# Lecture 13: Solution of Lyapunov Equation for Continuous and Discrete Systems

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

**ECE 6435**
**Adv Numerical Methods in Sci Comp**

Fall 2008
November 12, 2008

# Solution of Lyapunov Equation For Continuous and Discrete Systems

- What is a Lyapunov Equation?
- Application of Lyapunov Equation
- Computational methods for solving the Lyapunov Equation
  - Direct method
  - Iterative methods
  - Semi-iterative methods

# What is Lyapunov Equation?

## What is Lyapunov Equation?

- Continuous-time Lyapunov Equation: $A^T X + XA + S = 0$
- Discrete-time Lyapunov Equation: $X = A^T XA + S$
- The equations are linear and arise in the context of stability of linear systems

## Lyapunov Equation and Stability of Linear Systems

- Continuous-time system: $\dot{\underline{x}} = A\underline{x} + B\underline{u};\quad \underline{y} = C\underline{x}$
- Discrete-time system: $\underline{x}_{i+1} = A\underline{x}_i + B\underline{u}_i\ ;\ \underline{y}_i = C\underline{x}_i$
- Original Lyapunov theorem: "$\dot{\underline{x}} = A\underline{x}$ is asymptotically stable iff

for $S > 0, \exists$ a PD solution $X$ for $A^T X + XA + S = 0$".

$$\Rightarrow X = \int_0^\infty e^{A^T\sigma} S e^{A\sigma} d\sigma$$

# Lyapunov Equation and Stability

- "For discrete-time systems, $\underline{x}_{i+1} = A\underline{x}_i$ is asymptotically stable

  iff for S > 0, ∃ a PD solution $X$ for $X = A^T XA + S$"

$$X = \sum_{i=0}^{\infty} (A^T)^i S A^i$$

- In the above cases, $v(\underline{x}) = \underline{x}^T X \underline{x}$ is a Lyapunov function.
- Lyapunov equation is used in estimating the rates at which $\|x\| \to 0$
- Lyapunov function is used to analyze Lyapunov controllers, observers, etc.
- Our interest in Lyapunov equation stems from control and filtering applications rather than stability

# Lyapunov Equation and LQR - 1

## ❑ Lyapunov Equation & Linear Quadratic Regulator (LQR) Problem

"For the continuous-time system, $\dot{\underline{x}} = A\underline{x} + B\underline{u}$;

find a linear feedback control $\underline{u}(t) = -L\underline{x}(t)$ that minimizes

$$J(\underline{u}) = \int_0^\infty [\underline{x}^T(t)Q\underline{x}(t) + \underline{u}^T(t)R\underline{u}(t)]dt; Q > 0, R > 0"$$

## ❑ What does $J(u)$ mean?

"We want $\underline{x} \to 0$ from $\underline{x}_0$ without using too much control."

## ❑ For a given $L$ computation of $\underline{J(u)}$ $\Rightarrow$ solution of Lyapunov equation

- Suppose we take any feedback control $\underline{u}(t) = -L\underline{x}(t)$, where $L$ is known
- What is the associated cost?
  - Obviously we need $\dot{\underline{x}} = (A - BL)\underline{x}$ stable, otherwise $\underline{x}^T Q \underline{x} \to \infty$
  - Let $\bar{A}_L = A - BL \Rightarrow \underline{x}(t) = e^{\bar{A}_L t}\underline{x}_0$

# Lyapunov Equation and LQR - 2

- Therefore,

$$J(u) = \underline{x}_0^T \int_0^\infty [e^{\bar{A}_L^T t} Q e^{\bar{A}_L t} + e^{\bar{A}_L^T t} L^T R L e^{\bar{A}_L t}] dt \, \underline{x}_0$$

$$= \underline{x}_0^T \int_0^\infty e^{\bar{A}_L^T t} [Q + L^T R L] e^{\bar{A}_L t} dt \, \underline{x}_0 = \underline{x}_0^T V_L \underline{x}_0$$

where $V_L$ = "cost matrix" associated with gain $L$.

