Lectures on Numerical Linear Algebra in Control

Laub, Alan

1985

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Laub, A. (1985). *Lectures on Numerical Linear Algebra in Control*. (Technical Reports TFRT-7312). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

CODEN: LUTFD2/(TFRT-7312)/1-56/(1985)

Lectures on Numerical Linear Algebra in Control

Alan Laub

Department of Automatic Control
Lund Institute of Technology
December 1985

Department of Automatic Control **Lund Institute of Technology** P.O. Box 118 S-221 00 Lund Sweden	*Document name* Report *Date of issue* November 1985 *Document Number* CODEN: LUTFD2/(TFRT-7312)/1-56/(1985)
Author(s) Alan Laub	*Supervisor* *Sponsoring organisation*

Title and subtitle

Lectures on Numerical Linear Algebra in Control

Abstract

The report contains notes from a series of lectures held at the Department of Automatic Control, LTH, during June 17-20, 1985.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title	*ISBN*

Language English	*Number of pages* 56	*Recipient's notes*
Security classification		

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

ECE 231 - NUMERICAL SYSTEMS THEORY

Prof. Alan J. Laub
ECE Dep't., UCSB

OUTLINE:

- introduction
- floating-point arithmetic, rounding error analysis, finite arithmetic
- stability and conditioning
- linear algebra review, pseudoinverses, linear equations, singular value decomposition, rank, matrix norms
- numerical matrix algebra, software considerations
- linear equations: algorithms and their numerical analysis, perturbation theory, condition estimation, LINPACK implementation
- linear least squares problems: algorithms and their numerical analysis, perturbation theory, condition estimation, LINPACK and other implementations
- eigenvalue problems: algorithms and their numerical analysis (esp. double Francis QR), perturbation theory, condition estimation, EISPACK implementation
- singular value decomposition, generalized eigenvalue/eigenvector problems
- applications: f(A)
    matrix equations: Lyapunov, Sylvester, Riccati
    various control and systems theory problems

(ref.: Laub, A.J., Numerical Linear Algebra Aspects of Control Design Computations, IEEE Trans. Aut. Contr., Feb. 1985)

ECE 231 - NUMERICAL SYSTEMS THEORY

Prof. Alan J. Laub
ECE Dep't., UCSB

Texts:

[1] Stewart, G.W., Introduction to Matrix Computations, Academic Press, 1973
[2] Rice, J.R., Matrix Computations and Mathematical Software, McGraw-Hill, 1981

Recommended References:

[3] Golub, G.H., and C.F. VanLoan, Matrix Computations, Johns Hopkins University Press, 1983
[4] Noble, B., and J.W. Daniel, Applied Linear Algebra, Prentice-Hall, 1977
[5] Strang, G., Linear Algebra and its Applications, 2nd Ed., Academic Press, 1980
[6] Forsythe, G., and C.B. Moler, Computer Solution of Linear Algebraic Systems, Prentice-Hall, 1967
[7] Lawson, C.L., and R.J. Hanson, Solving Least Squares Problems, Prentice-Hall, 1974
[8] Wilkinson, J.H., Rounding Errors in Algebraic Processes, Prentice-Hall, 1963
[9] Wilkinson, J.H., The Algebraic Eigenvalue Problem, Oxford University Press, 1965
[10] Parlett, B.N., The Symmetric Eigenvalue Problem, Prentice-Hall, 1980
[11] Householder, A.S., The Theory of Matrices in Numerical Analysis, Blaisdell, 1964 (Dover, 1975)
[12] Smith, B.T., et.al., EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, Springer-Verlag, 1976
[13] Garbow, B.S., et.al., EISPACK Guide Extension, Lecture Notes in Computer Science, Vol. 51, Springer-Verlag, 1977
[14] Wilkinson, J.H., and C. Reinsch, Handbook for Automatic Computation, Vol. II, Linear Algebra, Springer-Verlag, 1971
[15] Dongarra, J.J., et.al., LINPACK User's Guide, SIAM, 1979

Journals:

SIAM Review
SIAM Journal on Numerical Analysis
SIAM Journal on Scientific and Statistical Computing
Mathematics of Computation
Numerische Mathematik
BIT

1. INTRODUCTION

ANALYSIS OF NUMERICAL ALGORITHMS IN FINITE ARITHMETIC (FINITE PRECISION, FINITE RANGE)

ACCURACY OF COMPUTED SOLUTIONS DEPENDS ON:

- CONDITIONING OF THE PROBLEM
- NUMERICAL STABILITY OF THE ALGORITHM
- SPECIFICS OF THE SOFTWARE IMPLEMENTATION

CONDITIONING

PROBLEM IS WELL-CONDITIONED IF SMALL CHANGES IN DATA CAUSE ONLY SMALL CHANGES IN SOLUTION; IF SMALL CHANGES IN DATA HAVE THE POTENTIAL TO INDUCE LARGE CHANGES IN THE SOLUTION, PROBLEM IS ILL-CONDITIONED.

EXAMPLE

SOLVE $Ax = b$

with $A = \begin{pmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{pmatrix}$, $b = \begin{pmatrix} 0.217 \\ 0.254 \end{pmatrix}$

"TRUE SOLUTION" IS $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

LET $E = \begin{pmatrix} 0.001 & 0.001 \\ -0.002 & -0.001 \end{pmatrix} \leftarrow$ "SMALL"

THEN SOLUTION OF $(A+E)x = b$ IS $x = \begin{pmatrix} -5 \\ 7.3085\cdots \end{pmatrix}$!

SOLUTION OF $Ax = b$ IS ILL-CONDITIONED FOR THIS A.

* $\Longrightarrow$ RESIDUALS ARE UNRELIABLE INDICATORS OF RELATIVE SOLUTION ACCURACY

e.g. $x_{(1)} = \begin{pmatrix} 0.999 \\ -1.001 \end{pmatrix}$: $r_{(1)} := Ax_{(1)} - b \approx \begin{pmatrix} -0.001243 \\ -0.001572 \end{pmatrix}$

$x_{(2)} = \begin{pmatrix} 0.341 \\ -0.0870 \end{pmatrix}$: $r_{(2)} := Ax_{(2)} - b \approx \begin{pmatrix} -0.000001 \\ 0.0 \end{pmatrix}$

STABILITY

ALGORITHM IS <u>NUMERICALLY STABLE</u> IF IT DOES NOT INTRODUCE ANY MORE SENSITIVITY TO PERTURBATION THAN IS ALREADY INHERENT IN THE PROBLEM; MANY NLA ALGORITHMS CAN BE SHOWN TO BE <u>BACKWARD STABLE</u>, I.E. THE COMPUTED SOLUTION CAN BE SHOWN TO BE (NEAR) THE EXACT SOLUTION OF A SLIGHTLY PERTURBED ORIGINAL PROBLEM

<u>EXAMPLE</u>

SOLVE $Ax = b$ BY GAUSSIAN ELIMINATION WITH NO PIVOTING.

TAKE $A = \begin{pmatrix} 0.0001 & 1.000 \\ 1.000 & -1.000 \end{pmatrix}$, $b = \begin{pmatrix} 1.000 \\ 0.0000 \end{pmatrix}$ (4 DEC. PL. ARITH.)

"TRUE x" IS $\begin{pmatrix} 0.9999 \\ 0.9999 \end{pmatrix}$ (PROB. <u>IS</u> WELL-CONDITIONED)

G. ELIM.: $\begin{pmatrix} 0.0001 & 1.000 \\ 0 & -1.000 \times 10^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.000 \\ -1.000 \times 10^4 \end{pmatrix}$

$\Rightarrow x_2 = 1.000$ (A GOOD APPROX.)

$\Rightarrow 0.0001 x_1 = 1.000 - (1.000 \times 1.000) = 0.0000$

$\Rightarrow x_1 = 0.0000$ (A <u>BAD</u> APPROX.)

FOR G. ELIM. <u>WITH PIVOTING</u> COMPUTED $\bar{x}$ CAN BE SHOWN TO SATISFY $(A+E)\bar{x} = b$ WITH E "SMALL" (USUALLY).

EXAMPLES

(1) $$fl(x * y) = (x * y)(1+\delta), \quad |\delta| < \epsilon$$
$$= x(1+\delta)^{1/2} * y(1+\delta)^{1/2}$$

(2) SOLUTION OF $Ax = b$

$\hat{x}$ = COMPUTED SOLUTION

$(A+E)\hat{x} = b$

$(A+E)^{-1}b$ vs. $A^{-1}b$
PERTURBATION THEORY

WHERE $|e_{ij}| \leq g(n) \cdot \gamma \cdot \alpha \cdot \epsilon$ $(\|E\| \leq c \cdot \|A\| \cdot \epsilon)$

$g(n)$ = LOW DEG. FCN. OF n DEP. ON DETAILS OF ARITHMETIC USED

$\gamma = \dfrac{\text{MAG. OF LARGEST ELT. ENCOUNTERED IN REDUC.}}{\text{MAG. OF LARGEST ELT. OF } A}$

= "GROWTH FACTOR" ($\leq 2^{n-1}$, PAR. PIV.)

α = MAG. OF LARGEST ELT. OF A

ALG. IS STABLE IF γ SMALL BUT PROBLEM MAY STILL BE ILL-CONDITIONED; PERTURBATIONS IN A (AND/OR b) MAY BE MAGNIFIED BY AS MUCH AS $K(A) := \|A\| \cdot \|A^{-1}\|$ IN THE SOLUTION.

(3) EIGENVALUE / EIGENVECTOR PROBLEMS

DENSE E: "HARD ZEROS"

SOFTWARE

SHOULD BE <u>RELIABLE</u> AND <u>ROBUST</u> (MANY ASPECTS AND CHARACTERIZATIONS) ; SHOULD WARN USER (even another computer program) OF UNUSUAL SITUATIONS OR POTENTIALLY INACCURATE RESULTS (because of, e.g., ill-conditioning) ; MUST BE COGNIZANT OF THE FINITENESS OF THE UNDERLYING (FLOATING-POINT) ARITHMETIC.

EXAMPLES

- LINEAR EQUATION SOFTWARE SHOULD MONITOR CONDITION
- MAY NEED UNDERFLOW -/ OVERFLOW - PROOFING
  e.g. $\sqrt{x^2+y^2}$
- COLUMNWISE ORIENTATION FOR FORTRAN
- PORTABILITY ; USE OF "LOCAL TRICKS"

ACCURACY OF COMPUTED SOLUTIONS

$$\boxed{\text{WELL-CONDITIONED PROBLEM}} + \boxed{\text{NUMERICALLY STABLE ALGORITHM}} + \boxed{\text{ROBUST SOFTWARE IMPLEMENTATION}}$$

$$\longrightarrow \boxed{\text{ACCURATE ANSWERS}}$$

SCALING

$$\text{STABILITY} \Rightarrow \underbrace{\underline{\text{COMPUTED SOL'N.}}\text{ IS (NEAR) THE EXACT SOL'N. OF A NEARBY PROBLEM}}_{\text{NEAR "}\underline{\text{TRUE SOLUTION}}\text{" IF PROBLEM IS WELL-CONDITIONED}}$$

<u>IF</u> RELIABLY IMPLEMENTED IN SOFTWARE

2. SINGULAR VALUE DECOMPOSITION

A FUNDAMENTAL (THEORETICAL) WORKING TOOL IN NUMERICAL LINEAR ALGEBRA ; NOW USED AND MIS-USED IN CONTROL (AND SIGNAL PROC., STATISTICS, etc.)

THEOREM : LET $A \in \mathbb{R}_r^{m\times n}$ (OR $\mathbb{C}_r^{m\times n}$ WITH ANALOGOUS FORMULATION)

WHERE $r = \text{rank}(A) \le \min\{m,n\}$.

THEN THERE EXIST ORTHOGONAL MATRICES $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ SUCH THAT

$$\underset{m\times n}{A} = \underset{m\times m}{U}\,\underset{m\times n}{\Sigma}\,\underset{n\times n}{V^T}$$

WHERE $\Sigma = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}$, $S = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{pmatrix}$

AND $\sigma_1 \geqslant \cdots \geqslant \sigma_r > 0$.

MORE SPECIFICALLY,

$$A = \left(\underset{m\times r}{U_1}, \underset{m\times(m-r)}{U_2}\right)\begin{pmatrix} \underset{r\times r}{S} & \underset{r\times(n-r)}{0} \\ \underset{(m-r)\times r}{0} & \underset{(m-r)\times(n-r)}{0} \end{pmatrix}\begin{pmatrix} \underset{r\times n}{V_1^T} \\ \underset{(n-r)\times n}{V_2^T} \end{pmatrix} = U_1 S V_1^T$$

SVD IS NOT UNIQUE.

HOWEVER, THE COLUMNS OF U_1, U_2, V_1, V_2 DO SPAN UNIQUE SUBSPACES

EXAMPLES

1. $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = U \cdot I \cdot U^T$; U = ARB. ORTHOG. MATRIX

2. $A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}$

3. $\underset{\in \mathbb{R}_1^{3\times 2}}{A = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 2 & 2 \end{pmatrix}} = \left(\begin{array}{c:cc} 1/3 & -2/\sqrt{5} & 2\sqrt{5}/15 \\ 2/3 & 1/\sqrt{5} & 4\sqrt{5}/15 \\ 2/3 & 0 & -5\sqrt{5}/15 \\ U_1 & U_2 & \end{array}\right)\overset{S}{\left(\begin{array}{c:c} 3\sqrt{2} & 0 \\ \hdashline 0 & 0 \\ 0 & 0 \end{array}\right)}\left(\begin{array}{cc} \overset{V_1^T}{1/\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \hdashline 1/\sqrt{2} & -1/\sqrt{2} \\ & V_2^T \end{array}\right)$

4. $A = A^T > 0$;
   $A = V \Lambda V^T$

DEFN. : $\sigma_1 \geq \cdots \geq \sigma_r > 0$ ARE THE SINGULAR VALUES (DENOTED $\Sigma(A)$) OF A AND ARE THE POSITIVE SQUARE ROOTS OF THE EIGENVALUES OF

A^TA (ALSO $\sigma_{r+1} = \cdots = \sigma_n = 0$).

OR AA^T (ALSO $\sigma_{r+1} = \cdots = \sigma_m = 0$).

u_i : COLUMNS OF U ARE THE LEFT SINGULAR VECTORS OF A (ORTHONORMAL EIGENVECTORS OF AA^T).

v_i : COLUMNS OF V ARE THE RIGHT SINGULAR VECTORS OF A (ORTHONORMAL EIGENVECTORS OF A^TA).

$$\left.\begin{aligned} Av_i &= \sigma_i u_i \\ A^T u_i &= \sigma_i v_i \end{aligned}\right\} \quad i = 1, \ldots, r$$

EVALS. OF $\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$ ARE $\pm\sigma_i$

IF WORKING IN FINITE ARITHMETIC,

DON'T COMPUTE EIGENDECOMPOSITION OF A^TA OR AA^T

(cf. FOLK THEOREM) (KF vs. SQRT.FLT.)

EXAMPLE

$$A = \begin{pmatrix} 1 & 1 \\ \mu & 0 \\ 0 & \mu \end{pmatrix} \in \mathbb{R}^{3\times 2}_2$$

$0 < |\mu| < \sqrt{\epsilon}$ $\qquad \epsilon := \min\limits_{\delta\in F} \{\delta : fl(1+\delta) > 1\}$

$\Longrightarrow fl(1+\mu^2) = 1$

IN INFINITE PRECISION, $A^TA = \begin{pmatrix} 1+\mu^2 & 1 \\ 1 & 1+\mu^2 \end{pmatrix}$

$\sigma_1 = \sqrt{2+\mu^2}$, $\sigma_2 = |\mu|$ $\Longrightarrow \operatorname{rank}(A) = 2$

IN FINITE PRECISION, $fl(A^TA) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

$\hat{\sigma}_1 = 1.4142\cdots$, $\hat{\sigma}_2 = 0 \Longrightarrow \operatorname{rank}(A) = 1$

THERE EXISTS A BETTER (STABLE) ALGORITHM THAT WORKS DIRECTLY WITH A.

THEOREM: (i) rank$(A) = r$
= NO. OF NONZERO SINGULAR VALUES OF A

(ii) $\|A\|_2 = \sigma_1$

$$\|A^+\|_2 = \frac{1}{\sigma_r} \quad (= \|A^{-1}\|_2 \text{ IF } m = n = r)$$

WHERE $\|A\|_2 := \max\limits_{\|x\|_2 \neq 0} \dfrac{\|Ax\|_2}{\|x\|_2}$

A^+ = MOORE-PENROSE PSEUDO-INVERSE

$$= V_1 S^{-1} U_1^T$$

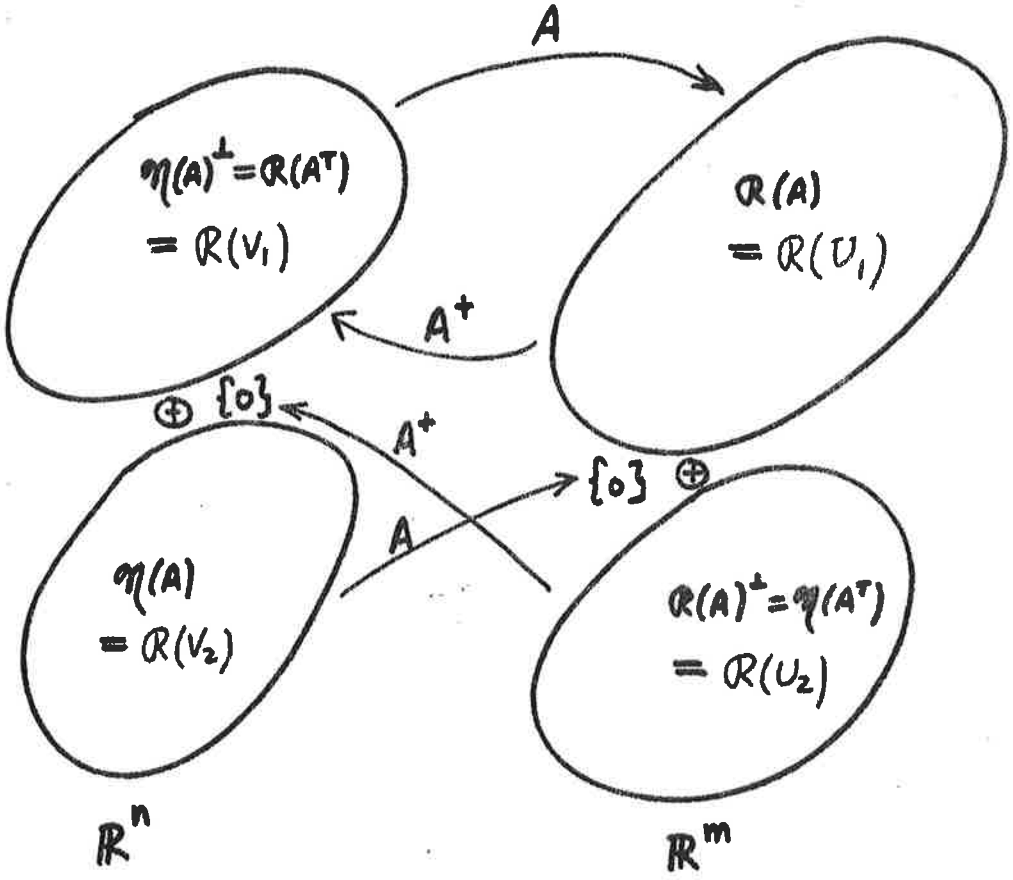
$$= V \Sigma^+ U^T \quad \text{WHERE } \Sigma^+ = \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

(iii) LET $U = (\underbrace{u_1, \cdots, u_r}_{U_1}, \underbrace{u_{r+1}, \cdots, u_m}_{U_2})$

$$V = (\underbrace{v_1, \cdots, v_r}_{V_1}, \underbrace{v_{r+1}, \cdots, v_n}_{V_2})$$

THEN $A = \sum\limits_{i=1}^{r} \sigma_i u_i v_i^T$

FOUR FUNDAMENTAL SUBSPACES



ⓡ IS IMPLICIT IN DEFINITION OF ALL 4 OF THESE SUBSPACES

SVD PROVIDES A NUMERICALLY SUPERIOR METHOD FOR FINDING BASES FOR THE 4 FUNDAMENTAL SUBSPACES THAN METHODS BASED ON REDUCTION TO ROW OR COLUMN ECHELON FORM.

