CHAPTER 8

NUMERICAL SOLUTIONS AND CONDITIONING OF LYAPUNOV AND SYLVESTER EQUATIONS

Topics covered

- Existence and Uniqueness Results for Solutions of Lyapunov and Sylvester Equations
- Perturbation Analyses and Condition Numbers
- The Schur and the Hessenberg–Schur Methods (Both Continuous and Discrete-Time Cases)
- Backward Error Analyses of the Schur and the Hessenberg–Schur Methods
- Direct Computations of Cholesky Factors of Symmetric Positive Definite Solutions of Lyapunov Equations

8.1 INTRODUCTION

In **Chapter 7**, we have seen that the Lyapunov equations arise in **stability** and **robust stability** analyses, in determining **controllability** and **observability Grammians**, and in **computing H_2-norm**. The solutions of Lyapunov equations are also needed for the implementation of some **iterative methods for solving algebraic Riccati equations** (**AREs**), such as Newton's methods (**Chapter 13**). The important role of Lyapunov equations in these practical applications warrants discussion of numerically viable techniques for their solutions.

The continuous-time **Lyapunov equation**:

$$XA + A^{\mathrm{T}}X = C \tag{8.1.1}$$

is a special case of another classical matrix equation, known as the **Sylvester equation**:

$$XA + BX = C. \tag{8.1.2}$$

Similarly, the discrete-time Lyapunov equation:

$$A^{\mathrm{T}}XA - X = C$$

is a special case of the discrete-time Sylvester equation:

$$BXA - X = C.$$

(Note *that the matrices A, B, C in the above equations are not necessarily the system matrices.*)

The Sylvester equations also arise in a wide variety of applications. For example, we will see in **Chapter 12** that a variation of the Sylvester equation, known as the **Sylvester-observer** equation, arises in the construction of **observers** and in solutions of the **eigenvalue assignment** (EVA) (or pole-placement) problems. The Sylvester equation also arises in other areas of applied mathematics. For example, the numerical solution of elliptic boundary value problems can be formulated in terms of the solution of the Sylvester equation (Starke and Niethammer 1991). The solution of the Sylvester equation is also needed in the block diagonalization of a matrix by a similarity transformation (see Datta 1995) and Golub and Van Loan (1996). Once a matrix is transformed to a block diagonal form using a similarity transformation, the block diagonal form can then be conveniently used to compute the matrix exponential e^{At}.

In this chapter, we will first develop the basic theories on the **existence** and **uniqueness** of solutions of the Sylvester and Lyapunov equations (**Section 8.2**), next discuss **perturbation theories (Section 8.3)**, and then finally describe **computational methods (Sections 8.5 and 8.6)**.

The continuous-time Lyapunov equation (8.1.1) and the continuous-time Sylvester equation (8.1.2) will be referred to as just the **Lyapunov** and **Sylvester equations**, respectively.

The following methods are discussed in this chapter. They have excellent numerical properties and are recommended for use in practice:

- The **Schur methods** for the Lyapunov equations (**Sections 8.5.2 and 8.5.4**).
- The **Hessenberg–Schur Method** for the Sylvester equations (**Algorithm 8.5.1 and Section 8.5.7**).
- The **modified Schur methods** for the Cholesky factors of the Lyapunov equations (**Algorithms 8.6.1 and 8.6.2**).

Besides, a **Hessenberg method** (method based on Hessenberg decomposition only) for the Sylvester equation $AX+XB = C$ has been described in **Section 8.5.6**. The method is more efficient than the Hessenberg–Schur method, but numerical stability of this method has not been investigated yet. At present, the method is mostly of theoretical interest only.

Because of possible numerical instabilities, solving the Lyapunov and Sylvester equations via the Jordan canonical form (JCF) or a companion form of the matrix A cannot be recommended for use in practice (see discussions in Section 8.5.1).

8.2 THE EXISTENCE AND UNIQUENESS OF SOLUTIONS

In most numerical methods for solving matrix equations, it is implicitly assumed that the equation to be solved has a unique solution, and the methods then construct the unique solution. Thus, the results on the existence and uniqueness of solutions of the Sylvester and Lyapunov equations are of importance. We present some of these results in this section.

8.2.1 The Sylvester Equation: $XA + BX = C$

Assume that the matrices A, B, and C are of dimensions $n \times n$, $m \times m$, and $m \times n$, respectively. Then the following is the fundamental result on the existence and uniqueness of the Sylvester equation solution.

Theorem 8.2.1. *Uniqueness of the Sylvester Equation Solution. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A, and $\mu_1, \ldots, \mu_m$, be the eigenvalues of B. Then the Sylvester equation* (8.1.2) *has a unique solution X if and only if $\lambda_i + \mu_j \neq 0$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.* **In other words, the Sylvester equation has a unique solution if and only if A and $-B$ do not have a common eigenvalue.**

Proof. The Sylvester equation $XA + BX = C$ is equivalent to the $nm \times nm$ linear system

$$Px = c, \tag{8.2.1}$$

where $P = (I_n \otimes B) + (A^{\mathrm{T}} \otimes I_m)$,

$$x = \mathrm{vec}(X) = (x_{11}, \ldots, x_{m1}, x_{12}, x_{22}, \ldots, x_{m2}, \ldots, x_{1n}, x_{2n}, \ldots, x_{mn})^{\mathrm{T}},$$

$$c = \mathrm{vec}(C) = (c_{11}, \ldots, c_{m1}, c_{12}, c_{22}, \ldots, c_{m2}, \ldots, c_{1n}, c_{2n}, \ldots, c_{mn})^{\mathrm{T}}.$$

Thus, the Sylvester equation has a unique solution if and only if P is nonsingular.

Here $W \otimes Z$ is the **Kronecker product** of W and Z. Recall from Chapter 2 that if $W = (w_{ij})$ and $Z = (z_{ij})$ are two matrices of orders $p \times p$ and $r \times r$, respectively, then their Kronecker product $W \otimes Z$ is defined by

$$W \otimes Z = \begin{pmatrix} w_{11}Z & w_{12}Z & \cdots & w_{1p}Z \\ w_{21}Z & w_{22}Z & \cdots & w_{2p}Z \\ \vdots & \vdots & & \vdots \\ w_{p1}Z & w_{p2}Z & \cdots & w_{pp}Z \end{pmatrix}. \tag{8.2.2}$$

Thus, the Sylvester equation (8.1.2) has a unique solution if and only if the matrix P of the system (8.2.1) is nonsingular.

Now, the eigenvalues of the matrix P are the nm numbers $\lambda_i + \mu_j$, where $i = 1, \ldots, n$ and $j = 1, \ldots, m$ (Horn and Johnson 1991). Since the determinant of a matrix is equal to the product of its eigenvalues, this means that P is nonsingular if and only if $\lambda_i + \mu_j \neq 0$, for $i = 1, \ldots, n$, and $j = 1, \ldots, m$. ■

8.2.2 The Lyapunov Equation: $XA + A^{\mathrm{T}}X = C$

Since the Lyapunov equation (8.1.1) is a special case of the Sylvester (8.1.2) equation, the following corollary is immediate.

Corollary 8.2.1. *Uniqueness of the Lyapunov Equation Solution. Let* λ_1, $\lambda_2, \ldots, \lambda_n$ *be the eigenvalues of* A. *Then the Lyapunov equation* (8.1.1) *has a unique solution* X *if and only if* $\lambda_i + \lambda_j \neq 0,\ i = 1, \ldots, n;\ j = 1, \ldots, n.$

8.2.3 The Discrete Lyapunov Equation: $A^{\mathrm{T}}XA - X = C$

The following result on the uniqueness of the solution X of the discrete Lyapunov equation

$$A^{\mathrm{T}}XA - X = C \tag{8.2.3}$$

can be established in the same way as in the proof of Theorem 8.2.1.

Theorem 8.2.2. *Uniqueness of the Discrete Lyapunov Equation Solution. Let* $\lambda_1, \ldots, \lambda_n$ *be the* n *eigenvalues of* A. *Then the discrete Lyapunov equation* (8.2.3) *has a unique solution* X *if and only if* $\lambda_i\lambda_j \neq 1, i = 1, \ldots, n; j = 1, \ldots, n.$

Remark

- In the above theorems, we have given results only for the uniqueness of solutions of the Sylvester and Lyapunov equations. However, there are certain control problems such as the **construction of Luenberger observer** and the **EVA problems**, etc., that require **nonsingular or full-rank solutions of the Sylvester equations** (see **Chapter 12**).

The nonsingularity of the unique solution of the Sylvester equation has been completely characterized recently by Datta *et al.* (1997). Also, partial results on nonsingularity of the Sylvester equation were obtained earlier by DeSouza and Bhattacharyya (1981), and Hearon (1977). **We will state these results in Chapter 12.**

8.3 PERTURBATION ANALYSIS AND THE CONDITION NUMBERS

8.3.1 Perturbation Analysis for the Sylvester Equation

In this section, we study perturbation analyses of the Sylvester and Lyapunov equations and identify the condition numbers for these problems. The results are important in assessing the accuracy of the solution obtained by a numerical algorithm. We also present an algorithm (**Algorithm 8.3.1**) for estimating the sep function that arises in computing the condition number of the Sylvester equation.

Let ΔA, ΔB, ΔC, and ΔX be the perturbations, respectively, in the matrices A, B, C, and X. Let $\hat{X}$ be the solution of the perturbed problem. That is, $\hat{X}$ satisfies

$$\hat{X}(A + \Delta A) + (B + \Delta B)\hat{X} = C + \Delta C. \tag{8.3.1}$$

Then, proceeding as in the case of perturbation analysis for the linear system problem applied to the system (8.2.1), the following result (see Higham 1996) can be proved.

Theorem 8.3.1. *Perturbation Theorem for the Sylvester Equation. Let the Sylvester equation $XA + BX = C$ have a unique solution X for $C \neq 0$.*

Let

$$\epsilon = \max\left\{\frac{\|\Delta A\|_F}{\alpha}, \frac{\|\Delta B\|_F}{\beta}, \frac{\|\Delta C\|_F}{\gamma}\right\} \tag{8.3.2}$$

where α, β, and γ are tolerances such that $\|\Delta A\|_F \leq \epsilon\alpha$, $\|\Delta B\|_F \leq \epsilon\beta$, and $\|\Delta C\|_F \leq \epsilon\gamma$.

Then,

$$\frac{\|\Delta X\|_F}{\|X\|_F} = \frac{\|\hat{X} - X\|_F}{\|X\|_F} \leq \sqrt{3}\epsilon\delta, \tag{8.3.3}$$

where $\delta = \|P^{-1}\|_2 \dfrac{(\alpha + \beta)\|X\|_F + \gamma}{\|X\|_F}$.

Sep Function and its Role in Perturbation Results for the Sylvester Equation

Definition 8.3.1. *The separation of two matrices A and B, denoted by* sep(A, B)*, is defined as:*

$$\operatorname{sep}(A, B) = \min_{X \neq 0} \frac{\|AX - XB\|_F}{\|X\|_F}$$

Thus, in terms of the sep *function, we have*

$$\|P^{-1}\|_2 = \frac{1}{\sigma_{\min}(P)} = \frac{1}{\mathrm{sep}(B, -A)}. \tag{8.3.4}$$

Using sep function, the inequality (8.3.3) can be re-written as:

$$\frac{\|\Delta X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} < \sqrt{3}\epsilon \frac{1}{\mathrm{sep}(B, -A)} \frac{(\alpha+\beta)\|X\|_{\mathrm{F}} + \gamma}{\|X\|_{\mathrm{F}}}.$$

The perturbation result (8.3.3) clearly depends upon the norm of the solution X. However, if the relative perturbations in A, B, and C are only of the order of the machine epsilon, then the following result, independent of $\|X\|$, due to Golub *et al.* (1979), can be established.

Corollary 8.3.1. *Assume that the relative perturbations in A, B, and C are all of the order of the machine precision μ, that is, $\|\Delta A\|_{\mathrm{F}} \le \mu\|A\|_{\mathrm{F}}$, $\|\Delta B\|_{\mathrm{F}} \le \mu\|B\|_{\mathrm{F}}$, and $\|\Delta C\|_{\mathrm{F}} \le \mu\|C\|_{\mathrm{F}}$.*

If X is a unique solution of the Sylvester equation $XA + BX = C$, C is nonzero and

$$\mu\frac{\|A\|_{\mathrm{F}} + \|B\|_{\mathrm{F}}}{\mathrm{sep}(B, -A)} \le \frac{1}{2}. \tag{8.3.5}$$

Then,

$$\frac{\|\hat{X} - X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} \le 4\mu\frac{\|A\|_{\mathrm{F}} + \|B\|_{\mathrm{F}}}{\mathrm{sep}(B, -A)}. \tag{8.3.6}$$

Example 8.3.1. Consider the Sylvester equation $XA + BX = C$ with

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} -0.9888 & 0 & 0 \\ 0 & -0.9777 & 0 \\ 0 & 0 & -0.9666 \end{pmatrix}.$$

Take $X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$. Then, $C = \begin{pmatrix} 0.0112 & 1.0112 & 2.0112 \\ 0.0223 & 1.0223 & 2.0223 \\ 0.0334 & 1.0334 & 2.0334 \end{pmatrix}$.

Now, change the entry (1, 1) of A to 0.999999. Call this perturbed matrix $\hat{A}$. The matrices B and C remain unperturbed.

The computed solution of the perturbed problem (computed by MATLAB function **lyap**)

$$\hat{X} = \begin{pmatrix} 1.0001 & 0.9920 & 1.7039 \\ 1.0000 & 0.9980 & 1.0882 \\ 1.0000 & 0.9991 & 1.0259 \end{pmatrix}.$$

The relative error in the solution:

$$\frac{\|\hat{X} - X\|_\mathrm{F}}{\|X\|_\mathrm{F}} = 0.2366.$$

On the other hand, the relative error in the data:

$$\frac{\|A - \hat{A}\|_\mathrm{F}}{\|A\|_\mathrm{F}} = 4.0825 \times 10^{-7}.$$

The phenomenon can be easily explained by noting that $\mathrm{sep}(B, -A)$ is small: $\mathrm{sep}(B, -A) = 1.4207 \times 10^{-6}$.

Verification of the Bound 8.3.3
Take $\alpha = \|A\|_\mathrm{F}$, $\beta = \|B\|_\mathrm{F}$, and $\gamma = \|C\|_\mathrm{F}$

Then,

$$\epsilon = \frac{\|\hat{A} - A\|_\mathrm{F}}{\|A\|_\mathrm{F}} = 4.0825 \times 10^{-7} \quad \text{(Note that } \|\Delta B\| = 0 \text{ and } \|\Delta C\| = 0\text{)}.$$

The right-hand side of (8.3.3) is 2.7133.

Since

$$\frac{\|\hat{X} - X\|_\mathrm{F}}{\|X\|_\mathrm{F}} = 0.2366,$$

the inequality (8.3.3) is satisfied.

8.3.2 The Condition Number of the Sylvester Equation

The perturbation bound for the Sylvester equation given in Theorem 8.3.1 does not take into account the Kronecker structure of the coefficient matrix P. The bound (8.3.3) can sometimes overestimate the effects of perturbations when A and B are only perturbed. A much sharper perturbation bound that exploits the Kronecker structure of P has been given by Higham (1996, p. 318).

Specifically, the following result has been proved.

Theorem 8.3.2. *Let*

$$\epsilon = \max\left\{\frac{\|\Delta A\|_{\mathrm{F}}}{\alpha}, \frac{\|\Delta B\|_{\mathrm{F}}}{\beta}, \frac{\|\Delta C\|_{\mathrm{F}}}{\gamma}\right\},$$

where α, β, and γ are tolerances given by $\|\Delta A\|_{\mathrm{F}} \leq \epsilon\alpha$, $\|\Delta B\| \leq \epsilon\beta$, and $\|\Delta C\| \leq \epsilon\gamma$. Let ΔX denote the perturbation in the solution X of the Sylvester equation (8.1.2). *Let P be defined by* (8.2.1).

Then,

$$\frac{\|\Delta X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} \leq \sqrt{3}\Psi\epsilon, \tag{8.3.7}$$

where

$$\Psi = \|P^{-1}[\beta(X^{\mathrm{T}} \otimes I_m), \alpha(I_n \otimes X), -\gamma I_{mn}]\|_2/\|X\|_{\mathrm{F}}. \tag{8.3.8}$$

The bound (8.3.7) can be attained for any A, B, and C and we shall call the number Ψ the **condition number** of the Sylvester equation.

Remark

- Examples can be constructed that show that the bounds (8.3.3) and (8.3.7) can differ by an arbitrary factor. For details, see Higham (1996).

MATCONTROL note: The condition number given by (8.3.7)–(8.3.8) has been implemented in MATCONTROL function **condsylvc**.

Example 8.3.2. We verify the results of Theorem 8.3.2 with Example 8.3.1. Take $\alpha = \|A\|_{\mathrm{F}} = 2.4494$. Then, $\epsilon = 4.0825 \times 10^{-7}$.

The condition number is $\Psi = 1.0039 \times 10^6$.

$$\frac{\|\Delta X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} = 0.2366 \quad \text{and} \quad \sqrt{3}\Psi\epsilon = 0.7099.$$

Thus, the inequality (8.3.7) is verified.

8.3.3 Perturbation Analysis for the Lyapunov Equation

Since the Lyapunov equation $XA + A^{\mathrm{T}}X = C$ is a special case of the Sylvester equation, we immediately have the following Corollary of Theorem 8.3.1.

