail. For Step 2.1, we first write the right-hand side as

$$\frac{1}{2}(TV_j + SV_j)D_j(TV_j + SV_j) - \frac{1}{2}(TV_j - SV_j)D_j(TV_j - SV_j) \ .$$

We then decompose

$$\frac{1}{2}[SV_j + TV_j, SV_j - TV_j] = QR$$

by a QR factorization and compute the eigendecomposition of $(R \cdot \mathrm{diag}(D_j, -D_j) \cdot R^*)U = UC_j$. Then $P_j = QU$.

In Step 2.3, we drop the $n-p$ smallest eigenvalues of Y_j. This reduces the number of columns of V_j to p. Let the diagonal elements of D_j be ordered in decreasing modulus, then we truncate

$$Y_j = [V_p \ V_{k-p}] \begin{bmatrix} D_p & \\ & D_{k-p} \end{bmatrix} [V_p \ V_{k-p}]^T \approx V_p D_p V_p^T \ ,$$

so that $\|D_{k-p}\|_2 \leq \tau \|D_p\|_2$.

An issue that is also of importance for the inexact inverse iteration method for the standard eigenvalue problem $Ax = \lambda x$, is that the error on the iterative solve should decrease when the residual norm of the approximate eigenpair decreases. For standard eigenproblems, inverse iteration requires the solution of

$$Ax_j = x_{j-1} \ .$$

Clearly, x_{j-1} would be an excellent starting vector (up to a scalar factor), since at some point, x_j makes a small angle with x_{j-1}. Using an Arnoldi based method, the Krylov space

$$\mathrm{span}\{x_{j-1}, Ax_{j-1}, \ldots, A^{k-1}x_{j-1}\}$$

is built. This space contains x_{j-1}, and so the solution x_j will pick this up. This implies that even for a fixed number of iterations the residual norm gradually decreases [23].

We have a similar situation here. The eigendecomposition of the right-hand side of (3.4) is $P_j C_j P_j^T$ where

$$\mathrm{Range}(P_j) = \mathrm{Range}(SV_j) + \mathrm{Range}(TV_j) \ .$$

Since

$$\begin{aligned} F_j &= SZ_jT^T + TZ_jS^T \\ &= -\lambda_j^{-1}(SZ_j + Z_jS^T) + \lambda_j^{-1}((I+\lambda_j T)Z_jS^T + SZ_j(I+\lambda_j T)^T \\ &= -\lambda_j^{-1}(SZ_j + Z_jS^T) + \lambda_j^{-1}A^{-1}R_jA^{-T} \ , \end{aligned}$$

where R_j is defined by (3.1) with $Z = Z_j$ and $\lambda = \lambda_j$. The last term in F_j is proportional to the eigenvalue residual norm and decreases to zero, whereas the first term becomes more and more dominant in the right-hand side F_j.

Suppose that $F_j = -\lambda_j^{-1}(SZ_j + Z_jS^T)$, then

$$\text{Range}(P_j) = \text{Range}(V_j) + \text{Range}(SV_j) \ .$$

Since P_j is used as starting vectors for the Block Arnoldi method, V_j is in the subspace. This provides a good starting guess for the Lyapunov solver in the Krylov space itself. From the numerical examples, we will indeed see a reducing residual norm for the Lyapunov equation.

The technically most difficult part is Step 2.2. We chose to solve (3.3) by the Lyapunov equation (3.4). In this paper, we solve (3.4) from the block Krylov space for the starting vectors P_j and the matrix S.

Methods for solving Lyapunov equations with rank one right-hand sides are sometimes solved by Arnoldi's method. For rank p right-hand sides, we use the block Arnoldi method with p starting vectors. The Arnoldi method builds an orthogonal basis for the Krylov space

$$\text{span}\{w_1, Sw_1, S^2w_1, \ldots, S^{k-1}w_1\}$$

and a $k \times k$ upper Hessenberg matrix $H_k = W_k^T S W_k$. Here is an outline of the Arnoldi algorithm:

ALGORITHM 4.2 (Arnoldi).

1. Given w_1 with $\|w_1\|_2 = 1$
2. For $j = 1, \ldots, k$
   - 2.1. Compute $\tilde{w}_j = Sw_j$.
   - 2.2. Compute $h_j = W_j^* \tilde{w}_j \in \mathbf{R}^j$ and $\hat{w}_j = \tilde{w}_j - W_j h_j$
   - 2.3. Compute $\beta_j = \|\hat{w}_j\|_2$ and define $w_{j+1} = \hat{w}_j / \beta_j$
   - 2.4. Let $W_{j+1} = [W_j \ w_{j+1}]$.

