SIAM J. MATRIX ANAL. APPL.
Vol. 31, No. 1, pp. 18–39

AN EFFICIENT METHOD FOR ESTIMATING THE OPTIMAL DAMPERS' VISCOSITY FOR LINEAR VIBRATING SYSTEMS USING LYAPUNOV EQUATION*

NINOSLAV TRUHAR† AND KREŠIMIR VESELIĆ‡

Abstract. This paper deals with an efficient algorithm for dampers' viscosity optimization in mechanical systems. Our algorithm optimizes the trace of the solution of the corresponding Lyapunov equation using an iterative method which calculates a low rank Cholesky factor for the solution of the corresponding Lyapunov equation. We have shown that the new algorithm calculates the trace in $\mathcal{O}(m)$ flops per iteration, where m is a dimension of matrices in the Lyapunov equation (our coefficient matrices are treated as dense).

Key words. damped vibration, Lyapunov equation, optimization of viscosities of dampers

AMS subject classifications. 70J25, 70J50, 11D04, 15A06, 15A24, 90C31

DOI. 10.1137/070683052

1. Introduction.

This paper can be considered as a certain continuation of the paper [16]. In [16] we derived some new estimates for the eigenvalue decay rate of the Lyapunov equation $AX + XA^T = B$ with a low rank right-hand side B; we also proposed a new choice of the ADI parameters for calculating X. All this was based on newly established bounds on the trace of a solution to the Lyapunov equation with a general stable coefficient matrix. The trace itself was calculated from the solution of the Lyapunov equation which has been obtained using low rank Cholesky factor ADI (LRCF-ADI) proposed in [12], [8].

In this paper we use the results from [16] to develop an efficient algorithm for dampers' viscosity optimization in mechanical systems. Our penalty function is the trace of the Lyapunov solution X (advantages of this choice were discussed in [4], [17], [18]). Our main issue here is to calculate *only the trace* and not the whole solution of the Lyapunov equation with obvious computational advantages.

We consider a damped linear vibrational system described by the differential equation

$$M\ddot{x} + D\dot{x} + Kx = 0, \tag{1.1}$$

$$x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0, \tag{1.2}$$

where M, D, K (called mass, damping, stiffness matrices, respectively) are real, symmetric matrices of order n with M, K positive definite and $D = C_u + C$, where C_u is positive definite and represents the internal damping, which is usually taken to be a small multiple of the critical damping; that is,

$$C_u = \alpha C_{crit},$$

*Received by the editors February 20, 2007; accepted for publication (in revised form) by F. Dopico October 22, 2007; published electronically February 27, 2009.
http://www.siam.org/journals/simax/31-1/68305.html

†Department of Mathematics, J.J. Strossmayer University of Osijek, Trg Ljudevita Gaja 6, 31000 Osijek, Croatia (ntruhar@mathos.hr). Part of this work was done while this author was visiting the Department of Mathematics, University of Kentucky, Lexington, Kentucky, under the support of a postdoctoral research award from the National Foundation for Science, Higher Education and Technological Development of the Republic of Croatia.

‡Lehrgebiet Mathematische Physik, Fernuniversität, 58084 Hagen, Germany (kresimir.veselic@fernuni-hagen.de).

$$C_{crit} = 2M^{1/2}\sqrt{M^{-1/2}KM^{-1/2}}M^{1/2}, \quad \alpha = 2\text{–}10\%$$

(see [11, pp. 26, 260]), and C is positive semidefinite.

A very important question arises in considerations of such systems: *for given mass and stiffness determine the available dampers' viscosities so as to ensure an optimal evanescence.*

This optimization problem has been recently considered in [17], [14], [4], [15].

For such optimization one can use different optimization criteria (see [15]). One of the frequently used criteria is the so-called spectral abscissa criterion, which requires that a maximal real part of the eigenvalues λ_k be minimal, symbolically

$$sp := \max_k \mathrm{Re}\lambda_k \to \min, \tag{1.3}$$

where λ_k are the complex eigenvalues of the system

$$\left(\lambda^2 M + \lambda D + K\right) x = 0, \tag{1.4}$$

obtained from (1.1), simply using the substitution $x(t) = e^{\lambda t}x$.

For example, this criterion was used in [5] and [7]. In [5] a nice result on optimal damping was presented, but as the authors pointed out in section 4, "the only situation for which it is feasible to compute explicitly all possible solutions of the optimization problem (2.3) by hand is when n equals 2," which means that they present an exact form of the optimal damping matrix for systems that are 2×2. In [7] the problem of optimal damping (optimal dampers' positions, or, more precisely, optimal regions) has been solved for a string vibration.

Another criterion, used in [19], [17], [15], [4], is given by requirement of the minimization of the total energy of the system, that is,

$$\int_0^\infty E(t)\,dt \to \min. \tag{1.5}$$

The advantages of this criterion are (i) its obvious closeness to the total energy of the vibration and (ii) its smoothness as the function of the damping parameters, which allows standard methods of minimization via gradient or Hessian. Note that the latter property is not shared by the spectral penalty function (1.3). On the other hand, Veselić in [20], [21], [22] has shown that the solution of the Lyapunov equation provides rigorous bounds to the energy decay of a vibrating system.

Since the criterion (1.5) depends on the initial condition, the simplest way to correct this is to take the average of (1.5) over all initial states of the unit total energy and a given frequency range. It can be shown that this average is the trace of the solution of the corresponding Lyapunov equation.

A general algorithm for the optimization of damping does not exist. Available algorithms optimize only viscosities of dampers, not their positions. Two types of algorithms are currently in use. The first are the Newton-type algorithms for higher-dimensional (constrained or unconstrained) problems which use some Lyapunov solvers, and the second are the algorithms which explicitly calculate the trace of the solution of the corresponding Lyapunov equation.

An algorithm of the second type was presented in [19] for the case when $C_u = 0$ and the rank of the matrix C is one. Moreover, in [19] Veselić has given an efficient

algorithm which calculates an optimal v, where $C = vcc^*$, and the optimal viscosity is given by a closed formula.

On the other hand, in [15] a Newton-type algorithm which calculates optimal viscosity v has been proposed. This algorithm covers the case with internal damping ($C_u \neq 0$) with $C = vC_0C_0^*$, where $r \equiv rank(C_0) > 1$; it calculates the trace of the solution of the corresponding Lyapunov equation as a function of viscosity v of dampers in $\mathcal{O}(r^3m^3)$ flops, where $m = 2n$ (dimension of the phase space). This means that if the number of degrees of freedom of dampers r is much less than n ($r = 2, 3, 4$), this algorithm can be more efficient than the standard methods which use Lyapunov solvers such as, e.g., Bartels–Stewart, which cost $\mathcal{O}(m^3)$ operations per iteration.

Unfortunately, all existing algorithms calculate the solution of the Lyapunov equation and do not take advantage of the fact that we need only the trace of the solution.

Thus, we propose a different approach for optimization of the trace of the solution of the corresponding Lyapunov equation. Our algorithm calculates only the trace of the solution of the Lyapunov equation using an iterative method for an LRCF of the solution of the corresponding Lyapunov equation. This fact allows a more efficient memory usage. Further, in the case when only a small part of undamped spectra (say, the first s smallest undamped eigenvalues) is dominant, our algorithm needs $\mathcal{O}(r^3)+\mathcal{O}(r\,m)+\mathcal{O}(s^3)$ flops per iteration. Since standard optimization processes, such as the golden section search (which has been implemented in the MATLAB function `fminbnd`), need 20–30 iterations, if $r \ll n$ and $s \ll n$, our algorithm minimizes the trace of the Lyapunov equation in $\mathcal{O}(r\,m)$ operations.

We also present a new error bound for the trace approximation, which shows that sometimes the structure of the right-hand side of the Lyapunov equation can greatly influence the accuracy of the solution.

This paper is organized as follows. Section 2 describes a mathematical model we will use and three different algorithms for optimization of the trace of the solution of the corresponding Lyapunov equation. Then, section 3 contains the algorithm which calculates the trace of the solution using LRCF-ADI proposed in [8] (we use the algorithm described in [12]). Since the proper choice of ADI parameters is crucial for efficiency of the LRCF-ADI method, we describe two different algorithms for selection of a suboptimal set of ADI parameters. One was proposed by Penzl in [12], and the other was proposed in [16] and is based on the result that the optimal set of ADI parameters in the case of "modal damping" is given by the set of $2s$ eigenvalues of the matrix $\mathbf{A}$ which correspond to $2s$ undamped eigenvalues (for more details, see [16]). In section 4 we present a new error bound for the trace obtained by the new algorithm.

Finally, in the last section we present two examples. The first example illustrates the efficiency and accuracy of the new algorithm with respect to the column rank of the right-hand side of the Lyapunov equation. The second example compares our new algorithm (applied by using two different suboptimal sets of ADI parameters) with algorithms from [18], [2], and [15].

We will use the following notation: matrices written in simple mathematical italic fonts (M, D, or K), for example, will have $\mathcal{O}(n^2)$ nonzero entries. Matrices written in mathematical bold fonts ($\mathbf{A}$, $\mathbf{B}$) will have $\mathcal{O}(m^2)$ nonzero entries, where $m = 2n$. The symbol $\|\cdot\|$ stands for the standard 2-norm, while $\|\cdot\|_F$ denotes a Frobenius norm. $\mathcal{R}(A)$ denotes a column space spanned by the columns of the matrix A.

2. Setting the scene. As described in [15], [14], [17], [4], minimization of the total energy (1.5) is equivalent to minimization of the trace of the solution of the

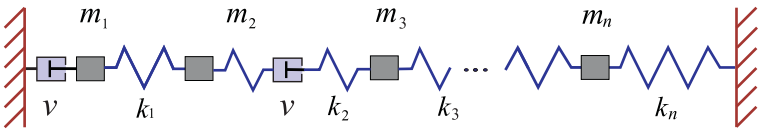

FIG. 2.1. *The n-mass oscillator with two dampers.*

Lyapunov equation. For the sake of completeness, we will shortly describe the basics of this approach.