Corollary 8.3.2. *Perturbation Theorem for the Lyapunov Equation. Let X be the unique solution of the Lyapunov equation $XA + A^{\mathrm{T}}X = C$; $C \neq 0$. Let $\hat{X}$ be* *the unique solution of the perturbed equation* $\hat{X}(A+\Delta A)+(A+\Delta A)^{\mathrm{T}}\hat{X} = C$,

$$\textit{where } \|\Delta A\|_{\mathrm{F}} \leq \mu\|A\|_{\mathrm{F}}. \tag{8.3.9}$$

Assume that

$$\frac{\mu\|A\|_{\mathrm{F}}}{\operatorname{sep}(A^{\mathrm{T}}, -A)} = \delta < \frac{1}{4}. \tag{8.3.10}$$

Then,

$$\frac{\|\hat{X}-X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} \leq 8\mu\frac{\|A\|_{\mathrm{F}}}{\operatorname{sep}(A^{\mathrm{T}}, -A)}.$$

8.3.4 The Condition Number of the Lyapunov Equation

For the Lyapunov equation, the **condition number** is (Higham (1996, p. 319):

$$\phi = \|(I_n \otimes A^{\mathrm{T}} + A^{\mathrm{T}} \otimes I_n)^{-1}[\alpha((X^{\mathrm{T}} \otimes I_n) + (I_n \otimes X)\Pi^{\mathrm{T}}), -\gamma I_{n^2}]\|_2/\|X\|_{\mathrm{F}},$$

where Π is the **vec-permutation matrix** given by

$$\Pi = \sum_{i,j=1}^{n}(e_i e_j^{\mathrm{T}}) \otimes (e_j e_i^{\mathrm{T}}),$$

and α and γ are as defined as:

$$\|\Delta A\|_{\mathrm{F}} \leq \epsilon\alpha \quad \text{and} \quad \Delta C = \Delta C^{\mathrm{T}} \quad \text{with} \quad \|\Delta C\|_{\mathrm{F}} \leq \epsilon\gamma.$$

8.3.5 Sensitivity of the Stable Lyapunov Equation

While Corollary 8.3.2 shows that the sensitivity of the Lyapunov equation under the assumptions (8.3.9) and (8.3.10) depends upon $\operatorname{sep}(A^{\mathrm{T}}, -A)$, Hewer and Kenney (1988) have shown that if A is a stable matrix, then the sensitivity can be determined by means of the 2-**norm** of the symmetric positive definite solution H of the equation

$$HA + A^{\mathrm{T}}H = -I.$$

Specifically, the following result has been proved:

Theorem 8.3.3. *Perturbation Result for the Stable Lyapunov Equation. Let A be stable and let X satisfy $XA + A^{\mathrm{T}}X = -C$. Let ΔX and ΔC, respectively,* *be the perturbations in X and C such that*

$$(A + \Delta A)^{\mathrm{T}}(X + \Delta X) + (X + \Delta X)(A + \Delta A) = -(C + \Delta C). \qquad (8.3.11)$$

Then,

$$\frac{\|\Delta X\|}{\|X + \Delta X\|} \leq 2\|A + \Delta A\|\|H\|\left[\frac{\|\Delta A\|}{\|A + \Delta A\|} + \frac{\|\Delta C\|}{\|C + \Delta C\|}\right], \qquad (8.3.12)$$

where H satisfies the following Lyapunov equation and $\|\cdot\|$ represents the 2-norm:

$$HA + A^{\mathrm{T}}H = -I.$$

Proof. Since A is stable, we may write $H = \int_0^\infty e^{A^{\mathrm{T}}t}e^{At}\,dt$. Now from $XA + A^{\mathrm{T}}X = -C$ and (8.3.11), we have

$$A^{\mathrm{T}}\Delta X + \Delta XA = -(\Delta C + \Delta A^{\mathrm{T}}(X + \Delta X) + (X + \Delta X)\Delta A). \qquad (8.3.13)$$

Since (8.3.13) is a Lyapunov equation in ΔX and A is stable, we may again write

$$\Delta X = \int_0^\infty e^{A^{\mathrm{T}}t}(\Delta C + \Delta A^{\mathrm{T}}(X + \Delta X) + (X + \Delta X)\Delta A)e^{At}\,dt.$$

Let u and v be the left and right singular vectors of unit length of ΔX associated with the largest singular value. Then multiplying the above equation by u^{T} to the left and by v to the right, we have

$$\begin{aligned}
\|\Delta X\| &= \int_0^\infty \Big|u^{\mathrm{T}}e^{A^{\mathrm{T}}t}(\Delta C + \Delta A^{\mathrm{T}}(X + \Delta X) \\
&\quad + (X + \Delta X)\Delta A)e^{At}v\Big|\,dt, \\
&\leq \|\Delta C + \Delta A^{\mathrm{T}}(X + \Delta X) + (X + \Delta X)\Delta A)\| \\
&\quad \int_0^\infty \|e^{At}u\|\|e^{At}v\|\,dt, \\
&\leq (\|\Delta C\| + 2\|\Delta A\|\|X + \Delta X\|)\int_0^\infty \|e^{At}u\|\|e^{At}v\|\,dt. \qquad (8.3.14)
\end{aligned}$$

Now, by the Cauchy–Schwarz inequality, we have

$$\int_0^\infty \|e^{At}u\|\|e^{At}v\|\,dt \leq \left[\int_0^\infty \|e^{At}u\|^2\,dt\right]^{1/2}\left[\int_0^\infty \|e^{At}v\|^2\,dt\right]^{1/2}.$$

Again

$$\begin{aligned}\int_0^\infty \|e^{At}u\|^2\,dt &= \int_0^\infty u^{\mathrm{T}} e^{A^{\mathrm{T}}t} e^{At} u\,dt,\\ &= u^{\mathrm{T}}\left[\int_0^\infty e^{A^{\mathrm{T}}t} e^{At}\,dt\right] u,\\ &= u^{\mathrm{T}} H u, \text{ where } H = \int_0^\infty e^{A^{\mathrm{T}}t} e^{At}\,dt\end{aligned}$$

Since $\|u\|_2 = 1$ and H is symmetric positive definite (because A is stable), we have

$$u^{\mathrm{T}} H u \le \|H\|,$$

and thus

$$\int_0^\infty \|e^{At}u\|^2\,dt \le \|H\|.$$

Similarly $\int_0^\infty \|e^{At}v\|^2\,dt \le \|H\|$.

Thus, from (8.3.14), we have

$$\|\Delta X\| \le (\|\Delta C\| + 2\|\Delta A\|\|X + \Delta X\|)\|H\|. \tag{8.3.15}$$

Again from (8.3.11), we have

$$\|C + \Delta C\| \le 2\|A + \Delta A\|\|X + \Delta X\|. \tag{8.3.16}$$

Combining (8.3.15) with (8.3.16), we obtain the desired result. ■

Remark

- The results of Theorem 8.3.3 hold for any perturbation.
  In particular, if

$$\|\Delta C\| \le \mu\|C\|, \qquad \|\Delta A\| \le \mu\|A\|,$$

and $8\mu\|A\|\|H\| \le (1-\mu)/(1+\mu)$, then it can be shown that

$$\frac{\|\Delta X\|}{\|X\|} \le 8\mu\|A\|\|H\|(1-\mu) \approx 8\mu\|A\|\|H\|. \tag{8.3.17}$$

Example 8.3.3. Consider the Lyapunov equation (8.1.1) with

$$A = \begin{pmatrix} -1 & 2 & 3 \\ 0 & -0.0001 & 3 \\ 0 & 0 & -3 \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} -2 & 0.9999 & 2 \\ 0.9999 & 3.9998 & 4.9999 \\ 2 & 4.9999 & 6 \end{pmatrix}.$$

A is stable. The exact solution X of the Lyapunov equation $XA + A^{\mathrm{T}}X = C$ is

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

The solution H of the Lyapunov equation $HA + A^{\mathrm{T}}H = -I$ is

$$H = 10^4 \begin{pmatrix} 0.0001 & 0.0001 & 0.0001 \\ 0.0001 & 2.4998 & 2.4999 \\ 0.0001 & 2.4999 & 2.5000 \end{pmatrix}.$$

Since $\|H\| = 4.9998 \times 10^4$ and $\|A\| = 5.3744$, according to Theorem 8.3.3, the Lyapunov equation with above A and C is expected to be ill-conditioned. We verify this as follows:

Perturb the (1, 1) entry of A to -0.9999999 and keep the other entries of A and those of C unchanged. Then the computed solution $\hat{X}$ with this slightly perturbed A is

$$\hat{X} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.006 & 1.006 \\ 1 & 1.006 & 1.006 \end{pmatrix}.$$

Let $\hat{A}$ denote the perturbed A, then the relative perturbation in A:

$$\frac{\|\hat{A} - A\|}{\|A\|} = 1.8607 \times 10^{-8}.$$

The relative error in the solution X:

$$\frac{\|\hat{X} - X\|}{\|X\|} = 0.0040.$$

Verification of the result of Theorem 8.3.3

$$\frac{\|\Delta X\|}{\|X + \Delta X\|} = 0.0040, \qquad \Delta C = 0,$$

$$2\|A + \Delta A\|\,\|H\|\frac{\|\Delta A\|}{\|A + \Delta A\|} = 0.0100.$$

Thus, the inequality (8.3.12) is satisfied.

Verification of the inequality (8.3.17)

Since $\|\Delta A\|/\|A\| = 1.8607 \times 10^{-8}$, we take $\mu = 1.8607 \times 10^{-8}$.

Then $8\mu\|A\|\|H\| = 0.04 \leq (1-\mu)/(1+\mu) = 0.9999996$. Thus, the hypothesis holds.

Also, $\|\Delta X\|/\|X\| = 0.004$, $8\mu\|A\|\|H\| = 0.04$. Therefore, the inequality (8.3.17) is satisfied.

8.3.6 Sensitivity of the Discrete Lyapunov Equation

Consider now the discrete Lyapunov equation:

$$A^{\mathrm{T}}XA - X = C.$$

This equation is equivalent to the linear system: $Rx = c$, where $R = A^{\mathrm{T}} \otimes A^{\mathrm{T}} - I_{n^2}$ (I_{n^2} is the $n^2 \times n^2$ identity matrix).

Applying the results of perturbation analysis to the linear system $Rx = c$, the following result can be proved.

Theorem 8.3.4. *Perturbation Result for the Discrete Lyapunov Equation. Let X be the unique solution of the discrete Lyapunov equation:*

$$A^{\mathrm{T}}XA - X = C.$$

Let $\hat{X}$ be the unique solution of the perturbed equation where the perturbation in A is of order machine precision μ.

Assume that

$$\frac{(2\mu + \mu^2)\|A\|_{\mathrm{F}}^2}{\mathrm{sep}_d(A^{\mathrm{T}}, A)} = \delta < 1,$$

where

$$\mathrm{sep}_d(A^{\mathrm{T}}, A) = \min_{x \neq 0} \frac{\|Rx\|_2}{\|x\|_2} = \min_{X \neq 0} \frac{\|A^{\mathrm{T}}XA - X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} = \sigma_{\min}(A^{\mathrm{T}} \otimes A^{\mathrm{T}} - I_{n^2}).$$

Then,

$$\frac{\|\hat{X} - X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} \leq \frac{\mu}{1-\delta} \frac{(3+\mu)\|A\|_{\mathrm{F}}^2 + 1}{\mathrm{sep}_d(A^{\mathrm{T}}, A)}. \tag{8.3.18}$$

8.3.7 Sensitivity of the Stable Discrete Lyapunov Equation

As in the case of the stable Lyapunov equation, it can be shown (Gahinet *et al.* 1990) that the sensitivity of the stable discrete Lyapunov equation can also be measured by the **2-norm** of the unique solution of the discrete Lyapunov equation: $A^{\mathrm{T}}XA - X = -I$. Specifically, the following result has been proved by Gahinet *et al.* (1990).

Theorem 8.3.5. *Sensitivity of the Stable Discrete Lyapunov Equation. Let A be discrete stable. Let H be the unique solution of*

$$A^{\mathrm{T}}HA - H = -I,$$

then $\operatorname{sep}_d(A^{\mathrm{T}}, A) \geq \sqrt{n}/\|H\|_2$.

Example 8.3.4. Let

$$A = \begin{pmatrix} 0.9990 & 1 & 1 \\ 0 & 0.5000 & 1 \\ 0 & 0 & 0.8999 \end{pmatrix}.$$

Set

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{and take} \quad C = A^{\mathrm{T}}XA - X.$$

Then $\|H\|_2 = 4.4752 \times 10^5$.

By Theorem 8.3.5, the discrete Lyapunov equation $A^{\mathrm{T}}XA - X = C$ is expected to be ill-conditioned. We verify this as follows.

Let $a(2, 2)$ be perturbed to 0.4990 and all other entries of A and of C remain unchanged. Let $\hat{A}$ denote the perturbed A. Let $\hat{X}$ be the solution of the perturbed problem. Then $\hat{X}$, computed by the MATLAB function **dlyap**, is

$$\hat{X} = \begin{pmatrix} 1 & 1 & 1.0010 \\ 1 & 1 & 1.0019 \\ 1.0010 & 1.0019 & 1.0304 \end{pmatrix}.$$

The relative error in X:

$$\frac{\|\hat{X} - X\|_2}{\|X\|_2} = 0.0102.$$

The relative perturbation in A:

$$\frac{\|\hat{A} - A\|_2}{\|A\|_2} = 4.8244 \times 10^{-5}.$$

8.3.8 Determining Ill-Conditioning from the Eigenvalues

Since $\|P^{-1}\|_2 = 1/\operatorname{sep}(B, -A)$ is not easily computable, and $\operatorname{sep}(B, -A) > 0$ if and only if B and $-A$ have no common eigenvalues, one may wonder if the ill-conditioning of P^{-1} (and therefore of the Sylvester or the Lyapunov equation) can be determined a priori from the eigenvalues of A and B.

The following result can be easily proved to this effect (Ghavimi and Laub 1995).

Theorem 8.3.6. *The Sylvester equation $XA + BX = C$ is ill-conditioned if both coefficient matrices A and B are ill-conditioned with respect to inversion.*

Example 8.3.5. Let

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0.001 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0.0001 & 1 \\ 0 & 0 & 0.0001 \end{pmatrix},$$

$$\text{and} \quad C = \begin{pmatrix} 8 & 8 & 8.001 \\ 3.0001 & 3.0001 & 3.0011 \\ 2.0001 & 2.0001 & 2.0011 \end{pmatrix}.$$

The exact solution $X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$.

Now change $a(3, 1)$ to 0.99999 and keep the rest of the data unchanged. Then the solution of the perturbed problem is

$$\hat{X} = \begin{pmatrix} 908.1970 & -905.2944 & -906.2015 \\ -452.6722 & 454.2208 & 454.6745 \\ 1.0476 & 0.9524 & 0.9524 \end{pmatrix},$$

which is completely different from the exact solution X.

Note that the relative error in the solution:

$$\frac{\|X - \hat{X}\|}{\|X\|} = 585.4190.$$

However, the relative perturbation in the data:

$$\frac{\|A - \hat{A}\|}{\|A\|} = 4.5964 \times 10^{-6} \ (\hat{A} \text{ is the perturbed matrix}).$$

This drastic change in the solution X can be explained by noting that **A and B are both ill-conditioned**:

$$\text{Cond}(A) = 6.1918 \times 10^{16}, \qquad \text{Cond}(B) = 8.5602 \times 10^{8}.$$

Remark

- The converse of the above theorem is, in general, not true. To see this, consider Example 8.3.1 once more. We have seen that the Sylvester equation with the data of this example is ill-conditioned. But note that $\text{Cond}(A) = 4.0489$, $\text{Cond}(B) = 1.0230$. Thus, neither A nor B is ill-conditioned.

Near Singularity of A and the Ill-conditioning of the Lyapunov Equation

From Theorem 8.3.6, we immediately obtain the following corollary:

Corollary 8.3.3. *If A is nearly singular, then the Lyapunov equation $XA + A^{\mathrm{T}}X = C$ is ill-conditioned.*

Example 8.3.6. Let

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0.0001 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 2 & 2.0001 & 4 \\ 2.0001 & 2.0002 & 4.0001 \\ 4 & 4.0001 & 6 \end{pmatrix}.$$

The exact solution

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Now perturb the $(1, 1)$ entry of A to 0.9880. Call the perturbed matrix $\hat{A}$. The computed solution of the perturbed problem

$$\hat{X} = \begin{pmatrix} 1.0121 & 0.9999 & 1.0000 \\ 0.9999 & 2.4750 & -0.4747 \\ 1.000 & -0.4747 & 2.4747 \end{pmatrix}.$$

The relative error in X:

$$\frac{\|\hat{X} - X\|}{\|X\|} = 0.9832.$$

The relative perturbation in A:

$$\frac{\|\hat{A} - A\|}{\|A\|} = 0.0060.$$

The ill-conditioning of the Lyapunov equation with the given data can be explained from the fact that A is nearly singular. Note that $\mathrm{Cond}(A) = 3.9999 \times 10^4$ and $\mathrm{sep}(A^{\mathrm{T}}, -A) = 5.001 \times 10^{-5}$.

8.3.9 A Condition Number Estimator for the Sylvester Equation: $A^{\mathrm{T}}X - XB = C$

We have seen in **Section 8.3.1** that

$$\mathrm{sep}(B, -A) = \frac{1}{\|P^{-1}\|_2} = \sigma_{\min}(P),$$

where P is the coefficient matrix of the linear system (8.2.1).

However, finding $\text{sep}(B, -A)$ by computing the smallest singular value of P^{-1} requires a major computational effort. Even for modest m and n, it might be computationally prohibitive from the viewpoints of both the storage and the computational cost. It will require $O(m^3n^3)$ flops and $O(m^2n^2)$ storage.

Byers (1984) has proposed an algorithm to estimate $\text{sep}(A, B^{\mathrm{T}})$ in the style of the LINPACK condition number estimator. The LINPACK condition number estimator for $\text{Cond}(P)$ is based on estimating $\|P^{-1}\|_2$ by $\|y\|/\|z\|$, where y, z, and w satisfy

$$P^{\mathrm{T}}z = w \quad \text{and} \quad Py = z;$$

the components of the vector w are taken to be $w_i = \pm 1$, where the signs are chosen such that the growth in z is maximized.