When we collect the matrices h_j in $H_k \in \mathbf{R}^{k\times k}$, i.e. $H_k = [h_1, \ldots, h_k]$, then we have

$$SW_k - W_k H_k = w_{k+1}\beta_k e_k^T \tag{4.1}$$

where e_k is the last column of the $k \times k$ identity matrix.

Suppose we want to solve the Lyapunov equation

$$SY + YS^* = w_1 w_1^* \ . \tag{4.2}$$

Let W_k and H_k be computed by the Arnoldi method with starting vector w_1. We look for a solution of the form $Y = W_k X W_k^*$. First substituting this into (4.2), and then using (4.1), we have

$$\begin{aligned} (SW_k)XW_k^* + W_kX(SW_k)^* &= w_1 w_1^T \\ W_k(H_kX + XH_k^* - e_1e_1^*)W_k^* &= -w_{k+1}\beta_k e_k^* X W_k^* - W_k X e_k \beta_k w_{k+1}^* \ . \end{aligned}$$

Multiplication on the left and the right by W_k^* and W_k respectively, produces the Lyapunov equation of order k

$$H_k X + X H_k^* = e_1 e_1^* \ .$$

If k is much smaller than n, this is a cheap problem to solve.

In the block Arnoldi method, we build the Krylov space

$$\text{span}\{P_j, SP_j, \ldots, S^{k/p-1} P_j\}$$

which is a subspace of dimension smaller than or equal to k. The extension to the block Arnoldi method is easy, although rather technical, and therefore omitted, see [31] for technical details.

5. Numerical examples. We present three examples to illustrate the theory in this paper. The first is contrived to illustrate two important points, the second and third are from physical applications.

5.1. Illustration of the theory by a small problem. We use $n \times n$ matrices with $n = 100$, $M = B = I$ and A a block diagonal matrix with 2×2 blocks on the main diagonal such that the eigenvalues of A are $\eta_{1,2} = 1 \pm 100i$, $\eta_{2j-1,2j} = j \pm i$ for $j = 2, \ldots, n/20$. Since $\Delta_0 = 2I$, the solutions of (1.4) are $\lambda = -\frac{1}{2}(\eta_j + \eta_i)$ for $i, j = 1, \ldots, n/2$. Each eigenvalue has multiplicity two. There are $n/2$ real λ's: $-j$ for $j = 1, \ldots, n/2$. Since we use real arithmetic in the inverse iteration method, we can only converge to a real eigenvalue. By construction, the λ's nearest zero are -1 and -2. The corresponding μ's are $\pm 100i$ and $\pm i$ respectively.

From earlier work [4] [10] [5] [25], we know that detecting right-most eigenvalues is not an easy task. The reason is that most eigenvalue eigenvalue solvers search for eigenvalues near a point, called a shift, or a target. A key advantage of the approach described here is that our method converges to the λ nearest zero (assuming we solve the Lyapunov equation exactly or, at least, accurately). The actual value of μ is not important, indeed it isn't used by the method, but arises as a by-product, described by (2.5). This is a significant advantage over other methods where a good starting value of μ is required. If we apply Algorithm 4.1 and solve the Lyapunov equation exactly, we have the convergence behaviour as is shown in Table 5.1. The results are in line with the theory in this paper.

The second important point illustrated by this model example concerns the accuracy of the solves in Step 2.2 of Algorithm 4.1. When we use a Krylov method in Step 2.2 of Algorithm 4.1, convergence behaviour may depend on k, the dimension of the Krylov subspace, as is shown in Tables 5.2 and 5.3. When we use twenty Krylov vectors, we converge to the wrong value of λ (namely to -2), and hence would calculate an incorrect value of μ. The reason for this is that the Lyapunov solver builds a Krylov space with $A^{-1}M$, which is rich in the eigenvectors associated with the eigenvalues near zero. The eigenvalues $1 \pm 100i$ lie quite far from zero and it may take a while before they converge. The associated eigenvectors do not participate in the Lyapunov solution, so the eigenvalue solver does not 'see' these eigenvectors. Indeed, Arnoldi's method for $A^{-1}M$, is, as we know, not reliable for computing eigenvalue with large imaginary parts [25]. However, when we repeat the experiment with $k = 40$, we recover the behaviour achieved in Table 5.1, and hence converge to the desired values of λ. As a consequence, the correct value of μ is found.