- Note that $V_L$ satisfies the Lyapunov equation:

$$\bar{A}_L^T V_L + V_L \bar{A}_L + Q + L^T R L = 0$$

- **How can we pick $L^* \ni V_L^* < V_L \forall L$ ? i.e., $V_L - V_L^* \geq 0$?**

=> Solving continuous-time Riccati equation!.....Lecture 14

- **Discrete-time case: "Given a discrete-time system,**

$$\underline{x}_{i+1} = A\underline{x}_i + B\underline{u}_i \; ;$$

find a linear feedback control $\underline{u}_i = -L\underline{x}_i$ that minimizes

$$J(\underline{u}) = \sum_{i=0}^{\infty} \underline{x}_i^T Q \underline{x}_i + \underline{u}_i^T R \underline{u}_i \text{"}$$

# Lyapunov Equation and LQR - 3

- Closed-loop system matrix

$$\bar{A}_L = A - BL$$

- State in terms of closed-loop system matrix

$$\underline{x}_i = \bar{A}_L^i \underline{x}_0$$

- Cost function can be rewritten in terms of closed-loop system matrix as:

$$J(\underline{u}) = \underline{x}_0^T [\sum_{i=0}^{\infty} (\bar{A}_L^T)^i (Q + L^T RL) \bar{A}_L^i ] \underline{x}_0 = \underline{x}_0^T V_L \underline{x}_0$$

- The cost matrix $V_L$ satisfies the discrete Lyapunov equation:

$$\bar{A}_L^T V_L \bar{A}_L + Q + L^T RL = V_L$$

- Again the problem of picking $L^* \ni V_L^* = \min_L \underline{x}_0^T V_L \underline{x}_0$ involves discrete-time Riccati equation.

# Lyapunov Equation and Estimation - 1

## Lyapunov Equation and the Estimation problem

- Consider a linear continuous-time system:

  $\dot{\underline{x}} = A\underline{x} + E\underline{w}$ where $\underline{w}$ is zero mean Gaussian noise

  with covariance matrix E{w(t)w($\tau$)}=W$\delta$(t-$\tau$)

- $\Rightarrow X(t) = \text{cov}\{x(t)\} = E\{x(t)\underline{x}^T(t)\}$, where

  $$\dot{X}(t) = AX(t) + X(t)A^T + EWE^T$$

- If $A$ is stable, then in the steady state

  $$0 = AX + XA^T + EWE^T$$

- For discrete-time systems

  $\underline{x}_{i+1} = A\underline{x}_i + E\underline{w}_i$ where $\underline{w}_i$ is zero mean white Gaussian noise sequence

  with covariance matrix $E\{\underline{w}_i \underline{w}_j^T\} = W\delta_{ij}$

- $\Rightarrow$ s.s covariance of x denoted by $X$ satisfies:

  $$X = AXA^T + EWE^T$$

# Lyapunov Equation and Estimation - 2

## Estimation/Filtering in Continuous-time Systems with Continuous-time Measurements

- Consider a continuous-time linear system:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} + E\underline{w}$$

- Measurement equation

$$\underline{y}(t) = C\underline{x}(t) + \underline{v}(t)$$

- Objective: Generate estimate $\hat{\underline{x}}(t)$ continuously.
- If there is no noise and we knew $\underline{x}_0$, then

$$\Rightarrow d\hat{\underline{x}}(t)/dt = A\hat{\underline{x}}(t) + B\underline{u}(t); \quad \hat{\underline{x}}(0) = \underline{x}_0$$

$$\dot{\underline{e}}(t) = A\underline{e}(t) \Rightarrow \underline{e}(t) = e^{At}\underline{e}_0 = \underline{0}; \quad \underline{e}(t) = \underline{x}(t) - \hat{\underline{x}}(t)$$