We consider a damped linear vibrational system described by the differential equation

$$M\ddot{x} + D\dot{x} + Kx = 0, \tag{2.1}$$

where M, C, K (called mass, damping, stiffness matrix, respectively) are real, symmetric matrices of order n with M, K positive definite and $D = C_u + C$ positive semidefinite, where C_u describes internal damping. Often the matrix C has a small rank. An example is the so-called *n-mass oscillator* or *oscillator ladder* (Figure 2.1), where

$$M = \operatorname{diag}(m_1, m_2, \ldots, m_n),$$

$$K = \begin{bmatrix} k_0 + k_1 & -k_1 & & & & \\ -k_1 & k_1 + k_2 & -k_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -k_{n-2} & k_{n-2} + k_{n-1} & -k_{n-1} & \\ & & & -k_{n-1} & k_{n-1} + k_n \end{bmatrix},$$

$$D \equiv C_u + C = C_u + ve_1e_1^T + v(e_3 - e_2)(e_3 - e_2)^T.$$

Here $m_i > 0$ are the masses, $k_i > 0$ are the spring constants or stiffnesses, e_i is the ith canonical basis vector, and v is the viscosity of the damper applied on the ith mass (in Figure 2.1, $k_0 = 0$). Note that all dampers have the same viscosity and that rank of the matrix C is two. In this paper we study the system with r equal dampers where we assume that $r \ll n$ (usually $r = 2, 3, 4$), which will allow us to use a one-dimensional optimization process (MATLAB function `fminbnd`).

To (2.1) there corresponds the eigenvalue problem

$$(\lambda^2 M + \lambda D + K)x = 0\,. \tag{2.2}$$

Obviously all eigenvalues of (2.2) lie in the left complex plane.

Using the eigenvalue decomposition

$$\Phi^T K \Phi = \Omega^2\,, \quad \Phi^T M \Phi = I, \tag{2.3}$$

where $\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_n)$, $\omega_1 < \cdots < \omega_n$, and setting

$$y_1 = \Omega\,\Phi^T x, \quad y_2 = \Phi^T \dot{x}, \tag{2.4}$$

(2.1) can be written as

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}, \tag{2.5}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\Phi^T D \Phi \end{bmatrix} \tag{2.6}$$

(we are now in a $2n$-dimensional phase space), with the solution

$$\mathbf{y} = e^{\mathbf{A}t}\,\mathbf{y}_0\,, \quad \text{where } \mathbf{y}_0 \text{ is the initial data.} \tag{2.7}$$

Note that the numbers

$$\omega_1, \omega_2, \ldots, \omega_n \tag{2.8}$$

are the eigenvalues of the corresponding undamped system

$$(\lambda^2 M + K)x = 0,$$

and we call them (undamped) eigenfrequencies of the system.

The eigenvalue problem $\mathbf{A}\mathbf{y} = \lambda\,\mathbf{y}$ is equivalent to (2.2). The energy of the system is given by

$$E(t) = \frac{1}{2}\,\dot{x}(t)^T M \dot{x}(t) + \frac{1}{2}\,x(t)^T K x(t) = \frac{1}{2}\,y^T y.$$

Now (1.5) can be written as

$$\mathbf{y}_0^T \mathbf{X} \mathbf{y}_0 \to \min, \tag{2.9}$$

where

$$\mathbf{X} = \int_0^\infty e^{\mathbf{A}^T t}\, e^{\mathbf{A}t} dt \tag{2.10}$$

is the solution of the Lyapunov equation

$$\mathbf{A}^T\mathbf{X} + \mathbf{X}\mathbf{A} = -\mathbf{I}\,. \tag{2.11}$$

An inconvenience of the criterion (2.9) is its dependence on the initial data $\mathbf{y}_0$. Thus, similarly as in [17], instead of the quantity $\mathbf{y}_0^T\mathbf{X}\mathbf{y}_0$ we are going to take its mean value over all initial data $\mathbf{y}$ with the unit energy $\|\mathbf{y}\|^2$. Therefore, instead of (2.9) we require

$$\int_{\|\mathbf{y}_0\|=1} \mathbf{y}_0^T \mathbf{X} \mathbf{y}_0 \; d\sigma \to \min, \tag{2.12}$$

where $d\sigma$ is a chosen probability measure on the unit sphere $S^{2n} = \{\mathbf{y}_0 \in \mathbb{R}^{2n}; \|\mathbf{y}_0\| = 1\}$.

In [17], [10], and [15] it has been shown that (2.12) is equivalent to

$$Tr(\mathbf{Z}\mathbf{X}) \to \min, \tag{2.13}$$

where $\mathbf{Z}$ is a symmetric positive semidefinite matrix which may be normalized to have a unit trace. If we take for the measure σ the measure generated by the Lebesgue measure on $\mathbb{R}^{2n}$, we obtain $Z = \frac{1}{2n} I$.

Further, it is easy to show that

$$Tr(\mathbf{ZX}) = Tr(\mathbf{Y}),$$

where $\mathbf{Y}$ is a solution of the so-called dual Lyapunov equation

$$\mathbf{AY} + \mathbf{YA}^T = -\mathbf{Z}.$$

The structure of the matrix $\mathbf{Z}$ has been studied in detail in [10], and some of these results are presented in [15].

Throughout this paper we will assume that the matrix $\mathbf{Z}$ has the form

$$\mathbf{Z} = \begin{bmatrix} 0_{t_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & I_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0_{t_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0_{t_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0_{t_2} \end{bmatrix}, \tag{2.14}$$

where I_s is the s-dimensional identity matrix and 0_{t_i} is the t_i-dimensional ($i = 1, 2$) zero matrix, where t_1 and s are defined such that eigenfrequencies from (2.8) smaller than ω_{t_1} and greater than ω_{t_1+s} are not dangerous (observe that $t_2 = n - t_1 - s$).

Now we will briefly describe the existing algorithms for optimization (2.13).

In [19] a solution of problem (2.13) has been given in the case when $C_u = 0$ and $\operatorname{rank}(C) = 1$. In particular,

$$Tr(\mathbf{ZX}(v)) = const + \frac{a}{v} + bv, \tag{2.15}$$

where $a, b > 0$ are constants which can be easily calculated (in $O(n)$ flops), which makes it possible to find the minimum explicitly by a simple formula. The case $\operatorname{rank}(C) > 1$ seems to be essentially more difficult to handle.

In [15], problem (2.13) with $C_u \neq 0$ and $\operatorname{rank}(C) > 1$ has been considered. In particular,

$$Tr(\mathbf{ZX}(v)) = -x_0 - v\, b_L^T (\mathbf{I} - v\, \mathbf{H}_s)^{-1} b_R, \tag{2.16}$$

where $\mathbf{H}_s$ denotes the upper Hessenberg matrix for whose construction one needs $\frac{112}{3}\, r^3 m^3 + \mathcal{O}(r^2 m^2)$ operations. Since from (2.16) one can find the first and the second derivative of the function $v \to Tr(\mathbf{ZX}(v))$ almost for free, the whole optimization process costs $\frac{112}{3}\, r^3 m^3 + \mathcal{O}(r^2 m^2)$.

On the other hand, a more general case with the damping matrix

$$D \equiv C_u + C = C_u + C_0 \operatorname{diag}(v_1, \ldots, v_r) C_0^T$$

has been considered in [2]. There, a Newton-type algorithm has been proposed, which uses the Bartels–Stewart Lyapunov solver.

As we will see in the last section, each of these algorithms has some advantages in certain situations. But all of them calculate the whole solution at every stage of the iteration and then use only the trace.

As we have mentioned in the introduction, our approach here consists of constructing an efficient algorithm which will derive the trace $Tr(\mathbf{ZX}(v))$ using the LRCF-ADI method, and then find the minimum of the function $v \rightarrowtail Tr(\mathbf{ZX}(v))$ using some standard minimization process such as the golden section search which has been implemented in the MATLAB function `fminbnd`. Since we calculate only the trace and not the whole solution, our algorithm is much faster than existing ones which calculate the whole solution first and then the trace. The next section contains a description of our new algorithm.

3. The main algorithm. As described in the previous section, our aim is to minimize the trace of the Lyapunov equation

$$\mathbf{AX} + \mathbf{XA}^T = -\mathbf{GG}^T, \tag{3.1}$$

where

$$\mathbf{A} \equiv \mathbf{A}_0 - v\,\mathbf{D} = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\,\Omega^k \end{bmatrix} - v \begin{bmatrix} 0 & 0 \\ 0 & C_0 C_0^T \end{bmatrix}, \tag{3.2}$$

where $\operatorname{rank}(\mathbf{G}) = 2s$, $s \ll n$, and

$$\mathbf{D} = \mathbf{D}_0 \mathbf{D}_0^T, \quad \mathbf{D}_0 = \begin{bmatrix} 0 \\ C_0 \end{bmatrix}, \quad \text{and} \quad C_0 = \Phi^T \left[e_{i_1}, \dots e_{i_r} \right]. \tag{3.3}$$

The vector e_{i_j} is the i_jth canonical basis vector and r is the number of dampers. We assume that $\Omega = \operatorname{diag}(\omega_1, \dots, \omega_n)$, where $\omega_1 < \cdots < \omega_n$.