Iteration	rank p	λ	$\|(\Delta_1 + \lambda\Delta_2)x\|_2$
1	12	-16.8607	$4 \cdot 10^{1}$
2	13	-1.75552	$1.7 \cdot 10^{0}$
3	13	-1.02289	$5.7 \cdot 10^{-1}$
4	12	-1.00186	$1.3 \cdot 10^{-1}$
5	11	-1.00024	$4.2 \cdot 10^{-2}$
6	10	-1.00004	$1.6 \cdot 10^{-2}$
7	9	-1.00001	$6.6 \cdot 10^{-3}$
8	8	-1	$3 \cdot 10^{-3}$
9	7	-1	$1.4 \cdot 10^{-3}$
10	7	-1	$6.6 \cdot 10^{-4}$
11	7	-1	$3.2 \cdot 10^{-4}$
12	6	-1	$1.6 \cdot 10^{-4}$
13	6	-1	$7.7 \cdot 10^{-5}$
14	5	-1	$3.8 \cdot 10^{-5}$
15	5	-1	$1.9 \cdot 10^{-5}$
16	4	-1	$9.3 \cdot 10^{-6}$
17	4	-1	$4.6 \cdot 10^{-6}$
18	4	-1	$2.3 \cdot 10^{-6}$
19	4	-1	$1.1 \cdot 10^{-6}$
20	4	-1	$5.7 \cdot 10^{-7}$
21	4	-1	$2.9 \cdot 10^{-7}$
22	4	-1	$1.4 \cdot 10^{-7}$
23	4	-1	$7.1 \cdot 10^{-8}$
24	4	-1	$3.6 \cdot 10^{-8}$
25	2	-1	$7.8 \cdot 10^{-14}$

TABLE 5.1
Convergence when using an exact Lyapunov solver

5.2. The Olmstead model. The mathematical model represents the flow of a layer of viscoelastic fluid heated from below [27] [15]. The equations are

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

where u represents the speed of the fluid and v is related to viscoelastic forces. The boundary conditions are $u(0) = u(1) = 0$ and $v(0) = v(1) = 0$. After discretization with central differences with grid-size $h = 1/(n/2+1)$, the equations may be written as $\dot{x} = f(x)$ with $x = [u_1, v_1, u_2, v_2, \ldots, u_{N/2}, v_{N/2}]^T$. We consider the Jacobian $A + \lambda B = \partial f/\partial x$ for $n = 2,500$, $B = 2$, $C/ = 0.1$ and $\lambda = R \in [0.6, 5]$, evaluated in the trivial steady state solution.

We used Algorithm 4.1 to solve this problem with the following parameters. We discretized the problem around $R = 4$. We used at most $k = 40$ Krylov vectors in the solution of the Lyapunov equation. We used the tolerance $\tau = 10^{-4}$ to drop the singular values of the solution of the Lyapunov equation in Step 3.3 in Algorithm 4.1.