Algorithm 8.3.1. *Estimating* $\text{sep}(A, B^{\mathrm{T}})$.
Input. $A_{m\times m}$, $B_{n\times n}$—*Both upper triangular matrices.*
Output. *Sepest—An estimate of* $\text{sep}(A, B^{\mathrm{T}})$.
Step 1.

```
For $i = m, m-1, \ldots, 1$ do
    For $j = n, n-1, \ldots, 1$ do
        $p \equiv \left(\sum_{h=i+1}^{m} a_{ih}z_{hj} - \sum_{h=j+1}^{n} z_{ih}b_{jh}\right)$
        $w \equiv -\text{sign}(p)$
        $z_{ij} \equiv (w - p)/(a_{ii} - b_{jj})$
    End
End
```

Step 2. *Compute* $Z \equiv Z/\|Z\|$, *where* $Z = (z_{ij})$.
Step 3. *Solve for* Y: $A^{\mathrm{T}}Y - YB = Z$.
Step 4. *Sepest* $= 1/\|Y\|$.

Example 8.3.7. Consider estimating $\text{sep}(B, -A)$ with

$$A = \begin{pmatrix} -1 & 2 & 3 \\ 0 & -2 & 1 \\ 0 & 0 & 0.9990 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -1 & 2 & 3 \\ 0 & -2.5 & 0 \\ 0 & 0 & 1.9999 \end{pmatrix}.$$

The algorithm produces sepest $(B, -A) = \mathrm{O}(10^{-5})$, whereas the actual value of $\text{sep}(B, -A) = 3.0263 \times 10^{-5}$.

Remarks

- If $p = 0$, $\text{sign}(p)$ can be taken arbitrarily.

- The major work in the algorithm is in the solution of the Sylvester equation in Step 3. Thus, once this equation is solved, the remaining part of the algorithm requires only a little extra work. Efficient numerical methods for solving the Sylvester and Lyapunov equations are discussed in the next section.

Flop-count: The algorithm requires $2(m^2n + mn^2)$ flops.

Remarks

- The algorithm must be modified when A and B are in quasi-triangular forms (real Schur forms, RSFs), or one is in Hessenberg form and the other is in RSF.
- There exists a LAPACK-style (rich in Basic Linear Algebra Subroutine-Level 3 operators), estimator for $\text{sep}(B, -A)$. For details, see Kågström and Poromaa (1989, 1992, 1996).

MATCONTROL notes: The sep function can be computed using the Kronecker product in MATCONTROL function **sepkr**. Algorithm 8.3.1 has been implemented in MATCONTROL function **sepest**, which calls the function **sylvhutc** for solving the upper triangular Sylvester equation in Step 3.

8.4 ANALYTICAL METHODS FOR THE LYAPUNOV EQUATIONS: EXPLICIT EXPRESSIONS FOR SOLUTIONS

There are numerous methods for solving Lyapunov and Sylvester equations. They can be broadly classified into two classes: **Analytical** and **Numerical** Methods.

By an analytical method, we mean a method that attempts to give an explicit expression for the solution matrix (usually the unique solution).

Recall from Chapter 7 that when A is a stable matrix, a unique solution X of the continuous-time Lyapunov equation requires computations of the matrix exponential e^{At} and evaluation of a matrix integral.

Similarly, when A is discrete-stable, a unique solution X of the discrete Lyapunov equation requires computations of various powers of A and many matrix multiplications. We have already seen that there are some obvious computational difficulties with these computations.

The other analytical methods include **finite and infinite series methods** (see Barnett and Storey (1970)).

These methods again have some severe computational difficulties. For example, consider the solution of the Lyapunov equation $XA + A^{\mathrm{T}}X = C$, using the **finite**

series method proposed by Jameson (1968). The method can be briefly described as follows:

Let the characteristic polynomial of A be $\det(\lambda I - A) = \lambda^n + c_1\lambda^{n-1} + \cdots + c_n$. Define the sequence of matrices $\{Q_k\}$ by

$$\begin{aligned} Q_{-1} &= 0, \qquad Q_0 = C \\ Q_k &= A^{\mathrm{T}} Q_{k-1} - Q_{k-1}A + A^{\mathrm{T}} Q_{k-2}A, \quad k = 1, 2, \ldots, n. \end{aligned} \tag{8.4.1}$$

Then it has be shown that

$$X = P^{-1}(Q_n - c_1 Q_{n-1} + \cdots + (-1)^n c_n Q_0), \tag{8.4.2}$$

where $P = \left(A^{\mathrm{T}}\right)^n - c_1 \left(A^{\mathrm{T}}\right)^{n-1} + \cdots + (-1)^n c_n I$.

It can be seen from the description of the method that it is not numerically effective for practical computations.

Note that for computation of the matrix P, various powers of A need to be computed and the matrix P can be ill-conditioned, which will affect the accuracy of X. This, together with the fact that the sensitivity of the characteristic polynomial of a matrix A (due to the small perturbations in the coefficients) grows as the order of the matrix grows (in general), lead us to believe that such methods will give unacceptable accuracy. **Indeed, the numerical experiments show that for random matrices of size** 14 × 14, **the errors are almost as large as the solutions themselves.**

Thus, we will not pursue further with the analytic methods. However, for reader's convenience, to compare this method with other numerically viable methods, the finite series method has been implemented in MATCONTROL function **lyapfns**.

8.5 NUMERICAL METHODS FOR THE LYAPUNOV AND SYLVESTER EQUATIONS

An obvious way to solve the Sylvester equation $XA + BX = C$ is to apply Gaussian elimination with partial pivoting to the system $Px = c$ given by (8.2.1). But, unless the special structure of P can be exploited, Gaussian elimination scheme for the Sylvester equation will be **computationally prohibitive**, since $O(n^3m^3)$ flops and $O(n^2m^2)$ storage will be required. One way to exploit the structure of P will be to transform A and B to some **simple forms** using similarity transformations.

Thus, if U and V are nonsingular matrices such that

$$U^{-1}AU = \hat{A}, \qquad V^{-1}BV = \hat{B}, \qquad \text{and} \qquad V^{-1}CU = \hat{C},$$

then $XA + BX = C$ is transformed to

$$Y\hat{A} + \hat{B}Y = \hat{C}, \tag{8.5.1}$$

where $Y = V^{-1}XU$.

If $\hat{A}$ and $\hat{B}$ are in simple forms, then the equation $Y\hat{A} + \hat{B}Y = \hat{C}$ can be easily solved, and the solution X can then be recovered from Y. The idea, therefore, is summarized in the following steps:

Step 1. Transform A and B to "**simple**" forms (e.g., diagonal, Jordan and companion, Hessenberg, real-Schur, and Schur etc.) using similarity transformations:

$$\hat{A} = U^{-1}AU, \qquad \hat{B} = V^{-1}BV.$$

Step 2. Update the right-hand side matrix: $\hat{C} = V^{-1}CU$.
Step 3. Solve the **transformed equation** for Y: $Y\hat{A} + \hat{B}Y = \hat{C}$.
Step 4. Recover X from Y by solving the system: $XU = VY$.

8.5.1 Numerical Instability of Diagonalization, Jordan Canonical Form, and Companion Form Techniques

It is true that the rich structures of Jordan and companion matrices can be nicely exploited in solving the reduced Sylvester equation (8.5.1). However, as noted before, the companion, and JCFs, in general, cannot be obtained in a numerically stable way. (For more on numerically computing the JCF, see Golub and Wilkinson (1976).) The transforming matrices will be, in some cases, ill-conditioned and this ill-conditioning will affect the computations of $\hat{A}$, $\hat{B}$, $\hat{C}$, and X (from Y), which require computations involving inverses of the transforming matrices. Indeed, **numerical experiments performed by us show that solutions of the Sylvester equation using companion form of A with A of sizes larger than 15 have errors almost as large as the solutions themselves.** We will illustrate below by a simple example how diagonalization technique yields an inaccurate solution.

Example 8.5.1. Consider solving the Lyapunov equation: $XA + A^{\mathrm{T}}X - C = 0$, with

$$A = \begin{pmatrix} 2.4618 & -1.5284 & 2.2096 & -0.3503 \\ 5.5854 & -1.2161 & 2.3825 & -1.2843 \\ 1.6935 & 2.5009 & 2.1131 & -1.2186 \\ -0.2686 & -3.2594 & 7.9205 & 0.6412 \end{pmatrix}.$$

Choose

$$X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{and take} \quad C = XA + A^{\mathrm{T}}X.$$

Let X_{Diag} be the solution obtained by the diagonalization procedure (using MATLAB Function **lyap2**).

The relative residual:

$$\frac{\|X_{\text{Diag}}A + A^{\mathrm{T}}X_{\text{Diag}} - C\|}{\|X_{\text{Diag}}\|} = 1.6418 \times 10^{-7}.$$

The solution $\hat{X}$ obtained by MATLAB function **lyap** (based on the numerically viable **Schur method**):

$$\hat{X} = \begin{pmatrix} 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{pmatrix}.$$

The relative residual:

$$\frac{\|\hat{X}A + A^{\mathrm{T}}\hat{X} - C\|}{\|\hat{X}\|} = 9.5815 \times 10^{-15}.$$

Solutions via Hessenberg and Schur Forms

In view of the remarks made above, our "**simple**" forms of choice have to be **Hessenberg forms** and the **(real) Schur forms**, since we know that the transforming matrices U and V in these cases can be chosen to be orthogonal, which are perfectly well-conditioned. Some such methods are discussed in the following sections.

8.5.2 The Schur Method for the Lyapunov Equation: $XA + A^{\mathrm{T}}X = C$

The following method, proposed by Bartels and Stewart (1972), is **now widely used as an effective computational method for the Lyapunov equation.** The method is based on reduction of A^{T} to RSF. It is, therefore, known as the **Schur method for the Lyapunov equation.** The method is described as follows:

Step 1. Reduction of the Problem. Let $R = U^{\mathrm{T}}A^{\mathrm{T}}U$ be the **RSF** of the matrix A^{T}. Then, employing this transformation, the Lyapunov matrix equation $XA + A^{\mathrm{T}}X = C$ is reduced to

$$YR^{\mathrm{T}} + RY = \hat{C}, \tag{8.5.2}$$

where $R = U^{\mathrm{T}}A^{\mathrm{T}}U$, $\hat{C} = U^{\mathrm{T}}CU$, and $Y = U^{\mathrm{T}}XU$.

Step 2. Solution of the Reduced Problem. The reduced equation to be solved is: $YR^{\mathrm{T}} + RY = \hat{C}$. Let

$$Y = (y_1, \ldots, y_n), \qquad \hat{C} = (c_1, \ldots, c_n), \qquad \text{and} \qquad R = (r_{ij}).$$

Assume that the columns y_{k+1} through y_n have been computed, and consider the following two cases.

Case 1: $r_{k,k-1} = 0$. Then y_k is determined by solving the quasi-triangular system

$$(R + r_{kk}I)y_k = c_k - \sum_{j=k+1}^{n} r_{kj}y_j.$$

If, in particular, R **is upper triangular**, that is there are no "**Schur bumps**" (see **Chapter 4**) on the diagonal, then each $y_i, i = n, n-1, \ldots, 2, 1$ can be obtained by solving an $n \times n$ upper triangular system as follows:

$$\begin{aligned}(R + r_{nn}I)y_n &= c_n,\\ (R + r_{n-1,n-1}I)y_{n-1} &= c_{n-1} - r_{n-1,n}y_n,\\ &\vdots\\ (R + r_{11}I)y_1 &= c_1 - r_{12}y_2 - \cdots - r_{1n}y_n.\end{aligned} \tag{8.5.3}$$

Remark

- If the complex Schur decomposition is used, that is, if $R_c = U_c^* A^T U_c$ is a complex triangular matrix, then the solution Y_c of the reduced problem is computed by solving n complex $n \times n$ linear systems (8.5.3). The MATLAB function **rsf2csf** converts an RSF to a complex triangular matrix. However, the use of complex arithmetic is more expensive and **not recommended in practice**.

Case 2: $r_{k,k-1} \neq 0$ **for some** k. This indicates that there is a "**Schur bump**" on the diagonal. This enables us to compute y_{k-1} and y_k simultaneously, by solving the following $2n \times 2n$ linear system:

$$\begin{aligned}&R(y_{k-1}, y_k) + (y_{k-1}, y_k)\begin{pmatrix} r_{k-1,k-1} & r_{k,k-1} \\ r_{k-1,k} & r_{kk} \end{pmatrix}\\ &\quad = (c_{k-1}, c_k) - \sum_{j=k+1}^{n} (r_{k-1,j}y_j, r_{kj}y_j) = (d_{k-1}, d_k).\end{aligned} \tag{8.5.4}$$

Remark

- To distinguish between Case 1 and Case 2 in a computational setting, it is recommended that some threshold, for example, Tol $= \mu\|A\|_F$ be used, where μ is the machine precision.
- Thus, to see if $r_{k,k-1} = 0$, accept $r_{k,k-1} = 0$, if $|r_{k,k-1}| <$ Tol.

An Illustration

We illustrate the above procedure with $n = 3$.

Assume that $r_{21} \neq 0$, that is,

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix}.$$

Since $r_{32} = 0$, by **Case 1**, y_3 is computed by solving the system:

$$(R + r_{33}I)\,y_3 = c_3.$$

Since $r_{21} \neq 0$, by **Case 2**, y_1 and y_2 are computed by solving

$$R(y_1, y_2) + (y_1, y_2)\begin{pmatrix} r_{11} & r_{21} \\ r_{12} & r_{22} \end{pmatrix} = (c_1 - r_{13}y_3, c_2 - r_{23}y_3). \tag{8.5.5}$$

Step 3. Recovery of the solution of the original problem from the solution of the reduced problem. Once Y is obtained by solving the reduced problem $YR^{\mathrm{T}} + RY = \hat{C}$, the solution X of the original problem $XA + A^{\mathrm{T}}X = C$, is recovered as

$$X = UYU^{\mathrm{T}}.$$

Example 8.5.2. Consider solving the Lyapunov equation: $XA + A^{\mathrm{T}}X = C$, with

$$A = \begin{pmatrix} 0 & 2 & -1 \\ -3 & -2 & 2 \\ -2 & 1 & -1 \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} -2 & 2 & -3 \\ -8 & -6 & -5 \\ 11 & 13 & -2 \end{pmatrix},$$

Step 1. *Reduction.* Using MATLAB function $[U, R]$ =**schur**(A^{T}), we obtain

$$R = \begin{pmatrix} -1.3776 & 3.8328 & 1.3064 \\ -1.0470 & 0.8936 & -1.2166 \\ 0 & 0 & -2.5160 \end{pmatrix}, \quad U = \begin{pmatrix} 0.7052 & 0.4905 & 0.5120 \\ 0.6628 & -0.7124 & -0.2304 \\ -0.2518 & -0.5019 & 0.8275 \end{pmatrix}.$$

Then $\hat{C} = U^{\mathrm{T}}CU$ is

$$\hat{C} = \begin{pmatrix} -9.3174 & 1.9816 & -7.5863 \\ -2.1855 & -1.0425 & 2.4422 \\ 16.2351 & -3.4886 & 0.3600 \end{pmatrix}.$$

Step 2. Solve $RY + YR^{\mathrm{T}} = \hat{C}$. Since $r_{32} = 0$, then by **Case 1**, y_3 is computed by solving the system:

$$\begin{pmatrix} -3.8936 & 3.8328 & 1.3064 \\ -1.0470 & -1.6224 & -1.2166 \\ 0 & 0 & -5.0320 \end{pmatrix} y_3 = \begin{pmatrix} -7.5863 \\ 2.4422 \\ 0.3600 \end{pmatrix}.$$

$$y_3 = \begin{pmatrix} 0.3030 \\ -1.6472 \\ -0.0715 \end{pmatrix}.$$

Since $r_{21} \neq 0$, then by **Case 2**, y_1 and y_2 are computed by solving the system (8.5.5):

$$(y_1^{\mathrm{T}}, y_2^{\mathrm{T}}) = (3.4969, 0.1669, -1.2379, 0.2345, 0.5746, 3.0027)^{\mathrm{T}}.$$

Step 3. *Recovery of the solution.*

$$X = UYU^{\mathrm{T}} = \begin{pmatrix} 2 & 0 & -2 \\ 2 & 2 & 1 \\ 0 & -3 & 0 \end{pmatrix}.$$

Example 8.5.3. We now solve the previous example (Example 8.5.2) using the **complex Schur form**

Step1. $R = \begin{pmatrix} -0.2420 + 1.6503j & -0.3227 + 3.5797j & -0.0927 - 0.9538j \\ 0 & -0.2420 - 1.6503j & 1.2113 - 0.8883j \\ 0 & 0 & -2.5160 \end{pmatrix}$,

$$U = \begin{pmatrix} -0.5814 + 0.5148j & 0.0802 - 0.3581j & 0.5120 \\ -0.0030 + 0.4839j & 0.6649 + 0.5202j & -0.2304 \\ 0.3590 - 0.1838j & 0.1355 + 0.3664j & -0.8275 \end{pmatrix},$$

$$\hat{C} = \begin{pmatrix} -7.5894 + 1.4093j & 1.8383 + 4.252j & 2.6799 + 5.5388j \\ 2.1139 - 1.1953j & -2.7706 - 1.4093j & -4.7410 + 1.783j \\ -6.5401 + 11.8534j & 9.2728 + 2.5470j & 0.3600 \end{pmatrix}$$

Step 2. Since R is triangular (complex), the columns y_1, y_2, y_3 of y are successively computed by solving **complex linear systems** (8.5.3). This gives

$$y_1 = (-2.9633 + 0.000229j, -0.7772 + 0.8659j, -0.7690 - 0.9038j)^{\mathrm{T}},$$
$$y_2 = (-0.7811 - 0.8163j, 1.1082 - 0.0229j, -2.0819 - 2.923j)^{\mathrm{T}},$$
$$y_3 = (0.6108 - 0.2212j, 0.9678 - 1.2026j, -0.0715)^{\mathrm{T}}.$$

Thus, with $Y = (y_1, y_2, y_3)$, we have

$$X = UYU^{\mathrm{T}} = \begin{pmatrix} 2 & 0 & -2 \\ 2 & 2 & 1 \\ 0 & -3 & 0 \end{pmatrix}.$$

Note: In practice, the system (8.5.4) is solved using Gaussian elimination with partial pivoting. The LAPACK and SLICOT routines (see **Section 8.10.4**) have used Gaussian elimination with complete pivoting (see Datta (1995) and Golub and Van Loan (1996)) and the structure of the RSF has been exploited there. For details of implementations, the readers may consult the book by Sima (1996).
MATCONTROL note: The Schur method for the Lyapunov equation has been implemented in MATCONTROL function **lyaprsc**.
MATLAB note: MATLAB function **lyap** in the form

$$X = \textbf{lyap}\ (A, C)$$

solves the Lyapunov equation

$$AX + XA^{\mathrm{T}} = -C$$

using the **complex Schur** triangularization of A.