Note that for $\mathbf{Z}$ defined as in (2.14), we have $\mathbf{Z} = \mathbf{GG}^T$, where

$$\mathbf{G} = \begin{bmatrix} 0 & I_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_s & 0 \end{bmatrix}^T, \tag{3.4}$$

$\mathbf{G} \in \mathbb{R}^{m \times 2s}$, $s \ll n$. This assumption and the fact that the solution of (3.1) is positive definite allow us to use the LRCF-ADI method proposed in [8] (see also [9]) and implemented in [12]. As we will see throughout this paper, the choice of this algorithm for computing the trace of the Lyapunov equation has many advantages. The most important fact is that by using this algorithm one can find the trace of the solution without calculating the whole solution, which can substantially speed up the calculation.

As we have mentioned above, since $s \ll n$, we are going to use the LRCF-ADI algorithm for solving the Lyapunov equation

$$\mathbf{AX} + \mathbf{XA}^T = -\mathbf{GG}^T.$$

The basic code taken from [12] reads as follows.

ALGORITHM 1 (LRCF-ADI).

INPUT: $\mathbf{A}$, $\mathbf{G}$, $\{p_1, p_2, \dots, p_{i_{max}}\}$

OUTPUT: $\mathbf{V} = \mathbf{V}_{i_{max}} \in \mathbb{C}^{m \times 2\,s\,i_{max}}$, *such that* $\mathbf{VV}^* \approx \mathbf{X}$.

1. $\mathbf{W}_1 = \sqrt{-2\mathrm{Re}p_1}\,(\mathbf{A} + p_1\mathbf{I})^{-1}\mathbf{G}$
2. $\mathbf{V} = \mathbf{W}_1$

FOR: $i = 2, 3, \dots, i_{max}$

3. $\mathbf{W}_i = \sqrt{\mathrm{Re}p_i / \mathrm{Re}p_{i-1}}\,\left(\mathbf{W}_{i-1} - (p_i + \overline{p}_{i-1})(\mathbf{A} + p_i\mathbf{I})^{-1}\mathbf{W}_{i-1}\right)$
4. $\mathbf{V}_i = [\mathbf{V}_{i-1}\,\mathbf{W}_i]$

END

Here $\{p_1, p_2, \ldots, p_{i_{max}}\}$ denotes a set of ADI parameters. As pointed out in [12], the proper choice of ADI parameters is crucial for efficiency of the LRCF-ADI method. There exist several routines for selection of ADI parameters. We will describe two of them.

The first has been presented in [12] and is based on the following two ideas. First, we generate a discrete set, which "approximates" the spectrum, which is done by a pair of Arnoldi processes (we calculate the set of Ritz values). Then we choose a set of shift parameters which is a subset of the set of Ritz values by a heuristic that delivers a suboptimal set of ADI shifts. As we will see in the last section, sometimes this choice can yield a poor approximation of the trace, especially in cases when the viscosity is of small magnitude or in the case when s is not small enough.

The second routine has been proposed in [16] and contains the following four steps:

1. Find the indices of 1's on the right-hand side (i.e., find positions of 1's in the matrix GG^T).
2. Find the corresponding submatrix of A using these indices (i.e., form the submatrix A_s).
3. Take a "little bit bigger block" A_{block} (which depends on a particular problem) which includes the submatrix A_s.
4. Eigenvalues of the chosen matrix A_{block} are ADI parameters ($p_1, \ldots, p_l \in \sigma(A_{block})$).

Figure 3.1 shows how we form the matrix $\mathbf{A}_{block}$.

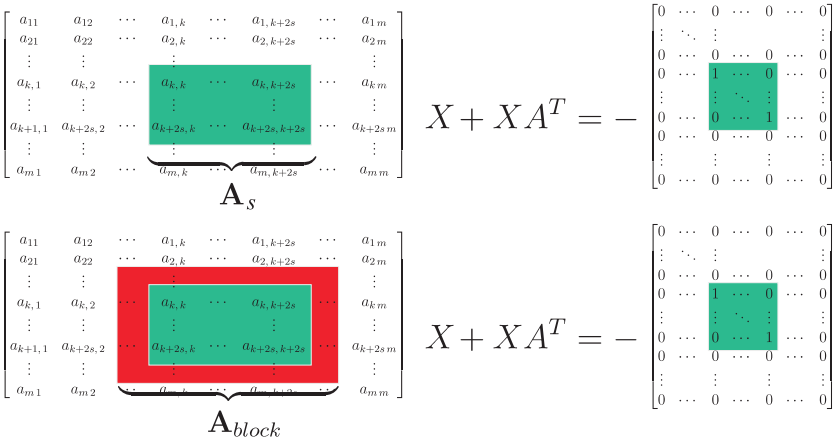

FIG. 3.1. *Choosing of* A_{block}.

Once we find a proper set of ADI parameters we can proceed with the implementation of Algorithm 1.

Before giving our algorithm for the trace of the solution of the Lyapunov equation (3.1), we will point out some facts and introduce some notation which will be used later.

First, in Algorithm 1 one has to compute the inverse of $(\mathbf{A} + p_i\mathbf{I})$. In our case (3.1)–(3.3)

$$\mathbf{A} \equiv \mathbf{A}_0 - v\,\mathbf{D}_0\,\mathbf{D}_0^T,$$

where

$$\mathbf{A}_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\,\Omega^k \end{bmatrix} \quad \text{and} \quad \mathbf{D}_0 = \begin{bmatrix} 0 \\ C_0 \end{bmatrix}.$$

Since we consider the problems with $\mathrm{rank}(\mathbf{D}_0) = r \ll n$, one can use the *Sherman–Morrison–Woodbury formula* for calculation of the inverse $(\mathbf{A}+p_i\mathbf{I})^{-1}$ [6, eq. (2.1.4), p. 51]. For this purpose we will need the notation

$$\mathbf{A}_0(p_i) = \mathbf{A}_0 + p_i\mathbf{I}\,. \tag{3.5}$$

Now, we can write

$$\begin{aligned} \mathbf{A}^{-1} &\equiv (\mathbf{A}_0(p_i) - v\mathbf{D}_0\,\mathbf{D}_0^T)^{-1} \\ &= \mathbf{A}_0(p_i)^{-1} + v\,\mathbf{A}_0(p_i)^{-1}\mathbf{D}_0\left(I_r - v\mathbf{D}_0^T\mathbf{A}_0(p_i)^{-1}\mathbf{D}_0\right)^{-1}\mathbf{D}_0^T\mathbf{A}_0(p_i)^{-1}\,. \end{aligned} \tag{3.6}$$

Note that the inverse $\mathbf{A}_0(p_i)^{-1}$ can be derived directly; that is,

(3.7)

$$\begin{aligned} \mathbf{A}_0(p_i)^{-1} &= \begin{bmatrix} p_iI & \Omega \\ -\Omega & p_iI - \alpha\,\Omega^k \end{bmatrix}^{-1} \\ &= \begin{bmatrix} (\Omega^2 + p_i^2I - p_i\alpha\,\Omega^k)^{-1}(p_iI - \alpha\,\Omega^k) & -(\Omega^2 + p_i^2I - p_i\alpha\,\Omega^k)^{-1}\,\Omega \\ (\Omega^2 + p_i^2I - p_i\alpha\,\Omega^k)^{-1}\,\Omega & p_i\,(\Omega^2 + p_i^2I - p_i\alpha\,\Omega^k)^{-1} \end{bmatrix}. \end{aligned}$$

This means that all matrices in (3.6) can be computed directly, except

$$Inv(v,p_i) \equiv \left(I_r - v\mathbf{D}_0^T\mathbf{A}_0(p_i)^{-1}\mathbf{D}_0\right)^{-1}\,. \tag{3.8}$$

The matrix $I_r - v\mathbf{D}_0^T\mathbf{A}_0(p_i)^{-1}\mathbf{D}_0$ is of order r.

Using the above considerations, we can adapt Algorithm 1 for calculating the trace of the solution of the Lyapunov equation (3.1) in the following way.

ALGORITHM 2 (calculating the trace using LRCF-ADI).

INPUT: Ω, C_0, v, $\mathbf{G}$, $\{p_1, p_2, \ldots, p_l\}$

OUTPUT: Tr, *Tr stands for trace(X)*

0. $Tr = 0$
1. $\mathbf{W}_1 = \sqrt{-2\mathrm{Re}p_1}\,\left(\mathbf{A}_0(p_1)^{-1}\mathbf{G} + v\,\mathbf{A}_0(p_1)^{-1}\mathbf{D}_0\,Inv(v,p_1)\,\mathbf{D}_0^T\mathbf{A}_0(p_1)^{-1}\mathbf{G}\right)$
2. $tr(1) = \sum_i^{2\,s} \|\mathbf{W}_1(:,i)\|^2$

FOR $j = 2, 3, \ldots, l$

3. $\mathbf{W}_j = \sqrt{\mathrm{Re}p_j/\mathrm{Re}p_{j-1}}$
$\cdot\left(\mathbf{W}_{j-1} - (p_j + \overline{p}_{j-1})\left(\mathbf{A}_0(p_j)^{-1}\right.\right.$
$\left.\left. + v\,\mathbf{A}_0(p_j)^{-1}\mathbf{D}_0\,Inv(v,p_j)\,\mathbf{D}_0^T\mathbf{A}_0(p_j)^{-1}\right)\mathbf{W}_{j-1}\right)$
4. $tr(j) = \sum_i^{2\,s} \|\mathbf{W}_j(:,i)\|^2$

END

5. $Tr = \sum_i^{l} tr(i)$

Assuming that we have a proper set of ADI parameters, we can calculate the costs for Algorithm 2. Note that every step in Algorithm 2 contains $\mathbf{A}_0(p_j)^{-1}\mathbf{G}$. It is easy to see that this multiplication costs $2s \cdot \mathcal{O}(m)$ flops. Further, the inner loop

contains a matrix $\mathbf{A}_0(p_j)^{-1}\mathbf{D}_0\, Inv(v,p_j)\,\mathbf{D}_0^T\mathbf{A}_0(p_j)^{-1}\mathbf{G}$, which can be calculated in $2\,s\,r\cdot(\,\mathcal{O}(m)+\mathcal{O}(r)\,)+2s(\,m\mathcal{O}(r)+2\mathcal{O}(m)\,)+\mathcal{O}(r^3)$ operations. Altogether this yields that Algorithm 2 calculates the trace of the solution of the Lyapunov equation in

$$l\cdot\big(\,s\,r\cdot(\,\mathcal{O}(m)+\mathcal{O}(r)\,)+s(\mathcal{O}(m\,r)+\mathcal{O}(m)\,)+\mathcal{O}(r^3)\big) \tag{3.9}$$

operations. In our applications we usually have $r\le 12$. Now from (3.9) it follows that Algorithm 2 with such r needs less than $\mathcal{O}(s\,m)$ flops.