Iteration	$\text{rank}(V_j)$	λ_j	$\|G_j\|_F$	$\|(\Delta_1 + \lambda\Delta_0)\text{vec}(V_j D_j V_j^*)\|_2$
1	10	-17.1375	$5.6 \cdot 10^{-4}$	$3.9 \cdot 10^{1}$
2	10	-3.16988	$4.6 \cdot 10^{-4}$	$1.9 \cdot 10^{0}$
3	10	-2.23436	$1.2 \cdot 10^{-3}$	$3 \cdot 10^{0}$
4	10	-2.67188	$6.9 \cdot 10^{-4}$	$2.2 \cdot 10^{0}$
5	10	-2.34886	$1.8 \cdot 10^{-4}$	$8.9 \cdot 10^{-1}$
6	10	-2.17747	$1.4 \cdot 10^{-5}$	$5.5 \cdot 10^{-1}$
7	10	-2.09318	$8.4 \cdot 10^{-7}$	$4.7 \cdot 10^{-1}$
8	10	-2.05086	$6.3 \cdot 10^{-8}$	$4.1 \cdot 10^{-1}$
9	10	-2.0288	$1.4 \cdot 10^{-8}$	$3.7 \cdot 10^{-1}$
10	10	-2.01681	$4.2 \cdot 10^{-9}$	$3.2 \cdot 10^{-1}$
11	10	-2.01005	$1.3 \cdot 10^{-9}$	$2.9 \cdot 10^{-1}$
12	9	-2.00611	$7.2 \cdot 10^{-10}$	$2.6 \cdot 10^{-1}$
13	9	-2.00376	$2.2 \cdot 10^{-10}$	$2.3 \cdot 10^{-1}$
14	8	-2.00234	$5.4 \cdot 10^{-10}$	$2.1 \cdot 10^{-1}$
15	8	-2.00147	$3.9 \cdot 10^{-11}$	$1.9 \cdot 10^{-1}$
16	7	-2.00092	$8.5 \cdot 10^{-10}$	$1.7 \cdot 10^{-1}$
17	6	-2.00058	$1.3 \cdot 10^{-9}$	$1.5 \cdot 10^{-1}$
18	6	-2.00037	$1.2 \cdot 10^{-10}$	$1.3 \cdot 10^{-1}$
19	6	-2.00023	$5.9 \cdot 10^{-11}$	$1.2 \cdot 10^{-1}$
20	6	-2.00015	$2.8 \cdot 10^{-11}$	$1.1 \cdot 10^{-1}$
21	6	-2.00009	$1.3 \cdot 10^{-11}$	$9.6 \cdot 10^{-2}$
22	6	-2.00006	$6 \cdot 10^{-12}$	$8.6 \cdot 10^{-2}$
23	6	-2.00004	$2.7 \cdot 10^{-12}$	$7.7 \cdot 10^{-2}$
24	5	-2.00002	$1.7 \cdot 10^{-9}$	$6.9 \cdot 10^{-2}$
25	5	-2.00002	$1.4 \cdot 10^{-10}$	$6.1 \cdot 10^{-2}$
26	5	-2.00001	$7.6 \cdot 10^{-11}$	$5.5 \cdot 10^{-2}$
27	5	-2.00001	$4.1 \cdot 10^{-11}$	$4.9 \cdot 10^{-2}$
28	4	-2	$1.1 \cdot 10^{-9}$	$4.4 \cdot 10^{-2}$
29	4	-2	$4.2 \cdot 10^{-11}$	$3.9 \cdot 10^{-2}$
30	4	-2	$2.1 \cdot 10^{-11}$	$3.5 \cdot 10^{-2}$
31	4	-2	$1 \cdot 10^{-11}$	$3.1 \cdot 10^{-2}$
32	4	-2	$5.1 \cdot 10^{-12}$	$2.8 \cdot 10^{-2}$
33	4	-2	$2.5 \cdot 10^{-12}$	$2.5 \cdot 10^{-2}$
34	4	-2	$1.2 \cdot 10^{-12}$	$2.3 \cdot 10^{-2}$
35	4	-2	$6 \cdot 10^{-13}$	$2 \cdot 10^{-2}$
36	4	-2	$2.9 \cdot 10^{-13}$	$1.8 \cdot 10^{-2}$
37	4	-2	$1.4 \cdot 10^{-13}$	$1.6 \cdot 10^{-2}$
38	4	-2	$6.9 \cdot 10^{-14}$	$1.4 \cdot 10^{-2}$
39	4	-2	$3.3 \cdot 10^{-14}$	$1.3 \cdot 10^{-2}$
40	4	-2	$1.6 \cdot 10^{-14}$	$1.2 \cdot 10^{-2}$
41	4	-2	$7.8 \cdot 10^{-15}$	$1 \cdot 10^{-2}$
42	4	-2	$3.8 \cdot 10^{-15}$	$9.2 \cdot 10^{-3}$
43	4	-2	$2.2 \cdot 10^{-15}$	$8.2 \cdot 10^{-3}$
44	4	-2	$9.1 \cdot 10^{-16}$	$7.4 \cdot 10^{-3}$
45	3	-2	$2 \cdot 10^{-9}$	$6.6 \cdot 10^{-3}$
46	2	-2	$1.7 \cdot 10^{-9}$	$5.9 \cdot 10^{-3}$
47	2	-2	$1.1 \cdot 10^{-11}$	$5.3 \cdot 10^{-3}$
48	2	-2	$6.7 \cdot 10^{-12}$	$4.7 \cdot 10^{-3}$
49	2	-2	$4.3 \cdot 10^{-12}$	$4.2 \cdot 10^{-3}$
50	2	-2	$2.8 \cdot 10^{-12}$	$3.8 \cdot 10^{-3}$