SIMILAR REMARKS APPLY TO COMPUTING:

ROW COMPRESSION :
$$U^TA = \Sigma V^T = \begin{pmatrix} SV_1^T \\ 0 \end{pmatrix} \longleftarrow \text{FULL ROW RANK}$$

COLUMN COMPRESSION :
$$AV = U\Sigma = (\underset{\nwarrow}{U_1S},\ 0)$$
FULL COLUMN RANK

ORTHOGONAL PROJECTIONS:
$$\begin{aligned} P_{\mathcal{R}(A)} &= AA^+ = U_1U_1^T \\ P_{\mathcal{R}(A)^\perp} &= I - AA^+ = U_2U_2^T \\ P_{\mathcal{N}(A)} &= I - A^+A = V_2V_2^T \\ P_{\mathcal{N}(A)^\perp} &= A^+A = V_1V_1^T \end{aligned}$$

APPROXIMATION OF $A \in \mathbb{R}_r^{m\times n}$ BY A LOWER RANK MATRIX :

THEOREM : LET A HAVE SVD $A = U\Sigma V^T$ AND DEFINE
$$A_k := \sum_{i=1}^{k} \sigma_i u_i v_i^T \quad ; \quad k < r$$
THEN
$$\min_{\text{rank}(B)=k} \|A-B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

COROLLARY : LET $A \in \mathbb{R}_n^{n\times n}$. THEN SMALLEST $\|E\|_2$ SUCH THAT $A+E$ IS SINGULAR IS GIVEN BY
$$E = U \begin{pmatrix} 0 & & & 0 \\ & \ddots & & \\ & & 0 & \\ 0 & & & -\sigma_n \end{pmatrix} V^T.$$

APPLICATIONS :

ROBUST STABILITY OF LINEAR SYSTEMS
IMAGE PROCESSING
MODEL REDUCTION

ACCURATE COMPUTATION OF SVD

SINGULAR VALUES ARE WELL-CONDITIONED :

THEOREM : LET $A \in \mathbb{R}^{m \times n}$ WITH SINGULAR VALUES $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ AND SUPPOSE $A+E$ HAS SINGULAR VALUES $\tau_1 \geq \cdots \geq \tau_n \geq 0$.

THEN $|\sigma_k - \tau_k| \leq \|E\|_2$; $k = 1, \ldots, n$

NOTE : THIS IS AN ABSOLUTE RESULT, NOT RELATIVE. CAN'T COMPUTE SING. VAL. SPREAD $> \frac{1}{\epsilon}$.

THERE IS A NUMERICALLY STABLE ALGORITHM TO COMPUTE SINGULAR VALUES : GOLUB & REINSCH, 1970, NUMER. MATH.

COMPUTED σ_i'S ARE EXACT FOR $A+E$ FOR SOME "SMALL" E ; $\|E\|_2 = O(\epsilon) \cdot \|A\|$

BASIC IDEA : 1. REDUCE A TO UPPER BIDIAGONAL FORM

e.g. $\begin{pmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 \end{pmatrix}$

2. QR-TYPE ITERATION TO "DIAGONALIZE"

CAN BE DONE IN $\approx 2mn^2 + 4n^3$ FLOPS $(m \geq n)$

A USEFUL TRICK : 1. DO HOUSEHOLDER : $QA = \begin{pmatrix} R \\ 0 \end{pmatrix} \leftarrow$ ASSUME FULL RANK FOR CONVENIENCE

2. $R = \hat{U} S \hat{V}^T \quad (n \times n)$

$\approx mn^2 + \frac{17}{3} n^3$

3. $A = (Q_1^T \hat{U}, Q_2^T) \begin{pmatrix} S \\ 0 \end{pmatrix} \hat{V}^T$ IS SVD OF A

WHERE $Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$

MANY OTHER SVD ALGORITHMS

THERE EXIST ROBUST, RELIABLE SOFTWARE IMPLEMENTATIONS :

EISPACK

SVD, MINFIT

LINPACK

SSVDC, DSVDC, CSVDC, ZSVDC

CORRECTION :

```
        .
        .
        .
550 CONTINUE
    F = (SL+SM)*(SL-SM) - SHIFT
```

SHOULD BE

+

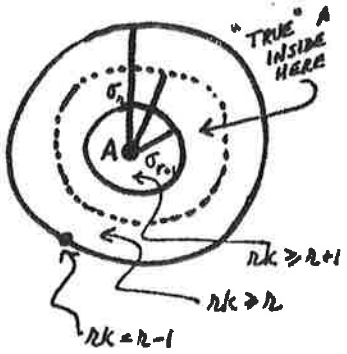
RANK

- MOST ALGS. IN LIN. SYS. & CONTROL INVOLVE AN EXPLICIT OR IMPLICIT DETERMINATION OF RANK
- ∃ NONTRIV. PROBLEMS IN FINITE ARITHMETIC
- DIFFICULTIES COMPOUNDED BY :
  - UNCERTAIN DATA / PARAMETERS
  - HIGH ORDER

EXAMPLE (OSTROWSKI, 1954)

$$A = \begin{pmatrix} -1 & +1 & \cdots & +1 \\ & -1 & +1 \cdots & +1 \\ & & \ddots & \vdots \\ \text{\Large 0} & & \ddots & +1 \\ & & & -1 \end{pmatrix} \in \mathbb{R}^{n \times n} \qquad A \cdot \begin{pmatrix} 1 \\ 1/2 \\ \vdots \\ 2^{-n+1} \end{pmatrix} = \underset{\sim 0}{\begin{pmatrix} 2^{-n+1} \\ \vdots \\ 2^{-n+1} \end{pmatrix}}$$

- SHAPE & SPECTRUM UNRELIABLE IN DETECTING NEAR-SINGULARITY
- COROLLARY : G. ELIM. TECHS. ALSO SIMILARLY UNRELIABLE
- STABILITY MARGIN
- $\sigma_n(A) \sim 2^{-n}$



- CAN USE S.V.'S TO MAKE A FAIRLY SENSIBLE NUMERICAL DETERM. OF RANK : "GAP"?
- SVD VS QR
- CONSTRAINED PERTURBATIONS

- MOST OBJECTS IN LINEAR ALGEBRA AND LINEAR SYSTEMS INVOLVE AN EXPLICIT OR IMPLICIT DETERMINATION OF RANK
- HENCE ∃ MANY NONTRIVIAL PROBLEMS <u>IN THE PRESENCE OF ROUNDING ERROR</u>
- DIFFICULTIES COMPOUNDED WHEN:
  - DATA / PARAMETERS UNCERTAIN
  - HIGH-ORDER
- NUMERICAL PROCEDURES ALSO INFLUENCED BY ULTIMATE USE OF THE CALCULATIONS
- CONSTRAINED PERTURBATIONS

A "MIS-USE" OF SVD

(A, B) CONTROLLABLE IF

$$\text{rank}\,(B, AB, \cdots, A^{n-1}B) = n$$

SUPPOSE $A = \begin{pmatrix} 1 & \mu \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ \mu \end{pmatrix}$

$$|\mu| < \sqrt{\epsilon}$$

THEN $fl\,(B, AB) = \begin{pmatrix} 1 & 1 \\ \mu & \mu \end{pmatrix}$.

APPLICATION OF SVD TO DETERMINE rank $fl\,(B, AB)$ YIELDS THE ERRONEOUS CONCLUSION OF UNCONTROLLABILITY.

THE DAMAGE IS DONE (IRRECOVERABLY) IN SIMPLY FORMING A^kB (cf. NORMAL EQUATIONS A^TA, etc.)

COMPUTATION OF SYSTEM BALANCING TRANSFS.

HEATH, LAUB, PAIGE, WARD
ORNL UCSB McGILL ORNL

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

+ ASSUMPTIONS

REACHABILITY GRAMMIAN : $\int_0^{+\infty} e^{tA} BB^T e^{tA^T} dt =: Y$

$$AY + YA^T + BB^T = 0$$

OBSERVABILITY GRAMMIAN : $\int_0^{+\infty} e^{tA^T} C^T C e^{tA} dt =: X$

$$A^T X + XA + C^T C = 0$$

FIND BALANCING (OR OTHER CONTRAGREDIENT) $T \ni$

$$T^{-1} Y T^{-T} = \Lambda = \text{diag.}$$
$$T^T X T = \Lambda$$

and $A_b := T^{-1} A T$, $B_b := T^{-1} B$, $C_b := CT$

ALGORITHM OUTLINE

(1) SOLVE LYAPUNOV EQUATIONS FOR THE CHOLESKY FACTORS OF THE GRAMMIANS

$$Y = L_r L_r^T \quad , \quad X = L_o L_o^T$$

L_r, L_o FOUND DIRECTLY (HAMMARLING) WITHOUT "SQUARING UP"

(2) COMPUTE SVD OF $L_o^T L_r$

$$L_o^T L_r = U \Lambda V^T$$

(HLPW, 1984)

PRODUCT OF L_o^T AND L_r IS NEVER FORMED EXPLICITLY

$$\Lambda = \{ \sigma_i (L_o^T L_r) \} = \{ \lambda_i^{1/2} (YX) \}$$

(3) BALANCING TRANSFORMATION

$$T = L_r V \Lambda^{-\frac{1}{2}}$$

$$T^{-1} = \Lambda^{-\frac{1}{2}} U^T L_o^T$$

$$A_b = \Lambda^{-\frac{1}{2}} U^T L_o^T A L_r V \Lambda^{-\frac{1}{2}}$$

$$B_b = \Lambda^{-\frac{1}{2}} U^T L_o^T B$$

$$C_b = C L_r V \Lambda^{-\frac{1}{2}}$$

SIMILAR TECHNIQUE TO GET BEST CONDITIONED CONTRAGREDIENT TRANSFORMATION

OTHER APPLICATIONS :
- $ABx = \lambda x$
- RICCATI BALANCING
- DISCRETE-TIME PROBS.

I. INTRODUCTION

- LINEAR MODELS

$$M\ddot{q} + G\dot{q} + Kq = \hat{B}u$$
$$y = Pq + R\dot{q}$$

u : INPUT

y : OUTPUT

$M = M^T > 0$

$K = K^T \geqslant 0$

$G = G_1 + G_2$

$G_1 = G_1^T$

$G_2 = -G_2^T$

- FIRST-ORDER REALIZATIONS

$$\frac{d}{dt}\begin{pmatrix} q \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -M^{-1}K & -M^{-1}G \end{pmatrix}\begin{pmatrix} q \\ \dot{q} \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1}\hat{B} \end{pmatrix} u$$

$$y = (P, R)\begin{pmatrix} q \\ \dot{q} \end{pmatrix}$$

- STANDARD $\dot{x} = Ax + Bu$, $y = Cx$ FORM

- OTHER REALIZATIONS

$$\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix} \frac{d}{dt} \begin{pmatrix} q \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -K & -G \end{pmatrix} \begin{pmatrix} q \\ \dot{q} \end{pmatrix} + \begin{pmatrix} 0 \\ \hat{B} \end{pmatrix} u$$

$$y = (P, R) \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$$

OR $$\underset{\text{SYMM.}}{\begin{pmatrix} -K & 0 \\ 0 & M \end{pmatrix}} \dot{x} = \underset{\text{SYMM. IF } G_2 = 0}{\begin{pmatrix} 0 & -K \\ -K & -G \end{pmatrix}} x + \begin{pmatrix} 0 \\ \hat{B} \end{pmatrix} u$$

OR $$\underset{\substack{\text{SKEW-SYMM.} \\ \text{IF } G_1 = 0}}{\begin{pmatrix} G & M \\ -M & 0 \end{pmatrix}} \dot{x} = \underset{\text{SYMM.}}{\begin{pmatrix} -K & 0 \\ 0 & -M \end{pmatrix}} x + \begin{pmatrix} \hat{B} \\ 0 \end{pmatrix} u$$

etc.

- ALL OF THE "STANDARD FORM":

$$E\dot{x} = Ax + Bu$$
$$y = Cx \; (+Du)$$

- DISCRETE-TIME; TIME-VARYING; ...

- NEED ALGORITHMS — WHICH ARE RELIABLE IN FINITE ARITHMETIC — FOR ANALYSIS AND SYNTHESIS PROBLEMS ARISING FROM MODELS SUCH AS

$$E\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- (E, A, B, C, D) SMALL, DENSE vs. LARGE, SPARSE

- (M, G, K, B, P, R, D) MODELS

- THESE ALGORITHMS, ROBUSTLY IMPLEMENTED IN SOFTWARE, MAY BE COMPONENTS OF A MORE COMPLEX DESIGN OR ANALYSIS TOOL INVOLVING I/O, DATABASES, GRAPHICS, OTHER HARDWARE, etc.

- WELL-CONDITIONED PROBLEM + NUMERICALLY STABLE ALGORITHM + ROBUST SOFTWARE IMPLEMENTATION

  $\longrightarrow$ ACCURATE ANSWERS

- MORE DIFFICULT TO GET CANONICAL FORMS FOR THE

$$M\ddot{q} + G\dot{q} + Kq = \hat{B}u \qquad (*)$$

MODEL THAT RESPECT THE STRUCTURE

  - e.g. freq. resp.

- MODAL TESTS

LET $W := \{ \lambda_i : \lambda_i$ is a generalized eigenvalue of the problem $\begin{pmatrix} 0 & I \\ -K & -G \end{pmatrix} - \lambda \begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix} \}$

(or various equivalent problems)

= MODES OF THE MODEL (*)

THEN (*) IS CONTROLLABLE IFF

$$\text{rank}\left[\lambda^2 M + \lambda G + K, \hat{B}\right] = n \qquad \forall \lambda \in W.$$

- VARIOUS SPECIAL CASES FOLLOW EASILY
- CAN CHECK PARTICULAR MODES SELECTIVELY

- STATUS OF CONTROL ALGORITHMS :
  - MANY PROBLEMS NOT WELL-UNDERSTOOD
  - MANY "OPEN" PROBLEMS (esp. in finite arith.; IEEE / non. IEEE)
  - MANY ALGORITHMS NOT COMPLETELY ANALYZED

- STATUS OF CONTROL SOFTWARE :
  - STANDARDS CONSTANTLY EVOLVING
  - AN ENORMOUS NUMBER AND VARIETY OF PROBLEMS TO BE SOLVED :

MTNS, 1985
Stockholm

THE GENERALIZED EIGENPROBLEM APPROACH TO THE NUMERICAL SOLUTION OF ALGEBRAIC RICCATI EQUATIONS

Alan J. Laub
Department of Electrical & Computer Engineering
University of California
Santa Barbara, CA 93106

Abstract

Recent techniques for the solution of various types of Riccati equations by means of associated matrix pencils will be discussed. The resulting associated generalized eigenproblem framework provides a unifying methodology which facilitates the solution of Riccati equations arising in optimal control or filtering problems with "nonstandard" features such as singular control weighting or measurement noise covariance matrices, cross-weighting or cross-correlation matrices, and singular transition matrices (discrete-time). Generalized state-space models are also considered and are shown to give rise to "generalized" Riccati equations.

Various algebraic features of the above problems are discussed along with pertinent numerical aspects. The generalized eigenproblem methodology has been implemented in a Fortran software package called RICPACK. A number of key features of RICPACK will be discussed along with several related algorithmic details.

AJL3/H

ALGORITHMS, ANALYSIS, AND SOFTWARE FOR RICCATI EQUATIONS

Principal Investigator: Professor Alan J. Laub
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106

The principal goal of this project is to investigate various aspects of the structure and numerical solution of general classes of matrix Riccati equations and related problems. Riccati equations arise naturally in applied mathematics and a wide variety of engineering and scientific applications and their role and use in systems and control theory, in particular, has been well-established over the past twenty-five years. This project will employ an interdisciplinary approach blending control and systems theory, numerical linear algebra, and mathematical software.

Riccati equation algorithms based on certain types of matrix pencils and associated generalized eigenvalue problems will be stressed initially as they offer simultaneously efficiency, numerical robustness, and flexibility to handle a broad range of problem types and applications. Topics proposed for research include mathematical extensions of the underlying linear algebraic structure, Riccati differential and difference equations, nonsymmetric and nondefinite Riccati equations, exploitation of special matrix structures, and dual results for filtering applications. In the beginning, the focus will be on small problems (say, of order a couple of hundred or so), solved on conventional computing machines. Later in the project our attention will shift more towards large-scale equations and parallel and sparse-matrix algorithms with implementations for more "exotic" computing environments of a parallel or concurrent nature as those mature with respect to both hardware and software.

Throughout the project considerable attention will be paid to the effects on numerical algorithms of finite-precision, finite range arithmetic. Finite arithmetic effects provide a compelling motivation for the use of numerically stable algorithms and further understanding of the condition of the Riccati equation problem. Algorithms for scaling and balancing must also be considered as must various approaches for iterative refinement.

Implementation of pertinent research results as robust mathematical software will be a primary vehicle for reliable technology transfer. The necessary software research and development will reflect recent advances in computer hardware and software environments. Software modules relating to the numerical solution of Riccati equations will be developed which will find numerous applications in computer-aided analysis and design efforts.