- But, if $\underline{e}_0 \neq 0$ and $A$ is unstable, then $\|\underline{e}(t)\| \to \infty$.
- Solution: Use measurements to stabilize.

e.g., $d\hat{\underline{x}}(t)/dt = A\hat{\underline{x}}(t) + B\underline{u}(t) + K[\underline{y}(t) - C\hat{\underline{x}}(t)]$

residual $\underline{\nu}(t) = \underline{y}(t) - C\hat{\underline{x}}(t) = C\underline{e}(t) + \underline{v}(t)$

# Lyapunov Equation and Estimation - 3

- In fact, we can accomplish more than stability!! We can choose $K$ to minimize a performance criterion.

  "minimize cov$\{e(t)\}$ or MMSE in s.s".

- Now $\dot{\underline{e}}(t) = (A - KC)\underline{e}(t) + E\underline{w} - K\underline{v}(t)$

  let $\Sigma = \text{cov}[e(t)]$ in ss $\Rightarrow 0 = (A - KC)\Sigma + \Sigma(A - KC)^T + EWE^T + KVK^T$

- We need $(A - KC)$ stable, so pick $K^* \ni \Sigma_{K^*} < \Sigma_K \; \forall K \neq K^*$

## ❑ Sensitivity Analysis

$$0 = AX_i + X_i A^T + (A_i X + E_i WE^T + EWE_i^T + XA_i^T); \quad X_i = \frac{\partial X}{\partial \theta_i}$$

output feedback problem, insensitive control design, PDE,....

# Computational Techniques

## Computational Techniques

- Direct methods $\Rightarrow$ solve $A\underline{x} = \underline{b}$
- Iterative methods $\Rightarrow$ sum up terms. Use doubling schemes
- Semi-iterative methods $\Rightarrow$ use QR metod to reduce $A$ to special form and then use $A\underline{x} = \underline{b}$ on the modified matrix.

1) Direct Method: Equation has $n(n+1)/2$ unknowns. Organize $X$ and $S$ as vectors.

- Rewrite as $A_V \underline{x}_V = -\underline{S}_V$;

Example:

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11}x_{11}+a_{21}x_{12} & a_{12}x_{11}+a_{22}x_{12} \\ a_{11}x_{12}+a_{21}x_{22} & a_{12}x_{12}+a_{22}x_{22} \end{bmatrix} + \begin{bmatrix} & \\ & \end{bmatrix}^T = -S$$

$$\Rightarrow \begin{bmatrix} 2a_{11} & 2a_{21} & 0 \\ a_{12} & a_{11}+a_{22} & a_{21} \\ 0 & 2a_{12} & 2a_{22} \end{bmatrix}\begin{bmatrix} x_{11} \\ x_{12} \\ x_{22} \end{bmatrix} = -\begin{bmatrix} s_{11} \\ s_{12} \\ s_{22} \end{bmatrix}$$

- need $\lambda_k(A_V) = \lambda_i + \lambda_j \neq 0$

# Direct Method

- Can see as follows:
  - $A = U\Lambda U^{-1}$; $A^T = V\Lambda V^{-1}$; $Note: V = (U^{-1})^T$ *and* $V^{-1} = U^T$
  - $XU\Lambda U^{-1} + V\Lambda V^{-1}X = -S$
  - $V^{-1}XU.\Lambda + \Lambda V^{-1}XU = -V^{-1}SU$
  - define $Y = V^{-1}XU = U^T XU$ and $\tilde{S} = V^{-1}SU = U^T SU$. then

    $Y\Lambda + \Lambda Y = -\tilde{S}$

    $\Rightarrow y_{ij} = -s_{ij} / (\lambda_i + \lambda_j)$

    $\Rightarrow$ need $\lambda_i + \lambda_j \neq 0$
- Can be solved via LU decomposition
- Can solve for multiple $S_i$
- Direct method requires $O(n^6/24)$ operations.
- Very bad approach for $n \geq 6$ due to round-off errors and/or CPU time.
- Accuracy is not very well controlled.