Flop-count

1. Transformation of A to RSF: $26n^3$ (Assuming that the QR iteration algorithm requires about two iterations to make a subdiagonal entry negligible). (This count includes construction of U.)
2. Solution of the reduced problem: $3n^3$
3. Recovery of the solution: $3n^3$ (using the symmetry of X).
   Total flops: $32n^3$ (**Approximate**).

8.5.3 The Hessenberg–Schur Method for the Sylvester Equation

The Schur method described above for the Lyapunov equation can also be used to solve the Sylvester equation $XA + BX = C$. The matrices A and B are, respectively, transformed to the lower and upper RSFs, and then back-substitution is used to solve the reduced Schur problem. Note, that the special form of the Schur matrix S can be exploited only in the solution of the $m \times m$ linear systems with S. Some computational effort can be saved if B, the larger of the two matrices A and B, is left in Hessenberg form, while the smaller matrix A is transformed further to RSF. The reason for this is that a matrix must be transformed to a Hessenberg matrix as an initial step in the reduction to RSF (see **Chapter 4**). The important outcome here is that back-substitution for the solution of the Hessenberg–Schur problem is still possible. Noting this, Golub *et al.* (1979) developed the following Hessenberg–Schur method for the Sylvester equation problem.

Step 1. *Reduction to the Hessenberg–Schur Problem.* Assume that m is larger than n. Let $R = U^{\mathrm{T}}A^{\mathrm{T}}U$ and $H = V^{\mathrm{T}}BV$ be, respectively, the upper RSF and the upper Hessenberg form of A and B. Then,

$$\begin{gathered} XA + BX = C \quad \text{becomes} \quad YR^{\mathrm{T}} + HY = \hat{C}, \\ \text{where} Y = V^{\mathrm{T}}XU, \; \hat{C} = V^{\mathrm{T}}CU. \end{gathered} \tag{8.5.6}$$

Step 2. *Solution of the Reduced Hessenberg–Schur Problem.* In the reduced problem $HY + YR^{\mathrm{T}} = \hat{C}$, let $Y = (y_1, y_2, \ldots, y_n)$ and $\hat{C} = (c_1, \ldots, c_n)$. Then, assuming that $y_{k+1}, \ldots, y_n$ have already been computed, y_k (or y_k and y_{k+1}) can be computed as in the case of the Lyapunov equation, by considering the following two cases.

Case 1. If $r_{k,k-1} = 0$, y_k is computed by solving the $m \times m$ Hessenberg system:

$$(H + r_{kk}I)y_k = c_k - \sum_{j=k+1}^{n} r_{kj}y_j.$$

Case 2. If $r_{k,k-1} \neq 0$, then equating columns $k-1$ and k in $HY + YR^{\mathrm{T}} = \hat{C}$, it is easy to see that y_{k-1} and y_k are simultaneously computed by solving the $2m \times 2m$ linear system:

$$\begin{aligned} &H(y_{k-1}, y_k) + (y_{k-1}, y_k)\begin{pmatrix} r_{k-1,k-1} & r_{k,k-1} \\ r_{k-1,k} & r_{kk} \end{pmatrix} \\ &\quad = (c_{k-1}, c_k) - \sum_{j=k+1}^{n} (r_{k-1,j}y_j, r_{kj}y_j) = (d_{k-1}, d_k) \end{aligned} \tag{8.5.7}$$

Note: The matrix of the system can be made upper triangular with two nonzero subdiagonals, by reordering the variables suitably. The upper triangular system can then be solved using Gaussian elimination with partial pivoting.

Step 3. *Recovery of the Original Solution.* The solution X is recovered from Y as

$$X = VYU^{\mathrm{T}}.$$

Algorithm 8.5.1. *The Hessenberg–Schur Algorithm for $XA + BX = C$*
Input: *The matrices A, B, and C, respectively, of order $n \times n, m \times m$, and $m \times n$; $n \leq m$.*
Output: *The matrix X satisfying $XA + BX = C$.*

Step 1. *Transform A^{T} to a real Schur matrix R, and B to an upper Hessenberg matrix H by orthogonal similarity:*

$$U^{\mathrm{T}}A^{\mathrm{T}}U = R, \qquad V^{\mathrm{T}}BV = H.$$

Form $\hat{C} = V^{\mathrm{T}}CU$, and partition $\hat{C} = (c_1, \dots, c_n)$ by columns.

Step 2. *Solve $HY + YR^{\mathrm{T}} = \hat{C}$:*

For $k = n, \dots, 1$ do until the columns of Y are computed
If $r_{k,k-1} = 0$, then compute y_k by solving the Hessenberg system:

$$(H + r_{kk}I)y_k = c_k - \sum_{j=k+1}^{n} r_{kj}y_j \tag{8.5.8}$$

Else, compute y_k and y_{k-1} by solving the system:

$$\begin{pmatrix} H + r_{k-1,k-1}I & r_{k-1,k}I \\ r_{k,k-1}I & H + r_{kk}I \end{pmatrix} \begin{pmatrix} y_{k-1} \\ y_k \end{pmatrix} = \begin{pmatrix} d_{k-1} \\ d_k \end{pmatrix}, \tag{8.5.9}$$

where

$$(d_{k-1}, d_k) = (c_{k-1}, c_k) - \sum_{j=k+1}^{n} (r_{k-1,j}y_j, r_{kj}y_j). \tag{8.5.10}$$

Step 3. *Recover X: $X = VYU^{\mathrm{T}}$.*

Example 8.5.4. Consider solving $XA + BX = C$ using Algorithm 8.5.1 with

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9 & 1 \\ 10 & 0 & 0 & 0 \end{pmatrix}$$

and

$$C = \begin{pmatrix} 12 & 10 & 12 \\ 24 & 22 & 24 \\ 27 & 25 & 27 \\ 12 & 10 & 12 \end{pmatrix}.$$

Step 1. Reduce A^{T} to RSF and B to Hessenberg form H:

$$U^{\mathrm{T}}A^{\mathrm{T}}U = R = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

$$V^{\mathrm{T}}BV = H = \begin{pmatrix} 1 & -5.3766 & -0.3709 & -0.0886 \\ -12.8452 & 7.6545 & 5.3962 & -0.7695 \\ 0 & 10.6689 & 4.7871 & -0.2737 \\ 0 & 0 & -5.3340 & 1.5584 \end{pmatrix}.$$

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -3.114 & -0.7398 & -0.5965 \\ 0 & -5.449 & -0.3752 & 0.7498 \\ 0 & -0.7785 & 0.5585 & -0.2863 \end{pmatrix},$$

Compute

$$\hat{C} = \begin{pmatrix} 12 & 10 & 12 \\ -31.5292 & -28.2595 & -31.5292 \\ -21.1822 & -20.0693 & -21.1822 \\ 2.4949 & 2.7608 & 2.4949 \end{pmatrix}.$$

Step 2. Solution of the reduced problem: $HY + YR^{\mathrm{T}} = \hat{C}$.
Case 1. Since $r(3,2)$ is 0, y_3 is obtained by solving: $(H + r_{33}I)y_3 = c_3$.

$$y_3 = (1, -1.6348, -0.5564, -0.1329)^{\mathrm{T}}.$$

Case 2. Since $r_{21} \neq 0$, y_1 and y_2 are simultaneously computed by solving the system:

$$\begin{pmatrix} H + r_{11}I & r_{12}I \\ r_{21}I & H + r_{22}I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix},$$

where $(d_1, d_2) = (c_1 - r_{13}y_3, c_2 - r_{23}y_3)$.

$$y_1 = \begin{pmatrix} 1 \\ -1.6348 \\ -0.5564 \\ -0.1329 \end{pmatrix}, \qquad y_2 = \begin{pmatrix} 1 \\ -1.6348 \\ -0.5564 \\ -0.1329 \end{pmatrix}.$$

So,

$$Y = (y_1, y_2, y_3) = \begin{pmatrix} 1 & 1 & 1 \\ -1.6348 & -1.6348 & -1.6348 \\ -0.5564 & -0.5564 & -0.5564 \\ -0.1329 & -0.1329 & -0.1329 \end{pmatrix}.$$

Step 3. $X = VYU^{\mathrm{T}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$

Flop-count:

1. Reduction to Hessenberg and RSFs: $\frac{10}{3}m^3 + 26n^3$.
2. Computation of $\hat{C}$: $2m^2n + 2mn^2$.
3. Computation of Y: $6m^2n + mn^2$.
   (To obtain Y, it was assumed that S has $n/2(2 \times 2)$ bumps, which is the worst case.)
4. Computation of X: $2m^2n + 2mn^2$.
   Total flops: **Approximately** $(10m^3/3 + 26n^3 + 10m^2n + 5mn^2)$.

Numerical Stability of the Schur and Hessenberg–Schur Methods: The round-off error analysis of the Hessenberg–Schur algorithm for the Sylvester equation $XA + BX = C$ performed by Golub *et al.* (1979) shows that "the errors no worse in magnitude than $O(\|\phi^{-1}\|\epsilon)$ will contaminate the computed $\hat{X}$, where $\|\phi^{-1}\| = 1/\text{sep}(B, -A)$, and ϵ is a small multiple of the machine precision μ."

Specifically, if

$$\frac{\epsilon(2+\epsilon)(\|A\|_2 + \|B\|_2)}{\text{sep}(B, -A)} < \frac{1}{2}.$$

Then,

$$\frac{\|X - \hat{X}\|_{\text{F}}}{\|X\|_{\text{F}}} \leq \frac{(9\epsilon + 2\epsilon^2)(\|A\|_{\text{F}} + \|B\|_{\text{F}})}{\text{sep}(B, -A)}. \tag{8.5.11}$$

The above result shows that the quantity sep$(B, -A)$ will indeed influence the numerical accuracy of the computed solution obtained by the Hessenberg–Schur algorithm for the Sylvester equation. (Note that sep$(B, -A)$ also appears in the perturbation bound (8.3.6).)

Similar remarks, of course, also hold for the Schur methods for the Lyapunov and Sylvester equations. We will have some more to say about the **backward error** of the computed solutions by these methods a little later in this chapter.

MATCONTROL notes: Algorithm 8.5.1 has been implemented in MATCONTROL function **sylvhrsc.** The function **sylvhcsc** solves the Sylvester equation using **Hessenberg decomposition** of B and **complex-Schur decomposition** of A.

MATLAB note: MATLAB function **lyap** in the form:

$$X = \textbf{lyap}\,(A, B, C)$$

solves the Sylvester equation

$$AX + XB = -C$$

using **complex-Schur decompositions** of both A and B.

8.5.4 The Schur Method for the Discrete Lyapunov Equation

We now briefly outline the Schur method for the discrete Lyapunov equation:

$$A^{\mathrm{T}}XA - X = C. \tag{8.5.12}$$

The method is due to Barraud (1977).

As before, we divide the process into three steps:

Step 1. Reduction of the problem. Let $R = U^{\mathrm{T}}A^{\mathrm{T}}U$ be the upper RSF of the matrix A^{T}. Then the equation:

$$A^{\mathrm{T}}XA - X = C$$

reduces to

$$RYR^{\mathrm{T}} - Y = \hat{C}, \tag{8.5.13}$$

where $Y = U^{\mathrm{T}}XU$ and $\hat{C} = U^{\mathrm{T}}CU$.

Step 2. Solution of the reduced equation. Let $R = (r_{ij})$, $Y = (y_1, y_2, \ldots, y_n)$, and $\hat{C} = (c_1, c_2, \ldots, c_n)$.

Consider two cases as before.

Case 1. $r_{k,k-1} = 0$, for some k.
In this case, y_k can be determined by solving the quasi-triangular system:

$$(r_{kk}R - I)y_k = c_k - R\sum_{j=k+1}^{n} r_{kj}y_j. \tag{8.5.14}$$

In particular, if R is an upper triangular matrix, then y_n through y_1 can be computed successively by solving the triangular systems:

$$(r_{kk}R - I)y_k = c_k - R\sum_{j=k+1}^{n} r_{kj}y_j, \quad k = n, n-1, \ldots, 2, 1. \tag{8.5.15}$$

Case 2. $r_{k,k-1} \neq 0$, for some k. In this case y_k and y_{k-1} can be simultaneously computed, as before.

For example, if $n = 3$, and $r_{2,1} \neq 0$, then y_2 and y_1 can be computed simultaneously by solving the system:

$$\begin{pmatrix} r_{11}R - I & r_{12}R \\ r_{21}R & r_{22}R - I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} c_1 - r_{13}Ry_3, \\ c_2 - r_{23}Ry_3 \end{pmatrix}. \tag{8.5.16}$$

Step 3. Recovery of X from Y. Once Y is computed, X is recovered from Y as

$$X = UYU^{\mathrm{T}}. \tag{8.5.17}$$

Example 8.5.5. Consider solving the discrete Lyapunov equation $A^{\mathrm{T}}XA - X = C$ with

$$A = \begin{pmatrix} 0 & 2 & -1 \\ -3 & -2 & 2 \\ -2 & 1 & -1 \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} -2 & 2 & -3 \\ -8 & -6 & -5 \\ 11 & 13 & -2 \end{pmatrix}.$$

Step 1. Reduction to: $RYR^{\mathrm{T}} - Y = \hat{C}$.

$$R = \begin{pmatrix} -2.5160 & -2.7102 & -1.6565 \\ 0 & -0.2420 & 3.2825 \\ 0 & -0.8298 & -0.2420 \end{pmatrix},$$

$$U = \begin{pmatrix} -0.1972 & 0.9778 & -0.0705 \\ -0.6529 & -0.1847 & -0.7346 \\ 0.7313 & 0.0988 & -0.6749 \end{pmatrix},$$

$$\hat{C} = \begin{pmatrix} -9.4514 & 11.1896 & -12.1503 \\ -4.5736 & -0.4260 & -1.7470 \\ 7.0475 & -0.0252 & -0.1226 \end{pmatrix}.$$

Step 2. Solution of the reduced equation: $RYR^{\mathrm{T}} - Y = \hat{C}$:

$$Y = (y_1, y_2, y_3) = \begin{pmatrix} 2.2373 & -5.9557 & 2.4409 \\ 3.6415 & -0.3633 & -0.2531 \\ -5.1720 & -0.1677 & 1.5570 \end{pmatrix}.$$

Step 3. Recovery of X from Y:

$$\begin{aligned} X &= UYU^{\mathrm{T}} \\ &= \begin{pmatrix} 0.1376 & -2.1290 & 2.4409 \\ 3.6774 & 0.1419 & -1.3935 \\ -5.1721 & -0.1678 & 1.5570 \end{pmatrix}. \end{aligned}$$

Verify: $\|A^{\mathrm{T}}XA - X - C\|_2 = O(10^{-14})$.

Flop-Count: The Schur method for the discrete Lyapunov equation requires about $34n^3$ flops ($26n^3$ for the reduction of A to the RSF).

Round-off properties: As in the case of the continuous-time Lyapunov equation, it can be shown (**Exercise 8.26**) that the computed solution $\hat{X}$ of the discrete Lyapunov equation $A^{\mathrm{T}}XA - X = C$ satisfies the inequality

$$\frac{\|\hat{X} - X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} \leq \frac{cm\mu}{\mathrm{sep}_d(A^{\mathrm{T}}, A)}, \tag{8.5.18}$$

where $m = \max(1, \|A\|_{\mathrm{F}}^2)$ and c is a small constant.

Thus, the accuracy of the solution obtained by the Schur method for the discrete Lyapunov equation depends upon the quantity $\mathrm{sep}_d(A^{\mathrm{T}}, A)$. (Note again that the $\mathrm{sep}_d(A^{\mathrm{T}}, A)$ appears in the perturbation bound (8.3.18).)

MATLAB note: X $=$***dlyap***(A, C) solves the discrete Lyapunov equation: $AXA^{\mathrm{T}} - X = -C$, using **complex-Schur** decomposition of A.

MATCONTROL notes: MATCONTROL functions **lyaprsd** and **lyapcsd** solve the discrete-time Lyapunov equation using **real-Schur** and **complex-Schur** decomposition of A, respectively.

8.5.5 Residual and Backward Error in the Schur and Hessenberg–Schur Algorithms

We consider here the following questions: How small are the relative residuals obtained by the Schur and the Hessenberg–Schur algorithms? **Does a small relative residual guarantee that the solution is accurate?**

To answer these questions, we note that there are two major computational tasks with these algorithms:

First. The reduction of the matrices to the RSF and/or to the Hessenberg form.

Second. Solutions of certain linear systems.

We know that the reduction to the RSF of a matrix by the QR iteration method, and that to the Hessenberg form by Householder's or Givens' method, are backward stable (**See Chapter 4**).

And, if the linear systems are solved using Gaussian elimination with partial pivoting, followed by the technique of iterative refinement (which is the most practical way to solve a dense linear system), then it can be shown (Golub *et al.* 1979, Higham 1996) that the relative residual norm obtained by the **Hessenberg–Schur algorithm** for the Sylvester equation satisfies

$$\frac{\|C - \left(\hat{X}A + B\hat{X}\right)\|_{\mathrm{F}}}{\|\hat{X}\|_{\mathrm{F}}} \leq c\mu\left(\|A\|_{\mathrm{F}} + \|B\|_{\mathrm{F}}\right), \tag{8.5.19}$$

where $\hat{X}$ is the computed solution and c is a small constant depending upon m and n.

This means that the relative residual is guaranteed to be small. Note that this bound does not involve $\text{sep}(B, -A)$.