OUTLINE

0. BACKGROUND

1. INTRODUCTION TO SCHUR METHODS

2. GENERALIZED EIGENVALUE PROBLEMS

3. ORTHOGONAL SYMPLECTIC REDUCTIONS

4. GENERALIZED RICCATI EQUATIONS — EXTENDED FORMULATIONS

5. NONSYMMETRIC RICCATI EQUATIONS

6. CONDITIONING

7. SOFTWARE - RICPACK

---

0. BACKGROUND

- RICCATI EQUATIONS :

  $\left\{ \begin{array}{l} \text{CONTINUOUS - TIME} \\ \text{DISCRETE - TIME} \end{array} \right.$

  $\left\{ \begin{array}{l} \text{ALGEBRAIC EQUATIONS} \\ \text{DIFFERENTIAL OR DIFFERENCE EQUATIONS} \end{array} \right.$

  $\left\{ \begin{array}{l} \text{SYMMETRIC} < \begin{array}{l} \text{DEFINITE} \\ \text{INDEFINITE} \end{array} \\ \\ \text{NONSYMMETRIC} \end{array} \right.$

- MANY SOLUTION TECHNIQUES
  - METHODS BASED ON SPECIAL CANONICAL FORMS
  - DOUBLING & OTHER DIRECT INTEGRATION TECHNIQUES
  - NEWTON'S METHOD
  - PARAMETER IMBEDDING METHODS
  - CHANDRASEKHAR - TYPE ALGORITHMS
  - METHODS BASED ON MATRIX SIGN FUNCTION
  - SPECTRAL FACTORIZATION TECHNIQUES
  - "SQUARE ROOT" FORMULATIONS
  - EIGENVECTOR TECHNIQUES

  ⋮

- EIGENVECTOR METHODS — REVIEW

$$\underset{u}{\text{Min}} \quad \frac{1}{2}\int_0^{+\infty} [y^TQy + u^TRu]\,dt$$

$$\text{s.t.} \quad \dot{x} = Fx + Bu$$
$$y = Cx$$

$$u^*(t) = \underbrace{-R^{-1}B^TX}_{=:K}\, x(t)$$

where $F^TX + XF - X\underbrace{BR^{-1}B^T}_{=:G}X + \underbrace{C^TQC}_{=:H} = 0$

Compute $T \ni$

$$\begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}^{-1} \begin{pmatrix} F & -G \\ -H & -F^T \end{pmatrix} \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} = \begin{pmatrix} -\Lambda & 0 \\ 0 & \Lambda \end{pmatrix}$$

$$\begin{pmatrix} F & -G \\ -H & -F^T \end{pmatrix} \begin{pmatrix} T_{11} \\ T_{21} \end{pmatrix} = \begin{pmatrix} T_{11} \\ T_{21} \end{pmatrix} (-\Lambda)$$

$$\Longrightarrow X = T_{21}T_{11}^{-1}$$

- NUMERICAL PROBLEMS (see Golub & Wilkinson, 1976)

→ • SCHUR METHODS :
  - EFFICIENT
  - NUMERICALLY ROBUST
  - FLEXIBILITY TO HANDLE BROAD RANGE OF PROBLEMS

1. INTRODUCTION TO SCHUR METHODS

CONTINUOUS-TIME ALGEBRAIC RICCATI EQUATION

$$F^TX + XF - XGX + H = 0$$

WITH $0 \le G = G^T \in \mathbb{R}^{n \times n}$

$0 \le H = H^T \in \mathbb{R}^{n \times n}$

CONSIDER $M = \begin{pmatrix} F & -G \\ -H & -F^T \end{pmatrix}$

M IS <u>HAMILTONIAN</u> : $J^TM^TJ = -M$ where $J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$

ASSUME (F, G_1) STABILIZABLE $(G_1G_1^T = G,\ \text{rk}\, G_1 = \text{rk}\, G)$

(H_1, F) DETECTABLE $(H_1^TH_1 = H,\ \text{rk}\, H_1 = \text{rk}\, H)$

$Mx = \lambda x$

THEN $\Lambda(M) = \{-\lambda_1, \cdots, -\lambda_n, \lambda_1, \cdots, \lambda_n\}$; $\text{Re}\,\lambda_i >$

$$\begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}^T M \overset{\text{ORTHOG.}}{\begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}} = \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} \quad ; \ \Lambda(S_{11}) \subseteq \text{LHP}$$

$\begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix}$ SPANS STABLE EIGENSP. ; $X = U_{21}U_{11}^{-1} = X^T \geq 0$

Laub, 1978

DISCRETE - TIME ALGEBRAIC RICCATI EQUATION

$$F^TXF - X - F^TXG_1(G_2 + G_1^TXG_1)^{-1}G_1^TXF + H = 0$$

WITH $0 \leq H = H^T \in \mathbb{R}^{n \times n}$
$0 < G_2 = G_2^T \in \mathbb{R}^{m \times m}$; $G := G_1G_2^{-1}G_1^T$

CONSIDER $M = \begin{pmatrix} F + GF^{-T}H & -GF^{-T} \\ -F^{-T}H & F^{-T} \end{pmatrix}$

M IS <u>SYMPLECTIC</u> : $J^TM^TJ = M^{-1}$

ASSUME (F, G_1) STABILIZABLE
(H_1, F) DETECTABLE

$Mx = \lambda x$
THEN $\Lambda(M) = \{ \frac{1}{\lambda_1}, \cdots, \frac{1}{\lambda_n}, \lambda_1, \cdots, \lambda_n \}$; $|\lambda_i| > 1$

$$\begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}^T M \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix}; \quad \Lambda(S_{11}) \text{ INSIDE UNIT CIRCLE}$$

$\begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix}$ SPANS STABLE EIGENSPACE

$$X = U_{21}U_{11}^{-1} = X^T \geq 0$$

Laub, 1978

---

2. GENERALIZED EIGENVALUE PROBLEMS

(DISCRETE-TIME CASE ; MORE GENERAL CASE LATER) Laub, 1978

CONSIDER GENERALIZED EVAL. / EVEC. PROBLEM

$$Nx = \lambda Lx$$

WHERE $L = \begin{pmatrix} I & G \\ 0 & F^T \end{pmatrix}$, $N = \begin{pmatrix} \overset{\text{SINGULAR}}{F} & 0 \\ -H & I \end{pmatrix}$

NOTE : $L^{-1}N = M$

GEN. EVALS. : $\underbrace{0, \cdots, 0, \frac{1}{\lambda_1}, \cdots, \frac{1}{\lambda_k}}_{\text{STABLE}}, \underbrace{\lambda_1, \cdots, \lambda_k}_{|\lambda_i| > 1}, \infty, \cdots, \infty$ GENERALIZED SYMPLECTIC PROPERTY

$QLZ = \begin{pmatrix} L_{11} & L_{12} \\ 0 & L_{22} \end{pmatrix}$, $QNZ = \begin{pmatrix} N_{11} & N_{12} \\ 0 & N_{22} \end{pmatrix}$

GEN. EVALS. OF $N_{11} - \lambda L_{11}$ ARE STABLE

$\begin{pmatrix} Z_{11} \\ Z_{21} \end{pmatrix}$ SPANS STABLE EIGENSPACE

$X = Z_{21}Z_{11}^{-1} = X^T \geq 0$ SOLVES DISCR.-TIME ARE

DEFN. : $N-\lambda L \ \left(\in \mathbb{R}^{2n\times 2n}\right)$ IS A **SYMPLECTIC PENCIL** IF

$$NJN^T = LJL^T \qquad \left[\omega(Nx,Ny)-\omega(Lx,Ly)=0\right]$$

THEOREM : If λ is a gen. eval. of a symplectic pencil $N-\lambda L$ then so also is $\frac{1}{\lambda}$ with the same multiplicity.

PROOF : $\pi(\lambda) := \det(N-\lambda L)$

$$\begin{aligned}
[\pi(\lambda)]^2 &= \det (N-\lambda L) J (N-\lambda L)^T \\
&= \det\big(\underbrace{NJN^T}_{=LJL^T}-\lambda LJN^T-\lambda NJL^T+\lambda^2 \underbrace{LJL^T}_{=NJN^T}\big) \\
&= \det(\lambda N-L)J(\lambda N-L)^T \\
&= \det(\lambda I)\det(N-\tfrac{1}{\lambda}L)J(N-\tfrac{1}{\lambda}L)^T\det(\lambda I) \\
&= \left[\lambda^{2n}\pi(\tfrac{1}{\lambda})\right]^2
\end{aligned}$$

$$\Rightarrow \pi(\lambda) = \pm \lambda^{2n}\pi(\tfrac{1}{\lambda}) \qquad \blacksquare$$

EXAMPLE: $L = \begin{pmatrix} I & G \\ 0 & F^T\end{pmatrix}$, $N = \begin{pmatrix} F & 0 \\ -H & I\end{pmatrix}$

$$NJN^T = LJL^T = \begin{pmatrix} 0 & F \\ -F^T & 0\end{pmatrix}$$

DEFN. : $N-\lambda L \ \left(\in \mathbb{R}^{2n\times 2n}\right)$ IS A **HAMILTONIAN PENCIL** IF

$$LJN^T = -NJL^T \quad \left(= (LJN^T)^T\right)$$

$$\left[\omega(Nx,Ly)+\omega(Lx,Ny)=0\right]$$

THEOREM : If λ is a gen. eval. of a Hamiltonian pencil $N-\lambda L$ then so also is $-\lambda$ with the same multiplicity.

$$\begin{aligned}
\text{PROOF : } [\pi(\lambda)]^2 &= \det(N-\lambda L)J(N-\lambda L)^T \\
&= \det\big(NJN^T-\lambda\underbrace{LJN^T}_{=-NJL^T}-\lambda NJL^T+\lambda^2 LJL^T\big) \\
&= \det(NJN^T+\lambda^2 LJL^T) \\
&= [\pi(-\lambda)]^2
\end{aligned}$$

$$\Rightarrow \pi(\lambda) = \pm\pi(-\lambda)$$

EIGENVECTORS : y = LEFT. EVEC. OF EVAL. λ OF A SYMPLECTIC [HAMILTONIAN] PENCIL $N-\lambda L$

$\Rightarrow JL^Ty$ = RIGHT EVEC. OF $\frac{1}{\lambda}$ $[-\lambda]$

RELATIONSHIPS BETWEEN HAMILTONIAN AND SYMPLECTIC PENCILS

THEOREM : IF $N-\lambda L$ IS A SYMPLECTIC PENCIL THEN
$(\alpha N+\beta L)-\lambda(\gamma N+\delta L)$ IS A HAMILTONIAN PENCIL
IFF $\beta = \pm\alpha$, $\delta = \mp\gamma$

THEOREM : IF $N-\lambda L$ IS A HAMILTONIAN PENCIL THEN
$(\alpha N+\beta L)-\lambda(\gamma N+\delta L)$ IS A SYMPLECTIC PENCIL
IFF $\gamma = \pm\alpha$, $\delta = \mp\beta$

COROLLARY : IF $N-\lambda L$ IS HAMILTONIAN [SYMPLECTIC]
THEN $(N\pm L)-\lambda(N\mp L)$ IS SYMPLECTIC [HAMILTONIAN]

COROLLARY : H HAMILTONIAN $\Rightarrow$ $(H-I)^{-1}(H+I)$ SYMPLECTIC
S SYMPLECTIC $\Rightarrow$ $(S+I)^{-1}(S-I)$ HAMILTONIAN

e^H SYMPLECTIC IF H HAMILTONIAN

- BILINEAR TRANSFORMATIONS OF EIGENVALUES BUT EIGENVECTORS (INVARIANT SUBSPACES, REDUCING SUBSPACES) ARE PRESERVED

---

3. ORTHOGONAL SYMPLECTIC REDUCTIONS

THEOREM : Let T be symplectic and M Hamiltonian [symplectic].

Then $T^{-1}MT$ is Hamiltonian [symplectic].

THEOREM : $\exists$ orthogonal and symplectic $U \ni$ — Paige + VanLoan 1981

$$U^TMU = \begin{pmatrix} S_{11} & S_{12}^{\,=S_{12}^T} \\ 0 & -S_{11}^T \end{pmatrix}$$, Hamiltonian (+ symplectic analog)

CONSTRUCTIVE PROCEDURE FOR DETERMINING U ?

THEOREM : Let U be just orthogonal as before with $\begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix}$ spanning the stable eigenspace. Then — L + Lee 1982

$$U' = \begin{pmatrix} U_{11} & -U_{21} \\ U_{21} & U_{11} \end{pmatrix}$$ is orthogonal and symplectic.

PROOF :
$$\begin{pmatrix} U_{11}^T & U_{21}^T \\ -U_{21}^T & U_{11}^T \end{pmatrix}\begin{pmatrix} U_{11} & -U_{21} \\ U_{21} & U_{11} \end{pmatrix} = \begin{pmatrix} U_{11}^TU_{11}+U_{21}^TU_{21} & U_{21}^TU_{11}-U_{11}^TU_{21} \\ \underbrace{U_{11}^TU_{21}-U_{21}^TU_{11}}_{=0 \text{ since } X=U_{21}U_{11}^{-1} \text{ symm.}} & \underbrace{U_{11}^TU_{11}+U_{21}^TU_{21}}_{=I \text{ since orig. } U \text{ was orthog.}} \end{pmatrix} = I$$

$\Rightarrow U'$ orthog.

$$J^{-1}U'^TJ = U'^T = U'^{-1} \Rightarrow U' \text{ symplectic}$$

THEOREM : Let Z be symplectic and $N - \lambda L$ a Hamiltonian [symplectic] pencil. Let Q be arbitrary. Then $QNZ - \lambda QLZ$ is a Hamiltonian [symplectic] pencil.

PROOF : Verify definitions.

SCHUR-SYMPLECTIC PENCIL
SCHUR-HAMILTONIAN PENCIL
cf. L., 1982

QLZ , QNZ REGULAR ORTHOG. REDUC. OF GN. EVAL./EVEC. PROB.

$$Z' = \begin{pmatrix} Z_{11} & -Z_{21} \\ Z_{21} & Z_{11} \end{pmatrix}$$

QLZ' , QNZ' ORTHOG. SYMPLECTIC REDUC.

⟶ ANALOGOUS RESULTS

CONDITIONING : via SVD of Z' vs. $K(Z_{11})$ } MORE LATER

SYMPLECTIC ORTHOGONAL : $\begin{pmatrix} C & S \\ -S & C \end{pmatrix}$, $C + jS$ UNITARY

HAMILTONIAN SKEW : $\begin{pmatrix} A & S \\ -S & A \end{pmatrix}$,
A SKEW-SYMMETRIC
S SYMMETRIC
$e^{\begin{pmatrix} A & S \\ -S & A \end{pmatrix}}$ SYMPL. ORTHOG.

IF $H = \begin{pmatrix} A & S_1 \\ S_2 & -A^T \end{pmatrix}$, S_i SYMMETRIC , IS AN ARBITRARY HAMILTONIAN MATRIX AND A IS WRITTEN $A = \underset{\substack{\uparrow \\ \text{STRICTLY} \\ \text{LOWER TRIANG.}}}{L} + R \nwarrow$ UPPER TRIANG.

THEN

$$H = \underbrace{\begin{pmatrix} L - L^T & -S_2 \\ S_2 & L - L^T \end{pmatrix}}_{\text{HAMILTONIAN SKEW}} + \underbrace{\begin{pmatrix} R + L^T & S_1 + S_2 \\ 0 & -R^T - L \end{pmatrix}}_{\text{HAMILTONIAN UPPER TRIANGULAR}}$$

IF M IS SYMPLECTIC THEN $M = QR$ WHERE
Q = SYMPLECTIC ORTHOGONAL
R = SYMPLECTIC UPPER TRIANGULAR = $\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{11}^{-T} \end{pmatrix}$, $R_{11} R_{12}^T$ SYMM.

ANALAGOUS RESULTS FOR HAMILTONIAN + SYMPLECTIC PENCILS

Campbell

4. GENERALIZED RICCATI EQUATIONS (CONTS. TIME CASE)

EXTENDED FORMULATIONS

S Lee, PhD Thesis 1982

$$\text{Min} \quad J(u) = \int_{t_0}^{t_f} \frac{1}{2}\left[x^T Q x + 2x^T S u + u^T R u\right] dt$$

$$\text{s.t.} \quad E\dot{x} = Fx + Gu \quad ; \quad x(t_0) = x_0$$

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0 \quad , \quad \overset{t_f = +\infty}{(\text{STABILIZABLE, DETECTABLE})}$$

E NONSINGULAR BUT MAY BE : — NEAR SINGULAR or
— SPARSE

(E SINGULAR)

$$M\ddot{q} + D\dot{q} + Kq = Bu$$

→ IMPLICIT STATE MODELS

SCALAR HAMILTONIAN :

$$H = \frac{1}{2}\left[x^T Q x + 2x^T S u + u^T R u\right] + p^T\left[Fx + Gu - E\dot{x}\right]$$

$$\frac{\partial H}{\partial p} - \frac{d}{dt}\left(\frac{\partial H}{\partial \dot{p}}\right) = 0$$

$$\frac{\partial H}{\partial x} - \frac{d}{dt}\left(\frac{\partial H}{\partial \dot{x}}\right) = 0$$

$$\frac{\partial H}{\partial u} - \frac{d}{dt}\left(\frac{\partial H}{\partial \dot{u}}\right) = 0$$

$$\begin{pmatrix} E & 0 & 0 \\ 0 & E^T & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} \dot{x} \\ \dot{p} \\ \dot{u} \end{pmatrix} = \begin{pmatrix} F & 0 & G \\ -Q & -F^T & -S \\ S^T & G^T & R \end{pmatrix}\begin{pmatrix} x \\ p \\ u \end{pmatrix}$$

SUPPOSE — TEMPORARILY — THAT R IS NONSINGULAR .

SOLVE FOR u :

$$u = -R^{-1}S^T x - R^{-1}G^T p$$

SUBST. IN x, p EQUATIONS AND MAKE THE RICCATI SUBST.

$$p(t) = \Pi(t) E x(t)$$

THEN GET <u>GENERALIZED RICCATI EQUATION</u>

$$-E^T \underset{z}{\dot{\Pi}} E = (F - GR^{-1}S^T)^T \Pi E + E^T \Pi (F - GR^{-1}S^T) - E^T \Pi G R^{-1} G^T \Pi E + Q - SR^{-1}S^T$$

$$= F^T \Pi E + \overset{2}{E^T} \Pi F - \overset{?}{(} E^T \Pi G + S) R^{-1} (G^T \overset{z}{\Pi} E + S^T) \overset{z}{+} Q$$

ALGEBRAICALLY,

$$\begin{pmatrix} I & 0 & -GR^{-1} \\ 0 & I & SR^{-1} \\ 0 & 0 & I \end{pmatrix}\left[\begin{pmatrix} F & 0 & G \\ -Q & -F^T & -S \\ S^T & G^T & R \end{pmatrix} - \lambda\begin{pmatrix} E & 0 & 0 \\ 0 & E^T & 0 \\ 0 & 0 & 0 \end{pmatrix}\right]$$

$$= \left(\begin{array}{cc:c} F - GR^{-1}S^T & -GR^{-1}G^T & 0 \\ SR^{-1}S^T - Q & -(F - GR^{-1}S^T)^T & 0 \\ \hdashline S^T & G^T & R \end{array}\right) - \lambda\left(\begin{array}{cc:c} E & 0 & 0 \\ 0 & E^T & 0 \\ \hdashline 0 & 0 & 0 \end{array}\right)$$

IF R IS SINGULAR, CAN DO COMPRESSION (ORTHOGONAL)

Van Dooren, 1981

$$\overset{2n\times 2n}{\begin{pmatrix} U_1 & \vdots & U_2 \\ \cdots & & \cdots \\ U_3 & \vdots & U_4 \end{pmatrix}}\begin{pmatrix} G \\ -S \\ \cdots \\ R \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \tilde{R} \end{pmatrix} \quad , \quad \tilde{R} \in \mathbb{R}^{m\times m}_m$$

THEN WORK WITH

$$\Big[\underbrace{U_1\begin{pmatrix} F & 0 \\ -Q & -F^T \end{pmatrix} + U_2\,(S^T, G^T)}_{\text{THM.}}\Big] - \lambda\Big[U_1\begin{pmatrix} E & 0 \\ 0 & E^T \end{pmatrix}\Big]$$

$$= \quad U_1 M - \lambda U_1 \begin{pmatrix} E & 0 \\ 0 & E^T \end{pmatrix}$$

$$\text{where} \quad M := \begin{pmatrix} F - GR^{-1}S^T & -GR^{-1}G^T \\ SR^{-1}S^T - Q & -(F-GR^{-1}S^T)^T \end{pmatrix}$$

$$:= A - \lambda B$$

<u>NOTE</u>: (A,B) IS NOT A [redacted] HAMIL. PENCIL BUT ITS GEN. EVALS. DO HAVE THE HAMILTONIAN SYMMETRY

<u>Pf</u>: $$\begin{pmatrix} E^{-1} & 0 \\ 0 & E^{-T} \end{pmatrix} M = \begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix}^{-1}\left[\begin{pmatrix} I & 0 \\ 0 & E^{-T} \end{pmatrix} M \begin{pmatrix} E^{-1} & 0 \\ 0 & I \end{pmatrix}\right]\begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix}^{-1}\underbrace{\begin{bmatrix} (F-GR^{-1}S^T)E^{-1} & -GR^{-1}G^T \\ E^{-T}(SR^{-1}S^T-Q)E^{-1} & -E^{-T}(F-GR^{-1}S^T)^T \end{bmatrix}}_{:=\hat{M}\text{ , Hamiltonian}}\begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix}$$

<u>THEOREM</u>: *The generalized Riccati solution Π is the "regular" Riccati solution corresponding to the system*

$$\begin{pmatrix} \dot{\hat{x}} \\ \dot{\hat{p}} \end{pmatrix} = \hat{M}\begin{pmatrix} \hat{x} \\ \hat{p} \end{pmatrix}$$

GENERALIZED ALGEBRAIC RICCATI EQUATION — SUMMARY

$$(F-GR^{-1}S^T)^T\Pi E + E^T\Pi(F-GR^{-1}S^T) - E^T\Pi GR^{-1}G^T\Pi E + Q - SR^{-1}S^T = 0$$

OR

$$F^T\Pi E + E^T\Pi F - (E^T\Pi G + S)R^{-1}(G^T\Pi E + S^T) + Q = 0$$

FIND ORTHOGONAL Y AND Z $\ni$

$$Y\begin{pmatrix} E & 0 \\ 0 & E^T \end{pmatrix}Z \quad \text{AND} \quad Y\begin{pmatrix} F-GR^{-1}S^T & -GR^{-1}G^T \\ SR^{-1}S^T - Q & -(F-GR^{-1}S^T)^T \end{pmatrix}Z$$

ARE ∇ AND ∇ WITH STABLE GEN. EVALS. IN UPPER LEFT.