# Iterative Method: DLyap

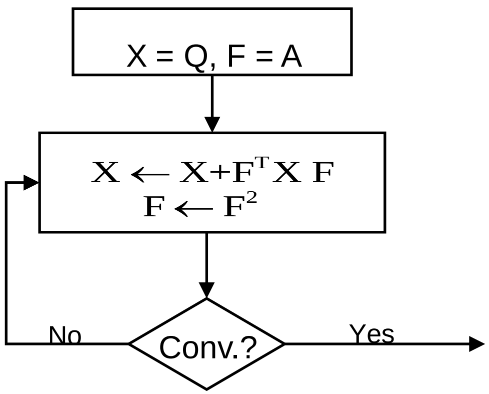
2) "Iterative Method" ....Generate $X_1 \to X_2 \to X_3 \to .... \to X_n \to X$.

- First, consider the discrete Lyapunov equation:

  $X = A^T XA + Q$

  – We know that the solution is $X = \sum_{i=0}^{\infty} (A^T)^i QA^i$.

  – So, sum via the doubling algorithm.




Convergence test $\|\Delta X\| \leq \text{TOL}\,\|X\|$

$\Delta x_{ii}, i = 1, 2, \ldots$

TOL=$10^{-5}$ or so

or test on diagonal elements

- Convergence rate $\|\Delta X_k\| \leq \|(A^2)^k\| . \|X_k\|$
- Actually growth is governed by $\lambda_{\max}(F)$
- So, $\|\Delta X_k\| \leq |\lambda_{\max}(A)|^{2^k} \|X\|$
- Rarely, if ever, do we need more than 10 iterations ($k = 10$ will handle $\lambda_{\max} \approx .99$)
  So, expect $< 25n^3$ operations.

# Iterative Method: Continuous Lyap - 1

## Application to Continuous Lyapunov Equation

Recall $X$ is s.s solution of $dX/dt = A^T X + XA + S$

There are basically two approaches

(a) Discrete-time representation

(b) Bilinear transformation

(a) Discrete-time representation

$$X(t+\Delta) = e^{A^T\Delta} X(t) e^{A\Delta} + \int_0^\Delta e^{A^T\sigma} S e^{A\sigma} d\sigma \text{ valid for } \forall\ \Delta.$$

- So, pick $\Delta \ni \Delta \le .5/\|A\|$ and use algorithms for $e^{A\Delta}$ and $\int_0^\Delta e^{A^T\sigma} S e^{A\sigma} d\sigma$
- Then use doubling scheme with $F = e^{A\Delta}$ and $Q = \int_0^\Delta e^{A^T\sigma} S e^{A\sigma} d\sigma$
- Need ~ $25n^3$ MADDS just for set up.
- Also, truncation errors in $e^{A\Delta}$ and Q will give same order of magnitude errors in $X$
- Convergence rate depends on $|\lambda_{\max}(e^{A\Delta})| = e^{\sigma_{\min}(A)\Delta}$; $\sigma_{\min}$ = min.real part < 0

# Iterative Method: Continuous Lyap - 2

– So, bigger $\Delta$=> faster convergence rate; $\Delta$too small => trouble.

- **Comments:**
  - Very simple method (needs only matrix multiplication routine)
  - Safe and robust, but too costly in initialization
  - Keep $\Delta \geq .1/\|A\|$

(b) Bilinear transformation:

- In (a), we have used an exponential transformation:

$$\Phi = e^{A\Delta}$$

- Idea: Suppose, we define $\Phi = (\tau A + I)(\tau A - I)^{-1}; \tau > 0$

  since functions of $A$ commute, we also have:

$$\Phi = (\tau A - I)^{-1}(\tau A + I)$$

  This is called **bilinear transformation**. Why?