At this point it is important to emphasize that the fact that Algorithm 2 needs less operations than existing algorithms which calculate the whole solution of the Lyapunov equation ($\mathcal{O}(s\,m)$ contrary to $\mathcal{O}(m^3)$) is not its only advantage. The fact that Algorithm 2 calculates only the trace of the solution of the Lyapunov equation implies much more efficient memory usage. Indeed, in each iteration step of Algorithm 2 we have to save only one $m\times 2s$ matrix (in each step we overwrite the old one) instead of standard LRCF-ADI where we have to form the factor which is a matrix of dimension $m\times 2s\cdot l$, where l is the number of iteration steps.

Since the efficiency and accuracy of Algorithm 2 depend on a proper choice of ADI parameters, in the next section we will analyze accuracy of the solution obtained by Algorithm 2 using a new suboptimal set of ADI parameters.

4. Quality of the new choice of ADI parameters. In this section we present an error bound for the approximation of the trace of the solution of the Lyapunov equation obtained by Algorithm 1 (Algorithm 2) generated by ADI parameters $\{p_1,\dots,p_l\}$ obtained by a new suboptimal choice proposed in the last section.

The error bound contains two parts: the first belongs to the approximation of the solution $\mathbf{X}$ of Lyapunov equation (3.1) with its lth approximation $\mathbf{X}_l$ obtained by Algorithm 1 (Algorithm 2) with the set of ADI parameters which corresponds to a certain subset of the spectrum of the matrix $\mathbf{A}$. This bound was presented in [16, Theorem 2.1].

The second part of the bound belongs to the approximation of a suboptimal set of ADI shifts ("exact eigenvalues" of the matrix $\mathbf{A}$) with some approximative values. This approximation has to be done since the location of eigenvalues which represent a suboptimal set of ADI shifts is still an open problem.

Thus, let the matrix $\widetilde{\mathbf{X}}_l$ be the approximation of the solution $\mathbf{X}_l$ by Algorithm 1 (Algorithm 2) with the set of ADI parameters $\{p_1,\dots,p_l\}$ obtained by our new suboptimal choice.

Thus, we can write

$$|Tr(\mathbf{X})-Tr(\widetilde{\mathbf{X}}_l)|\le|Tr(\mathbf{X})-Tr(\mathbf{X}_l)|+|Tr(\mathbf{X}_l)-Tr(\widetilde{\mathbf{X}}_l)|. \tag{4.1}$$

As pointed out above, the bound for $|Tr(\mathbf{X})-Tr(\mathbf{X}_l)|$ will be taken from [16, Theorem 2.1], assuming that $\mathbf{A}$ is diagonalizable with eigendecomposition:

$$\mathbf{A}=\mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}.$$

Let $\mathbf{X}_l$ be the lth approximation obtained by Algorithm 1 (Algorithm 2) with the set of ADI parameters which correspond to any subset of exact eigenvalues of the matrix $\mathbf{A}$ (i.e., $\lambda_{k_i}\in\sigma(\mathbf{A})$ for $i=1,\dots,l$). Then the following bound holds [16, Theorem 2.1]:

$$|Tr(\mathbf{X})-Tr(\mathbf{X}_l)|\le\|\mathbf{S}\|^2\sum_{j=l+1}^{m}(-2\mathrm{Re}(\lambda_{k_j}))\sum_{k=1}^{m}|\sigma(j,k)|^2\cdot\|\widehat{g}_k\|^2, \tag{4.2}$$

where

$$\sigma(1,k)=\frac{1}{\lambda_k+\overline{\lambda}_{k_1}} \quad \text{and} \quad \sigma(j,k)=\frac{1}{\lambda_k+\overline{\lambda}_{k_1}}\prod_{t=2}^{j}\frac{\lambda_k-\lambda_{k_{t-1}}}{\lambda_k+\overline{\lambda}_{k_t}} \quad \text{for} \quad j>1, \tag{4.3}$$

and

$$\widehat{\mathbf{G}}=\mathbf{S}^{-1}\mathbf{G}=\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1s}\\ g_{21} & g_{22} & \cdots & g_{2s}\\ \vdots & \vdots & \vdots & \vdots\\ g_{m1} & g_{m2} & \cdots & g_{ms}\end{bmatrix}=\begin{bmatrix}\widehat{g}_1\\ \widehat{g}_2\\ \vdots\\ \widehat{g}_m\end{bmatrix}; \tag{4.4}$$

that is, $\widehat{g}_i$ denotes the ith row of the matrix $\widehat{\mathbf{G}}$.

As shown in [16], the right-hand side of (4.2) strongly depends on the magnitude of $\|\widehat{g}_k\|_F^2$, $k=1,\ldots,k_0$ (the structure of the matrix $\widehat{G}$ is important). For example, if

$$\|\widehat{g}_1\|\geq\cdots\geq\|\widehat{g}_l\|\gg\|\widehat{g}_{l+1}\|_F\approx\cdots\approx\|\widehat{g}_m\|_F\approx\sqrt{\varepsilon}, \tag{4.5}$$

then we can choose $\lambda_{k_1},\ldots,\lambda_{k_l}$ such that $\sigma(j,1)=\cdots=\sigma(j,l)=0$ for $j\geq 2$. This is fulfilled for $k_i=i$. If $\|S\|$, $\mathrm{Re}(\lambda_j)$, and the rest of $\sigma(j,k)$'s have modest magnitudes, then from (4.2) we have

$$|Tr(\mathbf{X})-Tr(\mathbf{X}_l)|\leq\mathcal{O}(\varepsilon)\,.$$

With this assumption we will continue with bounding the second part of the right-hand side of (4.1). Without loss off generality, we will assume that the matrix $\mathbf{G}$ from (3.1) has the form $\mathbf{G}=[I_s,0]^T$, where I_s is an identity matrix of dimension s, that is,

$$\mathbf{G}\mathbf{G}^T=\begin{bmatrix} I_s & 0\\ 0 & 0\end{bmatrix}. \tag{4.6}$$

It is important to note that in the case when $\mathbf{G}$ has the form defined as in (4.6), our choice of ADI parameters is given as the set of eigenvalues of the matrix $\mathbf{A}_{block}$, where $\mathbf{A}_{block}=(\mathbf{A}_0)_{11}-v\,\mathbf{d}_1\,\mathbf{d}_1^T$ and where, after perfect shuffle permutation, $\mathbf{A}$ has the following form:

$$\mathbf{A}\equiv\begin{bmatrix}(\mathbf{A}_0)_{11}-v\,\mathbf{d}_1\,\mathbf{d}_1^T & -v\,\mathbf{d}_1\,\mathbf{d}_2^T\\ -v\,\mathbf{d}_2\,\mathbf{d}_1^T & (\mathbf{A}_0)_{22}-v\,\mathbf{d}_2\,\mathbf{d}_2^T\end{bmatrix},\quad \text{where}\quad \mathbf{D}_0=\begin{bmatrix}\mathbf{d}_1\\ \mathbf{d}_2\end{bmatrix}.$$

Usually, the dimension l of the matrix $\mathbf{A}_{block}$ is taken as $3s\leq l\leq 5s$.

Note that we can write

$$\widetilde{\mathbf{A}}=\mathbf{A}-\Delta\mathbf{A}\equiv\begin{bmatrix}(\mathbf{A}_0)_{11}-v\,\mathbf{d}_1\,\mathbf{d}_1^T & 0\\ 0 & (\mathbf{A}_0)_{22}-v\,\mathbf{d}_2\,\mathbf{d}_2^T\end{bmatrix},$$

where

$$\Delta\mathbf{A}=v\cdot\begin{bmatrix}0 & \mathbf{d}_1\,\mathbf{d}_2^T\\ \mathbf{d}_2\,\mathbf{d}_1^T & 0\end{bmatrix},$$

which means that our ADI shifts are exact eigenvalues of the matrix $\widetilde{\mathbf{A}}$.

Recall that $\mathbf{X}_l$ is the lth approximation of the solution of the Lyapunov equation (3.1) obtained by Algorithm 1 (Algorithm 2) generated by the set $\{\lambda_1, \lambda_2, \ldots, \lambda_l\}$, where $\lambda_i \in \sigma(\mathbf{A})$ for $i = 1, \ldots, l$, while $\widetilde{\mathbf{X}}_l$ is the lth approximation of the solution of the Lyapunov equation (3.1) obtained by Algorithm 1 (Algorithm 2) generated by the set of ADI parameters obtained by our new suboptimal choice of ADI parameters, that is, with $p_i \in \sigma(\mathbf{A}_{block}) \subset \sigma(\widetilde{\mathbf{A}})$.