THEN $\quad \Pi = V_{21}V_{11}^{-1}$ WHERE $V = \begin{pmatrix} E & 0 \\ 0 & I \end{pmatrix}Z$

OPTIMAL CONTROL:

$$u^* = -R^{-1}S^Tx - R^{-1}G^Tp$$
$$= -R^{-1}\left[S^T + G^T\Pi E\right]x$$
$$= -R^{-1}\left[S^T + G^TZ_{21}Z_{11}^{-1}\right]x$$

REDUCING SUBSPACE:
$$\begin{pmatrix} I \\ X \\ K \end{pmatrix}$$
OF EXTENDED PENC

+ ANALOGOUS RESULTS WITHOUT EXPLICIT R^{-1}.

$$E^T XGXE + F^T XE + E^T XF + H = 0$$

$$\begin{pmatrix} F & G \\ -H & -F^T \end{pmatrix} - \lambda \begin{pmatrix} E & 0 \\ 0 & E^T \end{pmatrix}$$

NOT HAMILTONIAN PENCIL BY PREVIOUS DEFN.

BUT USE $J = \begin{pmatrix} 0 & E^{-1} \\ -E^{-T} & 0 \end{pmatrix}$ (AND IN DEFN. OF ω)

THEN $LJN^T = -NJL^T \quad \left(= \begin{pmatrix} G & -F \\ -F^T & H \end{pmatrix} \right)$

DISCRETE-TIME VERSIONS

- APPLICS. TO FILTERING ; COVAR. PROPAGATION WITH SINGULAR TRANSITION MATRICES, SINGULAR MEAS. NOISE COVAR., etc.

STEADY-STATE COVARIANCE EQUATION $(\Sigma_{k+1|k})$:

$$E\Sigma E^T = F\Sigma F^T - (F\Sigma H + GS)(H^T\Sigma H + R)^{-1}(F\Sigma H + GS)^T + GQG^T$$

$$E x_{k+1} = F x_k + \Gamma u_k + G w_k$$

$$z_k = H^T x_k + v_k$$

$$E \begin{pmatrix} w_k \\ v_k \end{pmatrix} (w_j^T, v_j^T) = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{kj}$$

EQUIVALENTLY :

$$E\Sigma E^T = \hat{F}\Sigma\hat{F}^T - \hat{F}\Sigma H(H^T\Sigma H + R)^{-1} H^T \Sigma \hat{F}^T$$
$$+ G(Q - SR^{-1}S^T)G^T$$

where $\hat{F} := F - GSR^{-1}H^T$

CONSIDER THE PENCILS:

$$\lambda \begin{pmatrix} E^T & 0 & 0 \\ 0 & F & 0 \\ 0 & -H^T & 0 \end{pmatrix} - \begin{pmatrix} F^T & 0 & H \\ -GQG^T & E & -GS \\ S^T G^T & 0 & R \end{pmatrix}$$

OR $\lambda \begin{pmatrix} E^T & HR^{-1}H^T \\ 0 & \hat{F} \end{pmatrix} - \begin{pmatrix} \hat{F}^T & 0 \\ -G(Q-SR^{-1}S^T)G^T & E \end{pmatrix}$

- ALSO WORKS FOR TIME-VARYING CASE

5. NONSYMMETRIC RICCATI EQUATIONS

GENERAL LINEAR TWO-PT. BOUNDARY VALUE PROBLEM :

$$\left.\begin{aligned} E\dot{v} &= Av + Bw \\ -F\dot{w} &= Cv + Dw \end{aligned}\right\} \text{ PLUS BOUNDARY CONDITIONS}$$

RICCATI TRANSFORMATION / INVARIANT IMBEDDING Vandevender, 1977

$$\rightarrow \quad E\dot{X}(t)F = B + AXF + EXD + EXCXF$$

PLUS INITIAL CONDITION

ALGEBRAIC EQUATION : Laub, 1982

$$0 = B + AXF + EXD + EXCXF$$

GENERALIZED EIGENVAL. / EIGENVEC. PROBLEM :

$$\begin{pmatrix} A & B \\ -C & -D \end{pmatrix} - \lambda \begin{pmatrix} E & 0 \\ 0 & F \end{pmatrix}$$

6. CONDITIONING

- $\kappa(U_{11})$ or $\kappa(Z_{11})$ or $\kappa(V_{11})$

  $XU_{11} = U_{21}$

  U_{11} singular if unstabilizable model

- SINGULAR VALS. OF $Z = \begin{pmatrix} Z_{11} & -Z_{21} \\ Z_{21} & Z_{11} \end{pmatrix} \xrightarrow[]{\text{orth. equiv.}} \begin{pmatrix} \Sigma & \Delta \\ -\Delta & \Sigma \end{pmatrix}$

  Paige + VanLoan 1981

- FIRST-ORDER PERTURBATION THEORY

  e.g. Byers 1983

$$F^TX + XF - XGX + H = 0$$

$$M = \begin{pmatrix} F & -G \\ -H & -F^T \end{pmatrix}$$

$$\text{Sep}[A,B] := \inf_{\|P\|=1} \|PA - BP\|$$

$$\text{Cond} = \frac{(1 + \|\hat{X}\|^2)\,\|M\|}{\|\hat{X}\|\,\text{Sep}[(F - G\hat{X})^T, -(F - G\hat{X})]}$$

$$\text{Cond} \approx \frac{(\|H\| + 2\|F\|\|\hat{X}\| + \|G\|\|\hat{X}\|^2)}{\|\hat{X}\|\,\text{Sep}[(F - G\hat{X})^T, -(F - G\hat{X})]}$$

e.g. Arnold 1983

$$F^TXE + E^TXF - E^TXGXE + H = 0$$

$$\text{cond} \approx \frac{\kappa(E)\,\kappa(E^T)\,\|H\|}{\|\hat{X}\|\cdot\|EH\cdot HE^T\|\cdot \text{sep}\left[(FE^{-1}-G\hat{X}),-(FE^{-1}-G\hat{X})^T\right]}$$

- MUST REFLECT :
  - near unstabilit.
  - small sep. of cl.-loop spectrum
  - near sing. of R (contr.)
  - near sing. of E

- NEWTON'S METHOD

$$\bar{A}_k^T X_{k+1} E + E^T X_{k+1} \bar{A}_k + \bar{C}_k = 0 \qquad \text{CONTS.}$$

$$E^T X_{k+1} E - \bar{A}_k^T X_{k+1} \bar{A}_k + \bar{C}_k = 0 \qquad \text{DISCRETE}$$

$$X_k = \begin{cases} \text{NEW soln.} \\ \text{PERTURBATION} \end{cases}$$

CAN IMPROVE ACCURACY OF COMPUTED GENERALIZED EIGENPROBLEM SOLUTION — OFTEN SUBSTANTIALLY

- GENERALIZATION OF MATRIX SIGN FUNCTION Gardiner + L. 1985

- SCALING

- BALANCING Ward 1981
  $A-\lambda B$

7. SOFTWARE - RICPACK

WHY ?

RICPACK : A SOFTWARE PACKAGE (Fortran) FOR SOLVING GENERALIZED ALGEBRAIC RICCATI EQUATIONS (Arnold + Laub, CDC 1983)

FEATURES OF THE SOFTWARE AND ITS CAPABILITIES

- PFORT
  - SUBROUTINES

- INTERACTIVE DRIVER
  - F10
  - → F77, C / UNIX

- MANY CONVENIENT DEFAULT OPTIONS
  - $E = I$
  - $S = 0$
  - $Q = I$, $R = I$ OR $R = \text{diagonal}$

- PARALLEL DEVELOPMENT FOR BOTH $\begin{cases}\text{CONTINUOUS-TIME} \\ \text{DISCRETE-TIME}\end{cases}$ GARE 's .

- <u>GENERALIZED RICCATI EQUATIONS</u>

e.g. $$\min_u \frac{1}{2}\int_0^{+\infty}\left[x^TQx + 2x^TSu + u^TRu\right]dt$$

s.t. $$E\dot{x} = Ax + Bu$$

CORRESPONDING GARE IS :

$$\hat{A}^TXE + E^TX\hat{A} - E^TXBR^{-1}B^TXE + Q - SR^{-1}S^T = 0$$

$$\hat{A} := A - BR^{-1}S^T$$

OPTIMAL FEEDBACK : $u = Kx$ WHERE $K = -R^{-1}(B^TXE + S^T)$

- GET BASIS FOR THE STABLE DEFLATING SUBSPACE OF

$$\begin{pmatrix} \hat{A} & -BR^{-1}B^T \\ -Q+SR^{-1}S^T & -\hat{A}^T \end{pmatrix} - \lambda\begin{pmatrix} E & 0 \\ 0 & E^T \end{pmatrix}$$

- SPECIAL CASE : $E = I$, $S = 0$ (cf. PREV. PAGE)

- TO AVOID R^{-1} CAN WORK WITH THE EXTENDED PENCIL

$$\begin{pmatrix} A & 0 & B \\ -Q & -A^T & -S \\ S^T & B^T & R \end{pmatrix} - \lambda\begin{pmatrix} E & 0 & 0 \\ 0 & E^T & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- DISCRETE-TIME PROBLEMS

GARE :

$$E^TXE = A^TXA - (A^TXB+S)(B^TXB+R)^{-1}(A^TXB+S)^{-1} + Q$$

$$= \hat{A}^TX\hat{A} - \hat{A}^TXB\left(B^TXB+R\right)^{-1}B^TX\hat{A} + Q - SR^{-1}S^T$$

$$\left(\hat{A} := A - BR^{-1}S^T\right)$$

- IN THIS CASE, CONSIDER THE PENCIL

$$\begin{pmatrix} \hat{A} & 0 \\ -Q+SR^{-1}S^T & E^T \end{pmatrix} - \lambda\begin{pmatrix} E & BR^{-1}B^T \\ 0 & \hat{A}^T \end{pmatrix}$$

OR, IF R^{-1} IS TO BE AVOIDED, THE EXTENDED PENCIL

$$\begin{pmatrix} A & 0 & B \\ -Q & E^T & -S \\ S^T & 0 & R \end{pmatrix} - \lambda\begin{pmatrix} E & 0 & 0 \\ 0 & A^T & 0 \\ 0 & -B^T & 0 \end{pmatrix}$$

- FILTER EQUATIONS BY DUALITY

- CALCULATION OF THE STABILIZING, DESTABILIZING, OR JUST ANY SOLUTION TO A GARE

- WARD'S BALANCING FOR THE GENERALIZED EIGENVALUE PROBLEM
  - HEURISTIC SCALING
  - IMPROVES ACCURACY

- [CO-ORDINATE / SYSTEM BALANCING]

- DIRECT HANDLING OF SINGULAR CONTROL WEIGHTING OR SINGULAR MEASUREMENT NOISE COVARIANCE

- DIRECT HANDLING OF CROSS-WEIGHTING OR NOISE CORRELATION

- PROVISION FOR ROBUSTNESS RECOVERY PROCEDURES

- SPECTRAL FACTORIZATION, H_∞ - OPTIMIZATION APPLICATION

- ITERATIVE REFINEMENT OR NEW SOLUTIONS EFFICIENTLY FOR SMALL PROBLEM PARAMETER PERTURBATIONS
  - NEWTON'S METHOD / SYLVESTER EQUATIONS

    $$\mathcal{A}^T X \mathcal{E} + \mathcal{E}^T X \mathcal{A} = \mathcal{S} \quad , \text{etc.}$$

  - NEW SOLUTION OR CHANGE IN SOLUTION

- MODEL UNSTABILIZABILITY DETECTION
  - UNIQUE STABILIZING SOLUTION FOR STABILIZABLE MODELS WITH UNDETECTABLE MODES

- RELATIVE RESIDUAL CALCULATIONS

  $$\frac{\|\text{RESIDUAL}\|_1}{\|X\|_1}$$

- CONDITION ESTIMATES
  - INFORMATION ON EFFECTS OF :
    - — NEAR-UNSTABILIZABILITY OF MODEL
    - — SMALL (REL.) SEPARATION OF CLOSED-LOOP SPECTRUM FROM $\operatorname{Re} z = 0$ OR $|z| = 1$
    - — NEAR-SINGULARITY OF R
    - — NEAR-SINGULARITY OF E
  - ??

HESSENBERG FORMS IN LINEAR SYSTEMS THEORY

Alan J. Laub
Arno Linnemann
(Universität Bremen)

ECE DEPARTMENT
UNIV. OF CALIFORNIA
SANTA BARBARA, CA 93106

MTNS - 85
STOCKHOLM

---

1. INTRODUCTION

DEFN.: $A \in \mathbb{R}^{n \times n}$ IS UPPER HESSENBERG IF $\alpha_{ij} = 0$ for $i - j \geq 2$; $i \in \underline{n}$, $j \in \underline{n}$.

$$\begin{matrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{matrix}$$

- SIMILARLY LOWER HESSENBERG; ALSO FOR $\mathbb{C}^{n \times n}$
- $\mathcal{H} := \{ H \in \mathbb{R}^{n \times n} : H$ IS UPPER HESSENBERG $\}$
  $\mathcal{H} \subseteq \mathbb{R}^{n \times n}$
- $\forall A \in \mathbb{R}^{n \times n}$, $\exists T \in \mathbb{R}_n^{n \times n} \ni T^{-1} A T \in \mathcal{H}$

  IN FACT, A IS SIMILAR TO INFINITELY MANY U.H. MATRICES; e.g. $D^{-1} H D$, $H \in \mathcal{H}$
  $D = \diagdown$

- ALGORITHMS

T = PROD. STABILIZED ELEMENTARY (EISPACK : ELMHES)

T = PROD. SIMPLE ORTHOGONAL (EISPACK : ORTHES)
(e.g. HOUSEHOLDER REFLECTIONS)

<u>FINITE</u> PROCESS

DEFN. : $H \in \mathcal{H}$ IS UNREDUCED IF $h_{i+1,i} \neq 0$, $i \in \underline{n-1}$,
I.E., H HAS NO ZERO SUBDIAGONAL ENTRIES

$$\begin{array}{ccc|cc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ \hdashline 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{array}$$

- DECOMPOSITION INTO UNREDUCED HESSENBERG MATRICES
- USEFUL IN EIGENPROBLEM

- APPLICATIONS

1. EIGENPROBLEM FOR $A \in \mathbb{R}^{n \times n}$

QR STEP $O(n^3)$, FULL ALG. $O(n^4)$

$$\underset{\nwarrow O(n^3)}{A \longrightarrow H \in \mathcal{H}} \; ; \quad \text{QR STEP } O(n^2), \text{ FULL ALG. } O(n^3)$$

2. FREQUENCY RESPONSE

e.g. COMPUTE $G(j\omega) = C(j\omega I - A)^{-1}B$ $\quad (j = e^{\frac{\pi}{2}i})$

FOR MANY VALUES OF ω

FIND $T \ni T^{-1}AT \in \mathcal{H}$.

$$G(j\omega) = CT(j\omega I - T^{-1}AT)^{-1}T^{-1}B$$

$$(j\omega I - T^{-1}AT) \in \mathcal{H} \quad , \forall \omega$$

$\nwarrow$ CAN BE LU- (OR QR) FACTORED VERY STABLY IN $O(n^2)$ FLOPS

$\Rightarrow$ $G(j\omega)$ COMPUTED IN $O(n^2)$ FLOPS FOR EACH NEW VALUE OF ω.

- NUMERICAL REASONS TO PREFER HESSENBERG REDUCTION BY ORTHOGONAL SIMILARITIES

$$U^TAU = H \in \mathcal{H}$$

$U = (u_1, \cdots, u_n)$ ORTHOGONAL ; $\{u_i\}$ ORTHONORMAL

COMPARE FIRST COLUMNS IN $AU = UH$:

$$Au_1 = h_{11}u_1 + h_{21}u_2$$

$\Rightarrow u_1^TAu_1 = h_{11}$ u_1 IS ARBITRARY UNIT VECTOR

$$\|h_{21}u_2\| = |h_{21}| = \|Au_1 - h_{11}u_1\|$$

$= \|\cdot\|_2$ $$= \|Au_1 - (u_1^TAu_1)u_1\|$$

IF $h_{21} > 0$ IS CHOSEN, h_{21} IS UNIQUELY DETERMINED AND SO IS u_2.

IF $h_{21} = 0$, u_2 IS AN ARB. UNIT VECTOR ORTHOGONAL TO u_1 (AND u_1 IS AN EIGENVECTOR OF A FOR EIGENVAL. h_{11})

COMPARE SECOND COLUMNS : $Au_2 = h_{12}u_1 + h_{22}u_2 + h_{32}u_3$

BY ORTHONORMALITY : $h_{12} = u_1^TAu_2$, $h_{22} = u_2^TAu_2$

$$\|h_{32}u_2\| = |h_{32}| = \|Au_2 - h_{12}u_1 - h_{22}u_2\|$$

$$\vdots$$

- IN GENERAL, ONCE u_1 IS FIXED, THEN H WITH $h_{k+1,k} > 0$ IS DETERMINED UNIQUELY.

  IF $h_{k+1,k} = 0$ THEN u_{k+1} IS AN ARBITRARY UNIT VECTOR ORTHOGONAL TO $u_1, \cdots, u_k$ AND THE REMAINING $n-k$ COLUMNS OF H ARE NONUNIQUE

- IMPLICIT Q THEOREM (Golub + VanLoan, p.223)

- EISPACK (Martin + Wilkinson) CHOOSES $u_1 = e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

- HOW TO USE THE FREEDOM IN u_1 ?

2. SYSTEM HESSENBERG FORMS

LET $A \in \mathbb{R}^{n\times n}$, $u_1 \in \mathbb{R}^n$ WITH $\|u_1\|_2 = 1$ $\ni$ (A, u_1) IS CONTROLLABLE. THEN $\exists$ UNIQUELY DETERMINED VECTORS $u_2, \cdots, u_n$ $\ni$ $U := (u_1, \cdots, u_n)$ IS ORTHOGONAL AND $U^TAU \in \mathcal{H}$. (POS. SUBDIAGONAL)

SINGLE-INPUT CASE

$(A, b) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$, CONTROLLABLE

CHOOSE $\frac{1}{\|b\|} b$ AS THE FIRST COLUMN OF AN ORTHOGONAL MATRIX U WHICH TRANSFORMS A TO HESSENBERG FORM.