- Because if the transformation is solved for $A$:

# Iterative Method: Continuous Lyap - 3

$$(\tau A - I)\Phi = \tau A + I$$
$$\tau A\Phi - \Phi = \tau A + I \;\Rightarrow \tau A(\Phi - I) = \Phi + I$$
$$\Rightarrow A = (\Phi + I).(\Phi - I)^{-1}/\tau$$

$\Rightarrow$ same form as original $\Rightarrow$ Bilinear

- If we substitute $A$ into $A^T X + XA + S = 0$, we obtain

$$X(\Phi + I)(\Phi - I)^{-1}/\tau + (\Phi^T - I)^{-1}(\Phi^T + I)X/\tau + S = 0$$
$$(\Phi^T - I)X(\Phi + I) + (\Phi^T + I)X(\Phi - I) + \tau(\Phi^T - I)S(\Phi + I) = 0$$
$$\Rightarrow 2\Phi^T X\Phi - 2X + \tau(\Phi^T - I)S(\Phi + I) = 0$$

Note that since $\Phi = (\tau A - I)^{-1}(\tau A + I)$
$$= (\tau A - I)^{-1}(\tau A - I + 2I)$$
$$\Rightarrow \Phi = I + 2(\tau A - I)^{-1}$$
$$\Rightarrow \Phi^T X\Phi - X + 2\tau(\tau A^T - I)^{-1}S(\tau A - I)^{-1} = 0$$

$\Rightarrow$ So, this is a discrete-time Lyapunov equation with

$$Q = 2\tau(\tau A^T - I)^{-1}S(\tau A - I)^{-1}$$

# Iterative Method: Continuous Lyap - 4

## ❑ How to pick $\tau > 0$?

- Recall that $\lambda_i(\Phi) = (\tau\lambda_i(A)+1)/(\tau\lambda_i(A)-1)$.
- We would like to pick $\tau \ni \lambda_{\max}(\Phi)$ is minimized to speed up convergence.
- For real roots $\frac{1}{\tau^*} = \sqrt{\lambda_{\min}(A).\lambda_{\max}(A)} \sim$ geometric mean
- For arbitary case $1/\tau \sim |\,\mathrm{tr}(A)\,|/n$ can be argued "heuristically"

since want $\tau \approx \frac{1}{|\lambda_i|}$. Use $\tau = \min[1, 4n/\sum_i |a_{ii}|]$

# Iterative Method: Continuous Lyap - 5

- **Algorithm using bilinear transformation**

1. Pick $\tau$
2. Compute:

$$\Phi = I + 2(\tau A - I)^{-1}$$

$$Q = 2\tau(\tau A^T - I)^{-1} S(\tau A - I)^{-1}$$

3. Solve for $X$ using doubling scheme

- Note that the set up requires ~ 1.5 multiplications +1 inversion $\approx 2.5n^3$ operations.
- Excellent method for solving Lyapunov equation!

- **Can extend to generalized Lyapunov equation in a straightforward manner**

$$AX + XB + C = 0; A, B \text{ stable}$$

$A$ is $n$ x $n$, $B$ is $m$ x $m$, $C$ is $n$ x $m$ and $X$ is $n$ x $m$

# Semi-iterative Bartels-Stewart Algorihtm - 1

3) Semi iterative methods

Bartels and Stewart "Solution of matrix equation $AX + XB = C$" Comm. of the ACM, vol.15, No.9, sept.1972.

- Consider $A^T X + XA + S = 0$ (*)
  - Idea : Find an orthogonal matrix $Q \ni Q^T AQ$ = upper Schur form

$$Q^T AQ = \tilde{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ & A_{22} & \cdots & A_{2p} \\ & & & A_{pp} \end{bmatrix}$$

diagonals are 1x1 or 2x2 blocks and there are $p$ such blocks

  - Pre- and post multipy(*) by $Q^T$ and $Q$ to obtain
    $Q^T XQ.Q^T AQ + (Q^T AQ)^T Q^T XQ + Q^T SQ = 0$
    $\Rightarrow \tilde{X}\tilde{A} + \tilde{A}^T \tilde{X} + \tilde{S} = 0$
  - Q: Is it easier to solve this equation?
  - A: Yes!! Can be solved in pieces. Recall forward elimination!!