Does a small relative residual imply a small backward error? We will now consider this question.

To this end, let's recall that by **backward error** we mean the amount of perturbations to be made to the data so that an approximate solution is the exact solution to the perturbed problem. If the perturbations are small, then the algorithm is backward stable.

For the Sylvester equation problem, let's define (following Higham 1996) the backward error of an approximate solution Y of the Sylvester equation $XA + BX = C$ by

$$\begin{aligned}\nu(Y) = \min\{\varepsilon : {} & Y(A + \Delta A) + (B + \Delta B)Y = C + \Delta C, \\ & \|\Delta A\|_F \le \varepsilon\alpha,\ \|\Delta B\|_F \le \varepsilon\beta,\ \|\Delta C\|_F \le \varepsilon\gamma\},\end{aligned}$$

where α, β, and γ are tolerances. The most common choice is

$$\alpha = \|A\|_F, \qquad \beta = \|B\|_F, \qquad \gamma = \|C\|_F.$$

This choice yields the **normwise relative backward error.**

As earlier, we assume that A is $n \times n$ and B is $m \times m$, and $m \ge n$.

It has been shown by Higham (1996) that

$$\nu(Y) \le \delta \frac{\|\text{Res}(Y)\|_F}{(\alpha + \beta)\|Y\|_F + \gamma}, \tag{8.5.20}$$

where $\text{Res}(Y) = C - (YA + BY)$ is the residual and

$$\delta = \frac{(\alpha + \beta)\|Y\|_F + \gamma}{\sqrt{(\alpha^2\sigma_m^2 + \beta^2\sigma_n^2 + \gamma^2)}}. \tag{8.5.21}$$

Here $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_n \ge 0$ are the singular values of Y, and $\sigma_{n+1} = \cdots = \sigma_m = 0$.

The special case when $m = n$ is interesting. In this case

$$\delta = \frac{(\|A\|_F + \|B\|_F)\,\|Y\|_F + \|C\|_F}{\left((\|A\|_F^2 + \|B\|_F^2)\,\sigma_{\min}^2(Y) + \|C\|_F^2\right)^{1/2}} \tag{8.5.22}$$

Thus, δ is large only when

$$\|Y\|_F \gg \sigma_{\min}(Y) \quad \text{and} \quad \|Y\|_F \gg \frac{\|C\|_F}{\|A\|_F + \|B\|_F}. \tag{8.5.23}$$

In other words, **δ is large when Y is ill-conditioned and $\|Y\|_F$ is large.**

In the general case ($m \neq n$), δ can also be large if $\|B\|$ is large compared to the rest of the data. In these cases, the Sylvester equation is badly scaled.

Also, **note that if only A and B are perturbed, then δ is large whenever Y is ill-conditioned.**

This is because in this case,

$$\delta \geq \|Y\|_F \|Y^{\dagger}\|_2 \approx \text{Cond}_2(Y)$$

(for any m and n); so, δ is large whenever Y is ill-conditioned.

From above discussions, we see that "**the backward error of an approximate solution to the Sylvester equation can be arbitrarily larger than its relative residual**" (Higham 1996). The same remark, of course, also holds for the Lyapunov equation, as we will see below.

Backward Error for the Lyapunov Equation

In case of the Lyapunov equation, $B = A^{\mathrm{T}}$ (and thus $\beta = \alpha$), we have the following bound for the backward error for the Lyapunov equation.

Let Y be an approximate solution of the Lyapunov equation $XA + A^{\mathrm{T}}X = C$, and let $\nu(Y)$ denote the backward error. Assume that C is symmetric. Then

$$\begin{aligned}\nu(Y) = \min\{\epsilon : Y(A + \Delta A) + (A + \Delta A)^{\mathrm{T}} Y = C + \Delta C,\ \|\Delta A\| \leq \epsilon\alpha,\\ \Delta C = (\Delta C)^{\mathrm{T}},\ \|\Delta C\|_F \leq \epsilon\gamma\}.\end{aligned}$$

Thus,

$$\nu(Y) \leq \delta \frac{\|\text{Res}(Y)\|_F}{2\alpha\|Y\|_F + \gamma}. \qquad (8.5.24)$$

The expression for δ in (8.5.24) can now be easily written down by specializing (8.5.22) to this case.

8.5.6 A Hessenberg Method for the Sylvester Equation: $AX + XB = C$

Though the Schur and the Hessenberg–Schur methods are numerically effective for the Lyapunov and the Sylvester equations and are widely used in practice, it would be, however nice to have methods that would require reduction of the matrices A and B to Hessenberg forms only. Note that the reduction to a Hessenberg form is preliminary to that of the RSF. Thus, such Hessenberg methods will be more efficient than the Hessenberg–Schur method. We show below how a Hessenberg method for the Sylvester equation can be developed. The method is an extension of a Hessenberg method for the Lyapunov equation by Datta and Datta (1976), and is an efficient implementation of an idea of Kreisselmeier (1972). *It answers affirmatively a question raised by Charles Van Loan* (1982) *as to whether a method* *can be developed to solve the Lyapunov equation just by transforming A to a Hessenberg matrix.*

Step 1. *Reduction of the problem to a Hessenberg problem.* Transform A to a lower Hessenberg matrix H_1, and B to another lower Hessenberg matrix H_2:

$$U^{\mathrm{T}}AU = H_1, \qquad V^{\mathrm{T}}BV = H_2.$$

(Assume that both H_1 and H_2 are **unreduced**.)

Then, $AX + XB = C$ becomes

$$H_1 Y + Y H_2 = C', \qquad \text{where } Y = U^{\mathrm{T}}XV, \quad C' = U^{\mathrm{T}}CV.$$

Step 2. *Solution of the reduced problem.* $H_1Y + YH_2 = C'$ Let $Y = (y_1, y_2, \ldots, y_n)$ and $H_2 = (h_{ij})$.

Then the equation $H_1Y + YH_2 = C'$ is equivalent to

$$\begin{aligned} H_1 y_n + h_{n-1,n} y_{n-1} + h_{nn} y_n &= c'_n, \\ H_1 y_{n-1} + h_{n-2,n-1} y_{n-2} + h_{n-1,n-1} y_{n-1} + h_{nn-1} y_n &= c'_{n-1}, \\ &\vdots \\ H_1 y_1 + h_{11} y_1 + h_{21} y_2 + \cdots + h_{n1} y_n &= c'_1. \end{aligned}$$

Eliminating y_1 through y_{n-1}, we have,

$$R y_n = d,$$

where

$$R = \frac{1}{\prod_{i=2}^{n} h_{i-1,i}} \phi(-H_1),$$

$\phi(x)$, being the characteristic polynomial of H_1 and the vector d is defined in **Step 4** below.

Thus, once y_n is obtained by solving the system $Ry_n = d$, y_{n-1} through y_1 are computed recursively as follows:

$$y_{i-1} = -\frac{1}{h_{i-1,i}} \left(H_1 y_i + \sum_{j=i}^{n} h_{ji} y_j - c'_i \right), \quad i = n, n-1, \ldots, 2.$$

Step 3. *Computing the matrix R of Step 2.* It is well known (see Datta and Datta (1976)) that by knowing only one row or a column of a polynomial matrix in an unreduced Hessenberg matrix, the other rows or columns of the matrix polynomial can be generated recursively.

Realizing that the matrix R is basically a polynomial matrix in the lower Hessenberg matrix H_1, its computation is greatly facilitated.

Thus, if $R = (r_1, \dots, r_n)$, then, knowing r_n, r_{n-1} through r_1 can be generated recursively as follows:

$$r_{k-1} = \frac{1}{h'_{k-1,k}} \left(H_1 r_k - \sum_{i=k}^{n} h'_{ik} r_i \right),$$

where $H_1 = (h'_{ij})$; $k = n, n-1, \dots, 2$

It therefore remains to know how to compute r_n. This can be done as follows.

Set $\theta_n = e_n = (0, 0, 0, \dots, 0, 1)^{\mathrm{T}}$ and then compute θ_{n-1} through θ_0 recursively by using

$$\theta_{i-1} = -\frac{1}{h_{i-1,i}} \left(H_1 \theta_i + \sum_{j=i}^{n} h_{ji} \theta_j \right), \quad i = n, n-1, \dots, 1.$$

Then, it can be shown (Datta and Datta 1976) that

$$r_n = \theta_0, \quad \text{setting } h_{01} = 1.$$

Step 4. *Computing the vector d of Step 2.* The vector d can also be generated from the above recursion. Thus, starting with $z_n = 0$ (a zero vector), if z_{n-1} through z_0 are generated recursively using

$$z_{i-1} = -\frac{1}{h_{i-1,i}} \left(H_1 z_i + \sum_{j=i}^{n} h_{ji} z_j - c'_i \right), \quad i = n, \cdots, 2, 1,$$

then $d = -z_0$.

Step 5. *Recovery of the original solution X from Y.*

$$X = UYV^{\mathrm{T}}.$$

Remarks

- It is to be noted that the method, as presented above, is of theoretical interest only at present. There are possible numerical difficulties. For example, if one or more of the entries of the subdiagonal of the Hessenberg matrix H_2 are small, a large round-off error can be expected in computing y_{i-1} in Step 2. A detailed study on the numerical behavior of the method is necessary, before recommending it for practical use. Probably, some modification will be necessary to make it a working numerical algorithm. The reason for including this method here is to show that *a method for the Sylvester equation can be developed just by passing through the Hessenberg transformations* *of the matrices A and B only; no real Schur or Schur transformation is necessary.*

Example 8.5.6. Consider solving the Sylvester equation $AX + XB = C$ with the following data

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9 & 1 \\ 10 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad C = \begin{pmatrix} 12 & 10 & 12 \\ 24 & 22 & 24 \\ 27 & 25 & 27 \\ 12 & 10 & 12 \end{pmatrix}.$$

Step 1. Reduction of A and B to lower Hessenberg forms:

$$H_1 = \begin{pmatrix} 1.0000 & -5.3852 & 0 & 0 \\ -12.8130 & 8.7241 & 5.1151 & 0 \\ 0.8337 & 10.3127 & 4.6391 & 0.1586 \\ 0.3640 & 1.3595 & -4.8552 & 0.6368 \end{pmatrix},$$

$$U = \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & -0.3714 & -0.6009 & -0.7078 \\ 0 & -0.5571 & -0.4657 & 0.6876 \\ 0 & -0.7428 & 0.6497 & -0.1618 \end{pmatrix}.$$

$$H_2 = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$C' = \begin{pmatrix} 12.0000 & -10.0000 & 12.0000 \\ -32.8681 & 29.5256 & -32.8681 \\ -19.1978 & 18.3641 & -19.1978 \\ -0.3640 & -0.0000 & -0.3640 \end{pmatrix}.$$

Step 2. Solution of the reduced problem: Since the matrix H_2 is **reduced** ($h_{23} = 0$), instead of an algorithm breakdown, the set of equations for y_1, y_2, y_3 decouple and we obtain:

$$\begin{aligned} H_1 y_3 + h_{33} y_3 &= c_3', \\ H_1 y_2 + h_{12} y_1 + h_{22} y_2 &= c_2' - h_{32} y_3 = \hat{c}_2, \\ H_1 y_1 + h_{11} y_1 + h_{21} y_2 &= c_1' - h_{31} y_3 = \hat{c}_1. \end{aligned}$$

The vector y_3 is obtained as the solution of the first system, and once y_3 is known, $\hat{c}_2$ and $\hat{c}_3$ can be easily computed.

$$y_3 = \begin{pmatrix} 1.0000 \\ -1.6713 \\ -0.4169 \\ -0.1820 \end{pmatrix}, \quad \hat{c}_2 = \begin{pmatrix} -10.0000 \\ 29.5256 \\ 18.3641 \\ -0.0000 \end{pmatrix}, \quad \hat{c}_1 = \begin{pmatrix} 12.0000 \\ -32.8681 \\ -19.1978 \\ -0.3640 \end{pmatrix}.$$

We now proceed to compute y_2 and y_1 as follows:

Step 3. Computation of the vector d: starting from $z_2 = (0\ \ 0\ \ 0\ \ 0)^{\mathrm{T}}$,

$$\begin{aligned} z_1 &= -\frac{1}{h_{12}}(H_1 z_2 + h_{22} z_2 - \hat{c}_2) \\ &= (-10.0000\ \ 29.5256\ \ 18.3641\ \ -0.0000)^{\mathrm{T}}, \end{aligned}$$

$$\begin{aligned} d &= -z_0 = \frac{1}{1}(H_1 z_1 + h_{11} z_1 + h_{21} z_2 - \hat{c}_1) \\ &= (-191.0000\ \ 542.0447\ \ 418.9050\ \ -52.2985)^{\mathrm{T}}. \end{aligned}$$

Step 4. Computation of the matrix R. Starting from $\theta_2 = (0\ \ 0\ \ 0\ \ 1)^{\mathrm{T}}$,

$$\begin{aligned} \theta_1 &= -\frac{1}{h_{12}}(H_1\theta_2 + h_{22}\theta_2) = (0\ \ 0\ \ -0.1586\ \ -1.6368)^{\mathrm{T}}, \\ \theta_0 &= -\frac{1}{1}(H_1\theta_1 + h_{11}\theta_1 + h_{21}\theta_2) = (0\ \ 0.8112\ \ 1.1538\ \ 2.9092)^{\mathrm{T}} \end{aligned}$$

and now, starting from $r_4 = \theta_0$, we obtain

$$\begin{aligned} r_3 &= \frac{1}{h'_{34}}(H_1 r_4 - h'_{44} r_4) \\ &= (-27.5462\ \ 78.5858\ \ 84.7805\ \ \ -28.3717)^{\mathrm{T}}, \end{aligned}$$

$$\begin{aligned} r_2 &= \frac{1}{h'_{23}}(H_1 r_3 - h'_{33} r_3 - h'_{43} r_4) \\ &= (-63.1364\ \ 217.3103\ \ 154.1618 - 36.5856)^{\mathrm{T}}, \end{aligned}$$

$$\begin{aligned} r_1 &= \frac{1}{h'_{12}}(H_1 r_2 - h'_{22} r_2 - h'_{32} r_3 - h'_{42} r_4) \\ &= (74.0000\ \ -145.9565\ \ -125.7098\ \ -20.1428)^{\mathrm{T}}, \end{aligned}$$

which gives

$$R = \begin{pmatrix} 74.0000 & -63.1364 & -27.5462 & 0 \\ -145.9565 & 217.3103 & 78.5858 & 0.8112 \\ -125.7098 & 154.1618 & 84.7805 & 1.1538 \\ -20.1428 & -36.5856 & -28.3717 & 2.9092 \end{pmatrix}$$

and now $Ry_2 = d$ gives

$$y_2 = (-1.0000\ \ 1.6713\ \ 0.4169\ \ 0.1820)^{\mathrm{T}}$$

and finally we compute

$$y_1 = (1.0000\ \ -1.6713\ \ -0.4169\ \ -0.1820)^{\mathrm{T}}.$$

Therefore, the solution of the reduced problem is

$$Y = \begin{pmatrix} 1.0000 & -1.0000 & 1.0000 \\ -1.6713 & 1.6713 & -1.6713 \\ -0.4169 & 0.4169 & -0.4169 \\ -0.1820 & 0.1820 & -0.1820 \end{pmatrix}.$$

The original solution X is then recovered via $X = UYV^{\mathrm{T}}$:

$$X = \begin{pmatrix} 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 \end{pmatrix}.$$

Verification: $\|AX + XB - C\|_2 = 5.6169 \times 10^{-14}$.

MATCONTROL note: The Hessenberg methods for the Sylvester and Lyapunov equations have been implemented in MATCONTROL functions **sylvhess** and **lyaphess**, respectively. Both Hessenberg matrices are assumed to be unreduced. The above example shows that the method, however, works if one of them is reduced, but in that case the codes need to be modified.

8.5.7 The Hessenberg–Schur Method for the Discrete Sylvester Equation

In some applications, one needs to solve a general discrete Sylvester equation:

$$BXA + C = X.$$

The Schur method for the discrete Lyapunov equation described in **Section 8.5.4** can be easily extended to solve this equation.

Assume that **the order of A is smaller than that of B**. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$. Let the matrices A^{T} and B be transformed, respectively, to an upper real Schur matrix R and an upper Hessenberg matrix H by orthogonal similarity:

$$U^{\mathrm{T}}A^{\mathrm{T}}U = R,$$
$$V^{\mathrm{T}}BV = H.$$

Then,

$$BXA + C = X \quad \text{becomes} \quad HYR^{\mathrm{T}} + \hat{C} = Y,$$

where $Y = V^{\mathrm{T}}XU, \hat{C} = V^{\mathrm{T}}CU$. Let $Y = (y_1, \ldots, y_n)$, and $\hat{C} = (c_1, c_2, \ldots, c_n)$.

The reduced equation can now be solved in exactly the same way as in the Hessenberg–Schur algorithm for the Sylvester equation (**Algorithm 8.5.1**). This is left as an exercise (**Exercise 8.27**) for the readers.

MATCONTROL note: MATCONTROL function **sylvhcsd** solves the discrete-time Sylvester equation, based on **complex Schur** decomposition of A.

8.6 DIRECT COMPUTATIONS OF THE CHOLESKY FACTORS OF SYMMETRIC POSITIVE DEFINITE SOLUTIONS OF LYAPUNOV EQUATIONS

In this section we describe methods for finding the Cholesky factors of the symmetric positive definite solutions of both continuous-time and discrete-time Lyapunov equations, without explicitly computing such solutions.

8.6.1 Computing the Cholesky Factor of the Positive Definite Solution of the Lyapunov Equation

Consider first the Lyapunov equation:

$$XA + A^{\mathrm{T}}X = -C^{\mathrm{T}}C, \tag{8.6.1}$$

where A is an $n \times n$ stable matrix (i.e., all the eigenvalues $\lambda_1, \ldots, \lambda_n$ have negative real parts), and C is an $r \times n$ matrix.