CLEARLY, $U^Tb = \|b\| e_1$

DEFN.: $(H_A, H_b) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$ IS IN CONTROLLABLE SYSTEM HESSENBERG FORM IF

$$H_b = \begin{pmatrix} h_{10} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad H_A = \begin{pmatrix} h_{11} & \cdots & \cdots & h_{1n} \\ h_{21} & \ddots & & \vdots \\ & \ddots & \ddots & \\ \mathbf{0} & & h_{n,n-1} & h_{nn} \end{pmatrix}$$

WITH $h_{i,i-1} > 0$, $i \in \underline{n}$

THEOREM: $\forall$ CONTROLLABLE (A, b), $\exists$ UNIQUE ORTHOGONAL U AND UNIQUE SYSTEM HESSENBERG FORM (H_A, H_b) $\ni$

$$(U^TAU, U^Tb) = (H_A, H_b)$$

- CONSTRUCTIVE PROOF; OTHER DIRECT PROOFS AVAILABLE
- CAN BE DONE WITH STANDARD SOFTWARE (e.g. ORTHES)

1. $(A, b) \longrightarrow (U_1^TAU_1, U_1^Tb)$

   U_1 = "STANDARD" HOUSEHOLDER $\ni$ $U_1^Tb = U_1b = \begin{pmatrix} \|b\| \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

2. $(U_1^TAU_1, U_1^Tb) \longrightarrow (U_2^TU_1^TAU_1U_2, U_2^TU_1^Tb) = (H_A, H_b)$

   U_2 = PRODUCT OF HOUSEHOLDERS TO REDUCE $U_1^TAU_1$ TO U.H. FORM

   ↑ EACH OF FORM $\begin{pmatrix} I_k & 0 \\ 0 & \tilde{U}_2 \end{pmatrix}$, $k \geq 1$

   $\Longrightarrow U_2^TU_1^Tb = U_1^Tb$

- LOWER HESSENBERG: $A = \begin{pmatrix} x & x & 0 & \cdots & 0 \\ x & x & x & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ x & x & x & \cdots & x \\ x & x & x & \cdots & x \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ x \end{pmatrix}$

  SPECIAL CASE: $A = \begin{pmatrix} 0 & 1 & & \mathbf{0} \\ & & \ddots & \\ & \mathbf{0} & & 1 \\ x & x & \cdots & x \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$

MULTI-INPUT CASE

- MANY AUTHORS HAVE PROPOSED A VARIETY OF HESSENBERG-TYPE STATE SPACE REPRESENTATIONS
- "SHAPES" OF THESE REPRESENTATIONS DETERMINED BY
  - KRONECKER INDICES : BLOCK HESSENBERG ; $H_{i,i-1}$ FULL ROW RANK

OR

  - HERMITE INDICES : HESSENBERG H_A ; VARIOUS SIMPLIFICATIONS OF H_B .

APPLICATIONS

- CONTROLLABILITY, OBSERVABILITY
- MINIMAL REALIZATION
- POLE PLACEMENT, OBSERVER DESIGN
- FREQUENCY RESPONSE
- $\mathcal{V}^*$
- MODEL REDUCTION

EXAMPLE

COMPUTE $g(j\omega) = c^T(j\omega I - A)^{-1} b$

W.L.O.G., $A \in \mathcal{H}$. REQUIRES APPROX. $n^2 + n$ FLOPS TO GET $g(j\omega)$ FOR EACH NEW VALUE OF ω .

IF (A, b) REDUCED TO <u>SYSTEM</u> HESSENBERG FORM , CAN SAVE $O(\frac{1}{2}n^2)$ FLOPS BY MEANS OF A "TRICK" :

x SOLVES $(j\omega I - A)x = b$ IF AND ONLY IF

$$\bar{x} := \begin{pmatrix} 1/x_n \\ -x_1/x_n \\ \vdots \\ -x_{n-1}/x_n \end{pmatrix} \quad \text{SOLVES} \quad (b, \tilde{A})\,\bar{x} = \tilde{a}$$

(CAN PROVE $x_n \neq 0$)
IF (A, b) CONTROLLABLE

WHERE $\tilde{A} \in \mathbb{C}^{n \times n-1}$ CONSISTS OF FIRST $n-1$ COLS. OF $(j\omega I - A)$
$\tilde{a} \in \mathbb{C}^n$ IS LAST COL. OF $(j\omega I - A)$

$(b, \tilde{A})$ IS UPPER TRIANGULAR $\Rightarrow$ $\bar{x}$ CAN BE FOUND IN $O(\frac{n^2}{2})$ FLOPS

AVOIDS EXTRA $\frac{1}{2}n^2$ FLOPS FOR LU FACTORIZATION OF U.H. MATRIX

3. SENSITIVITY OF SYSTEM HESSENBERG FORMS

CONSIDER ARBITRARY $A \in \mathbb{R}^{n \times n}$, $E \in \mathbb{R}^{n \times n}$

$\|E\|$ SMALL FOR SOME $\|\cdot\|$

LET H_A = A FIXED HESSENBERG FORM FOR A

CONSIDER PERTURBED MATRIX $A+E$ AND ORTHOGONAL U

$\ni \quad H_{A+E} := U^T(A+E)U \in \mathcal{H}$.

IS $\| H_A - H_{A+E} \|$ SMALL ?

SYSTEM HESSENBERG FORM : MAYBE NOT

"FREE" HESSENBERG FORM : ?

SCHUR FORMS : $\| S_A - S_{A+E} \|$ NEED NOT BE SMALL

$$A = S_A = \begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix}, \quad A+E = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \delta & & & 0 \end{pmatrix}$$

EXAMPLE 1 – SMALL SUBDIAGONAL ELEMENTS

$$A = \begin{pmatrix} 1 & & & & \\ \alpha & \ddots & & 0 & \\ & \ddots & \ddots & & \\ 0 & & \ddots & 1 & \\ & & & \alpha & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = e_1$$

$$A+E = \begin{pmatrix} 1 & & & & \\ \alpha & \ddots & & 0 & \\ & \ddots & \ddots & & \\ 0 & & \ddots & 1 & \\ \epsilon & & & \alpha & 0 \end{pmatrix}, \quad b = e_1$$

RETURNING THE PERTURBED A AND b TO SYSTEM HESSENBERG FORM

$$A_H = \begin{pmatrix} h_{11} & \cdots & \cdots & h_{1n} \\ h_{21} & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ 0 & & h_{n,n-1} & h_{nn} \end{pmatrix}, \quad b_H = b = e_1$$

WE FIND $$h_{n,n-1} = \alpha + \underbrace{(-1)^{n-1} \alpha^{2-n} \epsilon}_{\text{LARGE FOR SMALL } \alpha} + O(\epsilon^2)$$

IN THIS CASE $u_1 = e_1$ TO PRESERVE THE FORM OF b.

- FREE HESSENBERG FORM MAY BE NUMERICALLY PREFERRED TO SYSTEM HESSENBERG FORM FOR CERTAIN APPLICATIONS

- FIRST-ORDER PERTURBATION THEORY
  - ROLE OF $h_{i,i-1}$
  - SUGGESTS CONDITION NUMBER

4. HESSENBERG / TRIANGULAR FORMS

- ORTHOGONAL $U, V \ni (E, A) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ REDUCED TO

$$VEU \in \mathcal{H}$$
$$VAU \in \nabla^{n\times n}$$

- UNIQUELY DETERMINED BY FIRST ROW OF V OR COLUMN OF U

- FREEDOM CAN BE USED TO SIMPLIFY $B \in \mathbb{R}^{n\times m}$

  FREE H/T FORMS
  SYSTEM H/T FORMS

- APPLICATIONS

- SIMILAR SENSITIVITY ANALYSIS

7. MATHEMATICAL SOFTWARE AS A CAD COMPONENT

I. INTRODUCTION

- NUMERICAL ANALYSIS AND MATHEMATICAL SOFTWARE CRITICAL COMPONENTS OF CAD PACKAGE
- MATHEMATICAL SOFTWARE : an implementation on a computer of an algorithm for solving a mathematical problem
- NUMERICAL ANALYSIS, ALGORITHMS
- PERFORMANCE OF ALGORITHMS IN FINITE ARITHMETIC (FINITE PRECISION, FINITE RANGE)
- MAKES MATH SOFTWARE SOMETIMES SUBTLE AND SEEMINGLY COMPLICATED (e.g. solving a quadratic eqn.)
- IEEE FLOATING POINT STANDARD $P\underset{8}{7}54$ (8087)
- NUMERICAL STABILITY OF ALGORITHMS
- PROBLEM CONDITION AND CONDITION ESTIMATION
- RAPID AND FUNDAMENTAL CHANGES :
  - STANDARD COMPUTING ENVIRONMENTS VS. MICROPROCESSOR BASED, VLSI / PARALLEL ARCHITECTURES
  - LANGUAGES

- STATUS OF CONTROL ALGORITHMS
  - IMPROVING BUT MANY OPEN PROBLEMS
    e.g. CONDITION OF RICCATI EQNS.
  - MOST WORK FOR "SMALL", DENSE PROBLEMS ; $O(10^2)$
    LESS DONE FOR LARGE, SPARSE PROBLEMS (FCN. OF MODEL : AEROSPACE, PROCESS CONTROL, STRUCTURES)
  - MANY ALGORITHMS ARE NOT "CLEAN" ; BETTER ANALOGIES IN NUMER. ODE'S/PDE'S THAN NLA

- STATUS OF CONTROL SOFTWARE
  - AN IMMENSE NUMBER OF VARIETIES
  - NICE FEATURES IN SOME BUT WIDELY VARYING QUALITY
  - LARGELY FORTRAN BASED

- $$\boxed{\text{WELL-CONDITIONED PROBLEM}} + \boxed{\text{NUMERICALLY STABLE ALGORITHM}} + \boxed{\text{ROBUST SOFTWARE IMPLEMENTATION}} \rightarrow \boxed{\text{ACCURATE ANSWERS}}$$

II. MATHEMATICAL SOFTWARE

- OFTEN THE ONLY WAY TO REALLY COMMUNICATE AN ALGORITHM PROPERLY
- NOT "JUST PROGRAMMING"
- MUST BE USED INTELLIGENTLY ; EDUCATION GAP
- PROTOTYPE EXAMPLES
  - EISPACK, LINPACK, FUNPACK, MINPACK, ROSEPACK,
  - var. ODE, PDE CODES
  - ACM Trans. Math. Software
- SOFTWARE EVALUATION : a highly nontrivial task
  - a function of the operational specification
  - influence of compiler
  - numerical aspects (e.g. stability, condn. est.)
  - robustness
  - different implementations can have markedly different properties and behavior
- LANGUAGES
  - PFORT, FORTRAN 77, PASCAL, ADA, C
- HARDWARE

- SOME DESIRABLE CHARACTERISTICS OF "GOOD" SOFTWARE :
  - RELIABILITY
    - – amount and kind of testing and use ; certification ; corroboration
    - – greatly enchanced by widespread use
    - – function of algorithmic features
  - PORTABILITY
  - HIGH STANDARDS OF DOCUMENTATION AND STYLE
  - EASE OF USE ; ABILITY OF THE USER TO INTERACT WITH ALGORITHM
  - CONSISTENCY / COMPATIBILITY / MODULARITY IN THE CONTEXT OF A LARGER PACKAGE OR MORE COMPLEX PROBLEM
  - ERROR CONTROL
  - ROBUSTNESS W.R.T. UNUSUAL SITUATIONS
  - GRACEFUL PERFORMANCE DEGRADATION AS PROBLEM DOMAIN BOUNDARIES ARE APPROACHED
  - "TRICKS"
    - underflow - /overflow - proofing
    - columnwise (rowwise) linear algebra
  - USE OF SOFTWARE TOOLS / AIDS (e.g. PFORT, POLISH, TAMPR, WATFIV)
    - mechanical handling offers many advantages w.r.t., for example, modifications, updates, versions, maintenance
  - EFFICIENCY, PROGRAM SIZE (fcn. of intended use : e.g. low acc., real time)
  - UTILITY
  - AVAILABILITY

III. MATHEMATICAL SOFTWARE IN CONTROL

- DESIGN / SYNTHESIS ASPECT
- CAN PROVIDE GOOD SOFTWARE TOOLS AND ROBUSTLY IMPLEMENTED ALGORITHMS TO ENABLE A CONTROL DESIGNER TO CARRY OUT A DESIGN EASILY AND SENSIBLY
- A GOOD DEAL OF CURRENT MATH SOFTWARE CAN BE USED DIRECTLY (e.g. SUBSETS OF EISPACK, LINPACK); EFFECTIVELY "GRAY-BOXED" FOR CONTROL ENGINEERS
- EDUCATION : LEARN TO BECOME INTELLIGENT, DISCRIMINATING USERS OF CURRENTLY AVAILABLE SOFTWARE
- INCREASINGLY DIFFICULT FOR NONSPECIALISTS TO COMPETE

- NEED TO BE ABLE TO CHOOSE APPROPRIATE TOOLS FROM A "TOOLBOX" TO EASILY IMPLEMENT NEW OR MODIFIED THEORIES

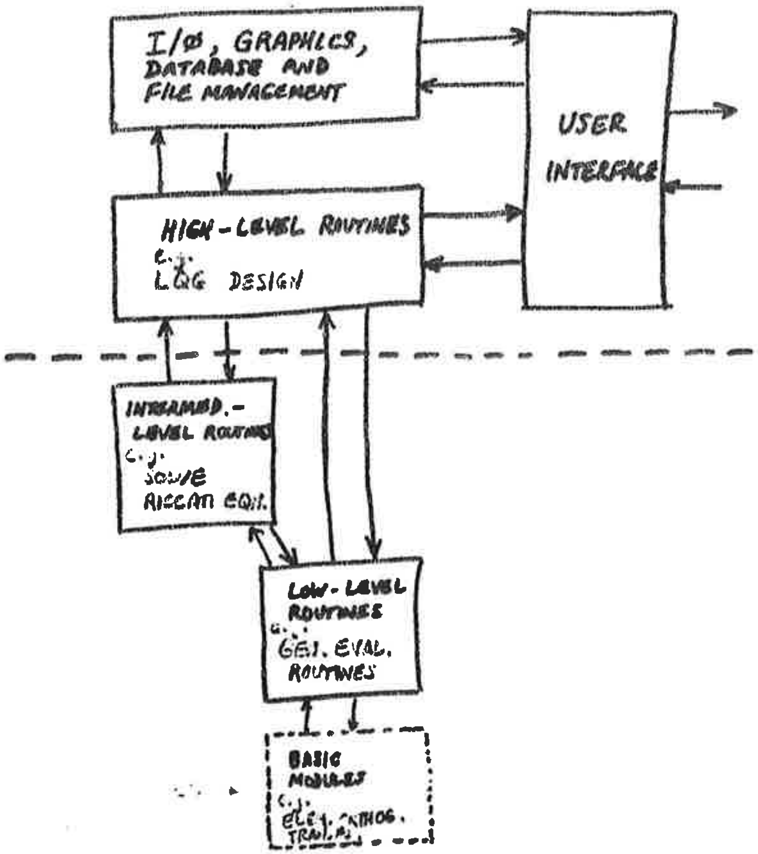
  e.g. plot $\|(I+G^{\pm 1}(j\omega))^{-1}\|$ vs. ω

$$\left.\begin{array}{l}\bullet\ \text{ALGORITHMS}\\ \bullet\ \text{THEORY}\end{array}\right\}\longrightarrow \text{TOOLBOX}\begin{array}{l}\nearrow \text{ANALYSIS}\\ \searrow \text{SYNTHESIS}\end{array}$$

INTERACTIVE

OVERALL PACKAGE SET-UP

CASCADE : COMPUTER-AIDED SYSTEMS AND CONTROL ANALYSIS AND DESIGN ENVIRONMENT



SOME DESIRED CAPABILITIES FOR A CACSD PACKAGE

MATRIX ARITHMETIC

add, subtract, multiply, $X^TSX \rightarrow Y$, save (copy), scale,
$A \cdot B \rightarrow C$
$A \cdot B \rightarrow A$
$A \cdot B \rightarrow B$
$X^TSX \rightarrow S$
αA

transpose, norms, BLAS, Householder, Givens, partitioned matrices,
$A^T \rightarrow A$
$A^T \rightarrow B$
(var. options)

(sort vector array, random no. gen.)

LINPACK-LIKE SUBSET

general linear equations, triangular lin. eqns., Cholesky factorization,
$AX = B$
$A^TX = B$

orthogonal factorizations, SVD
condition estimation

EISPACK-LIKE SUBSET

real general evals./evecs., real general generalized evals./evecs.,
real symmetric evals./evecs., real symm. p.d. generalized evals./evecs.,
ordered versions

"CLASSICAL" CONTROL (CONTS. & DISCRETE)

root locus, Bode, Nyquist, Nichols, partial fractions, poly. manipulations

STATE VARIABLES (CONTS. & DISCRETE) ...

poles/evecs. (damp. ratio, nat. freq.), zeros/zero directions,
realization, canonical forms, geometric objects/misc. "subspaces",
matrix exponential & integrals involving e^{tA},
time response: solve $\dot{x} = Ax + f$ stiff & nonstiff
time-varying
transformations & conversions among different representations,
(st. sp., tr. fcns., matrix fractions, etc.)
DC gains & transfer fcn. gains, frequency response from st. sp. descrip.,
rep. & manipula. of composite systems,
pole placement/observers, decoupling, spectral factorization,
Lyapunov and Sylvester equations

LQ, LQG (CONTS. & DISCRETE)

utilities (e.g. gain calc. from RE),
Riccati equations (alg. & DE's): direct and iterative (need initializing routine)
simulate lin. sys. with white noise input, KF,
transient, steady-state

DIFFERENTIAL & DIFFERENCE EQUATIONS

IVP'S, BVP'S
singular systems

ESTIMATION

linear and nonlinear least squares, MLE (solve sys. of nonlin. eqns.), square root filtering/smoothing, identification (param. est.)

OPTIMIZATION

unconstrained, constrained, "special" problems, ...

NONLINEAR

extended KF, optimal control, solve $\dot{x} = f(t, x, u)$
IVP's, BVP's, etc.

DIGITAL, SAMPLED-DATA

MODEL REDUCTION

OTHER (e.g., POLYNOMIAL MATRIX BASED THEORIES)

I/O, DATABASE MANAGEMENT

GRAPHICS

-
- ... EXTENSIONS ...
-

---

2. SOME RECENT INFLUENCES ON MATHEMATICAL SOFTWARE RESEARCH AND DEVELOPMENT

- ARITHMETIC
  - IEEE FLOATING-POINT STANDARD
    - DRAFT 10.0 OF P754
    - P854
    - SOME IMPLEMENTATIONS AVAILABLE
  - FUNDAMENTAL INFLUENCE ON ALGORITHMS AND THEIR IMPLEMENTATION
    - e.g.: could avoid underflow/overflow buffering
    - e.g.: solving normal equations for (linear) least squares could be numerically viable.

- ARCHITECTURE
  - "SUPER-COMPUTERS" AND LARGE-SCALE SCIENTIFIC COMPUTING
  - VECTOR COMPUTERS
    - MUCH CAN BE GAINED EVEN AT HIGH-LEVEL LANGUAGE LEVEL
    - NOT FOR EVERYONE
  - MULTI-PROCESSOR ENVIRONMENTS
  - "PARALLEL ALGORITHMS"
  - NLA ON A CHIP

- LANGUAGES AND SOFTWARE TOOLS
  - FORTRAN (77, 8X, ⋯)
  - ADA

- DENSE VS. SPARSE

- STRUCTURED VS. UNSTRUCTURED

4. CONCLUDING REMARKS

- SYMBIOTIC RELATIONSHIPS AMONG MEMBERS OF THE TEAM : ENGINEERING APPLICATION(S)
  COMPUTER SCIENCE & ENGINEERING
  NUMERICAL ANALYSIS / MATHEMATICS
  - CO-ORDINATION
  - REDUCE THE COMMUNICATION "GAP"
  - NEW RESEARCH AREAS
- GOOD APPLICATIONS SOFTWARE BASED ON GOOD ALGORITHMS WITH SUPPORTING ANALYSIS
- REQUIRES DISCIPLINE AND ATTENTION TO DETAIL; AIDED BY SOFTWARE TOOLS
- STANDARDS CONSTANTLY EVOLVING
- MUST EXPLOIT HARDWARE ECONOMICALLY BUT
- EXCEPTIONALLY HIGH QUALITY SOFTWARE IS EXCEPTIONALLY EXPENSIVE (BOTH <u>TIME</u> AND <u>MONEY</u>)
- MATHEMATICAL SOFTWARE IS A PRIMARY VEHICLE OF TECHNOLOGY TRANSFER
- BE DISCRIMINATING CONSUMERS OF SOFTWARE

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-30, NO. 2, FEBRUARY 1985

Numerical Linear Algebra Aspects of Control Design Computations

ALAN J. LAUB, SENIOR MEMBER, IEEE

***Abstract*—The interplay between recent results and methodologies in numerical linear algebra and mathematical software and their application to problems arising in systems, control, and estimation theory is discussed. The impact of finite precision, finite range arithmetic [including the implications of the proposed IEEE floating point standard(s)] on control design computations is illustrated with numerous examples as are pertinent remarks concerning numerical stability and conditioning. Basic tools from numerical linear algebra such as linear equations, linear least squares, eigenproblems, generalized eigenproblems, and singular value decomposition are then outlined. A selected list of applications of the basic tools then follows including algorithms for solution of problems such as matrix exponentials, frequency response, system balancing, and matrix Riccati equations. The implementation of such algorithms as robust mathematical software is then discussed. A number of issues are addressed including characteristics of reliable mathematical software, availability and evaluation, language implications (Fortran, Ada, etc.), and the overall role of mathematical software as a component of computer-aided control system design.**

I. INTRODUCTION

THIS tutorial paper provides an introduction to various aspects of the numerical solution of selected problems of interest in systems, control, and estimation theory. Space limitations preclude an exhaustive survey; rather, a compact "introduction to the literature" will lead the interested reader to sources of additional detailed information.