Our choice of ADI parameters can be written as

$$p_i \equiv \overline{\widetilde{\lambda}}_i = \overline{\lambda}_i \pm \overline{\delta\lambda}_i\,, \qquad i = 1, \ldots, l. \tag{4.7}$$

Further, $\mathbf{X}_l$ and $\widetilde{\mathbf{X}}_l$ can be written as

$$\mathbf{X}_l = \sum_{j=1}^{l} \|\mathbf{W}_j\|_F^2\,, \qquad\qquad \widetilde{\mathbf{X}}_l = \sum_{j=1}^{l} \|\widetilde{\mathbf{W}}_j\|_F^2,$$

where $\mathbf{W}_j$ and $\widetilde{\mathbf{W}}_j$ are matrices obtained by Algorithm 1 (Algorithm 2).

Then, if we write

$$\widetilde{\mathbf{W}}_j = \mathbf{W}_j + \delta\mathbf{W}_j,$$

it is easy to show that the following first order bound holds:

$$Tr(\mathbf{X}_l) - Tr(\widetilde{\mathbf{X}}_l) \le 2\sum_{j=1}^{l} \|\mathbf{W}_j\|_F \, \|\delta\mathbf{W}_j\|_F + \mathcal{O}(\|\delta\mathbf{W}_j\|_F{}^2)\,. \tag{4.8}$$

We will continue with bounding $\|\delta\mathbf{W}_j\|_F$.

Let $\mathbf{W}_j$ be the jth matrix obtained by Algorithm 1 (Algorithm 2) with ADI parameters $\{\lambda_1, \ldots, \lambda_l\}$, with input matrices $\mathbf{A}$ and $\mathbf{G}$, where $\mathbf{G}$ is defined as in (4.6). In [16] it has been shown that $\mathbf{W}_j$ can be written as

$$\mathbf{W}_j = \sqrt{-2\,\mathrm{Re}(\lambda_j)}\,\mathbf{S}\cdot \mathrm{diag}\left(\sigma(j,1), \sigma(j,2), \ldots\,, \sigma(j,m)\right)\,\mathbf{S}^{-1}\mathbf{G}, \tag{4.9}$$

where $\sigma(j,k)$ are given by

$$\sigma(1,k) = \frac{1}{\lambda_k + \overline{\lambda}_1} \quad \text{and} \quad \sigma(j,k) = \frac{1}{\lambda_k + \overline{\lambda}_j}\prod_{t=1}^{j-1}\frac{\lambda_k - \lambda_t}{\lambda_k + \overline{\lambda}_{t+1}} \quad \text{for} \quad j > 1.$$

Indeed, from Algorithm 1 (Algorithm 2) (for more details see the proof of Theorem 2.1 in [16]), it follows that

$$\begin{aligned}\mathbf{W}_j = \sqrt{-2\,\mathrm{Re}(\lambda_j)}\,\mathbf{S}\cdot &\left(\mathbf{I} - (\overline{\lambda}_j + \lambda_{j-1})(\mathbf{\Lambda} + \overline{\lambda}_j\mathbf{I})^{-1}\right)\\ &\cdot \left(\mathbf{I} - (\overline{\lambda}_{j-1} + \lambda_{j-2})(\mathbf{\Lambda} + \overline{\lambda}_{j-1}\mathbf{I})^{-1}\right)\cdots\left(\mathbf{I} - (\overline{\lambda}_2 + \lambda_1)(\mathbf{\Lambda} + \overline{\lambda}_2\mathbf{I})^{-1}\right)\\ &\cdot (\mathbf{\Lambda} + \overline{\lambda}_1\mathbf{I})^{-1}\mathbf{S}^{-1}\mathbf{G}\,,\end{aligned}$$

which together with the fact that in the above equality we have a $j-1$ diagonal matrix of the form

$$\left(\mathbf{I} - (\overline{\lambda}_k + \lambda_{k-1})(\mathbf{\Lambda} + \overline{\lambda}_k\mathbf{I})^{-1}\right) = \mathrm{diag}\left(\frac{\lambda_i - \lambda_{k-1}}{\lambda_i + \overline{\lambda}_k}\right)_i,$$

$i = 1, \ldots, m$, $k = 2, \ldots, j$, gives (4.9).

Here it is important to note that all eigenvalues of the matrix $\mathbf{A}$ from (3.2) are given in complex conjugate pairs. Thus, if we choose ADI parameters as the first l exact eigenvalues of $\mathbf{A}$, then the structure of $\sigma(j,k)$ implies

$$\sigma(j,k) = 0 \quad \text{for} \quad j = 1,\dots,l, \quad j > k.$$

Similarly, let $\widetilde{\mathbf{W}}_j$ be the jth matrix obtained by Algorithm 1 (Algorithm 2) with ADI parameters $\{p_1,\dots,p_l\}$, with the same input matrices $\mathbf{A}$ and $\mathbf{G}$, where p_i is defined by (4.7):

$$\widetilde{\mathbf{W}}_j = \sqrt{-2\,\mathrm{Re}(\lambda_j \pm \delta\lambda_j)}\,\mathbf{S}\cdot \mathrm{diag}\left(\widetilde{\sigma}(j,1),\widetilde{\sigma}(j,2),\dots,\widetilde{\sigma}(j,m)\right)\,\mathbf{S}^{-1}\mathbf{G}, \tag{4.10}$$

where $\widetilde{\sigma}(j,k)$ are given by

$$\widetilde{\sigma}(1,k) = \frac{1}{\lambda_k + \overline{\lambda}_1 \pm \overline{\delta\lambda}_1} \quad \text{and}$$
$$\widetilde{\sigma}(j,k) = \frac{1}{\lambda_k + \overline{\lambda}_j \pm \overline{\delta\lambda}_j} \prod_{t=1}^{j-1} \frac{\lambda_k - \lambda_t \mp \delta\lambda_t}{\lambda_k + \overline{\lambda}_{t+1} \pm \overline{\delta\lambda}_{t+1}} \quad \text{for} \quad j > 1\,.$$

Now it is easy to see that from (4.9) and (4.10) it follows that

$$\delta\mathbf{W}_j = \mathbf{S}\cdot \mathrm{diag}\left(\delta\zeta(j,1)\dots,\delta\zeta(j,k),\dots,\delta\zeta(j,m)\right)\,\mathbf{S}^{-1}\mathbf{G},$$

where

$$\delta\zeta(j,k) = \sqrt{-2\,\mathrm{Re}(\lambda_j \pm \delta\lambda_j)}\cdot\widetilde{\sigma}(j,k) - \sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot\sigma(j,k). \tag{4.11}$$

Now $\delta\mathbf{W}_j$ can be written as

$$\delta\mathbf{W}_j = \mathbf{S}\begin{bmatrix} \delta\zeta(j,1)\,\widehat{g}_1 \\ \delta\zeta(j,2)\,\widehat{g}_2 \\ \vdots \\ \delta\zeta(j,m)\,\widehat{g}_m \end{bmatrix}. \tag{4.12}$$

Recall that we have assumed that $\widehat{\mathbf{G}}$ satisfy (4.5) and that all $\mathrm{Re}(\lambda_j)$, $\sigma(j,k)$'s for $k \geq l$ have modest magnitudes such that

$$|\sigma(j,k)|\cdot\|\widehat{g}_k\| = \mathcal{O}(\sqrt{\varepsilon}), \qquad k \geq l, \tag{4.13}$$

and $\sqrt{-2\,\mathrm{Re}(\lambda_j)}\,\mathcal{O}(\varepsilon) = \mathcal{O}(\varepsilon)$. Then, as we have already pointed out, there holds $|Tr(\mathbf{X}) - Tr(\mathbf{X}_l)| \leq \mathcal{O}(\varepsilon)$.

Thus, if

$$|\eta_j| \equiv \left|\frac{\delta\lambda_j}{\lambda_j}\right| < 1, \tag{4.14}$$

then we have the first order approximation

$$\sqrt{-2\,\mathrm{Re}(\lambda_j \pm \delta\lambda_j)} = \sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot\left(1 \pm \frac{|\eta_j|}{2}\right) + \mathcal{O}(|\eta_j|^2),$$

which implies

$$\delta\zeta(j,k) \approx \sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot(\widetilde{\sigma}(j,k)-\sigma(j,k)) \mp \frac{\eta_j}{2}\cdot\sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot\widetilde{\sigma}(j,k)\,.$$

Thus, the above consideration implies that we have to bound more carefully the first l components of the matrix $\delta\mathbf{W}_j$ from (4.4) than the rest of the components. That is, we are going to bound $|\delta\zeta(j,i)|$ for $j=1,\ldots,l$.