TABLE 5.2
Convergence when using a Krylov Lyapunov solver with $k = 20$

Iteration	rank(V_j)	λ_j	$\|G_j\|_F$	$\|(\Delta_1 + \lambda\Delta_0)\text{vec}(V_j D_j V_j^*)\|_2$
1	12	-16.8607	$3.4 \cdot 10^{-9}$	$4 \cdot 10^{1}$
2	13	-1.75552	$4.8 \cdot 10^{-10}$	$1.7 \cdot 10^{0}$
3	13	-1.02289	$5.9 \cdot 10^{-10}$	$5.7 \cdot 10^{-1}$
4	12	-1.00186	$8.4 \cdot 10^{-10}$	$1.3 \cdot 10^{-1}$
5	11	-1.00024	$1.1 \cdot 10^{-9}$	$4.2 \cdot 10^{-2}$
6	10	-1.00004	$9 \cdot 10^{-10}$	$1.6 \cdot 10^{-2}$
7	9	-1.00001	$9 \cdot 10^{-10}$	$6.6 \cdot 10^{-3}$
8	8	-1	$1.4 \cdot 10^{-9}$	$3 \cdot 10^{-3}$
9	7	-1	$2.8 \cdot 10^{-9}$	$1.4 \cdot 10^{-3}$
10	7	-1	$3.3 \cdot 10^{-10}$	$6.6 \cdot 10^{-4}$
11	7	-1	$1 \cdot 10^{-10}$	$3.2 \cdot 10^{-4}$
12	6	-1	$1.3 \cdot 10^{-9}$	$1.6 \cdot 10^{-4}$
13	6	-1	$9.8 \cdot 10^{-11}$	$7.7 \cdot 10^{-5}$
14	5	-1	$3.8 \cdot 10^{-9}$	$3.8 \cdot 10^{-5}$
15	5	-1	$2.3 \cdot 10^{-10}$	$1.9 \cdot 10^{-5}$
16	4	-1	$2.3 \cdot 10^{-9}$	$9.3 \cdot 10^{-6}$
17	4	-1	$5.5 \cdot 10^{-11}$	$4.6 \cdot 10^{-6}$
18	4	-1	$1.7 \cdot 10^{-11}$	$2.3 \cdot 10^{-6}$
19	4	-1	$5.3 \cdot 10^{-12}$	$1.1 \cdot 10^{-6}$
20	4	-1	$1.7 \cdot 10^{-12}$	$5.7 \cdot 10^{-7}$
21	4	-1	$5.2 \cdot 10^{-13}$	$2.9 \cdot 10^{-7}$
22	4	-1	$1.6 \cdot 10^{-13}$	$1.4 \cdot 10^{-7}$
23	4	-1	$5 \cdot 10^{-14}$	$7.1 \cdot 10^{-8}$
24	4	-1	$1.6 \cdot 10^{-14}$	$3.6 \cdot 10^{-8}$
25	2	-1	$7.1 \cdot 10^{-9}$	$9.1 \cdot 10^{-13}$

TABLE 5.3
Convergence when using a Krylov Lyapunov solver with $k = 40$

Table 5.4 shows rank(V_j), the residual norm

$$\|(\Delta_1 + \lambda\Delta_0)\text{vec}(V_j D_j V_j^*)\|_2$$

versus the iteration count. Note that $\lambda = 0.447833$ corresponds to $R = 4.447833$. The final value of μ is $\pm 4.18512i$. The residual norm

$$\|Ax + \lambda Bx + \mu Mx\|_2 = 8.3\,10^{-10}\ .$$

From Table 5.4, we notice that the Lyapunov residual norm $\|G_j\|_F$ tends to zero and that the rank of the eigenvector goes to two.

We analyse the iterations 3 and 15 in more detail. Iteration three uses a block Krylov method with a block size $28 = 2 \times 14$. We perform only 1 iteration, so that the number of vectors is bounded by $k = 40$. Iteration fifteen uses a block Krylov method with block size 4. We thus perform 10 iterations, which requires $k = 40$ vectors. Table 5.5 shows the eigenvalues of Z_j for two different iteration numbers: one in the first iterations, where p is large and one in the last iterations, where $p = 2$.

5.3. The 2D Brusselator model. The trimolecular reaction scheme in a two-dimensional square reactor can be studied by the Brusselator model. For more details, see [6, Chapter 5].

TABLE 5.4
Result for the Olmstead equation for $R = 4$

Iteration	rank(V_j)	λ_j	$\|G_j\|_F$	$\|(\Delta_1 + \lambda\Delta_0)\mathrm{vec}(V_j D_j V_j^*)\|_2$
1	14	0.173873	$2.5 \cdot 10^{-3}$	$4.9 \cdot 10^{3}$
2	10	2.82629	$2.8 \cdot 10^{-5}$	$4.3 \cdot 10^{-1}$
3	8	2.67418	$2.2 \cdot 10^{-5}$	$4.1 \cdot 10^{-1}$
4	6	1.16866	$3.4 \cdot 10^{-5}$	$1.4 \cdot 10^{0}$
5	4	0.474148	$5.5 \cdot 10^{-4}$	$9.2 \cdot 10^{-1}$
6	4	0.448515	$5.9 \cdot 10^{-5}$	$1.8 \cdot 10^{-1}$
7	4	0.447851	$1.2 \cdot 10^{-5}$	$2.9 \cdot 10^{-2}$
8	4	0.447834	$1.2 \cdot 10^{-6}$	$4.6 \cdot 10^{-3}$
9	2	0.447833	$2.1 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$
10	2	0.447833	$1.8 \cdot 10^{-13}$	$2.1 \cdot 10^{-6}$
11	2	0.447833	$1.2 \cdot 10^{-14}$	$1.1 \cdot 10^{-7}$
12	2	0.447833	$3.3 \cdot 10^{-15}$	$4.4 \cdot 10^{-8}$
13	2	0.447833	$4.2 \cdot 10^{-15}$	$1.1 \cdot 10^{-8}$
14	2	0.447833	$3.4 \cdot 10^{-15}$	$1.4 \cdot 10^{-9}$
15	2	0.447833	$5.2 \cdot 10^{-15}$	$8.2 \cdot 10^{-10}$