Many of the problems considered here arise in the study of the "standard" linear model

$$\dot{x}(t)=Ax(t)+Bu(t) \tag{1}$$

$$y(t)=Cx(t)+Du(t). \tag{2}$$

Here, $x(t)$ is an n-vector of states, $u(t)$ is an m-vector of controls or inputs, and $y(t)$ is an r-vector of outputs. The standard discrete-time analog of (1), (2) takes the form

$$x_{k+1}=Ax_k+Bu_k \tag{3}$$

$$y_k=Cx_k+Du_k. \tag{4}$$

Of course, considerably more elaborate models are also studied, including time-varying, stochastic, and nonlinear versions of the above, but these will not be discussed here. In fact, the above linear models are usually derived from linearizations of nonlinear models about selected nominal points. The interested reader is referred to standard textbooks such as [2], [8], [45], and [98] for further details.

The matrices considered here will, for the most part, be assumed to have real coefficients and be small (of order a few hundred or less) and dense, with no particular exploitable structure. Calculations for most problems in classical single-input, single-output control fall into this category. It must be emphasized that consideration of large, sparse matrices or matrices with a special, exploitable structure may involve significantly different concerns and methodologies than those to be discussed here.

The systems, control, and estimation literature is replete with ad hoc algorithms to solve the computational problems which arise in the various methodologies. Many of these algorithms work quite well on some problems (e.g., "small order" matrices) but encounter numerical difficulties, often severe, when "pushed" (e.g., on larger order matrices). The reason for this is that little or no attention has been paid to how the algorithms will perform in "finite arithmetic," i.e., on a finite-word-length digital computer.

A simple example due to Moler and Van Loan [64] will illustrate a typical pitfall. Suppose it is desired to compute the matrix e^A in single precision arithmetic on an IBM 370 computer. In this particular computing environment we have, roughly speaking, about 6 decimal places of precision in the fraction part of floating-point numbers. Consider the case

$$A=\begin{pmatrix} -49 & 24 \\ -64 & 31 \end{pmatrix}$$

and suppose the computation is attempted using the Taylor series formula

$$e^A=\sum_{k=0}^{+\infty}\frac{1}{k!}A^k. \tag{5}$$

This is easily coded and it is determined that the first 60 terms in the series suffice for the computation, in the sense that terms for $k \geq 60$ are of the order of 10^{-7} and no longer add anything significant to the sum. The resulting answer is

$$\begin{pmatrix} -22.2588 & -1.43277 \\ -61.4993 & -3.47428 \end{pmatrix}.$$

Unfortunately, the true answer is (correctly rounded)

$$\begin{pmatrix} -0.735759 & 0.551819 \\ -1.47152 & 1.10364 \end{pmatrix}$$

and one sees a rather alarming disparity. What happened here was that the intermediate terms in the series got very large before the factorial began to dominate. In fact, the 17th and 18th terms, for example, are of the order of 10^7 but of opposite signs so that the less significant parts of these numbers—while significant for the final answer—are "lost" because of the finiteness of the arithmetic.

Now for this particular example various fixes and remedies are available. But in more realistic examples one seldom has the luxury of having the "true answer" available so that it is not always easy simply to inspect or test an answer such as the one

---

Manuscript received February 22, 1983; revised December 8, 1983. Paper recommended by R. A. DeCarlo, Past Chairman of the Surveys and Tutorials Committee. This work was supported by the U.S. Army Research Office under Contract DAAG29-81-K-0131.

The author is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

0018-9286/85/0200-0097$01.00 ©1985 IEEE

obtained above and determine it to be in error. Mathematical analysis (truncation of the series, in the example above) alone is simply not sufficient when a problem is analyzed or solved in finite arithmetic (truncation of the arithmetic). Clearly, a great deal of care must be taken.

The finiteness inherent in representing real or complex numbers as floating-point numbers on a digital computer manifests itself in two important ways: floating-point numbers have only finite precision and finite range. In fact, it is the degree of attention paid to these two considerations that distinguishes many reliable algorithms from more unreliable counterparts. Reference [96] still provides the definitive introduction to the vagaries of floating-point computation while [60] and the references therein may be consulted to bring the interested reader up to date on roundoff analysis.

An even more recent development in floating-point arithmetic has been the work of the Floating-Point Working Group of the Microprocessor Standards Subcommittee of the IEEE Computer Society Standards Committee. An early draft (Draft 8.0) of the IEEE Standard 754 for Binary Floating-Point Arithmetic appears in [11] along with several related articles. A final draft, 10.0, of P754 was completed in December 1982 and forwarded to appropriate committees for further action. A parallel effort (Task 854) is also under way to draft a radix-free standard. These standards define families of commercially feasible ways for new systems (originally microprocessor, but also now minicomputers and large mainframes) to perform (binary) floating-point arithmetic in a numerically "sensible" way. The adoption of these standards will have a major impact on algorithmic development and software. In fact, early versions of the binary standard are already available for certain microprocessor systems (e.g., the Intel 8087) which will come into ever-increasing use in control and estimation in the 1980's and 1990's.

The development in systems, control, and estimation theory, of stable, efficient, and reliable algorithms which respect the constraints of finite arithmetic is only now in its infancy. Much of current research in numerical analysis is directly applicable but there are many computational issues in control (e.g., the presence of hard or structural zeros) where numerical analysis does not yet provide a ready answer or guide. A symbiotic relationship has already developed, particularly between numerical linear algebra and linear control and systems theory, which is sure to provide a continuing source of challenging research areas.

The abundance of numerically fragile algorithms is partly explained by the following observation which will be emphasized by calling it a "folk theorem":

> If an algorithm is amenable to "easy" hand calculation, it's probably a poor method if implemented in the finite floating-point arithmetic of a digital computer.

For example, when confronted with finding the eigenvalues of a 2 × 2 matrix most people would find the characteristic polynomial and solve the resulting quadratic equation. But when extrapolated as a general method for computing eigenvalues and implemented on a digital computer this turns out to be a very poor procedure indeed for a variety of reasons (such as roundoff and overflow/underflow). Of course the preferred method now would generally be the double Francis QR algorithm (see [82], [97] for the messy details) but few would attempt that by hand—even for very small order problems.

In fact, it turns out that many algorithms which are now considered fairly reliable in the context of finite arithmetic are not amenable to hand calculation (e.g., various classes of orthogonal similarities). This is sort of a converse to the folk theorem. Particularly in linear control and systems theories, we have been too easily seduced by the ready availability of closed-form solutions and numerically naive methods to implement those solutions. For example, in solving the initial value problem

$$\dot{x}(t) = Ax(t); \qquad x(0) = x_0 \tag{6}$$

it is not at all clear that one should explicitly want to compute the intermediate quantity e^{tA}. Rather, it is the vector $e^{tA}x_0$ that is desired, a quantity that may be computed more reasonably by treating (6) as a system of (possibly stiff) differential equations and using, say, an implicit method for numerical integration of the differential equation. But such techniques are definitely not attractive for hand computation.

Remedying the present situation is largely a matter of awareness and education. While it is a slow process, we are now just beginning to see some of the background material (well known to numerical analysts) mentioned in this paper filter down to the undergraduate and graduate curriculum in mathematics and engineering. Introductory textbooks such as [23], [41], [77], [78] are now also reflecting a strong software component. This process is certain to have a significant impact on the future directions and development of control and systems theory and applications as witness the growth of computer-aided control system design (CACSD) as an intrinsic tool. Algorithms implemented as mathematical software are a critical "inner" component of a CACSD system and the remainder of this paper will address some of the issues involved.

Before proceeding further we shall list here some notation to be used in the sequel:

$\mathbb{F}^{n \times m}$	the set of all $n \times m$ matrices with coefficients in the field $\mathbb{F}$ ($\mathbb{F}$ will generally be $\mathbb{R}$ or $\mathbb{C}$)
$\mathbb{F}_r^{n \times m}$	the set of all $n \times m$ matrices of rank r with coefficients in the field $\mathbb{F}$
A^T	the transpose of $A \in \mathbb{R}^{n \times m}$
A^H	the complex-conjugate transpose of $A \in \mathbb{C}^{n \times m}$
A^+	the Moore–Penrose pseudoinverse of A
$\\|A\\|$	the spectral norm of A (i.e., the matrix norm subordinate to the Euclidean vector norm: $\\|A\\| = \max_{\\|x\\|_2=1} \\|Ax\\|_2$)
diag $(a_1, \cdots, a_n)$	the diagonal matrix $\begin{pmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{pmatrix}$
$\Lambda(A)$	the list of eigenvalues $\lambda_1, \cdots, \lambda_n$ (not necessarily distinct) of $A \in \mathbb{F}^{n \times n}$
$\lambda_i(A)$	the ith eigenvalue of A
$\Sigma(A)$	the list of singular values $\sigma_1, \cdots, \sigma_m$ (not necessarily distinct) of $A \in \mathbb{F}^{n \times m}$
$\sigma_i(A)$	the ith singular value of A.

Finally, let us define a particular number to which we shall make frequent reference in the sequel. The *machine epsilon* or *relative machine precision* can be defined, roughly speaking, as the smallest positive number ϵ which, when added to 1 on our computing machine, gives a number greater than 1. In other words, any machine representable number δ less than ϵ gets "rounded off" when (floating-point) added to 1 to give exactly 1 again as the rounded sum. The number ϵ varies, of course, depending on the kind of computer being used and the precision with which the computations are being done (single precision, double precision, etc.). But the fact that there exists such a positive number ϵ is entirely a consequence of finite word length.

II. NUMERICAL STABILITY AND CONDITIONING

In this section we give a very brief discussion of two concepts of fundamental importance in numerical analysis: numerical stability and conditioning. While this material is standard in introductory textbooks such as [16], [38], [83], [87] it is presented here both for completeness and because the two concepts are frequently confused in the systems/control/estimation literature.

Suppose we have some mathematically defined problem represented by f which acts on data $d \in \mathfrak{D}$ = some set of data, to produce a solution $f(d) \in \mathcal{S}$ = some set of solutions. These notions are kept deliberately vague for expository purposes. Given $d \in \mathfrak{D}$ we desire to compute $f(d)$. Suppose d^* is some approximation to d. If $f(d^*)$ is "near" $f(d)$ the problem is said to be well-conditioned. If $f(d^*)$ may potentially differ greatly from $f(d)$ even when d^* is near d, the problem is said to be ill-conditioned. The concept of "near" can be made precise by introducing norms in the appropriate spaces. We can then define the condition of the problem f with respect to these norms as

$$\text{cond } (f) := \sup_{d,\, d^* \in \mathfrak{D}} \frac{\|f(d^*) - f(d)\|_{\mathcal{S}}}{\|d^* - d\|_{\mathfrak{D}}}.$$

Thus, if cond (f) is small, the problem is well-conditioned while if it is large, the problem is ill-conditioned. Further details can be found in [76].

A simple example of an ill-conditioned problem is the following. Consider the $n \times n$ matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & \cdot & \cdot & 1 \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{pmatrix}$$

with n eigenvalues at 0. Now consider a small perturbation of the data (the n^2 elements of A) consisting of adding the number 2^{-n} to the first element in the last (nth) row of A. This perturbed matrix then has n distinct eigenvalues $\lambda_1, \cdots, \lambda_n$ with $\lambda_k = \frac{1}{2} \exp(2k\pi j/n)$. Thus, we see that this small perturbation in the data has been magnified by a factor on the order of 2^n to result in a rather large perturbation in the solution (the eigenvalues of A). Further details and related examples are to be found in [97].

Note that we have thus far made no mention of how the problem f above (computing $\Lambda(A)$ in the example) was to be solved. Conditioning is a function solely of the problem itself. To solve a problem numerically we typically must implement some numerical procedure or algorithm which we shall denote by f^*. Thus, given d, $f^*(d)$ represents the result of applying the algorithm to d (for simplicity, we assume d is "representable"; a more general definition can be given when some approximation d^{**} to d must be used). The algorithm f^* is said to be numerically stable if, for all $d \in \mathfrak{D}$, there exists $d^* \in \mathfrak{D}$ near d such that $f^*(d)$ is near $f(d^*)$ (= the exact solution of a nearby problem). If the problem is well-conditioned, then $f(d^*)$ will be near $f(d)$ so that $f^*(d)$ will be near $f(d)$. In other words, f^* does not introduce any more sensitivity to perturbation than is inherent in the problem. Example 1 below will further illuminate this definition of stability which, on a first reading, can seem somewhat confusing.

Of course, one cannot expect a stable algorithm to solve an ill-conditioned problem any more accurately than the data warrant but an unstable algorithm can produce poor solutions even to well-conditioned problems. Example 2, below, will illustrate this phenomenon. There are thus two separate factors to consider in determining the accuracy of a computed solution $f^*(d)$. First, if the algorithm is stable, $f^*(d)$ is near $f(d^*)$, for some d^*, and second, if the problem is well-conditioned then, as above, $f(d^*)$ is near $f(d)$. Thus, $f^*(d)$ is near $f(d)$ and we have an "accurate" solution.

Roundoff errors can cause unstable algorithms to give disastrous results. However, it would be virtually impossible to account for every roundoff error made at every arithmetic operation in a complex series of calculations such as those involved in most linear algebra calculations. This would constitute a forward error analysis. As a more practical alternative, Wilkinson and others have advanced the notion of backward error analysis to account for roundoff error. Specifically, for many problems (particularly in numerical linear algebra), it is possible to show that what is actually computed is near the exact solution of a nearby problem. One then attempts to show that the nearby problem is near enough which, if the problem is well-conditioned, can be translated into a quantitative statement regarding the accuracy of the solution. Examples of this will be quoted in later sections.

We close this section with two simple examples to illustrate some of the concepts introduced above.

Example 1: Let x and y be two floating-point computer numbers and let $fl(x^*y)$ denote the result of multiplying them in floating-point computer arithmetic. In general, the product x^*y will require more precision to be represented exactly than was necessary to represent x or y. But what can be shown for most computers is that

$$fl(x^*y) = x^*y(1+\delta) \tag{7}$$

where $|\delta| < \epsilon$ (= relative machine precision). In other words, $fl(x^*y)$ is x^*y correct to within a unit in the last place. Now, another way to write (7) is as

$$fl(x^*y) = x(1+\delta)^{1/2} * y(1+\delta)^{1/2} \tag{8}$$

where $|\delta| < \epsilon$. This can be interpreted as follows: the computed result $fl(x^*y)$ is the exact product of the two slightly perturbed numbers $x(1 + \delta)^{1/2}$ and $y(1 + \delta)^{1/2}$. Note that the slightly perturbed data (not unique) may not even be representable floating-point numbers. The representation (8) is simply a way of accounting for the roundoff incurred in the algorithm by an initial (small) perturbation in the data.

Example 2: Gaussian elimination with no pivoting for solving the linear system

$$Ax = b \tag{9}$$

is known to be numerically unstable. The following data will illustrate this phenomenon. Let

$$A = \begin{pmatrix} 0.0001 & 1.000 \\ 1.000 & -1.000 \end{pmatrix}, \quad b = \begin{pmatrix} 1.000 \\ 0.000 \end{pmatrix}.$$

All computations will be carried out in four-significant-figure decimal arithmetic. The "true answer" $x = A^{-1}b$ is easily seen to be

$$\begin{pmatrix} 0.9999 \\ 0.9999 \end{pmatrix}.$$

Using row 1 as the "pivot row" (i.e., subtracting 10 000 × row 1 from row 2) we arrive at the equivalent triangular system

$$\begin{pmatrix} 0.0001 & 1.000 \\ 0 & -1.000 \times 10^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.000 \\ -1.000 \times 10^4 \end{pmatrix}.$$

Note that the coefficient multiplying x_2 in the second equation should be $-10\,001$, but because of roundoff, becomes $-10\,000$. Thus, we compute $x_2 = 1.000$ (a good approximation) but back-substitution in the equation

$$0.0001x_1 = 1.000 - fl(1.000^*1.000)$$

yields $x_1 = 0.000$. This extremely bad approximation to x_1 is the result of numerical instability. The problem itself can be shown to be quite well-conditioned.

III. FUNDAMENTAL PROBLEMS IN NUMERICAL LINEAR ALGEBRA

In this section we give a brief overview of some of the fundamental problems in numerical linear algebra which serve as building blocks or "tools" for the solution of problems in systems, control, and estimation.

A. Linear Algebraic Equations and Linear Least Squares Problems

Probably the most fundamental problem in numerical computing is the calculation of a vector x which satisfies the linear system

$$Ax=b \tag{10}$$

where $A \in \mathbb{R}_n^{n\times n}$ (or $\mathbb{C}_n^{n\times n}$). A great deal is now known about solving (10) in finite arithmetic both for the general case and for a large number of special situations. Some of the standard references include [17], [22], [40], and [83].

The most commonly used algorithm for solving (10) with general A and small n (say $n \leq 200$) is Gaussian elimination with some sort of pivoting strategy, usually "partial pivoting." This essentially amounts to factoring some permutation of the rows of A into the product of a unit lower triangular matrix L and an upper triangular matrix U. The algorithm is effectively stable, i.e., it can be proved that the computed solution is near the exact solution of the system

$$(A+E)x=b \tag{11}$$

with $|e_{ij}| \leq \phi(n)\cdot\gamma\cdot\beta\cdot\epsilon$ where $\phi(n)$ is a modest function of n depending on details of the arithmetic used, γ is a "growth factor" (which is a function of the pivoting strategy and is usually—but not always—small), β behaves essentially like $\|A\|$, and ϵ is the machine precision. In other words, except for moderately pathological situations, E is "small"—on the order of $\|A\|\cdot\epsilon$. See [83] for further details.

The following question then arises. If, because of roundoff errors, we are effectively solving (11) rather than (10), what is the relationship between $(A+E)^{-1}b$ and $A^{-1}b$? To answer this question we need some elementary perturbation theory and this is where the notion of condition number arises. A condition number for the problem (10) is given by

$$\kappa(A):=\|A\|\,\|A^{-1}\|. \tag{12}$$

Simple perturbation results can be used to show that perturbation in A and/or b can be magnified by as much as $\kappa(A)$ in the computed solution. Estimation of $\kappa(A)$ (since, of course, A^{-1} is unknown) is thus a crucial aspect of assessing solutions of (10) and the particular estimation procedure used is usually the principal difference between competing linear equation software packages. One of the more sophisticated and reliable condition estimators presently available is based on [9] and is implemented in LINPACK [17]. In addition to the l_1 condition estimator of [9], LINPACK features many codes for solving (10) in case A has certain special structures (e.g., banded, symmetric, positive definite).