For $1<j$ and $k\le l$ we have

$$\widetilde{\sigma}(j,k) = \frac{1}{\lambda_k+\overline{\lambda}_1\pm\overline{\delta\lambda}_1}\prod_{\substack{t=1\\ t\neq k}}^{j-1}\frac{\lambda_k-\lambda_t\mp\delta\lambda_t}{\lambda_k+\overline{\lambda}_{t+1}\pm\overline{\delta\lambda}_{t+1}}\cdot\frac{\lambda_k\ \ \eta_k}{\lambda_k+\overline{\lambda}_{k+1}\pm\overline{\delta\lambda}_{k+1}},$$

which can be written as

$$\widetilde{\sigma}(j,k) = \delta\sigma(j,k)\cdot\eta_k, \qquad \text{where}$$
$$\delta\sigma(j,k) = \frac{1}{\lambda_k+\overline{\lambda}_1\pm\overline{\delta\lambda}_1}\prod_{\substack{t=1\\ t\neq k}}^{j-1}\frac{\lambda_k-\lambda_t\mp\delta\lambda_t}{\lambda_k+\overline{\lambda}_{t+1}\pm\overline{\delta\lambda}_{t+1}}\cdot\frac{\lambda_k}{\lambda_k+\overline{\lambda}_{k+1}\pm\overline{\delta\lambda}_{k+1}},$$

Further, let $\delta\mathbf{W}_j(k,:)$ be the kth row of the matrix $\delta\mathbf{W}_j$ defined by (4.12). Assumption (4.5) implies that the entries in $\delta\mathbf{W}_j(k,:)$ will be small in magnitude for $k>l$. Thus for $k>l$ the following simple bound is quite satisfactory to us:

$$|\delta\zeta(j,k)| \lesssim \sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot|\sigma(j,k)+\widetilde{\sigma}(j,k)| + \frac{|\eta_j|}{2}\cdot\sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot|\widetilde{\sigma}(j,k)|\,.$$

Finally, we can bound the right-hand side in (4.8). Indeed,

$$|Tr(\mathbf{X}_l)-Tr(\widetilde{\mathbf{X}}_l)| \lesssim 2\sum_{j=1}^{l}\|\mathbf{W}_j\|_F\,\|\delta\mathbf{W}_j\|_F, \tag{4.15}$$

where

$$\|\delta\mathbf{W}_j\|_F \le \|\mathbf{S}\|\sum_{k=1}^{m}|\delta\zeta(j,k)|\cdot\|\widehat{g}_k\|,$$

with

$$|\delta\zeta(j,k)| \lesssim \sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot\left(|\delta\sigma(j,k)|\cdot|\eta_k| + \frac{|\eta_j|}{2}\cdot|\widetilde{\sigma}(j,k)|\right), \quad k\le l, \tag{4.16}$$

and

$$|\delta\zeta(j,k)| \lesssim \sqrt{-2\,\mathrm{Re}(\lambda_j)}\cdot\left(|\sigma(j,k)|+|\widetilde{\sigma}(j,k)| + \frac{|\eta_j|}{2}\cdot|\widetilde{\sigma}(j,k)|\right), \quad k>l. \tag{4.17}$$

Now, from (4.2) and (4.15) it follows that

$$\begin{aligned}|Tr(\mathbf{X})-Tr(\widetilde{\mathbf{X}}_l)| \le{}& \|\mathbf{S}\|^2\sum_{j=l+1}^{m}(-2\mathrm{Re}(\lambda_{k_j}))\sum_{k=1}^{m}|\sigma(j,k)|^2\cdot\|\widehat{g}_k\|^2\\ &+2\,\|\mathbf{S}\|\sum_{j=1}^{l}\|\mathbf{W}_j\|_F\,\left(\sum_{k=1}^{m}|\delta\zeta(j,k)|\cdot\|\widehat{g}_k\|\right),\end{aligned} \tag{4.18}$$

where all quantities used in the above bound are defined in the above consideration.

From (4.16) and (4.17) it follows that an important part in our bound is played by the perturbation of eigenvalues. Since we have assumed that all eigenvalues of the matrix $\mathbf{A}$ are simple, the following (see [3] or [13]) holds:

$$|\delta\lambda_k| \leq \frac{t_k^* \Delta A s_k}{t_k^* s_k} = \varepsilon_k, \tag{4.19}$$

where s_k and t_k are right and left eigenvectors belonging to λ_k normalized so that $\|s_k\| = \|t_k\| = 1$ and $|t_k^* s_k| = t_k^* s_k$. Now from (4.19) and (4.14) it follows that

$$|\eta_k| \leq \frac{t_k^* \Delta A s_k}{|\lambda_k|\ t_k^* s_k}\,.$$

As the last issue in this section, we are going to discuss how realistic is our assumption (4.5). It is obvious that assumption (4.5) will be fulfilled if $\mathcal{R}(\mathbf{G})$ is close to column space $\mathcal{R}(\mathbf{S})$.

We are going to derive a bound for viscosity v, from which will be possible to conclude when our assumption,

$$\|\widehat{g}_j\| \approx \sqrt{\varepsilon} \qquad \text{for} \quad j = l+1, \ldots, m,$$

will be feasible.

Recall that we have denoted

$$\widehat{\mathbf{G}} = \mathbf{S}^{-1}\mathbf{G} = \left[\widehat{g}_1, \ldots, \widehat{g}_l, \widehat{g}_{l+1}, \ldots \widehat{g}_m\right]^T.$$

Note that for (4.6) $\widehat{\mathbf{G}}$ contains only the first s columns of the matrix $\mathbf{T}^* = \mathbf{S}^{-1}$. Further, let $\widetilde{\mathbf{A}} = \widetilde{\mathbf{S}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{S}}^{-1}$ be the eigenvalue decomposition of the matrix $\widetilde{\mathbf{A}}$. Note that since $\widetilde{\mathbf{A}}$ is block diagonal, $\widetilde{\mathbf{S}}$ will be block diagonal, too. Thus for v of modest magnitude one can expect that $\mathbf{T}^*$ will have an almost block diagonal structure.

If we write

$$\mathbf{E}_S = \mathbf{S}^{-1}\widetilde{\mathbf{S}},$$

using $\widehat{\mathbf{G}} = \mathbf{T}^*\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^{-1}\mathbf{G} \equiv \mathbf{E}_S\widetilde{\mathbf{T}}^*_{(1:s,:)}$ we can bound $\widehat{\mathbf{G}}_2 \equiv \left[\widehat{g}_{l+1}, \ldots, \widehat{g}_m\right]^T$ in the following way:

$$\|\widehat{\mathbf{G}}_2\| \leq \|(\mathbf{E}_S)_{21}\|\,\|\widetilde{\mathbf{T}}^*_{(1:s,:)}\|,$$

where $(\mathbf{E}_S)_{21}$ denotes an off-diagonal block which contains rows from $l+1$ up to m and the first s columns of $\mathbf{E}_S$.

Now using the simple equality

$$\mathbf{\Lambda}\mathbf{E}_S - \mathbf{E}_S\widetilde{\mathbf{\Lambda}} = -\mathbf{S}^{-1}\Delta\mathbf{A}\widetilde{\mathbf{S}},$$

one can easily see that

$$|(\mathbf{E}_S)_{21}|_{ij} = \frac{\mathbf{T}^*_{(:,i)}\Delta\mathbf{A}\widetilde{\mathbf{S}}_{(:,j)}}{|\lambda_i - \widetilde{\lambda}_j|} = v \cdot \frac{\mathbf{T}^*_{(:,i)}\mathbf{d}_2\mathbf{d}_1^T\widetilde{\mathbf{S}}_{(:,j)}}{|\lambda_i - \widetilde{\lambda}_j|} \equiv v \cdot \frac{\Psi_{ij}}{|\lambda_i - \widetilde{\lambda}_j|}\,, \qquad \text{where}$$
$$\lambda_i \in \mathbf{\Lambda}(l+1:m, l+1:m)\,, \qquad \widetilde{\lambda}_j \in \widetilde{\mathbf{\Lambda}}(1:s, 1:s).$$

Altogether this implies

$$\|\widehat{\mathbf{G}}_2\|_F \leq v \cdot \frac{\|\Psi\|_F \, \|\widetilde{\mathbf{T}}^*_{(1:s,:)}\|}{\mathrm{gap}(\widetilde{\mathbf{\Lambda}}(1:s,1:s), \mathbf{\Lambda}(l+1:m, l+1:m))}, \tag{4.20}$$

$$\mathrm{gap}(\widetilde{\mathbf{\Lambda}}(1:s,1:s), \mathbf{\Lambda}(l+1:m,l+1:m)) = \min_{i,j} |\lambda_i - \widetilde{\lambda}_j|, \qquad \text{where}$$

$$\lambda_i \in \mathbf{\Lambda}(l+1:m, l+1:m)\,, \qquad \widetilde{\lambda}_j \in \widetilde{\mathbf{\Lambda}}(1:s,1:s)\,.$$

Note that since $l > s$, the gap function can be large in magnitude; thus if $\|\widetilde{\mathbf{T}}^*_{(1:s,:)}\|$ has a modest magnitude, then for $l > 4\,s$ from (4.20) it follows that if

$$v \lesssim \sqrt{\varepsilon} \cdot \frac{\mathrm{gap}(\widetilde{\mathbf{\Lambda}}(1:s,1:s), \mathbf{\Lambda}(l+1:m,l+1:m))}{\|\Psi\|_F \, \|\widetilde{\mathbf{T}}^*_{(1:s,:)}\|},$$

our assumption (4.5) will be fulfilled, which means that $\|\widehat{g}_k\| \approx \sqrt{\varepsilon}$ for $k = l+1, \ldots, m$.

5. Numerical illustration. The first example in this section illustrates the efficiency and accuracy of the new algorithm in respect to the column rank of the right-hand side of the Lyapunov equation (3.1).

Example 1. We consider the Lyapunov equation

$$\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T = -\mathbf{G}\mathbf{G}^T,$$

where

$$\mathbf{A} = \mathbf{A}_0 - v\,\mathbf{D} = \begin{bmatrix} 0 & \Omega \\ -\Omega & 0 \end{bmatrix} - v \begin{bmatrix} 0 & 0 \\ 0 & C_0 C_0^T \end{bmatrix},$$

where $C_0 = \mathrm{rand}(n, r)$ and $r = 4$. The matrix $\mathbf{A}$ is $m \times m$, with $m = 400$ (note that $n = 200$). The matrix $\mathbf{G}$ we construct as

$$\mathbf{G} = 0.001 \cdot \mathrm{rand}(m, 2\,s), \qquad \mathbf{G}(1:s, 1:s) = \mathrm{rand}(s),$$
$$\mathbf{G}(n+1:n+s, s+1:2\,s) = \mathrm{rand}(s).$$

where $\mathrm{rank}(\mathbf{G}) = 2s$.

The ADI shifts will be chosen as the eigenvalues of the matrix $\mathbf{A}_p(1:l_p, 1:l_p)$, where $\mathbf{A}_p$ is obtained from $\mathbf{A}$ using the perfect shuffle permutation.