Eigenvalues of Z_j at iteration $j = 3$				
-0.252217	0.251277	-0.0851209	-0.00119439	-0.000393022
0.000340506	-0.000137845	$4.20454\,10^{-5}$	$2.16208\,10^{-5}$	$-6.69661\,10^{-6}$
$1.08423\,10^{-6}$	$-3.45564\,10^{-7}$	$3.14863\,10^{-7}$	$-2.37621\,10^{-7}$	$-3.67591\,10^{-8}$
$3.37953\,10^{-8}$	$1.73052\,10^{-8}$	$-1.24566\,10^{-8}$	$-5.25292\,10^{-9}$	$2.33332\,10^{-9}$
$1.08314\,10^{-9}$	$-1.05302\,10^{-9}$	$-5.54074\,10^{-10}$	$-5.37935\,10^{-11}$	$4.62913\,10^{-11}$
$1.54928\,10^{-11}$	$-2.0519\,10^{-12}$	$4.46693\,10^{-13}$	$5.98481\,10^{-14}$	$-4.35833\,10^{-14}$
$3.68477\,10^{-14}$	$2.28224\,10^{-15}$	$-2.03297\,10^{-15}$	$1.69454\,10^{-16}$	$-8.16802\,10^{-17}$
$1.24491\,10^{-17}$				
Eigenvalues of Z_j at iteration $j = 15$				
-2.23276	-0.0309657	$6.01592\,10^{-16}$	$-4.9945\,10^{-16}$	$-4.39358\,10^{-16}$
$-4.26782\,10^{-16}$	$4.19653\,10^{-16}$	$4.04914\,10^{-16}$	$-3.96199\,10^{-16}$	$3.37393\,10^{-16}$
$-3.28595\,10^{-16}$	$-2.86148\,10^{-16}$	$-2.80031\,10^{-16}$	$-2.77081\,10^{-16}$	$-2.61391\,10^{-16}$
$2.60575\,10^{-16}$	$2.57387\,10^{-16}$	$-2.40079\,10^{-16}$	$2.27563\,10^{-16}$	$2.16993\,10^{-16}$
$2.06197\,10^{-16}$	$-1.8499\,10^{-16}$	$-1.70789\,10^{-16}$	$1.61236\,10^{-16}$	$-1.6067\,10^{-16}$
$1.31704\,10^{-16}$	$-1.16259\,10^{-16}$	$9.44279\,10^{-17}$	$-8.17497\,10^{-17}$	$7.49816\,10^{-17}$
$5.37598\,10^{-17}$	$3.81614\,10^{-17}$	$2.22188\,10^{-17}$	$3.13134\,10^{-18}$	$5.74329\,10^{-20}$
$9.15542\,10^{-23}$	$-9.15143\,10^{-23}$	$5.5889\,10^{-26}$	$-3.92555\,10^{-26}$	$-6.94225\,10^{-31}$

TABLE 5.5
Eigenvalues of Z_j before truncation to order p for iterations 3 and 15

In this example, α and β are the concentrations of a continuous input of both reactants where the unknowns X and Y are the concentrations of the other components in the chemical reaction. Under certain conditions of the parameters, a steady state solution is reached for $X \equiv \alpha$ and $Y \equiv \beta/\alpha$. The stability of the steady state can be analysed by first perturbing the solution into $X = \alpha + x$ and $Y = \beta/\alpha + y$ and then analysing the stability of the linear equations