Another important class of linear algebra problems and one for which codes are available in LINPACK is the linear least squares problem

$$\min \|Ax-b\|_2 \tag{13}$$

where $A \in \mathbb{R}_k^{m\times n}$ with (in the simplest case) $k = n \leq m$. The solution of (13) can be written formally as $x = A^+b$. Here, standard references include [17], [55], and [83]. The method of choice is generally based upon the QR factorization of A (for simplicity, $A \in \mathbb{R}_n^{m\times n}$)

$$A=QR \tag{14}$$

where $R \in \mathbb{R}_n^{n\times n}$ is upper triangular and $Q \in \mathbb{R}^{m\times n}$ has orthonormal columns, i.e., $Q^TQ = I$. With special care and analysis the case $k < n$ can also be handled similarly. The factorization is effected through a sequence of Householder transformations H_i applied to A. Each H_i is symmetric and orthogonal and of the form $I - 2uu^T/u^Tu$ where $u \in \mathbb{R}^m$ is specially chosen to introduce zeros at appropriate places in A when premultiplied by H_i. After n such transformations we have

$$H_nH_{n-1}\cdots H_1A=\begin{pmatrix} R \\ 0 \end{pmatrix}$$

from which the factorization (14) follows. Defining c and d by

$$H_nH_{n-1}\cdots H_1b=:\begin{pmatrix} c \\ d \end{pmatrix}$$

where $c \in \mathbb{R}^n$, it is easily shown that the least squares solution x of (13) is given by the solution of the linear system

$$Rx=c. \tag{15}$$

The above algorithm can be shown to be numerically stable and, again, a well-developed perturbation theory exists from which condition numbers can be obtained, this time in terms of

$$\kappa(A):=\|A\|\,\|A^+\|.$$

Least squares perturbation theory is fairly straightforward in case $A \in \mathbb{R}_n^{m\times n}$ but is considerably more complicated when A is rank-deficient. The reason for this is that while the inverse is a continuous function of the data (i.e., the inverse is a continuous function in a neighborhood of a nonsingular matrix), the pseudoinverse is discontinuous. For example, consider

$$A=\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}=A^+$$

and perturbations

$$E_1=\begin{pmatrix} 0 & 0 \\ \delta & 0 \end{pmatrix}$$

and

$$E_2=\begin{pmatrix} 0 & 0 \\ 0 & \delta \end{pmatrix}$$

with δ small. Then

$$(A+E_1)^+=\begin{pmatrix} \dfrac{1}{1+\delta^2} & \dfrac{\delta}{1+\delta^2} \\ 0 & 0 \end{pmatrix}$$

which is close to A^+ but

$$(A+E_2)^+=\begin{pmatrix} 1 & 0 \\ 0 & \dfrac{1}{\delta} \end{pmatrix}$$

which gets arbitrarily far from A^+ as δ is decreased towards 0. For a complete survey of perturbation theory for the least squares problem and related questions, see [84].

In lieu of Householder transformations, Givens transformations (elementary rotations or reflections) may also be used to solve the linear least squares problem. Details can be found in [17], [55], [72], [85], and [97]. Recently, Givens transformations have received considerable attention for the solution of both linear least squares problems as well as systems of linear equations in a parallel computing environment. The capability of introducing zero elements selectively and the need for only local interprocessor communication make the technique ideal for "parallelization." Indeed, there have been literally dozens of "parallel Givens" algorithms proposed and we include [26], [28], [36], [57], and [81] as representative references.

B. Eigenvalue and Generalized Eigenvalue Problems

In the algebraic eigenvalue/eigenvector problem for $A \in \mathbb{R}^{n \times n}$ one seeks nonzero solutions $x \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$ which satisfy

$$Ax = \lambda x. \tag{16}$$

The classic reference on the numerical aspects of this problem is Wilkinson [97] with Parlett [72] providing an equally thorough and up to date treatment of the case of symmetric A (in which $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$). A more brief textbook introduction is given in [83].

It is really only rather recently that some of the computational issues associated with solving (16)—in the presence of rounding error—have been resolved or even understood. Even now some problems such as the invariant subspace problem continue to be active research areas. For an introduction to some of the difficulties which may be encountered in trying to make numerical sense out of mathematical constructions such as the Jordan canonical form the reader is urged to consult [32].

The most common algorithm now used to solve (17) for general A is the QR algorithm of Francis [25]. A shifting procedure (see [83, sects. 7.3, 7.4] for a brief explanation) is used to enhance convergence and the usual implementation is called the double-Francis-QR algorithm. Before the QR process is applied, A is initially reduced to upper Hessenberg form $A_H(a_{ij} = 0$ if $i - j \geq 2)$ [58]. This is accomplished by a finite sequence of similarities which can be chosen to be of the Householder form discussed above. The QR process then yields a sequence of matrices which are orthogonally similar to A and which converge (in some sense) to a so-called quasi-upper-triangular matrix S which is also called the real Schur form (RSF) of A. The matrix S is block-upper-triangular with 1×1 diagonal blocks corresponding to real eigenvalues of A and 2×2 diagonal blocks corresponding to complex-conjugate pairs of eigenvalues. Quasi-upper-triangular form permits all arithmetic done to be real rather than complex as would be necessary for convergence to an upper triangular matrix. The orthogonal transformations from both the Hessenberg reduction and the QR process may be accumulated into a single orthogonal transformation U so that

$$U^TAU = S \tag{17}$$

compactly represents the entire algorithm.

An analogous process can be applied in the case of symmetric A and considerable simplifications and specializations result. Moreover, [72] and [97] may be consulted regarding an immense literature concerning stability of the QR and related algorithms and conditioning of eigenvalues and eigenvectors. Both subjects are vastly more complex for the eigenvalue/eigenvector problem than for the linear equation problem.

Quality mathematical software for eigenvalues and eigenvectors has been only recently available—in the 1970's—and the EISPACK [27], [82] collection of subroutines represents a pivotal point in the history of mathematical software. This collection is primarily based on the algorithms collected in [95].

Closely related to the QR algorithm is the QZ algorithm [63] for the generalized eigenvalue problem

$$Ax = \lambda Mx \tag{18}$$

where $A, M \in \mathbb{R}^{n \times n}$. Again, a Hessenberg-like reduction, then an iterative process is implemented with orthogonal transformations to reduce (18) to the form

$$QAZy = \lambda QMZy \tag{19}$$

where QAZ is quasi-upper-triangular and QMZ is upper triangular. For a review and references to results on stability, conditioning, and software related to (18) and the QZ algorithm, see [48]. The generalized eigenvalue problem is both theoretically and numerically more difficult to handle than the ordinary eigenvalue problem but it finds numerous applications in control and systems theory.

C. The Singular Value Decomposition and Some Applications

One of the basic and most important tools of modern numerical analysis, particularly numerical linear algebra, is the singular value decomposition. We shall define it here and make a few comments about its properties and computation as well as its significance in various numerical problems.

Singular values and the singular value decomposition have a long history particularly in statistics and more recently in numerical linear algebra. Even more recently the ideas are finding applications in the control and signal processing literature, although their use there has been overstated somewhat in certain applications. For a survey of the singular value decomposition, its history, numerical details, and some applications in control and systems theory, see [44].

The fundamental result can be stated as follows for the real case. For complex matrices the result is virtually identical with complex-conjugate transposes replacing transposes and unitary matrices replacing orthogonal matrices.

Theorem 1: Let $A \in \mathbb{R}_r^{m \times n}$. Then there exist orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ such that

$$A = U\Sigma V^T \tag{20}$$

where

$$\Sigma = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}$$

and $S = \operatorname{diag}(\sigma_1, \cdots, \sigma_r)$ with $\sigma_1 \geq \cdots \geq \sigma_r > 0$.

The proof of Theorem 1 is straightforward and can be found in, for example, [30] and [83]. Geometrically, the theorem says that bases can be found (separately) in the domain and codomain spaces of a linear map with respect to which matrix representation of the linear map is diagonal.

The numbers $\sigma_1, \cdots, \sigma_r$ together with $\sigma_{r+1} = 0, \cdots, \sigma_n = 0$ are called the singular values of A and they are the positive square roots of the eigenvalues of A^TA. The columns $\{u_k, k = 1, \cdots, m\}$ of U are called the left singular vectors of A (the orthonormal eigenvectors of AA^T) while the columns $\{v_k, k = 1, \cdots, n\}$ of V are called the right singular vectors of A (the orthonormal eigenvectors of A^TA). The matrix A can then also be written (as a dyadic expansion) in terms of the singular vectors as follows:

$$A = \sum_{k=1}^{r} \sigma_k u_k v_k^T.$$

The matrix A^T has m singular values, the positive square roots of the eigenvalues of AA^T. The $r[= \operatorname{rank}(A)]$ nonzero singular values of A and A^T are, of course, the same. The choice of A^TA rather than AA^T in the definition of singular values is arbitrary. Only the nonzero singular values are usually of any real interest and their number, given the SVD, is the rank of the matrix. Naturally, the question of how to distinguish nonzero from zero singular values in the presence of rounding error is a nontrivial task.

It is not generally advisable to compute the singular values of A by first finding the eigenvalues of A^TA (remember the folk theorem!), tempting as that is. Consider the following example with μ a real number with $|\mu| < \sqrt{\epsilon}$ (so that $fl(1 + \mu^2) = 1$ where $fl(\cdot)$ denotes floating-point computation). Let

$$A = \begin{pmatrix} 1 & 1 \\ \mu & 0 \\ 0 & \mu \end{pmatrix}.$$

Then

$$fl(A^TA)=\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}$$

so we compute $\hat{\sigma}_1 = \sqrt{2}$, $\hat{\sigma}_2 = 0$ leading to the (erroneous) conclusion that the rank of A is 1. Of course, if we could compute in infinite precision we would find

$$A^TA=\begin{pmatrix}1+\mu^2 & 1\\ 1 & 1+\mu^2\end{pmatrix}$$

with $\sigma_1 = \sqrt{2+\mu^2}$, $\sigma_2 = |\mu|$ and thus rank $(A) = 2$. The point is that by working with A^TA we have unnecessarily introduced μ^2 into the computations. The above example illustrates a potential pitfall in attempting to form and solve the normal equations in a linear least squares problem and is at the heart of what makes square root filtering so attractive numerically. Very simplistically speaking, square root filtering involves working directly on an "A-matrix," for example updating it, as opposed to working on (updating, say) an "A^TA-matrix." See [6] for further details and references.

Square root filtering is usually implemented using QR factorization (or some closely related algorithm) as described previously rather than SVD. The key thing to remember is that in most current computing environments the condition of the least squares problem is squared, unnecessarily, in solving the normal equations and, moreover, critical information may be lost, irrecoverably, by simply forming A^TA. These caveats may not be of such great concern, however, if one has available certain computing environments which implement, for example, IEEE arithmetic with extended length registers (e.g., the Intel 8087 floating-point processor chip).

Returning now to the SVD there are two features of this matrix factorization that make it so attractive in finite arithmetic: first, it can be computed in a numerically stable way and second, singular values are well-conditioned. Specifically, there is an efficient and numerically stable algorithm due to Golub and Reinsch [31] (based on [30]) which works directly on A to give the SVD. The computed U and V are orthogonal to approximately the working precision and the computed singular values can be shown to be the exact σ_i's for $A + E$ where $\|E\|/\|A\|$ is a modest multiple of ϵ. Fairly sophisticated implementations of this algorithm can be found in [17] and [27]. The well-conditioned nature of the singular values follows from the fact that if A is perturbed to $A + E$, then it can be proved that

$$|\sigma_i(A+E)-\sigma_i(A)|\le\|E\|.$$

Thus, the singular values are computed with small absolute error although the relative error of sufficiently small singular values is not guaranteed small.

It is now acknowledged that the singular value decomposition is the most generally reliable method of determining rank numerically (see [32] for a more elaborate discussion). However, it is considerably more expensive to compute than, for example, the QR factorization which, with column pivoting [17], can usually give equivalent information with less computation. Thus, while the SVD is a useful theoretical tool, its use for actual computations should be weighed carefully against other approaches.

Only rather recently has the problem of numerical determination of rank become well-understood. One of the best treatments of the subject, including a careful definition of numerical rank, is a paper by Golub, Klema, and Stewart [33]; see also [86]. The essential idea is to try to determine a "gap" between "zero" and the "smallest nonzero singular value" of a matrix A. Since the computed values are exact for a matrix near A it makes sense to consider the rank of all matrices in some δ-ball (w.r.t. the spectral norm $\|\cdot\|$, say) around A. The choice of δ may also be based on measurement errors incurred in estimating the coefficients of A or the coefficients may be uncertain because of roundoff errors incurred in a previous computation to get them. We refer to [33], [86] for further details. We must emphasize, however, that even with SVD, numerical determination of rank in finite arithmetic is a highly nontrivial problem.

That other methods of rank determination are potentially unreliable is demonstrated by the following example which is a special case of a general class of matrices studied by Ostrowski [68]. Consider the matrix $A \in \mathbb{R}^{n\times n}$ whose diagonal elements are all -1, whose upper triangle elements are all $+1$, and whose lower triangle elements are all 0. This matrix is clearly of rank n, i.e., is invertible. It has a good "solid" upper triangular shape. All of its eigenvalues (all $= -1$) are well away from zero. Its determinant is $(-1)^n$—definitely not close to zero. But this matrix is, in fact, very near singular and gets more nearly so as n increases. Note, for example, that

$$\begin{pmatrix}-1 & +1 & \cdots & +1\\ & \ddots & \ddots & \vdots\\ & 0 & \ddots & +1\\ & & & -1\end{pmatrix}\begin{pmatrix}1\\ 2^{-1}\\ \vdots\\ 2^{-n+1}\end{pmatrix}$$

$$=\begin{pmatrix}-2^{-n+1}\\ -2^{-n+1}\\ \vdots\\ -2^{-n+1}\end{pmatrix}\to\begin{pmatrix}0\\ 0\\ \vdots\\ 0\end{pmatrix}\quad(n\to+\infty).$$

Moreover, adding 2^{-n+1} to every element in the first column of A gives an exactly singular matrix. Arriving at such a matrix by, say Gaussian elimination, would give no hint as to the near-singularity. However, it is easy to check that $\sigma_n(A)$ behaves as 2^{-n+1}. A corollary for control theory: eigenvalues do not necessarily give a reliable measure of "stability margin." As an aside it is useful to note here that in this example of an invertible matrix, the crucial quantity, $\sigma_n(A)$, which measures nearness to singularity, is simply $1/\|A^{-1}\|$ and the result is familiar from standard operator theory. There is nothing intrinsic about singular values in this example and, in fact, $\|A^{-1}\|$ might be more cheaply computed or estimated in another matrix norm. This is precisely what is done in estimating the condition of linear systems in LINPACK where $\|\cdot\|_1$ is used [9].

Since rank determination, in the presence of roundoff error, is a nontrivial problem, all the same difficulties will naturally arise in any problem equivalent to or involving rank determination such as determining the independence of vectors, finding subspace bases, etc. Such problems arise as basic calculations throughout systems, control, and estimation theory. Selected applications are discussed in more detail in [44].

Finally, let us close this section with a brief example illustrating a totally inappropriate use of SVD. The rank condition

$$\text{rank }[B,\ AB,\ \cdots,\ A^{n-1}B]=n \tag{21}$$

for the controllability of (1) is too well-known. Suppose

$$A=\begin{pmatrix}1 & \mu\\ 0 & 1\end{pmatrix},\quad B=\begin{pmatrix}1\\ \mu\end{pmatrix}$$

with $|\mu| < \sqrt{\epsilon}$. Then

$$fl[B,\ AB]=\begin{pmatrix}1 & 1\\ \mu & \mu\end{pmatrix}$$

and now even applying SVD the erroneous conclusion of uncontrollability is reached. Again the problem is in just forming

AB; not even SVD can come to the rescue after that numerical faux pas.

IV. Applications

In this section we shall present a representative selection of numerical problems which arise in linear systems, control, and estimation theory and which have been examined using some of the techniques described in Sections II and III. Some of the topics are described in more detail in [44] and [54] while still other topics are surveyed in [89].

A. Numerical Solution of Linear Ordinary Differential Equations and Matrix Exponentials

The "simulation" or numerical solution of linear systems of ordinary differential equations (ODE's) of the form

$$\dot{x}(t)=Ax(t)+f(t); \qquad x(0)=x_0 \tag{22}$$

is a standard problem. However, there is still debate as to what is the most effective numerical algorithm, particularly when A is defective (a deficiency of eigenvectors) or near-defective. The most common approach involves computation of the matrix exponential e^{tA}, since the solution of (22) can be written simply as

$$x(t)=e^{tA}x_0+e^{tA}\int_0^t e^{-sA}f(s)\ ds.$$

A delightful survey of computational techniques for matrix exponentials is given in [64]. Nineteen "dubious" ways are explored (there exist many more ways which are not discussed) but no clearly superior algorithm is singled out. Methods based on Padé approximation or reduction of A to real Schur form are seen as generally attractive while methods based on Taylor series or the characteristic polynomial of A are generally found to be unattractive. An interesting open problem is the design of a special algorithm for the matrix exponential when the matrix is known *a priori* to be stable ($\Lambda(A)$ in the left half of the complex plane).

The reason for the adjective "dubious" in the title of [64] is that in many (maybe even most) circumstances, it is better to treat (22) as a system of differential equations, typically stiff, and to apply various ODE techniques, specially tailored to the linear case. This approach is discussed in [21]. ODE techniques are clearly to be preferred when A is large and sparse for, in general, e^{tA} will be unmanageably large and dense. The relationship between ODE techniques and matrix exponential techniques when A has an ill-conditioned eigenstructure or when the "exponential problem" is ill-conditioned [42], [92] is not well-understood.

B. Controllability and Other "Abilities"

Basic to the study of linear control and systems theory are the various "abilities" such as controllability, observability, reachability, reconstructibility, stabilizability, and detectability [98]. Our remarks here will be confined, but are not limited, to the notion of controllability.

A large number of algebraic and dynamic characterizations of controllability have been given; see [54] for a sample. But each and every one of these has difficulties when implemented in finite arithmetic. For a survey of this topic and numerous examples, see [70]. Part of the difficulty in dealing with controllability numerically lies in the intimate relationship with the invariant subspace problem [32]. The controllable subspace associated with (1) is the smallest A-invariant subspace (subspace spanned by eigenvectors or principal vectors) containing the range of B. Since A-invariant subspaces are extremely sensitive to perturbation, it follows that, so too, is the controllable subspace. Similar remarks apply to the computation of the so-called controllability indexes. The example discussed in the third paragraph of Section II can be used to illustrate graphically these remarks. The matrix A has but one eigenvector (associated with 0) whereas the slightly perturbed A has n eigenvectors associated with the n distinct eigenvalues.

Recently, attempts have been made to provide numerically stable algorithms for the pole placement problem; we would cite [61], [73], [74], and [93] as an undoubtedly incomplete list of examples (only one representative and recent reference is chosen for each author or group). The methods are based on reduction of A to a Hessenberg form rather than a controllable or Luenberger canonical form whose computation is known to be numerically unstable [97]. For example, in the single-input case it can easily be shown, using the tools developed in Section III-A, that there exists an orthogonal transformation U such that U^TAU is upper Hessenberg and U^TB = a multiple of $(1, 0, \cdots, 0)^T$. The pair (A, B) is then controllable if and only if all $(n-1)$ subdiagonal elements of U^TAU are nonzero. If a subdiagonal element is 0, the system is uncontrollable and a basis for the uncontrollable subspace is easily constructed. The transfer function gain or first nonzero Markov parameter is also easily constructed from this "canonical form." In fact, the numerically more robust general Hessenberg form will probably play an ever-increasing role in systems theory in replacing the numerically more fragile special case of the companion or rational canonical or Luenberger canonical form.