The above equation admits a unique symmetric positive semidefinite solution X. Thus, such a solution matrix X has the Cholesky factorization $X = Y^{\mathrm{T}}Y$, where Y is upper triangular.

In several applications, all that is needed is the matrix Y; X is not needed as such. One such application is **model reduction problem via internal balancing and the Schur method for model reduction (Chapter 14)**, where the Cholesky factors of the controllability and observability Grammians are needed.

In these applications, it might be computationally more attractive to obtain the matrix Y directly without solving the equation for X, because X can be considerably more ill-conditioned than Y. Note that $\mathrm{Cond}_2(X) = (\mathrm{Cond}_2(Y))^2$. Also, it may not be computationally desirable to form the right-hand side matrix $-C^{\mathrm{T}}C$ explicitly; there may be a significant loss of accuracy in this explicit formation.

We describe below a procedure due to Hammarling (1982) for finding the Cholesky factor Y without explicitly computing X and without forming the matrix product $C^{\mathrm{T}}C$.

Reduction of the Problem

Substituting $X = Y^{\mathrm{T}}Y$ in Eq. (8.6.1), we have

$$(Y^{\mathrm{T}}Y)A + A^{\mathrm{T}}(Y^{\mathrm{T}}Y) = -C^{\mathrm{T}}C. \tag{8.6.2}$$

The challenge is now to compute Y without explicitly forming the product $C^{\mathrm{T}}C$.

Let $S = U^{\mathrm{T}}AU$, where S is in upper RSF and U is orthogonal. Let

$$CU = QR$$

be the QR factorization of CU.

Then Eq. (8.6.2) becomes

$$S^{\mathrm{T}}\left(\hat{Y}^{\mathrm{T}}\hat{Y}\right)+\left(\hat{Y}^{\mathrm{T}}\hat{Y}\right)S=-R^{\mathrm{T}}R, \tag{8.6.3}$$

where $\hat{Y}=YU$ and $R^{\mathrm{T}}R=(CU)^{\mathrm{T}}CU$.

Solution of the Reduced Equation

To obtain $\hat{Y}$ from (8.6.3) without explicitly forming $R^{\mathrm{T}}R$, we partition $\hat{Y}$, R, and S as follows:

$$\hat{Y}=\begin{pmatrix}\hat{y}_{11} & \hat{y}^{\mathrm{T}}\\ 0 & Y_1\end{pmatrix},\qquad R=\begin{pmatrix}r_{11} & r^{\mathrm{T}}\\ 0 & R_1\end{pmatrix},\qquad S=\begin{pmatrix}s_{11} & s^{\mathrm{T}}\\ 0 & S_1\end{pmatrix}, \tag{8.6.4}$$

where s_{11} is a scalar (a real eigenvalue in RSF S) or a 2×2 matrix ("Schur bump," corresponding to a pair of complex conjugate eigenvalues in the matrix S); and $\hat{y}$, r, and s are either column vectors or matrices with two columns.

Since $\hat{Y}$ satisfies (8.6.3) we can show, after some algebraic manipulations, that $\hat{y}_{11}$, $\hat{y}$, and Y_1 satisfy the following equations:

$$s_{11}^{\mathrm{T}}\left(\hat{y}_{11}^{\mathrm{T}}\hat{y}_{11}\right)+\left(\hat{y}_{11}^{\mathrm{T}}\hat{y}_{11}\right)s_{11}=-r_{11}^{\mathrm{T}}r_{11}, \tag{8.6.5}$$

$$S_1^{\mathrm{T}}\hat{y}+\hat{y}\left(\hat{y}_{11}s_{11}\hat{y}_{11}^{-1}\right)=-r\alpha-s\hat{y}_{11}^{\mathrm{T}}, \tag{8.6.6}$$

$$S_1^{\mathrm{T}}\left(Y_1^{\mathrm{T}}Y_1\right)+\left(Y_1^{\mathrm{T}}Y_1\right)S_1=-\hat{R}_1{}^{\mathrm{T}}\hat{R}_1, \tag{8.6.7}$$

where $\alpha=r_{11}\hat{y}_{11}^{-1}$, $\hat{R}_1^{\mathrm{T}}\hat{R}_1=R_1^{\mathrm{T}}R_1+uu^{\mathrm{T}}$, and $u=r-\hat{y}\alpha^{\mathrm{T}}$.

Since $R_1^{\mathrm{T}}R_1$ is positive definite, so is $\hat{R}_1^{\mathrm{T}}\hat{R}_1$.

Note that the matrix $\hat{R}_1$ can be easily computed, once R_1 and u are known, from the QR factorization:

$$\begin{pmatrix}u^{\mathrm{T}}\\ R_1\end{pmatrix}=\hat{Q}\hat{R}_1. \tag{8.6.8}$$

Equation (8.6.7) is of the same form as the original reduced equation (8.6.2), **but is of smaller order. This is the key observation.**

The matrices S_1, Y_1, and $\hat{R}_1$ can now be partitioned further as in (8.6.4), and the whole process can be repeated. The process is continued until $\hat{Y}$ is completely determined.

Recovery of the Solution

Once $\hat{Y}$ is obtained, the "R-matrix" $\tilde{Y}$ of the QR factorization $\tilde{Q}\tilde{Y}=\hat{Y}U^{\mathrm{T}}$ will be an upper triangular matrix that will satisfy Eq. (8.6.7). Let $\tilde{Y}=(y_{ij})$.

Since Y is required to have positive diagonal entries, we will take

$$Y = \operatorname{diag}(\operatorname{sign}(\tilde{y}_{11}), \ldots, \operatorname{sign}(\tilde{y}_{nn}))\, \tilde{Y}.$$

Algorithm 8.6.1. *Algorithm for the Direct Cholesky Factor of the Symmetric Positive Definite Solution of the Lyapunov Equation*
Inputs. *A—An $n \times n$ matrix*
C—An $r \times n$ matrix.

Output. *Y—The Cholesky factor of the symmetric positive definite solution of the Lyapunov equation: $XA + A^{\mathrm{T}}X = -C^{\mathrm{T}}C$.*

Assumption. *A is stable.*
Step 1. *Find the RSF S of A: $U^{\mathrm{T}}AU = S$.*
Step 2. *Find the QR factorization of the $r \times n$ matrix CU : $CU = QR$.*
Step 3. *Partition $R = \begin{pmatrix} r_{11} & r^{\mathrm{T}} \\ 0 & R_1 \end{pmatrix}$, $S = \begin{pmatrix} s_{11} & s^{\mathrm{T}} \\ 0 & S_1 \end{pmatrix}$.*
Step 4. *Find $\hat{Y} = \begin{pmatrix} \hat{y}_{11} & \hat{y}^{\mathrm{T}} \\ 0 & Y_1 \end{pmatrix}$ as follows:*
4.1 *Compute $\hat{y}_{11}$ from $s_{11}^{\mathrm{T}}(\hat{y}_{11}^{\mathrm{T}}\hat{y}_{11}) + (\hat{y}_{11}^{\mathrm{T}}\hat{y}_{11})s_{11} = -r_{11}^{\mathrm{T}}r_{11}$.*
4.2 *Compute $\alpha = r_{11}\hat{y}_1^{-1}$.*
4.3 *Solve for $\hat{y}$: $S_1^{\mathrm{T}}\hat{y} + \hat{y}(\hat{y}_{11}s_{11}\hat{y}_{11}^{-1}) = -r\alpha - s\hat{y}_{11}^{\mathrm{T}}$.*
4.4 *Compute $u = r - \hat{y}\alpha^{\mathrm{T}}$ and then find the QR factorization of $\begin{pmatrix} u^{\mathrm{T}} \\ R_1 \end{pmatrix}$ to find $\hat{R}_1$:*

$$\hat{Q}\hat{R}_1 = \begin{pmatrix} u^{\mathrm{T}} \\ R_1 \end{pmatrix}.$$

Step 5. *Set $S = S_1$, $R = \hat{R}_1$ and return to Step 3 and continue until $\hat{Y}$ is completely determined.*

Step 6. *Compute $\tilde{Y}$ from the QR factorization of $\hat{Y}U^{\mathrm{T}}$:$\hat{Y}U^{\mathrm{T}} = Q\tilde{Y}$. Let $\tilde{Y} = (\tilde{y}_{ij})$.*

Step 7.*Compute* $Y = \begin{pmatrix} \operatorname{sign}(\tilde{y}_{11}) & & 0 \\ & \ddots & \\ 0 & & \operatorname{sign}(\tilde{y}_{nn}) \end{pmatrix} \tilde{Y}.$

Example 8.6.1. Consider solving Eq. (8.6.1) for the Cholesky factor Y with

$$A = \begin{pmatrix} -0.9501 & 0.5996 & 0.2917 \\ 0.6964 & -1.0899 & -0.6864 \\ 0 & 0.0571 & -6.6228 \end{pmatrix}, \qquad C = (1, 1, 1).$$

Step 1. Reduction of A to RSF: $[U, S] =$ **schur** (A) gives

$$U = \begin{pmatrix} -0.7211 & -0.6929 & 0.0013 \\ -0.6928 & 0.7210 & -0.0105 \\ -0.0063 & 0.0084 & 0.9999 \end{pmatrix},$$

$$S = \begin{pmatrix} -0.3714 & 0.0947 & 0.3040 \\ 0 & -1.6762 & -0.7388 \\ 0 & 0 & -6.6152 \end{pmatrix}.$$

Step 2. The QR factorization of CU:$[Q, R] =$ **qr** (CU) gives

$$R = (-1.4202 \quad 0.0366 \quad 0.9908).$$

Step 3.

$$r_{11} = -1.4202, \qquad s_{11} = -0.3714,$$

$$r = \begin{pmatrix} 0.0366 \\ 0.9908 \end{pmatrix}, \qquad S_1 = \begin{pmatrix} -1.6762 & -0.7388 \\ 0 & -6.6152 \end{pmatrix}, \qquad s = \begin{pmatrix} 0.0947 \\ 0.3040 \end{pmatrix}.$$

Step 4. Compute $\hat{y}_{11}$ and α:

$$\hat{y}_{11} = 1.6479, \qquad \alpha = r_{11}\hat{y}_{11}^{-1} = -0.8619.$$

Solve for $\hat{y}$:

$$(S_1^{\mathrm{T}} + \hat{y}_{11}s_{11}\hat{y}_{11}^{-1}I)\hat{y} = -r\alpha - s\hat{y}_{11}^{\mathrm{T}}$$

or

$$\begin{pmatrix} -2.0476 & 0 \\ -0.7388 & -6.9866 \end{pmatrix} \hat{y} = \begin{pmatrix} -0.1245 \\ 0.3530 \end{pmatrix},$$

$$\hat{y} = \begin{pmatrix} 0.0608 \\ -0.0569 \end{pmatrix}.$$

$$u = r - \hat{y}\alpha^{\mathrm{T}} = \begin{pmatrix} 0.0890 \\ 0.9418 \end{pmatrix}, \qquad R_1 = 0$$

$$\hat{R}_1 = (0.0890, 0.9418).$$

Step 5. Solution of the reduced 2 × 2 problem:

$$S \equiv S_1 = \begin{pmatrix} -1.6762 & -0.7388 \\ 0 & -6.6152 \end{pmatrix}$$

$$R \equiv \hat{R}_1 = (0.0890,\ 0.9418)$$

$$\hat{y}_{11} = 0.0486, \qquad \hat{y} = (0.2036), \qquad \hat{R}_1 = 0.5689.$$

Solution of the final 1 × 1 problem:

$$S = -6.6152, \qquad R = 0.5689, \qquad \hat{y}_{11} = 0.1564.$$

Thus,

$$\hat{Y} = \begin{pmatrix} 1.6479 & 0.0608 & -0.0569 \\ 0 & 0.0486 & 0.02036 \\ 0 & 0 & 0.1564 \end{pmatrix}.$$

Step 6. Compute $\tilde{Y}$:$[Q_1, \tilde{Y}] = \mathbf{qr}\ (\hat{Y}U^T)$ (Using QR factorization):

$$\tilde{Y}_1 = \begin{pmatrix} 1.2309 & 1.0960 & 0.0613 \\ 0 & -0.0627 & -0.2011 \\ 0 & 0 & 0.1623 \end{pmatrix}.$$

Step 7. $Y = \begin{pmatrix} 1.2309 & 1.0960 & 0.0613 \\ 0 & 0.0627 & 0.2011 \\ 0 & 0 & 0.1623 \end{pmatrix}.$

MATCONTROL note: Algorithm 8.6.1 has been implemented in MATCONTROL functions **lyapchlc**.

Remark

- Note that it is possible to arrange the computation of Y with a different form of partitioning than as shown in (8.6.4). For example, let us partition matrices $\hat{Y}$, R, and S as follows:

$$\hat{Y} = \begin{pmatrix} Y_{11} & y \\ 0 & y_1 \end{pmatrix}, \qquad R = \begin{pmatrix} R_{11} & r \\ 0 & r_1 \end{pmatrix}, \qquad S = \begin{pmatrix} S_{11} & s \\ 0 & s_1 \end{pmatrix}, \quad (8.6.9)$$

  where y_1, r_1, and s_1 are scalars or 2 × 2 matrices and y, r, and s are either column vectors or matrices with two columns.

Then, similar to Eqs. (8.6.5)–(8.6.7), one will obtain three equations. For example, the first one will be just the deflated version of the original equation.

$$S_{11}^{\mathrm{T}} \left(Y_{11}^{\mathrm{T}} Y_{11}\right) + \left(Y_{11}^{\mathrm{T}} Y_{11}\right) S_{11} = -R_{11}^{\mathrm{T}} R_{11}. \tag{8.6.10}$$

Suppose that the solution Y_{11} of this deflated equation has been computed, then the second and third equations will give us the expressions for y and y_1.

By using this new partitioning, the original algorithm of Hammarling (1982) can be slightly improved.

In the following, we will use this partitioning to solve the discrete equation.

8.6.2 Computing the Cholesky Factor of the Positive Definite Solution of the Discrete Lyapunov Equation

Consider now the discrete Lyapunov equation:

$$A^{\mathrm{T}} X A + C^{\mathrm{T}} C = X, \tag{8.6.11}$$

where A is an $n \times n$ discrete–stable matrix (i.e., all the eigenvalues $\lambda_1, \ldots, \lambda_n$ are inside the unit circle) and C is an $r \times n$ matrix.

Then Eq. (8.6.11) admits a unique symmetric positive semidefinite solution X. Such a solution matrix X has the Cholesky factorization: $X = Y^{\mathrm{T}} Y$, where Y is upper triangular.

We would obtain the matrix Y directly without solving the equation (8.6.11) for X. Substituting $X = Y^{\mathrm{T}} Y$ into the Eq. (8.6.11), we have

$$A^{\mathrm{T}} (Y^{\mathrm{T}} Y) A + C^{\mathrm{T}} C = Y^{\mathrm{T}} Y. \tag{8.6.12}$$

As in the case of the continuous-time Lyapunov equation (8.6.1), we now outline a method for finding Y of (8.6.12) without computing X and without forming the matrix $C^{\mathrm{T}} C$.

Reduction of the Problem

Let $S = U^{\mathrm{T}} A U$, where S is in upper RSF and U is an orthogonal matrix. Let

$$Q_1 R = CU$$

be the economy QR factorization of the matrix CU.

Then Eq. (8.6.12) becomes

$$S^{\mathrm{T}} \left(\hat{Y}^{\mathrm{T}} \hat{Y}\right) S + R^{\mathrm{T}} R = \hat{Y}^{\mathrm{T}} \hat{Y}, \tag{8.6.13}$$

where $\hat{Y} = YU$ and $R^{\mathrm{T}} R = (CU)^{\mathrm{T}} CU$.

Solution of the Reduced Equation

To obtain $\hat{Y}$ from (8.6.13) without forming $R^{\mathrm{T}}R$ explicitly, we partition $\hat{Y}$, R, and S as

$$\hat{Y} = \begin{pmatrix} Y_{11} & y \\ 0 & y_1 \end{pmatrix}, \qquad R = \begin{pmatrix} R_{11} & r \\ 0 & r_1 \end{pmatrix}, \qquad S = \begin{pmatrix} S_{11} & s \\ 0 & s_1 \end{pmatrix}.$$

From (8.6.13), we see that Y_{11}, y, and y_1 satisfy the following equations:

$$S_{11}^{\mathrm{T}}\left(Y_{11}^{\mathrm{T}}Y_{11}\right)S_{11} + R_{11}^{\mathrm{T}}R_{11} = \left(Y_{11}^{\mathrm{T}}Y_{11}\right), \tag{8.6.14}$$

$$Y_{11}^{\mathrm{T}}y - (Y_{11}S_{11})^{\mathrm{T}}\,ys_1 = S_{11}^{\mathrm{T}}Y_{11}^{\mathrm{T}}Y_{11}s + R_{11}^{\mathrm{T}}r, \tag{8.6.15}$$

$$s_1^{\mathrm{T}}y_1^{\mathrm{T}}y_1s_1 + \left(r_1^{\mathrm{T}}r_1 + r^{\mathrm{T}}r + (Y_{11}s + ys_1)^{\mathrm{T}}(Y_{11}s + ys_1) - y^{\mathrm{T}}y\right) = y_1^{\mathrm{T}}y_1. \tag{8.6.16}$$

Equation (8.6.14) is of the same form as the original reduced equation (8.6.12), **but is of smaller order**.

Suppose that we have already computed the solution Y_{11} of this equation. Then y can be obtained from (8.6.15) by solving a linear system and, finally, (8.6.16) gives us y_1.

Recovery of the Solution

Once $\hat{Y}$ is obtained, the "R-matrix" $\tilde{Y}$ of the QR factorization of the matrix $\hat{Y}U^{\mathrm{T}}$: $\tilde{Q}\tilde{Y} = \hat{Y}U^{\mathrm{T}}$ will be the upper triangular matrix that will solve the equation (8.6.12). Let $\tilde{Y} = (\tilde{y}_{ij})$.