$$\frac{\partial x}{\partial t} = (\beta - 1)x + \alpha^2 y + D_x \nabla^2 x$$

TABLE 5.6
Result for the 2D Brusselator equation

Iteration	rank(V_j)	λ_j	$\|G_j\|_F$	$\|(\Delta_1 + \lambda\Delta_0)\text{vec}(V_j D_j V_j^*)\|_2$
1	6	-1.9543	$5.8 \cdot 10^{-8}$	$1.6 \cdot 10^{3}$
2	2	1	$2.1 \cdot 10^{-11}$	$1.3 \cdot 10^{-3}$
3	2	1	$2.3 \cdot 10^{-11}$	$1.1 \cdot 10^{-6}$
4	2	1	$9.5 \cdot 10^{-16}$	$3.6 \cdot 10^{-9}$
5	2	1	$1 \cdot 10^{-15}$	$6.1 \cdot 10^{-9}$

$$\frac{\partial y}{\partial t} = -\beta x - \alpha^2 y + D_y \nabla^2 y$$

where

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial s^2} \ ,$$

and r and s are the spatial coordinates. The boundary conditions are

$$\frac{\partial x(0,s,t)}{\partial r} = \frac{\partial x(r,0,t)}{\partial s} = \frac{\partial x(L,s,t)}{\partial r} = \frac{\partial x(r,L,t)}{\partial s} = 0$$
$$\frac{\partial y(0,s,t)}{\partial r} = \frac{\partial y(r,0,t)}{\partial s} = \frac{\partial y(L,s,t)}{\partial r} = \frac{\partial y(r,L,t)}{\partial s} = 0$$

The parameters are chosen as $D_x = 1.6 \cdot 10^{-3}$, $D_y = 8.0 \cdot 10^{-3}$, $\alpha = 2.0$ and $\beta = 4.6$. Discretization by finite differences with gridsize $h = L/(N+1)$ leads to the linear system of ODE's $\dot{u} = (A + \lambda B)u$ of dimension $n = 2N^2$. Note that for this problem B is again a singular matrix: it has eigenvalues 0 and 1. We have chosen $L = 0.0798443$. In our analysis, we chose $\lambda = \beta$ as parameter.

We used $k = 40$ Arnoldi vectors, tolerance $\tau = 1.\,10^{-4}$ and $n = 3,200$. Table 5.6 shows the results per iteration. The value of $\lambda = 1$ corresponds to $\beta = 5$ and $\mu = \pm 2i$.

6. Conclusions. We have described a method for the computation of good starting values for use in algorithms to compute Hopf bifurcations in large scale dynamical systems. This work was originally motivated by the use of the bialternate product in [16] and [13], which has the significant advantage that a good starting guess for an unknown complex eigenvalue is not required. The disadvantage of the bialternate product is that it is an n^2 dimensional matrix, though the restriction to the anti-symmetric subspace reduces this to $n(n-1)/2$. We overcome this by a reformulation as an inexact inverse iteration algorithm that requires a sequence of $n-$dimensional Lyapunov-type equations, with the key feature that the right hand sides are of low rank. This results in an efficient procedure, provided the Lyaponuv equations are solved accurately. Numerical results illustrate the power of the method.

The results of this paper could also be used for computing eigenvalues of $Ax = \theta Mx$ nearest the imaginary axis. Shift-and-invert based methods are not always reliable when a rough idea of $\text{Im}(\theta)$ is unknown. Our method could be applied for finding the smallest λ for which $\theta = \lambda + i\mu$, $\lambda, \mu \in \mathbf{R}$, is an eigenvalue without computing μ.

Acknowledgement. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

The authors also thank Bart Vandereycken for discussions on the solution of Lyapunov equations.

REFERENCES

[1] A. Antoulas, D. Sorensen, and Y. Zhou, *On the decay rate of hankel singular values and related issues*, Systems & Control Letters, 46 (2002), pp. 323–342.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, PA, USA, 2000.

[3] R. Bartels and G. Stewart, *Algorithm 432: Solution of the matrix equation $ax + xb = c$*, Comm. ACM, 15 (1972), pp. 820–826.

[4] K. Cliffe, T. Garratt, and A. Spence, *Calculation of eigenvalues of the discretised Navier-Stokes and related equations*, in The mathematics of finite elements and applications VII MAFELAP, J. Whiteman, ed., Academic Press, 1990, pp. 479–486.

[5] ———, *Eigenvalues of block matrices arising from problems in fluid mechanics*, SIAM Journal on Matrix Analysis and Applications, 15 (1994), pp. 1310–1318.

[6] B. De Dier, *An investigation of a reaction-diffusion model exhibiting dissipative structures*, PhD thesis, K.U.Leuven, Leuven (Heverlee), Belgium, 1998.

[7] D. Fokkema, G. Sleijpen, and H. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM Journal on Scientific Computing, 20 (1999), pp. pp.94–125.

[8] M. A. Freitag and A. Spence, *Convergence rates for inexact inverse iteration with application to preconditioned iterative solves*, BIT, 47 (2007), pp. 27–44.