# Bartels-Stewart Algorihtm - 2

– Partition blocks of $X$ conformal with $A$ so that

$$= \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ X_{k1} & X_{k2} & \cdots & X_{kp} \\ X_{p1} & & \cdots & X_{pp} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \,.\, A_{1l} & . & A_{1p} \\ 0 & & A_{22} \cdots & A_{2p} \\ & & A_{ll} & \\ 0 & \cdots & & A_{pp} \end{bmatrix}$$

$$+ \begin{bmatrix} A_{11}^T & & & \\ A_{12}^T & A_{22}^T & 0 & \\ A_{1k}^T & A_{2k}^T \cdots & A_{kk}^T & \\ A_{1p}^T & \cdots & & A_{pp}^T \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ X_{k1} & X_{k2} & \cdots & X_{kp} \\ X_{p1} & & \cdots & X_{pp} \end{bmatrix} + \begin{bmatrix} S_{11} & \cdots & S_{1p} \\ S_{21} & \cdots & S_{2p} \\ S_{p1} & \cdots & S_{pp} \end{bmatrix} = 0$$

– Solve for each sub-block $X_{kl}$.

– Note that:

1. $X_{11}A_{11} + A_{11}^T X_{11} + S_{11} = 0 (k=1, l=1)$

   $\Rightarrow X_{11}$ via algebraic symmetric formula. $X_{11}$ either 2x2 or 1x1.

2. $X_{11}A_{12} + X_{12}A_{22} + A_{11}X_{12} + S_{12} = 0 \; (k=1, l=1)$

   $X_{11}A_{12}$ is known, so the unknown $X_{12}$ can be solved via:

# Bartels-Stewart Algorihtm - 3

$$X_{12}A_{22} + A_{11}X_{12} + (S_{12} + X_{11}A_{12}) = 0$$

Must solve an equation of the form $XA + B^T X + C = 0$

1. $A$ 1x1 and $B$ 1x1
2. $A$ 1x1 and $B$ 2x2
3. $A$ 2x2 and $B$ 1x1
4. $A$ 2x2 and $B$ 2x2

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}; C = \begin{bmatrix} c_1 & c_4 \\ c_2 & c_3 \end{bmatrix}; X = \begin{bmatrix} x_1 & x_4 \\ x_2 & x_3 \end{bmatrix}$$

- For each case, we can solve for $x_1 .. x_4$ algebraically by expanding equation.
- Solution exists provided $\lambda_i(A) + \lambda_j(B) \neq 0$.

Case 1: $x_1 = -c_1 / (a_{11} + b_{11})$

Case 2: $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} a_{11} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0$

$$\Rightarrow \begin{bmatrix} a_{11}+b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Case 3: Similar to 2

# Bartels-Stewart Algorihtm - 4

Case4: $$\begin{bmatrix} a_{11}+b_{11} & b_{21} & 0 & a_{21} \\ b_{12} & a_{11}+b_{22} & a_{21} & 0 \\ 0 & a_{12} & a_{22}+b_{22} & b_{12} \\ a_{12} & 0 & b_{21} & a_{22}+b_{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = - \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

$\Rightarrow$ So, we work across rows $x_{11} \rightarrow x_{12} \rightarrow \ldots x_{1p}, \quad x_{22} \rightarrow \ldots$

– In general, consider block $kl$; $l \geq k$. solve by writing subequation for $X_{kl}$. Have $X_{ij}$ that have already been computed

$$\sum_{i=1}^{l} X_{ki} A_{il} + \sum_{j=1}^{k} A_{jk}^{T} X_{jl} + S_{kl} = 0$$

$$X_{kl} A_{ll} + A_{kk}^{T} X_{kl} + S_{kl} + \sum_{i=1}^{l-1} X_{ki} A_{il} + \sum_{j=1}^{k-1} A_{jk}^{T} X_{jl} = 0$$

– Solution for subblock is fast and accurate via $LU$ decomposition
– Then, desired solution $X = Q\tilde{X}Q^{T}$.