We are going to compute a relative error `relerr` and the number of floating point operations `flops` obtained by the new algorithm, Algorithm 2, for the different s—the rank of the matrix $\mathbf{G}$.

The relative error is defined as

$$\mathrm{relerr} = \frac{|Tr_X - Trnew_X|}{Tr_X},$$

where Tr_X is the trace of the solution of the Lyapunov equation (3.1) obtained by MATLAB function `Lyap` (which is based on the Bartels–Stewart method), while $Trnew_X$ is the trace of the solution obtained by the new algorithm, Algorithm 2. Further, the number of floating point operations `flops` is defined by

$$\texttt{flops} = l_p \cdot \big(s\,r\,(\,\mathcal{O}(m) + \mathcal{O}(r)\,) + s(\,\mathcal{O}(m\,r) + \mathcal{O}(m)\,) + \mathcal{O}(r^3)\big) + \mathcal{O}(l_p^3),$$

which has been obtained using (3.9) with additional $\mathcal{O}(l_p^3)$ flops, which corresponds to the number of calculations needed for the l_p suboptimal ADI shifts.

s	10	10	10	10
l_p	20	40	60	200
relerr	0.0016	3.2430e-004	2.1210e-004	3.3638e-005
`flops`	$\mathcal{O}(10^5)$	$\mathcal{O}(10^6)$	$\mathcal{O}(10^6)$	$\mathcal{O}(10^7)$
s	30	30	30	30
l_p	60	80	100	200
relerr	0.0015	6.2926e-004	4.0017e-004	1.1102e-004
`flops`	$\mathcal{O}(10^6)$	$\mathcal{O}(10^6)$	$\mathcal{O}(10^7)$	$\mathcal{O}(10^7)$
s	50	50	50	50
l_p	50	100	140	200
relerr	0.5525	0.0016	5.2693e-004	2.3062e-004
`flops`	$\mathcal{O}(10^6)$	$\mathcal{O}(10^7)$	$\mathcal{O}(10^7)$	$\mathcal{O}(10^7)$

As one can see from the above table, the accuracy and efficiency of the new algorithm, Algorithm 2, strongly depends on s—the number of damped modes. If we are interested in a result of a certain accuracy, then as s grows, the required number of ADI parameters grows, too, which slows performance of the new algorithm.

Further, we will compare different algorithms for dampers' viscosity optimization considering a simple mechanical system consisting of three rows of n masses connected with $n+1$ springs on the left-hand side on the fixed base and on the right-hand side on the mass m_0 connected to the fixed base with the spring with stiffness k_0 (see Figure 5.1).

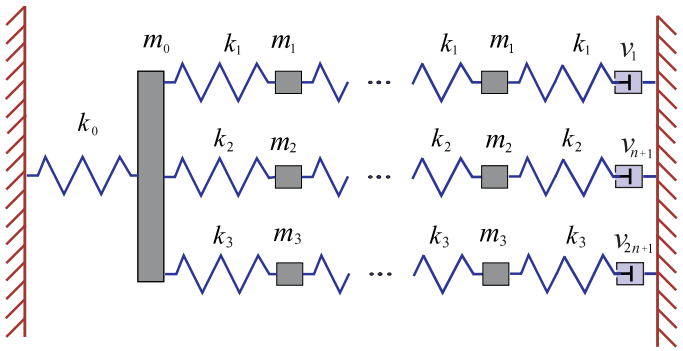

FIG. 5.1. *$(3n+1)$-mass oscillator with three dampers.*

Example 2. Consider a damped linear vibrational mechanical system consisting of three rows of n masses connected with $n+1$ springs on the left-hand side on the fixed base and on the right-hand side on the mass m_0 connected to the fixed base with the spring with stiffness k_0 (see Figure 5.1). Then one can write

$$M\ddot{x} + D\dot{x} + Kx = 0,$$

where M, D, and K are defined as

$$M = \operatorname{diag}(m_1, \ldots, m_1, m_2, \ldots, m_2, m_3, \ldots, m_3, m_0)\,, \tag{5.1}$$

(5.2)

$$K = \begin{bmatrix} K_{11} & & & -\kappa_1 \\ & K_{22} & & -\kappa_2 \\ & & K_{33} & -\kappa_3 \\ -\kappa_1^T & -\kappa_2^T & -\kappa_3^T & k_1{+}k_2{+}k_3{+}k_0 \end{bmatrix}, \quad K_{ii} = k_i \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix},$$

and $\kappa_i = \begin{bmatrix} 0 & \dots & 0 & k_i \end{bmatrix}^T$, $K_{ii} \in \mathbb{R}^{n\times n}$ and $\kappa_i \in \mathbb{R}^{n\times 1}$, for $i = 1, 2, 3$,

$$D \equiv C_u + C = C_u + ve_1e_1^T + ve_ne_n^T + ve_{2n+1}e_{2n+1}^T.$$

Note that M, D, and K are matrices of order $3n + 1 \times 3n + 1$. We will set $n = 40$.

Let $m_2 = k_2 = 2$ and $m_3 = k_3 = 4$ be fixed, and let m_0, m_1, k_0, k_1 be chosen such that

$$m_0, k_0 \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\} \quad \text{and} \quad m_1, k_1 \in \{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}.$$

This means that we have 1296 different configurations defined by different sets $\{m_0, m_1, m_2, m_3\}$ and $\{k_0, k_1, k_2, k_3\}$. For each of these configurations we have derived the optimal trace of the solution of the corresponding Lyapunov equation (3.1), where G is defined by (3.4) and $G(1 : s, 1 : s) = I_s$, $G(n + 1 : n + s, s + 1 : 2\,s) = I_s$ with $s = 20$ and the optimal viscosity. We have used the following four algorithms:

1. Minimization process based on Newton iteration process for higher-dimensional problems which use the Bartels–Stewart Lyapunov solver [1]. For the starting point we have used the one proposed in [2] (*New.-Bart.-Stew.*).
2. Minimization process based on Newton iteration process for one-dimensional problems which use a new Lyapunov solver proposed in [15]. For the starting point we have used $v_0 = 0.01$ (*New.-new*).
3. Minimization process with the standard MATLAB function `fminbnd` (using the Golden section search and parabolic interpolation) based on LRCF-ADI Lyapunov solver Algorithm 2 generated by the set of ADI parameters proposed by Penzl in [12] (*LRCF-ADI-Penzl*). Minimization has been performed on the interval $[0, 5000]$.
4. Minimization process with the standard MATLAB function `fminbnd` (using the Golden section search and parabolic interpolation) based on LRCF-ADI Lyapunov solver Algorithm 2 generated by the new set of ADI parameters proposed in [16] (*LRCF-ADI, new*). Minimization has been performed on the interval $[0, 500]$.

Before we continue with the analysis of all 1296 configurations, we will illustrate the quality of error bound (4.18) on one particular configuration. The chosen configuration has some interesting properties (later the same configuration will be considered as the case of a nonconvex energy curve), and it is far away from the best possible case for our error analysis.

Let

$$m_0 = 100, \quad m_1 = 0.01, \quad m_2 = 2, \quad m_4 = 4;$$
$$k_0 = 100, \quad k_1 = 0.01, \quad k_2 = 2, \quad k_4 = 4\,.$$

Let $v = 4$, $s = 20$ and let $l = 60$ be the number of ADI shifts generated by a new algorithm proposed in section 3. Simple calculation gives that the first part in the

bound (4.18) is bounded with 0.7876, while the second part is bounded with 100.7001. Altogether this gives

$$\frac{|Tr(\mathbf{X}) - Tr(\widetilde{\mathbf{X}}_l)|}{Tr(\mathbf{X})} \leq 0.0205\,.$$

At the same time real relative error for the lth approximation of the trace is

$$\frac{|Tr(\mathbf{X}) - Tr(\widetilde{\mathbf{X}}_l)|}{Tr(\mathbf{X})} \leq 2.55 \cdot 10^{-4}\,.$$

It has to be pointed out that for the considered configuration a relative error in eigenvalues satisfies

$$\max_{1\leq k\leq m} |\eta_k| \leq \max_{1\leq k\leq m} \frac{t_k^* \Delta A s_k}{|\lambda_k|\ t_k^* s_k} \leq 0.76,$$

while for the norms of the rows of the matrix $\widehat{\mathbf{G}} = \mathbf{S}^{-1}\mathbf{G}$ it holds that

$$10^{-7} \leq |\widehat{g}_k| \leq 0.77\,, \qquad l+1 \leq k \leq m.$$

The above example shows that, although we do not have a very accurate approximation for all eigenvalues and eigenvectors (which was expected), we still have 4 exact digits in our approximation of the trace, while our bound predicts 2 exact digits.

We continue with the analysis of all 1296 configurations. Table 5.1 contains the ratios between optimal traces obtained by algorithms *LRCF-ADI-Penzl* and *LRCF-ADI-new* (3 and 4) and algorithms *New.-new* and *New.-Bart.-Stew.* (2 and 1).

TABLE 5.1

	LRCF-ADI-Penzl/New.-new	*LRCF-ADI-new/New.-new*
> 1.02	15.7 %	5.2 %
< 0.98	2.7 %	2.7 %
	LRCF-ADI-Penzl/New.-Bart.-Stew.	*LRCF-ADI-new/New.-Bart.-Stew.*
> 1.05	42 %	37 %
< 1	0.75 %	6 %

As one can see from Table 5.1 in 5.2% of our experiments the optimal trace obtained by the algorithm *LRCF-ADI-new* is more than 2% larger than the optimal trace obtained by *New.-new*, while in 15.7% of our experiments the optimal trace obtained by *LRCF-ADI-Penzl* is more than 2% larger than the optimal trace obtained by *New.-new*. At the same time, in 2.7% of our experiments, both algorithms (*LRCF-ADI-new* and *LRCF-ADI-Penzl*) obtain at least 2% a larger optimal trace than the optimal trace obtained by *LRCF-ADI-new*.