Since Y has to have positive diagonal entries, we take

$$Y = \mathrm{diag}(\mathrm{sign}(\tilde{y}_{11}), \ldots, \mathrm{sign}(\tilde{y}_{nn}))\,\tilde{Y}.$$

Algorithm 8.6.2. *Algorithm for the Direct Cholesky Factor of the Symmetric Positive Definite Solution of the Discrete Lyapunov Equation*

Inputs. *A—An $n \times n$ matrix*

C—An $r \times n$ matrix.

Output. *Y—The Cholesky factor Y of the symmetric positive definite solution X of the discrete Lyapunov Equation: $A^{\mathrm{T}}XA + C^{\mathrm{T}}C = X$.*

Assumption. *A is discrete-stable, that is all its eigenvalues have moduli less than 1.*

Step 1. *Find the RSF S of A:$U^{\mathrm{T}}AU = S$.*

Step 2. *Find the (economy size) QR factorization of the $r \times n$ matrix CU: $QR = CU$.*

Step 3. *Partition*

$$S = \begin{pmatrix} S_{11} & * \\ 0 & * \end{pmatrix}, \quad R = \begin{pmatrix} R_{11} & * \\ 0 & * \end{pmatrix}, \quad Y = \begin{pmatrix} Y_{11} & * \\ 0 & * \end{pmatrix},$$

where S_{11} *is a scalar or* 2×2 *matrix (Schur bump).*

Compute Y_{11} *from* $S_{11}^{\mathrm{T}}(Y_{11}^{\mathrm{T}}Y_{11})S_{11} + R_{11}^{\mathrm{T}}R_{11} = Y_{11}^{\mathrm{T}}Y_{11}$.

Step 4. *Do while dimension of* Y_{11} $<$ *dimension of* S

4.1. *Partition*

$$Y = \begin{pmatrix} Y_{11} & y & * \\ 0 & y_1 & * \\ 0 & 0 & * \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s & * \\ 0 & s_1 & * \\ 0 & 0 & * \end{pmatrix}, \quad R = \begin{pmatrix} R_{11} & r & * \\ 0 & r_1 & * \\ 0 & 0 & * \end{pmatrix},$$

where s_1 *is* 1×1 *scalar or* 2×2 *Schur bump.*

4.2. *Compute* y *from* $Y_{11}^{\mathrm{T}}y - (Y_{11}S_{11})^{\mathrm{T}}ys_1 = S_{11}^{\mathrm{T}}Y_{11}^{\mathrm{T}}Y_{11}s + R_{11}^{\mathrm{T}}r$.

4.3. *Compute* y_1 *from*

$$s_1^{\mathrm{T}}y_1^{\mathrm{T}}y_1s_1 + (r_1^{\mathrm{T}}r_1 + r^{\mathrm{T}}r + (Y_{11}s + ys_1)^{\mathrm{T}}(Y_{11}s + ys_1) - y^{\mathrm{T}}y) = y_1^{\mathrm{T}}y_1.$$

4.4. *Go to Step* 4 *with* $Y_{11} \equiv \begin{pmatrix} Y_{11} & y \\ 0 & y_1 \end{pmatrix}$.

Step 5. *Compute* $\tilde{Y}$ *from the* QR *factorization of* $Y_{11}U^{\mathrm{T}}$: $Q\tilde{Y} = Y_{11}U^{\mathrm{T}}$. *Let* $\tilde{Y} = (\tilde{y}_{ij})$.

Step 6. *Compute* $Y = \begin{pmatrix} \mathrm{sign}(\tilde{y}_{11}) & & 0 \\ & \ddots & \\ 0 & & \mathrm{sign}(\tilde{y}_{nn}) \end{pmatrix} \tilde{Y}$.

Example 8.6.2. Consider solving the equation (8.6.12) for the Cholesky factor Y with

$$A = \begin{pmatrix} -0.1973 & -0.0382 & 0.0675 \\ -0.1790 & -0.3042 & -0.0544 \\ 0.0794 & 0.0890 & -0.1488 \end{pmatrix}$$

and

$$C = \begin{pmatrix} 0.0651 & 0.1499 & 0.2917 \\ 0.1917 & 0.0132 & 0.4051 \end{pmatrix}.$$

Step 1. Reduction of A to the RSF: $[U, S] =$ **schur** (A) gives

$$U = \begin{pmatrix} -0.3864 & -0.7790 & 0.4938 \\ -0.7877 & 0.5572 & 0.2627 \\ 0.4798 & 0.2875 & 0.8290 \end{pmatrix},$$

$$S = \begin{pmatrix} -0.3589 & -0.0490 & 0.1589 \\ 0 & -0.1595 & -0.0963 \\ 0 & 0.0173 & -0.1319 \end{pmatrix}.$$

Step 2. The **economy size** QR factorization of CU: $[Q, R] = \mathbf{qr}\ (CU, 0)$ gives

$$R = \begin{pmatrix} 0.1100 & -0.0289 & 0.4245 \\ 0 & 0.1159 & 0.3260 \end{pmatrix}.$$

Step 3. Partitioning of R and S gives $S_{11} = (-0.3589)$, $R_{11} = (0.1100)$, which enables us to compute $Y_{11} = (0.1178)$.

Step 4. Dimension of $Y_{11} = 1 <$ dimension of $S = 3$. So we do:

4.1.

$$s = \begin{pmatrix} -0.0490 \\ 0.1589 \end{pmatrix}^{\mathrm{T}}, \quad s_1 = \begin{pmatrix} -0.1595 & -0.0963 \\ 0.0173 & -0.1319 \end{pmatrix},$$

$$r = \begin{pmatrix} -0.0289 \\ 0.4245 \end{pmatrix}^{\mathrm{T}}, \quad r_1 = (0.1159\ \ 0.3260).$$

4.2. $y = \begin{pmatrix} -0.0291 \\ 0.4078 \end{pmatrix}^{\mathrm{T}}.$

4.3. Solve for upper triangular y_1 with positive diagonal:

$$y_1 = \begin{pmatrix} 0.1167 & 0.3242 \\ 0 & 0.1392 \end{pmatrix}.$$

4.4. Form Y_{11} :

$$Y_{11} = \begin{pmatrix} 0.1178 & -0.0291 & 0.4078 \\ 0 & 0.1167 & 0.3242 \\ 0 & 0 & 0.1392 \end{pmatrix}$$

and the loop in **Step 4** ends.

Step 5. Find the QR factorization of $Y_{11}U^{\mathrm{T}}$: $[Q_1, \tilde{Y}] = \mathbf{qr}(Y_{11}U^{\mathrm{T}})$ to obtain $\tilde{Y}$:

$$\tilde{Y} = \begin{pmatrix} -0.2034 & -0.0618 & -0.4807 \\ 0 & -0.1417 & -0.1355 \\ 0 & 0 & -0.0664 \end{pmatrix}.$$

Step 6. Compute the solution:

$$Y = \begin{pmatrix} 0.2034 & 0.0618 & 0.4807 \\ 0 & 0.1417 & 0.1355 \\ 0 & 0 & 0.0664 \end{pmatrix}.$$

MATCONTROL Note: Algorithm 8.6.2 has been implemented in MATCONTROL function **lyapchld**.

8.7 COMPARISONS OF DIFFERENT METHODS AND CONCLUSIONS

The analytical methods such as the ones based on evaluating the integral

$$X = \int_0^{\alpha} e^{A^{\mathrm{T}}t} C e^{At}\, dt$$

for the Lyapunov equation, evaluating the infinite sum $\sum (A^k)^{\mathrm{T}} C A^k$ for the discrete Lyapunov equation, and the finite series methods for the Sylvester and Lyapunov equations are not practical for numerical computations.

The methods, based on the reduction to Jordan and companion forms, will give inaccurate solutions when the transforming matrices are ill-conditioned. *The methods based on the reduction to Jordan and companion forms, therefore, in general should be avoided for numerical computations.*

From numerical viewpoints, the methods of choice are:

- The **Schur method (Section 8.5.2)** for the Lyapunov equation: $XA + A^{\mathrm{T}}X = C$.
- The **Hessenberg–Schur method (Algorithm 8.5.1)** for the Sylvester equation: $XA + BX = C$.
- The **Schur method (Section 8.5.4)** for the discrete Lyapunov equation: $A^{\mathrm{T}}XA - X = C$
- The **modified Schur methods (Algorithms 8.6.1 and 8.6.2)** for the Cholesky factors of the Lyapunov equation: $XA + A^{\mathrm{T}}X = -C^{\mathrm{T}}C$ and the discrete Lyapunov equation: $A^{\mathrm{T}}XA + C^{\mathrm{T}}C = X$.

8.8 SOME SELECTED SOFTWARE

8.8.1 MATLAB Control System Toolbox

Matrix equation solvers

lyap	Solve continuous Lyapunov equations
dlyap	Solve discrete Lyapunov equations.

8.8.2 MATCONTROL

CONDSYLVC	Finding the condition number of the Sylvester equation problem
LYAPCHLC	Finding the Cholesky factor of the positive definite solution of the continuous-time Lyapunov equation
LYAPCHLD	Find the Cholesky factor of the positive definite solution of the discrete-time Lyapunov equation
LYAPCSD	Solving discrete-time Lyapunov equation using complex Schur decomposition of A

LYAPFNS	Solving continuous-time Lyapunov equation via finite series method
LYAPHESS	Solving continuous-time Lyapunov equation via Hessenberg decomposition
LYAPRSC	Solving the continuous-time Lyapunov equation via real Schur decomposition
LYAPRSD	Solving discrete-time Lyapunov equation via real Schur decompostion
SEPEST	Estimating the sep function with triangular matrices
SEPKR	Computing the sep function using Kronecker product
SYLVHCSC	Solving the Sylvester equation using Hessenberg and complex Schur decompositions
SYLVHCSD	Solving the discrete-time Sylvester equation using Hessenberg and complex Schur decompositions
SYLVHESS	Solving the Sylvester equation via Hessenberg decomposition
SYLVHRSC	Solving the Sylvester equation using Hessenberg and real Schur decompositions
SYLVHUTC	Solving an upper triangular Sylvester equation.

8.8.3 CSP-ANM

Solutions of the Lyapunov and Sylvester matrix equations

- The Schur method for the Lyapunov equations is implemented as LyapunovSolve [a, b] SolveMethod → SchurDecomposition] (continuous-time case) and DiscreteLyapunovSolve [a, b, SolveMethod → SchurDecomposition] (discrete-time case).
- The Hessenberg–Schur method for the Sylvester equations is implemented as LyapunovSolve [a, b, c, SolveMethod → HessenbergSchur] (continuous-time case) and Discrete LyapunovSolve [a, b, c, SolveMethod → HessenbergSchur] (discrete-time case).
- The Cholesky factors of the controllability and observability Grammians of a stable system are computed using CholeskyFactorControllabilityGramian [system] and CholeskyFactorObservabilityGramian [system].

8.8.4 SLICOT

Lyapunov equations

SB03MD	Solution of Lyapunov equations and separation estimation
SB03OD	Solution of stable Lyapunov equations (Cholesky factor)
SB03PD	Solution of discrete Lyapunov equations and separation estimation

SB03QD	Condition and forward error for continuous Lyapunov equations
SB03RD	Solution of continuous Lyapunov equations and separation estimation
SB03SD	Condition and forward error for discrete Lyapunov equations
SB03TD	Solution of continuous Lyapunov equations, condition and forward error estimation
SB03UD	Solution of discrete Lyapunov equations, condition and forward error estimation

Sylvester equations

SB04MD	Solution of continuous Sylvester equations (Hessenberg–Schur method)
SB04ND	Solution of continuous Sylvester equations (one matrix in Schur form)
SB04OD	Solution of generalized Sylvester equations with separation estimation
SB04PD	Solution of continuous or discrete Sylvester equations (Schur method)
SB04QD	Solution of discrete Sylvester equations (Hessenberg–Schur method)
SB04RD	Solution of discrete Sylvester equations (one matrix in Schur form)

Generalized Lyapunov equations

SG03AD	Solution of generalized Lyapunov equations and separation estimation
SG03BD	Solution of stable generalized Lyapunov equations (Cholesky factor)

8.8.5 MATRIX$_X$

Purpose: Solve a discrete Lyapunov equation.

Syntax: P = DLYAP (A, Q)

Purpose: Solve a continuous Lyapunov equation.

Syntax: P = LYAP (A, Q)

8.8.6 LAPACK

The Schur method for the Sylvester equation, $XA + BX = C$, can be implemented in LAPACK by using the following routines in sequence: GEES to compute the Schur decomposition, GEMM to compute the transformed right-hand side, TRSY L to solve the (quasi-)triangular Sylvester equation, and GEMM to recover the solution X.

8.9 SUMMARY AND REVIEW

Applications

The applications of the Lyapunov equations include:

- Stability and robust stability analyses **(Chapter 7)**.
- Computations of the controllability and observability Grammians for stable systems (needed for internal balancing and model reduction) **(Chapter 14)**.
- Computations of the H_2 norm **(Chapter 7)**.
- Implementation of Newton's methods for Riccati equations **(Chapter 13)**.

The applications of the Sylvester equations include:

- Design of Luenberger observer **(Chapter 12)**
- Block-diagonalization of a matrix by similarity transformation.

Existence and Uniqueness Results

(1) The Sylvester equation $XA + BX = C$ has a unique solution if and only A and $-B$ do not have an eigenvalue in common (**Theorem 8.2.1**).

(2) The Lyapunov equation $XA + A^{\mathrm{T}}X = C$ has a unique solution if and only if A and $-A$ do not have an eigenvalue in common (**Corollary 8.2.1**).

(3) The discrete Lyapunov equation $A^{\mathrm{T}}XA - X = C$ has a unique solution if and only if the product of any two eigenvalues of A is not equal to 1 or A does not have an eigenvalue of modulus 1 (**Theorem 8.2.2**).

Sensitivity Results

(1) sep $(B, -A)$ defined by

$$\operatorname{sep}(B, -A) = \min_{X \neq 0} \frac{\|XA + BX\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} = \sigma_{\min}(P),$$

where $P = I_n \otimes B + A^{\mathrm{T}} \otimes I_m$, m and n are, respectively, the orders of B and A, plays an important role in the sensitivity analysis of the Sylvester equation $XA + BX = C$ (**Theorem 8.3.1**).

(2) sep $(A^{\mathrm{T}}, -A)$ has an important role in the sensitivity analysis of the Lyapunov equation: $XA + A^{\mathrm{T}}X = C$ (**Corollary 8.3.2**).

(3) $\operatorname{sep}_d(A^{\mathrm{T}}, A) = \sigma_{\min}(A^{\mathrm{T}} \otimes A^{\mathrm{T}} - I_{n^2})$ has an important role in the sensitivity analysis of the discrete Lyapunov equation $A^{\mathrm{T}}XA - X = C$ (**Theorem 8.3.4**).

(4) If A is stable, then the sensitivity of the Lyapunov equation can be determined by solving the Lyapunov equation $HA + A^{\mathrm{T}}H = -I.\|H\|_2$ is an

indicator of the sensitivity of the stable Lyapunov equation $XA + A^{\mathrm{T}}X = -C$ (**Theorem 8.3.3**).

(5) If A and B are ill-conditioned, then the Sylvester equation $XA + BX = C$ is ill-conditioned (**Theorem 8.3.6**). Thus, if A is ill-conditioned, then the Lyapunov equation is also ill-conditioned. **But the converse is not true in general.**

Sep-Estimation

The LINPACK style algorithm (**Algorithm 8.3.1**) gives an estimate of sep $(A, B)^{\mathrm{T}}$ without computing the Kronecker product sum P, which is computationally quite sensitive.

Methods for Solving the Lyapunov and Sylvester Equations

- The analytical methods such as the finite-series method or the method based on evaluation of the integral involving the matrix exponential are not practical for numerical computations **(Section 8.4)**.
- The methods based on reduction to the JCF and the companion form of a matrix should be avoided **(Section 8.5.1)**.
- The Schur methods for the Lyapunov equations **(Sections 8.5.2 and 8.5.4)** and the Hessenberg–Schur method **(Algorithms 8.5.1 and Section 8.5.7)** for the Sylvester equations are by far the best for numerical computations.
- If only the Cholesky factors of stable Lyapunov equations are needed, the modified Schur methods **(Algorithms 8.6.1 and 8.6.2)** should be used. These algorithms compute the Cholesky factors of the solutions without explicitly computing the solutions themselves. The algorithms are numerically stable.

8.10 CHAPTER NOTES AND FURTHER READING

The results on the existence and uniqueness of the Lyapunov and Sylvester equations are **classical**. For proofs of these results, see Horn and Johnson (1991), Lancaster and Rodman (1995). See also Barnett and Cameron (1985), and Barnett and Storey (1970). The sensitivity issues of these equations and the perturbation results given in Section 8.3 can be found in Golub *et al.* (1979) and in Higham (1996).

The sensitivity result of the stable Lyapunov equation is due to Hewer and Kenney (1988). The sensitivity result of the stable discrete Lyapunov equation is due to Gahinet *et al.* (1990). The perturbation result of the discrete Lyapunov equation appears in Petkov *et al.* (1991). The results relating the ill-conditioning of the Sylvester equation and eigenvalues can be found in Ghavimi and Laub (1995). The LINPACK-style sep-estimation algorithm is due to Byers (1984). See Kağström and Poromaa (1996)) for LAPACK-style algorithms. For perturbation results on generalized Sylvester equation, see Kågström (1994) and Edelman *et al.* (1997, 1999). For description of LAPACK, see Anderson *et al.* (1999). A recent book by Konstantinov *et al.* (2003) Contains many results on perturbation theory for matrix equations.

The Schur method for the Lyapunov equation is due to Bartels and Stewart (1972). The Schur method for the discrete Lyapunov equation is due to Barraud (1977). Independently of Barraud, a similar algorithm was developed by Kitagawa (1977). The Hessenberg–Schur algorithms for the Sylvester and discrete Sylvester equations are due to Golub *et al.* (1979). A good account of the algorithmic descriptions and implementational details of the methods for solving the discrete Lyapunov equations appears in the recent book of Sima (1996).