# Bartels-Stewart Algorihtm - 5

- Computational load = $2n^3 + 4\sigma n^3 + 7n^3/2$
  - $A \to$ Hessenberg form $\approx 2n^3$
  - Hessenberg $\to$ Schur form $\approx 4\sigma n^3$
  - Algebraic solution + forming $S$, $X$ from $\tilde{X} \approx 7n^3/2$
- Note if want to solve $XA + A^T X + S_i = 0$, $i = 1, 2, \ldots$ This can be accomplished in ~ $7n^3/2$ (~30% of time to solve for $i = 1$).
- Solution of adjoint equation $XA^T + AX + C = 0$ after having solved original
  - This arises in optimal output feedback and insensitive control system design problems
  - Have $A = Q^T A Q$ = upper Schur form
  - So, need to solve

$$Q^T XQ\left(Q^T AQ\right)^T + Q^T AQ\,Q^T XQ + Q^T CQ = 0$$

$$\tilde{X} = Q^T XQ; \;\; \tilde{A}^T = Q^T AQ = \text{lower Schur form}$$

  - Can transform $\tilde{A}^T$ to upper Schur form via $E\tilde{A}^T E$, where $n$ x $n$ Exchange matrix. $E$ is orthogonal and symmetric $\Rightarrow E^2 = I$ and $E^{-1} = I$

$$E = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

# Bartels-Stewart Algorihtm - 6

- So, solve

$$E\tilde{X}EE\tilde{A}^T E + E\tilde{A}EE\tilde{X}E + E\tilde{C}E = 0$$

$$\Rightarrow \tilde{X}_1 A_1 + \tilde{A}_1^T \tilde{X}_1 + \tilde{C}_1 = 0$$

❑ Algorithm steps:

1) $\tilde{A}_1 = E\tilde{A}^T E = \tilde{A}^T$ with rows and columns in reverse order.
2) $E\tilde{C}E$
3) Solve for $\tilde{X}_1$
4) $X = QE\tilde{X}_1 EQ^T$

❑ Advantages of Barter-Stewart algorithm:

1) Faster than iterative method
2) Excellent for repeated solutions and adjoint (also $C$ need not be equal to $C^T$)
3) Can solve when $A$ is not stable. Need only $\lambda_i + \lambda_j \neq 0$. of course, solution won't be PD in this case.

# Problems with Bartels-Stewart

- Problems:

  1) Disappointing accuracy ≈ 4 digits vs 5 digits for iterative. Why?
     because scheme generated to accuracy of QR and orthogonality of $Q$.
     What to do? **Use iterative improvement**.
     Let $X_1$=solution via Bartels-Stewart's algorithm.
     Let the true solution be $X = X_1 + \delta X$
     $(X_1 + \delta X) A + A^T(X_1 + \delta X) + C = 0$
     $\Rightarrow \delta X A + A^T \delta X + (C + X_1 A + A^T X_1) = 0$
     $C + X_1 A + A^T X_1 \ \rightarrow$ residual must be computed in DP
     solve for $\delta X$ using Bartels-Stewart's algorithm.
     already have $A \Rightarrow 7/2n^3$ ops + 1 matrix multiplication
  2) In case when $A$ = stable and $C \geq 0$, $X$ need not be PD.
  3) More storage needed (≈ 2-3 $n^2$ locations)
  4) More software <u>needed</u> (recall the need for QR algorithm to compute upper Schur form)

# Summary

- ❑ **Background on Lyapunov Equation**
- ❑ **Application of Lyapunov Equation**
- ❑ **Computational methods for solving the Lyapunov Equation**
  - Direct method
  - Iterative methods
  - Semi-iterative methods