Since the discrepancies in both cases were not expected, we will carefully consider the cases in which they appear. It turns out that by the algorithm *LRCF-ADI-new* in 5.2% of our experiments we have obtained a larger trace in comparison to the algorithm *New.-new*, whereas in 15.7% we have obtained a larger trace by the algorithm *LRCF-ADI-Penzl* in comparison to the algorithm *New.-new*. This has been caused by using the wrong intervals: $[0, 500]$ for *LRCF-ADI-new* and $[0, 5000]$ for *LRCF-ADI-Penzl*, respectively. For example, in the abovementioned situations, the algorithm *LRCF-ADI-new* has obtained the optimal trace for optimal viscosity $v = 500$, which is obviously wrong. On the other hand, in 2.7% of our experiments

the optimal trace obtained by the algorithms *LRCF-ADI-new* and *LRCF-ADI-Penzl* is smaller than the optimal trace obtained by algorithm *New.-new*. The reason for this is a wrong starting point $v_0 = 0.01$ for Newton iterations.

For illustration consider the case with

$$m_0 = 100, \quad m_1 = 0.01, \quad m_2 = 2, \quad m_4 = 4;$$
$$k_0 = 100, \quad k_1 = 0.01, \quad k_2 = 2, \quad k_4 = 4.$$

Figure 5.2 shows the trace as the function of viscosity v. It is obvious that starting point $v_0 = 0.01$ will lead to a wrong result. But if we take for the starting point any point $6 < v_0 < 10$, optimal viscosity is $v = 14.765$.

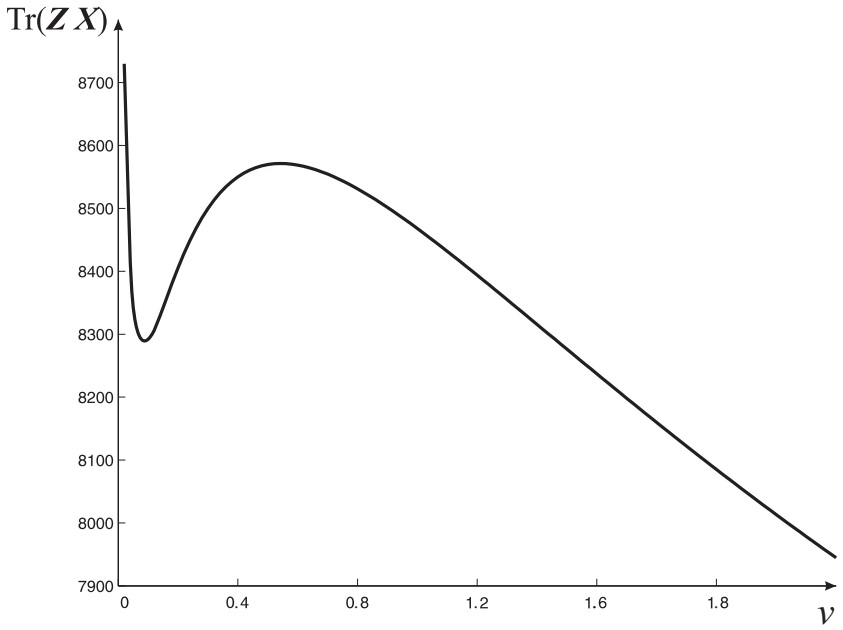

FIG. 5.2. *The graph of the trace function.*

Similar conclusions hold for the ratio between optimal traces obtained by algorithms *LRCF-ADI-new* and *LRCF-ADI-Penzl* and the optimal trace obtained by multidimensional optimization using *New.-Bart.-Stew.*

As expected (with a usage of correct intervals on which we perform minimization) in 30% of our experiments, the optimal trace obtained by *New.-Bart.-Stew.* is more than 5% smaller than the optimal trace obtained by algorithm *LRCF-ADI-new*, while in 35% of our experiments, the optimal trace obtained by *New.-Bart.-Stew.* is more than 5% smaller than optimal trace obtained by algorithm *LRCF-ADI-Penzl*. But in 6% of our experiments, the optimal trace obtained by algorithm *LRCF-ADI-new* is smaller than the optimal trace obtained by *New.-Bart.-Stew.*, which was definitively unexpected. The reason for this lies in the fact that in these particular situations the starting point for *New.-Bart.-Stew.*, obtained by the algorithm proposed in [2], is

wrong. For illustration we consider the case with

$$m_0 = 100, \quad m_1 = 0.01, \quad m_2 = 2, \quad m_4 = 4;$$
$$k_0 = 100, \quad k_1 = 10, \quad k_2 = 2, \quad k_4 = 4.$$

The starting point for optimization process *New.-Bart.-Stew.* obtained by routine `calcvisc` taken from [2] gives $\text{visc} = \big[0.0147, 2.7535, 5.5009\big]$, which corresponds to $Tr(\mathbf{ZX}_{opt}) = 4965.4$, at the same time the optimal trace for $v_{opt} = 16.41$ (obtained by algorithm *LRCF-ADI-new*) is $Tr(\mathbf{ZX}) = 4062.6$. But if we change a starting point to $\text{visc}_2 = \big[16.4, 16.4, 16.4\big]$, then the algorithm *New.-Bart.-Stew.* gives optimal trace $Tr(\mathbf{ZX}_{opt}) = 3997.1$ for viscosity $\text{visc}_{opt} = \big[20.6384, 11.5852, 23.183\big]$.

Considering the abovementioned, we can conclude that both algorithms based on the LRCF-ADI Lyapunov solver combined with some nonsmooth optimization give us very satisfactory results.

At the same time, the number of operations needed for one optimization with algorithm *LRCF-ADI-new* is much smaller than the number needed for optimization with algorithm *New.-new*. For illustration, to obtain optimal viscosity with algorithm *LRCF-ADI-new* one usually needs ~ 20 iterations, which together with (3.9) gives $20 \cdot (280\mathcal{O}(m) + 10/3(3s)^3)$ operations, where the second number in the bracket $10/3(3s)^3$ stands for the number of operations needed for calculating the ADI parameters (eigenvalues of a $3s \times 3s$ nonsymmetric matrix (see [6])). On the other hand, as shown in [15], the algorithm *New.-new* needs $14/3\,(2rm)^3 + \mathcal{O}(r^2m^2)$ operations, which is obviously much more.

Acknowledgment. We would like to thank the anonymous referees for their very careful reading of the manuscript and valuable comments.

REFERENCES

[1] R. H. Bartels and G. W. Stewart, *A solution of the matrix equation $AX + XB = C$*, Comm. ACM, 15 (1972), pp. 820–826.
[2] K. Brabender, *Optimale Dämpfung von linearen Schwingungssystemen*, Ph.D. thesis, Fernuniversität Hagen, Hagen, Germany, 1998.
[3] R. Byers and D. Kressner, *On the condition of a complex eigenvalue under real perturbations*, BIT, 44 (2004), pp. 209–214.
[4] S. J. Cox, I. Nakić, A. Rittmann, and K. Veselić, *Minimization of energy of a damped system*, Systems Control Lett., 53 (2004), pp. 187-194.
[5] P. Freitas and P. Lancaster, *On the optimal value of the spectral abscissa for a system of linear oscillators*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 195–208.
[6] G. H. Golub and Ch.F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1989.
[7] P. Hebrard and A. Henrot, *Optimal shape and position of the actuators for the stabilization of a string*, Systems Control Lett., 48 (2003), pp. 199–209.
[8] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.
[9] J. Li, F. Wang, and J. White, *An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect*, in Proceedings of the 36th Annual IEEE/ACM Design Automation Conference, New Orleans, LA, 1999.
[10] I. Nakić, *Optimal Damping of Vibrational Systems*, Ph.D. thesis, Fernuniversität Hagen, Hagen, Germany, 2002.
[11] M. Paz, *Structural Dynamics: Theory and Computation*, Van Nostrand Reinhold, New York, 1991.
[12] T. Penzl, *LYAPACK*, http://www.tu-chemnitz.de/sfb393/lyapack.
[13] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[14] N. Truhar and K. Veselić, *On some properties of the Lyapunov equation for damped systems*, Math. Commun., 9 (2004), pp. 189–197.

[15] N. Truhar, *An efficient algorithm for damper optimization for linear vibrating systems using Lyapunov equation*, J. Comput. Appl. Math., 172 (2004), pp. 169–182.
[16] N. Truhar and K. Veselić, *Bounds on the trace of solution to the Lyapunov equation with a general stable matrix*, Systems Control Lett., 56 (2007), pp. 493–503.
[17] K. Veselić, K. Brabender, and K. Delinić, *Passive control of linear systems*, in Applied Mathematics and Computation, M. Rogina et al., eds. Department of Mathematics, University of Zagreb, Zagreb, Croatia, 2001, pp. 39–68.
[18] K. Veselić, *On linear vibrational systems with one-dimensional damping*, Appl. Anal., 29 (1988), pp. 1–18.
[19] K. Veselić, *On linear vibrational systems with one-dimensional damping.* II, Integral Equations Operator Theory, 13 (1990), pp. 883–897.
[20] K. Veselić, *Exponential decay of semigroups in Hilbert space*, Semigroup Forum, 55 (1997), pp. 325–331.
[21] K. Veselić, *Estimating the operator exponential*, Linear Algebra Appl., 280 (1998), pp. 241–244.
[22] K. Veselić, *Bounds for exponentially stable semigroups*, Linear Algebra Appl., 358 (2003), pp. 309–333.