The Cholesky-factor algorithms for the stable Lyapunov equations are due to Hammarling (1982). The Hessenberg algorithm for the Sylvester equation is due to Datta and Datta (1976) and Kreisselmeier (1972). For numerical solutions of the generalized Sylvester equation $AXB^{\mathrm{T}} + CXD^{\mathrm{T}} = E$, see Gardiner *et al.* (1992a). For applications of generalized Sylvester equations of the above type including the computation of stable eigendecompositions of matrix pencils see Demmel and Kågström (1987, 1993a, 1993b), Kağström and Westin (1989), etc. See Kağström and Poromaa (1989, 1992) for block algorithms for triangular Sylvester equation (with condition estimator). See Gardiner *et al.* (1992b) for a software package for solving the generalized Sylvester equation.

Exercises

8.1 Prove that the equation $A^*XB + B^*XA = -C$ has a unique solution X if and only if $\lambda_i + \bar{\lambda}_j \neq 0$, for all i and j, where λ_i is an eigenvalue of the generalized eigenvalue problem: $Ax = \lambda Bx$. (Here $A^* = (\bar{A})^{\mathrm{T}}$ and $B^* = (\bar{B})^{\mathrm{T}}$.)

8.2 Let A be a normal matrix with $\lambda_1, \ldots, \lambda_n$ as the eigenvalues. Then show that $\max_i |\lambda_i| / \min_{ij} |\bar{\lambda}_i + \lambda_j)|$ can be regarded as the condition number of the Lyapunov equation $XA + A^*X = -C$, where $A^* = (\bar{A})^{\mathrm{T}}$. Using the result, construct an example of an ill-conditioned Lyapunov equation.

8.3 If $A = (a_{ij})$ and $B = (b_{ij})$ are upper triangular matrices of order $m \times m$ and $n \times n$ respectively, then show that $X = (x_{ij})$ satisfying the Sylvester equation $AX + XB = C$ can be found from

$$x_{ij} = \frac{c_{ij} - \sum_{k=i+1}^{m} a_{ik}x_{kj} - \sum_{k=1}^{n-1} x_{ik}b_{kj}}{a_{ii} + b_{jj}}.$$

8.4 Prove Theorems 8.3.1 and 8.3.4.

8.5 Using the perturbation results in Section 8.3, construct an example to show that the Sylvester equation problem $XA+BX=C$ can be very well-conditioned even when the eigenvector matrices for A and B are ill-conditioned.

8.6 Prove or disprove that if A and $-B$ have close eigenvalues, then the Sylvester equation $XA+BX=C$ is ill-conditioned.

8.7 Construct a 2×2 example to show that the bound (8.3.7) can be much smaller than the bound (8.3.3).

8.8 Derive the expression ϕ for the **condition number of the Lyapunov equation** given in Section 8.3.4.

8.9 Using the definition of the sep function, prove that if X is a unique solution of the Sylvester equation $XA+BX=C$, then

$$\|X\|_{\mathrm{F}} \leq \frac{\|C\|_{\mathrm{F}}}{\mathrm{sep}(B,-A)}.$$

8.10 Let

$$U^{\mathrm{T}}AU = T = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1p} \\ 0 & T_{22} & \cdots & T_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{pp} \end{pmatrix}$$

be the RSF of A, and assume that $T_{11}, \ldots, T_{pp}$ have disjoint spectra.

(a) Develop an algorithm to transform T to the block diagonal form:

$$Y^{-1}TY = \mathrm{diag}(T_{11}, \ldots, T_{pp}),$$

based on the solution of a Sylvester equation.

(b) Show that if the spectra of the diagonal blocks of T are not distinctly separated, then there will be a substantial loss of accuracy (consult Bavely and Stewart (1979)).

(c) Construct an example to support the statement in (b).

(d) Develop an algorithm to compute e^{At} based on the block diagonalization of A.

8.11 Construct a simple example to show that the Cholesky factor L of the solution matrix $X = L^{\mathrm{T}}L$ of the Lyapunov equation: $XA+A^{\mathrm{T}}X = BB^{\mathrm{T}}$, where A is a stable matrix, is less sensitive (with respect to perturbations in A) than X.

8.12 Construct your own example to show that the Lyapunov equation $XA + A^{\mathrm{T}}X = -C$ is always ill-conditioned if A is ill-conditioned with respect to inversion, but the converse is not true.

8.13 Repeat the last exercise with the Sylvester equation $XA+BX=C$, that is, construct an example to show that the Sylvester equation $XA+BX=C$ will be ill-conditioned if both A and B are ill-conditioned, but the converse is not true.

8.14 (a) Let A be a stable matrix. Show that the Lyapunov equation $XA + A^{\mathrm{T}}X = -C$ can still be ill-conditioned if A has one or more eigenvalues close to the imaginary axes.

(b) Construct an example to illustrate the result in (a).

8.15 Give an example to show that the backward error for the Sylvester equation $XA + BX = C$, where only A and B are perturbed, is large if an approximate solution Y of the equation is ill-conditioned.

8.16 Give an example to illustrate that the backward error of an approximate solution to the Sylvester equation $XA + BX = C$ can be large, even though the relative residual is quite small.

8.17 Prove that $\mathrm{sep}(A, -B) > 0$ if and only if A and $-B$ do not have common eigenvalues.

8.18 Let $K = I \otimes A^{\mathrm{T}} + A^{\mathrm{T}} \otimes I$ and $L = I \otimes S^{\mathrm{T}} + S^{\mathrm{T}} \otimes I$ be the Kronecker matrices, respectively, associated with the equations:

$$XA + A^{\mathrm{T}}X = -C$$

and

$$\hat{X}S + S^{\mathrm{T}}\hat{X} = -\hat{C},$$

where $S = U^{\mathrm{T}}AU$ is the RSF of A, and

$$\hat{C} = U^{\mathrm{T}}CU.$$

(a) Prove that $\|K^{-1}\|_2 = \|L^{-1}\|_2$

(b) Using the result in (a), find a bound for the error, when A is only perturbed, in terms of the norm of the matrix A and the norm of L^{-1}.

(c) Based on (a) and (b), develop an algorithm for estimating $\mathrm{sep}(A^{\mathrm{T}}, -A)$, analogous to the Byers' algorithm (Byers 1984) for estimating $\mathrm{sep}(A, B)$.

8.19 **Relationship of the distance to instability and sep** (A) (Van Loan 1985)
Define sep $(A) = \min\{\|AX + XA^*\|_{\mathrm{F}} \big| X \in \mathbb{C}^{n\times n}, \|X\|_{\mathrm{F}} = 1\}$
Then prove that

(a) sep $(A) = 0$, if and only if A has an eigenvalue on the imaginary axis.

(b) $\frac{1}{2}\mathrm{sep}(A) \le \beta(A) \le \sigma_{\min}(A)$, where $\beta(A)$ is the distance to instability (see Chapter 7).

(**Hint:** $\mathrm{sep}(A) = \sigma_{min}(I \otimes A + A \otimes I)$, and $\|B \otimes C\|_2 \le \|B\|_2\|C\|_2$.)

8.20 Construct an example of an ill-conditioned discrete Lyapunov equation based on Theorem 8.3.4.

8.21 Prove that if $p(x)$ is a real polynomial of degree n having no pair of roots conjugate with respect to the unit circle, and T is the lower companion matrix of $p(x)$, then the unique solution X of the discrete-time equation: $X - T^{\mathrm{T}}XT = \mathrm{diag}(1, 0, \ldots, 0)$ can be written explicitly as: $X = (I - \phi(S)^{\mathrm{T}}\phi(S))^{-1}$, where S is an unreduced lower Hessenberg matrix with 1s along the superdiagonal and zeros elsewhere, and $\phi(x) = p(x)/(x^n p(1/x))$.

Discuss the numerical difficulties of using this method for solving the discrete Lyapunov equation.

Work out an example to demonstrate the difficulties.

8.22 Develop an algorithm, analogous to Algorithm 8.6.1, to find the Cholesky factor of the symmetric positive definite solution of the Lyapunov equation $AX+XA^{\mathrm{T}} = -BB^{\mathrm{T}}$, where B is $n \times m$ and has full rank.

8.23 Compare the flop-count of the real Schur method and the complex Schur method for solving the Lyapunov equation: $XA + A^{\mathrm{T}}X = -C$.

8.24 Work out the flop-count of the Schur method for the discrete Lyapunov equation described in Section 8.5.4.

8.25 Develop a method to solve the Lyapunov equation $A^{\mathrm{T}}XA - X = -C$ based on the reduction of A to a companion form. Construct an example to show that the algorithm may not be numerically effective.

8.26 Establish the round-off error bound (8.5.18):

$$\frac{\|\hat{X} - X\|_{\mathrm{F}}}{\|X\|_{\mathrm{F}}} \leq \frac{cm\mu}{\mathrm{sep}_d(A^{\mathrm{T}}, A)}$$

for the Schur method to solve the discrete Lyapunov equation (8.5.12).

8.27 Develop a Hessenberg–Schur algorithm to solve the discrete Sylvester equation $BXA + C = X$.

8.28 Develop an algorithm to solve the Sylvester equation: $XA + BX = C$, based on the reductions of both A and B to RSFs.

Give a flop-count of this algorithm and compare this with that of Algorithm 8.5.1.

Research problems

8.1 Devise an algorithm for solving the equation:

$$A^{\mathrm{T}}XB + B^{\mathrm{T}}XA = -C$$

based on the **generalized real Schur decomposition** of the pair (A, B), described in Chapter 4.

8.2 Devise an algorithm for solving the equation:

$$AXB + LXC = D$$

using the **generalized real Schur decomposition** of the pairs (A, L) and $(C^{\mathrm{T}}, B^{\mathrm{T}})$.

8.3 Investigate if and how the norm of the solution of the discrete-stable Lyapunov equation:

$$A^{\mathrm{T}}XA - X = -I$$

provides information on the sensitivity of the discrete Lyapunov equation:

$$A^{\mathrm{T}}XA - X = C.$$

8.4 *Higham (1996).* Derive conditions for the Sylvester equation: $XA + BX = C$ to have a well-conditioned solution.

References

Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., and Sorensen D. *LAPACK Users' Guide*, 3rd edn, SIAM, Philadelphia, 1999.

Barnett S. and Cameron R.G. *Introduction to Mathematical Control Theory*, 2nd edn, Clarendon Press, Oxford, 1985.

Barnett S. and Storey C. *Matrix Methods in Stability Theory*, Nelson, London, 1970.

Barraud A.Y. "A numerical algorithm to solve $A^{\mathrm{T}}XA - X = Q$," *IEEE Trans. Autom. Control*, Vol. AC-22, pp. 883–885, 1977.

Bartels R.H. and Stewart G.W. "Algorithm 432: solution of the matrix equation $AX + XB = C$," *Comm. ACM*, Vol. 15, pp. 820–826, 1972.

Bavely C.A. and Stewart G.W. "An algorithm for computing reducing subspaces by block diagonalization," *SIAM J. Numer. Anal.*, Vol. 16, pp. 359–367, 1979.

Byers R. "A LINPACK-style condition estimator for the equation $AX - XB^{\mathrm{T}} = C$," *IEEE Trans. Autom. Control*, Vol. AC-29, pp. 926–928, 1984.

Datta B.N. and Datta K. "An algorithm to compute the powers of a Hessenberg matrix and it's applications," *Lin. Alg. Appl.* Vol. 14, pp. 273–284, 1976.

Datta B.N., *Numerical Linear Algebra and Applications*, Brooks/Cole Publishing Co., Pacific Grove, CA, 1995.

Datta K., Hong Y.P., and Lee R.B. "Applications of linear transformation to matrix equations," *Lin. Alg. Appl.*, Vol. 267, pp. 221–240, 1997.

DeSouza E. and Bhattacharyya S.P. "Controllability, observability and the solution of $AX - XB = C$," *Lin. Alg. Appl.*, Vol. 39, pp. 167–188, 1981.

Demmel J. and Bo Kågström "Computing stable eigendecompositions of matrix pencils," *Lin. Alg. Appl.*, Vol. 88/89, pp. 139–186, 1987.

Demmel J. and Kågström B. "The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications, Part I: Theory and algorithms," *ACM Trans. Math. Soft.*, Vol. 19, no. 2, pp. 160–174, 1993a.

Demmel J. and Kågström B. "The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and algorithms, Part II: Theory and algorithms," *ACM Trans. Math. Soft.*, Vol. 19, no. 2, pp. 175–201, 1993b.

Edelman A., Elmroth E., and Kågström B. "A geometric approach to perturbation theory of matrices and matrix pencils, Part I: Versal deformations," *SIAM J. Matrix Anal. Appl.*, Vol. 18, no. 3, pp. 653–692, 1997.

Edelman A., Elmroth E., and Kågström B. "A geometric approach to perturbation theory of matrices and matrix pencils, Part II: A stratification-enhanced staircase algorithm," *SIAM J. Matrix Anal. Appl.*, Vol. 20, no. 3, pp. 667–699, 1999.

Gahinet P.M., Laub A.J., Kenney C.S., and Hewer G. "Sensitivity of the stable discrete-time Lyapunov equation," *IEEE Trans. Autom. Control*, Vol. 35, pp. 1209–1217, 1990.

Gardiner J.D., Laub A.J., Amato J.J., and Moler C.B. "Solution of the Sylvester matrix equation $AXB^{\mathrm{T}} + CXD^{\mathrm{T}} = E$," *ACM Trans Math. Soft.*, Vol. 8, pp. 223–231, 1992a.

Gardiner J.D., Wette M.R., Laub A.J., Amato J.J., and Moler C.B. "Algorithm 705: A FORTRAN-77 Software package for solving the Sylvester matrix equation $AXB^{\mathrm{T}} + CXD^{\mathrm{T}} = E$," *ACM Trans. Math. Soft.*, Vol. 18, pp. 232–238, 1992b.

Ghavimi A.R. and Laub A.J. "An implicit deflation method for ill-conditioned Sylvester and Lyapunov equations," *Num. Lin. Alg. Appl.*, Vol. 2, pp. 29–49, 1995.

Golub G.H., Nash S., and Van Loan C.F. A Hessenberg–Schur method for the problem $AX + XB = C$, *IEEE Trans. Autom. Control*, Vol. AC-24, pp. 909–913, 1979.

Golub G.H. and Van Loan C.F. *Matrix Computations*, 3rd edn, Johns Hopkins University, Baltimore, MD, 1996.

Golub G.H. and Wilkinson J.H. "Ill-conditioned eigensystems and the computation of the Jordan canonical form," *SIAM Rev.*, Vol. 18, pp. 578–619, 1976.

Hammarling S.J. "Numerical solution of the stable nonnegative definite Lyapunov equation," *IMA J. Numer. Anal.*, Vol. 2, pp. 303–323, 1982.

Hearon J.Z. "Nonsingular solutions of $TA - BT = C$," *Lin. Alg. Appl.*, Vol. 16, pp. 57–63, 1977.

Hewer G. and Kenney C. "The sensitivity of the stable Lyapunov equation," *SIAM J. Contr. Optimiz.*, Vol. 26, pp. 321–344, 1988.

Higham N.J. *Accuracy and Stability of Numerical Algorithms*, SIAM Philadelphia, 1996.

Horn R.A. and Johnson C.R. *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

Jameson A. "Solution of the equation $AX + XB = C$ by the inversion of an $M \times M$ or $N \times N$ matrix," *SIAM J. Appl. Math.* Vol. 66, pp. 1020–1023, 1968.

Kağström B. "A perturbation analysis of the generalized Sylvester equation $(AR - LB, DR - LE) = (C, F)$," *SIAM J. Matrix Anal. Appl.*, Vol. 15, no. 4, pp. 1045–1060, 1994.

Kağström B. and Poromaa P., Distributed block algorithms for the triangular Sylvester equation with condition estimator, *Hypercube and Distributed Computers* (F. Andre and J.P. Verjus, Eds.), pp. 233–248, Elsevier Science Publishers, B.V. North Holland, 1989.

Kågström B. and Poromaa P. "Distributed and shared memory block algorithms for the triangular Sylvester equation with sep^{-1} estimators," *SIAM J. Matrix Anal. Appl.*, Vol. 13, no. 1, pp. 90–101, 1992.

Kågström B. and Poromaa P. "LAPACK-style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs," *ACM Trans. Math. Soft.*, Vol. 22, no. 1, pp. 78–103, 1996.

Kågström B. and Westin L. "Generalized Schur methods with condition estimators for solving the generalized Sylvester equation," *IEEE Trans. Autom. Control*, Vol. AC-34, no. 7, pp. 745–751, 1989.

Kitagawa G. "An algorithm for solving the matrix equation $X = FXF^{\mathrm{T}} + S$," *Int. J. Control*, Vol. 25, no. 5, pp. 745–753, 1977.

Konstantinov M., Gu, Da-Wei, Mehrmann Volker, Petkov Petko. Perturbation Theory for Matrix Equations, Elsevier Press, Amsterdam, 2003.

Kreisselmeier G. "A Solution of the bilinear matrix equation $AY + YB = -Q$," *SIAM J. Appl. Math.*, Vol. 23, pp. 334–338, 1972.

Lancaster P. and Rodman L., *The Algebraic Riccati Equation*, Oxford University Press, Oxford, UK, 1995.

Petkov P., Christov N.D., and Konstantinov M.M. *Computational Methods for Linear Control Systems*, Prentice Hall, London, 1991.

Sima V. *Algorithms for Linear-Quadratic Optimization*, Marcel Dekker, New York, 1996.

Starke G. and Niethammer W. "SOR for $AX - XB = C$," *Lin. Alg. Appl.*, Vol. 154–156, pp. 355–375, 1991.

Van Loan C.F. "Using the Hessenberg decomposition in control theory," in *Algorithms and Theory in Filtering and Control, Mathematical Programming Study* (Sorensen D.C. and Wets R.J., Eds.), pp. 102–111, no. 8, North Holland, Amsterdam, 1982.

Van Loan C.F. "How near is a stable matrix to an unstable matrix," *Contemporary Mathematics* (Brualdi R. *et al.*, Eds.), Vol. 47, pp. 465–477, American Mathematical Society, Providence, RI, 1985.