[9] ———, *Convergence theory for inexact inverse iteration applied to the generalised nonsymmetric eigenproblem*, ETNA, 28 (2007), pp. 40–64.

[10] T. Garratt, G. Moore, and A. Spence, *Two methods for the numerical detection of Hopf bifurcations*, in Bifurcation and Chaos : Analysis, Algorithms, Applications, R. Seydel, F. Schneider, and H. Troger, eds., Birkhäuser, 1991, pp. 119–123.

[11] G. Golub and C. Van Loan, *Matrix computations*, The Johns Hopkins University Press, 3rd ed., 1996.

[12] G. H. Golub and Q. Ye, *Inexact inverse iteration for generalized eigenvalue problems*, BIT, 40 (2000), pp. 671a–684.

[13] W. Govaerts, *Numerical methods for bifurcations of dynamical equilibria*, SIAM, Philadelphia, PA, USA, 2000.

[14] W. Govaerts, J. Guckenheimer, and A. Khibnik, *Defining functions for multiple Hopf bifurcations*, SIAM J Numer. Anal., 34 (1997), pp. 1269–1288.

[15] M. Grinfeld, *Dynamics of a model equation in viscoelasticity*, in Proceedings Equadiff 91, Singapore, 1993, World Scientific.

[16] J. Guckenheimer, S. Gueron, and R. Harris-Warrick, *Mapping the dynamics of a bursting neuron*, Philos. Trans. Roy. Soc. London Ser B, 341 (1993), pp. 345–359.

[17] J. Guckenheimer, M. Myers, and B. Sturmfels, *Computing Hopf bifurcations I*, SIAM J. Numer Anal., 34 (1997), pp. 1–21.

[18] S. Gugercin and J.-R. Li, *Smith-type Methods for Balanced Truncation of Large Sparse Systems*, Springer-Verlag, Berlin, Heidelberg, 2005.

[19] M. E. Hochstenbach, T. Kosir, and B. Plestenjak, *A Jacobi–Davidson type method for the two-parameter eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 477–497.

[20] M. E. Hochstenbach and B. Plestenjak, *A Jacobi–Davidson type method for a right definite two-parameter eigenvalue problem*, SIAM Journal on Matrix Analysis and Applications, 24 (2002), pp. 392–410.

[21] Y. Kuznetsov, *Elements of Applied Bifurcation Theory. 2nd ed.*, vol. 112 of Applied Math. Sci., Springer-Verlag, New York, 1998.

[22] R. Lehoucq and K. Meerbergen, *Using generalized Cayley transformations within an inexact rational Krylov sequence method*, SIAM Journal on Matrix Analysis and Applications, 20 (1998), pp. 131–148.

[23] K. Meerbergen and R. Morgan, *§11.2. Inexact methods*, in Templates for the solution of

algebraic eigenvalue problems: a practical guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., Philadelphia, PA, USA, 2000, SIAM.

[24] K. Meerbergen and D. Roose, *Finding eigenvalues to the right of a given line*, in Proceedings of the Second IMACS Symposium on Iterative Methods in Linear Algebra, P. Vassilevski and S. Margenov, eds., vol. 3 of IMACS Series in Computational and Applied Mathematics, Dept. of Computer Science, Rutgers University, Piscataway, NJ 08855, 1996, IMACS, pp. 402–412.

[25] ———, *Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems*, IMA Journal on Numerical Analysis, 16 (1996), pp. 297–346.

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

[27] W. Olmstead, W. Davis, S. Rosenblat, and W. Kath, *Bifurcation with memory*, SIAM J. Appl. Math., 46 (1986), pp. 171–188.

[28] B. Parlett and Y. Saad, *Complex shift and invert strategies for real matrices*, Linear Alg. Appl., 88/89 (1987), pp. 575–595.

[29] T. Penzl, *A cyclic low-rank smith method for large sparse lyapunov equations*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1401–1418.

[30] ———, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.

[31] M. Robbé and M. Sadkane, *Use of near-breakdowns in the block arnoldi method for solving large sylvester equations*, Applied Numerical Mathematics, 58 (2008), pp. 486–498.

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

[33] R. Seydel, *From Equilibrium to Chaos. Practical Bifurcation and Stability Analysis, 2nd ed.*, Springer-Verlag, Berlin, 1994.

[34] V. Simoncini, *A new iterative method for solving large-scale lyapunov matrix equations*, BIT, 36 (1996), pp. 814–830.

[35] V. Simoncini and V. Druskin, *Convergence analysis of projection methods for the numerical solution of large Lyapunov equations*, ??, ?? (2007), p. 18. Revised version, April 2008.