A more important aspect of controllability is to better understand topological notions such as "near-uncontrollability." But there are numerical difficulties lurking here, also, and we refer to [70] for further details. Related to this is an interesting new system-theoretic concept called "balancing"; see [66]. The computation of "balancing transformations" is discussed in [50].

There are at least two distinct notions of near-uncontrollability: in the parametric sense and in the energy sense. In the parametric sense a controllable pair (A, B) is said to be near-uncontrollable if the parameters of (A, B) need be perturbed by only a relatively small amount to become uncontrollable. In the energy sense, a controllable pair is near-uncontrollable if large amounts of control energy ($\int u^Tu$) are required to effect a state transfer. The pair

$$A=\begin{pmatrix} 0 & 1 & & 0 \\ \cdot & & \ddots & \\ \cdot & \ddots & & 1 \\ \cdot & & \ddots & \\ 0 & & \cdots & 0 \end{pmatrix}, \qquad B=\begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{pmatrix}$$

is near-uncontrollable in the energy sense but apparently not in the parametric sense. Of course, both measures are coordinate dependent and "balancing" is one attempt to try to remove this coordinate bias.

C. Computation of Objects Arising in the Geometric Theory of Linear Multivariable Control

A great many numerical problems arise in the geometric approach [98] to control of systems modeled as (1), (2). Some of these are discussed in [44] and [91], and [98] remains a fertile source of numerical problems. In fact, the power of the geometric approach derives in large part from its divorce from matrices and specific coordinate systems. Numerical issues are a separate concern. Two of the more elaborate but still fundamental objects in that theory are supremal (A, B)—invariant and controllability subspaces contained in a given subspace. An initial attempt at characterizing these spaces in terms of eigenvectors and a generalized eigenvalue problem was given in [65].

However, a rather different and very thorough numerical treatment of the problem has been done by Van Dooren [90], [91]. He has done the most definitive numerical study to date of the matrix pencil $(L - \lambda M)$ problem [43]. This work has implications for most calculations done with linear state-space models. For example, one byproduct is an extremely reliable algorithm (which is similar to an orthogonal version of Silverman's structure algorithm) for the computation of multivariable system zeros [20]. Like [48] this method involves a generalized eigenvalue problem (the Rosenbrock pencil) but the "infinite zeros" are first deflated out.

D. Frequency Response Calculations

Many properties of a linear system (1), (2) are known in terms of its frequency response matrix

$$G(j\omega):=C(j\omega I-A)^{-1}B+D; \qquad (\omega\geq 0) \qquad (23)$$

[or $G(e^{j\theta})$; $\theta\in[0,2\pi]$ for (3), (4)]. In fact, various norms of the return difference matrix $I+G(j\omega)$ and related quantities have recently been investigated as providing measures of robustness of a linear system with respect to stability, noise response, disturbance attenuation, sensitivity, etc. See [51] for some of the numerical aspects and [19], [80] for surveys of some of the control aspects.

It is thus a problem of considerable computational interest to compute $G(j\omega)$ efficiently, given A, B, and C for a (possibly) large number of values of ω (for convenience we shall take D to be 0 since if it is nonzero it is trivial to add to G). An efficient and generally applicable algorithm for this problem is presented in [49]. Rather than solve the linear equation (with dense, unstructured A) $(j\omega I-A)X=B$ which would require $0(n^3)$ operations for each successive value of ω, the new method does an initial reduction of A to upper Hessenberg form H. The orthogonal matrices used to effect the Hessenberg form of A are incorporated into B and C giving $\tilde{B}$ and $\tilde{C}$. Now as ω varies, the coefficient matrix in the linear equation $(j\omega I-H)X=\tilde{B}$ remains in upper Hessenberg form. The advantage is that X can now be found in $0(n^2)$ operations rather than $0(n^3)$ as before, a substantial savings. Moreover, the method is numerically very stable and has the advantage of being independent of the eigenstructure (possibly ill-conditioned) of A.

Portable mathematical software, in the sense to be discussed in Section V, is also available for this problem [47].

We note here that the above method can also be extended to state-space models in implicit form, e.g., (1) is replaced by

$$E\dot{x}=Ax+Bu. \qquad (24)$$

Then (23) is replaced with

$$G(j\omega)=C(j\omega E-A)^{-1}B+D \qquad (25)$$

and the initial triangular/Hessenberg reduction employed in [63] can be employed to again reduce the problem to one of updating the diagonal of a Hessenberg matrix and consequently an $0(n^2)$ linear equation problem.

E. Lyapunov, Sylvester, and Riccati Equations

Certain matrix equations arise naturally in linear control and systems theory. Among those frequently encountered in the analysis of continuous-time systems are the Lyapunov equation

$$FX+XF^T+H=0, \qquad (26)$$

and the Sylvester equation

$$FX+XG+H=0. \qquad (27)$$

The appropriate discrete-time analogs are

$$FXF^T-X+H=0 \qquad (28)$$

$$FXG-X+H=0. \qquad (29)$$

Various hypotheses are made on the coefficient matrices F, G, H to ensure certain properties of the solution X.

Surprisingly little attention has been paid to solution of these equations in the numerical linear algebra literature. There is, however, a voluminous literature in control and systems theory but most of that is ad hoc, at best, from a numerical point of view, with little attention paid to questions of numerical stability, conditioning, machine implementation, and the like.

For the Lyapunov equation the overall best algorithm in terms of efficiency, accuracy, reliability, availability, and ease of use appears to be that of Bartels and Stewart [5]. The basic idea is to reduce F to quasi-upper-triangular form [or real Schur form (RSF)] and perform a back substitution for the elements of X.

For the Sylvester equation the Bartels–Stewart algorithm reduces both F and G to real Schur form (RSF) and then a back substitution is done. It has been demonstrated in [34] that some improvement in this procedure is possible by only reducing the larger of F and G to upper Hessenberg form.

A promising new algorithm for solving Lyapunov equations has recently been proposed by Hammarling [35]. This algorithm is a variant of the Bartels–Stewart algorithm which solves directly for the Cholesky factor Y of X: $Y^TY=X$ and Y is upper-triangular. Clearly, given Y, X is easily recovered if necessary. But in many applications, for example [50], only the Cholesky factor is required.

Open questions remain concerning estimating the condition of Lyapunov and Sylvester equations efficiently and reliably in terms of the coefficient matrices.

A deeper analysis of the Lyapunov and Sylvester problems is probably a prerequisite to at least a better understanding of conditioning of the Riccati equation for which again, there is a considerable theoretical literature but rather little known from a purely numerical point of view. The symmetric $n\times n$ algebraic Riccati equation takes the form

$$XGX+FX+XF^T+H=0 \qquad (30)$$

for continuous-time systems and

$$FXF^T-X-FXG_1(G_2+G_1^TXG_1)^{-1}G_1^TXF^T+H=0 \qquad (31)$$

for discrete-time situations. Again, appropriate assumptions are made on the coefficient matrices to guarantee the existence and/or uniqueness of certain kinds of solutions X. Nonsymmetric Riccati equations of the form

$$XGX+F_1X+XF_2+H=0 \qquad (32)$$

for the continuous-time case (along with an analog for the discrete-time case) are also studied and can be solved numerically by the techniques discussed below.

One of the more reliable general-purpose methods for solving Riccati equations is the Schur method [46]. For the case of (30), for example, this method is based upon the reduction of the associated $2n\times 2n$ Hamiltonian matrix

$$\begin{pmatrix} F^T & G \\ -H & -F \end{pmatrix} \qquad (33)$$

to RSF. If the RSF is ordered so that its stable eigenvalues (there will be exactly n of them under certain assumptions) are in the upper left triangle, the corresponding first n vectors of the orthogonal matrix which effects the reduction will form a basis for the stable eigenspace from which the Riccati solution is then easily found.

Extensions to the basic Schur method have been made [71], [88] which were motivated by the following situations.

1) G in (30) is of the form $BR^{-1}B^T$ where R may be near-singular.

2) F in (31) is singular [F^{-1} is required in the classical

approach involving a symplectic matrix which assumes the role of (33)]. In fact, these extensions can be generalized even further, as the following problem will illustrate. Consider the optimal control problem

$$\min \frac{1}{2} \int_0^{+\infty} [x^TQx + 2x^TSu + u^TRu]\, dt \tag{34}$$

$$\text{subject to } E\dot{x} = Ax + Bu. \tag{35}$$

The Riccati equation associated with (34), (35) then takes the form

$$\begin{aligned} &E^TXBR^{-1}B^TXE - (A - BR^{-1}S^T)^TXE \\ &\quad - E^TX(A - BR^{-1}S^T) - Q + SR^{-1}S^T = 0 \end{aligned} \tag{36}$$

or

$$(E^TXB + S)R^{-1}(B^TXE + S^T) - A^TXE - E^TXA - Q = 0. \tag{37}$$

This so-called "generalized" Riccati equation can be solved by considering the associated matrix pencil

$$\begin{pmatrix} A & 0 & B \\ -Q & -A^T & -S \\ S^T & B^T & R \end{pmatrix} - \lambda \begin{pmatrix} E & 0 & 0 \\ 0 & E^T & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{38}$$

Note that S in (34) and E in (35) are handled directly and no inverses appear. It is worth emphasizing here that the presence of a nonsingular E in state-space models of the form (35) adds no particular difficulty to the solution process and is numerically the preferred form if E is, for example, near-singular or even sparse. Similar remarks apply to the frequency response problem in (24), (25) and, indeed, throughout all of linear control and systems theory. Numerical methods for handling (38) and a large variety of related problems are described in [3], [56], [91] and a thorough survey of the Schur method, generalized eigenvalue/eigenvector extensions, and the underlying algebraic structure in terms of "Hamiltonian pencils" and "symplectic pencils" is described in [53].

Schur techniques can also be applied to Riccati differential and difference equations [52] and to nonsymmetric Riccati equations which arise in, for example, invariant imbedding methods for solving linear two-point boundary value problems [53].

As with the linear Lyapunov and Sylvester equations there are few satisfactory results concerning conditioning of Riccati equations, a topic of great interest independent of what solution method is used, be it a Schur-type method or one of numerous alternatives. One fairly reliable method is to estimate the condition of U_{11} with respect to inversion where

$$\begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix}$$

is a basis for the stable eigenspace. This turns out to be essentially equivalent to the somewhat more elaborate procedure proposed in [69]. But it is easy to provide examples of ill-conditioned Riccati equations where U_{11} is well-conditioned and so much more sophisticated analysis needs to be performed. A number of different condition numbers are compared and evaluated in [3] along with numerous examples which illustrate deficiencies in each known condition estimate. Research continues by several groups on this topic.

A software package for Riccati equations called RICPACK has been developed by this author and Arnold essentially as a research tool to study conditioning of Riccati equations. Highlights of the capabilities of this general software (in Fortran) include the following:

i) Ward's balancing [94] for the generalized eigenvalue problem

ii) direct handling of singular control weighting or measurement noise covariance

iii) direct handling of cross-weighting or noise correlation

iv) direct handling of descriptor variable systems

v) iterative refinement by Newton's method and Sylvester equations

vi) residual calculations and various condition estimates.

Further details can be consulted in [3] and [4].

V. MATHEMATICAL SOFTWARE

A. General Remarks

The previous two sections have highlighted some topics from numerical linear algebra and their applications to numerical problems arising in systems, control, and estimation theories. Of course, these problems represent only a very small subset of numerical problems of interest in these fields but even for problems which are apparently "simple" from a mathematical point of view, the myriad of little details which constitute a sophisticated implementation become so overwhelming that the only effective means of communicating an algorithm is through its embodiment as mathematical software. Mathematical or numerical software simply means an implementation on a computing machine of an algorithm for solving a mathematical problem. Ideally, such software would be reliable, portable, and unaffected by the machine or system environment in which it is used.

The prototypical work on reliable, portable mathematical software for the standard eigenproblem was started in 1968. EISPACK [27], [82] Editions I and II were an outgrowth of that work. Subsequent efforts of interest to control engineers include LINPACK [17] for linear equations and linear least squares problems, FUNPACK (Argonne) [10] for certain function evaluations, MINPACK (Argonne) [67] for certain optimization problems, ROSEPACK (developed by Klema and others and available from International Mathematical and Statistical Libraries, Inc., Houston, TX) for robust statistical computations, and various ODE and PDE codes. High quality algorithms are published regularly in the ACM *Transactions on Mathematical Software*.

Moreover, many preprocessors that are, themselves, portable software have been designed and implemented to assist in instituting and verifying portability. Among such machine aids that are in use are the PFORT verifier [79] from Bell Laboratories and the Fortran Converter from International Mathematical and Statistical Libraries, Inc. [1]. Also available are machine aids for modifying and producing Fortran source code such as TAMPR [7] and POLISH [18]. Technology to aid in the development of mathematical software in Fortran is being assembled as a package called TOOLPACK [14], [59]. Mechanized code development offers other advantages with respect to, for example, modifications, updates, versions, and maintenance. Excellent references on portability and other aspects of mathematical software include [13], [24], [37], [59], and [75].

Inevitably numerical algorithms are strengthened when their mathematical software is made portable since their widespread use is greatly facilitated. Furthermore, such software has been shown to be markedly faster by factors ranging from 10 to 50 than earlier and less reliable code.

One can list many other features besides portability, reliability, and efficiency which are characteristic of "good" mathematical software. For example, one can include:

1) high standards of documentation and style so as to be easily understood and used

2) ease of use; ability of the user to interact with the algorithm

3) consistency/compatibility/modularity in the context of a larger package or more complex problem

4) error control, exception handling

5) robustness with respect to unusual situations

6) graceful performance degradation as problem domain boundaries are approached

7) appropriate program size (a function of intended use, e.g., low accuracy, real-time applications)

8) availability and maintenance

9) "tricks" such as underflow-/overflow- proofing if necessary and implementation of columnwise or rowwise linear algebra [62].

Clearly, the list can go on.

What becomes apparent from the above considerations is that the evaluation of mathematical software is a highly nontrivial task; see, for example, [24]. Clearly, the quality of software is largely a function of its operational specification. It must also reflect the numerical aspects of the algorithm being implemented. The language used and the compiler (e.g., optimizing or not) used for that language will both have an enormous impact on quality—both perceived and real—as will the underlying hardware and arithmetic. Different implementations of the same algorithm can have markedly different properties and behavior. Further discussions on this subject can be found in [15], [24], [37], and [59].

B. Mathematical Software in Control

Many aspects of systems, control, and estimation theory are ready for the research and design that is necessary to produce reliable, portable mathematical software that performs in finite arithmetic. Certainly many of the underlying linear algebra tools (for example, in EISPACK and LINPACK) are considered sufficiently reliable as to be used as black—or at least gray—boxes by control engineers. Much of that theory and methodology can and has been carried over to control problems. However, much of the work done in control, particularly the design and synthesis aspects, is simply not amenable to nice, "clean" algorithms and the ultimate software must have the capability to enable a dialogue between the computing machine and the control engineer but with the latter probably still making the final engineering decisions. We might never see a "control package" that will look like EISPACK or LINPACK. To even attempt it would be futile. Instead, a better analogy would be made by trying to emulate a good ODE or PDE package.

What mathematical software can provide is a "toolbox" from which the control engineer can choose software tools and robustly coded algorithms to implement easily new or modified theories or designs. Mathematical software forms the foundation of a computer-aided control system design (CACSD) package, but is only one of many interlocking parts.

Most CACSD packages—and there are now hundreds of them—divide fairly naturally into two fundamental levels, each with further subdivisions, of course. The lower level contains the numerical software and this can be written very portably. The upper level contains the basic control design and analysis procedures which are carried out by calling the low level procedures or subroutines for their actual implementation. The upper level also contains the most crucial part, as far as the control engineer is concerned, and that is the user interface. This interface, which can benefit from good human engineering considerations, interacts with the upper level procedures as well as the I/O, graphics, and file- and data-base management systems. Here the question of portability is considerably more complex and most packages aim for particular "target environments." For further comments and examples, see [12], [39].

Finally, we mention a few recent developments which are relevant to future control and estimation software developments. With respect to languages, Fortran is likely to remain the most common (and very efficient) language for quite some time. Fortran coding and portability is certain to be aided by the adoption and use of the Fortran 77 standard (ANSI X3.9-1978) and work continues on Fortran 8X. We may also see some movement towards Ada™. The use of other languages such as Pascal, C, PL/1, and APL seems somewhat more limited despite their attractiveness in particular environments. In hardware we are seeing more and more microprocessor-based systems. Parallel and pipeline architectures will both demand and suggest new algorithms and control strategies. Regarding arithmetic, the implementation of IEEE arithmetic in some computing environments could have a major beneficial impact on mathematical software. In graphics substantial progress is being made towards development of standards and graphics, with its enormous potentials, stands out as a highly under-utilized tool for CACSD environments.

VI. CONCLUDING REMARKS

Some numerical issues and techniques from numerical linear algebra together with some applications of these ideas have been outlined. A key question in these and other problems in systems, control, and estimation theory is what can be reliably computed and used in the presence of parameter uncertainty or structural constraints (e.g., certain "hard zeros") in the original model and roundoff errors in the calculations. However, the ultimate goal is to solve real problems and reliable tools (mathematical software) and experience must be available to effect real solutions or strategies. Only a serious interdisciplinary effort is capable of making substantial progress in improving the present state of affairs. As we move out of the "just programming" era we can soon expect to see some high quality control software. We have already witnessed a fruitful symbiosis between numerical analysis and numerical problems from control. We can expect a further symbiotic relationship as control engineering realizes the full potential of graphics, "cheap" memory, and substantial computing power. However, as in other applications areas, software will continue to dominate, both as a constraint and as a vehicle for progress. Unfortunately, exceptionally high quality software is exceptionally expensive, in terms of both money and time.

REFERENCES

[1] T. J. Aird, *The Fortran Converter User's Guide.* IMSL, 1975.

[2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[3] W. F. Arnold, "On the numerical solution of algebraic Riccati equations," Ph.D. dissertation, Dep. Elec. Eng.-Syst., Univ. Southern Calif., Dec. 1983.

[4] W. F. Arnold and A. J. Laub, "A software package for the solution of generalized algebraic Riccati equations," in *Proc. 22nd IEEE Conf. Decision Contr.*, San Antonio, TX, Dec. 1983.

[5] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation $AX + XB = C$," *Commun. Ass. Comput. Mach.*, vol. 15, pp. 820–826, 1972.

[6] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation.* New York: Academic, 1977.

[7] J. Boyle and K. Dritz, "An automated programming system to aid the development of quality mathematical software," in *IFIP Proceedings.* Amsterdam, The Netherlands: North-Holland, 1974, pp. 542–546.

[8] R. W. Brockett, *Finite Dimensional Linear Systems.* New York: Wiley, 1970.

[9] A. K. Cline, *et al.*, "An estimate for the condition number of a matrix," *SIAM J. Numer. Anal.*, vol. 16, pp. 368–375, 1979.

[10] W. J. Cody, " The FUNPACK package of special function subroutines," *ACM Trans. Math. Software*, vol. 1, pp. 13–25, 1975.

[11] *Computer*, vol. 14, Mar. 1981.

[12] *Contr. Syst. Mag.* (Special Issue on CACSD), vol. 2, Dec. 1982.

[13] W. Cowell, *Portability of Numerical Software*, Oak Brook, 1976" (Lecture Notes Comput. Sci., Vol. 57). New York: Springer-Verlag, 1977.

[14] W. Cowell and L. J. Osterweil, "The Toolpack/IST programming environment," Argonne National Lab., Appl. Math. Div., 1983, Rep. ANL/MCS-TM7.

[15] H. Crowder, R. S. Dembo, and J. M. Mulvey, "On reporting computational experiments with mathematical software," *ACM Trans. Math. Software*, vol. 5, pp. 193–203, 1979.

[16] G. Dahlquist and A. Bjorck, *Numerical Methods.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

™ Ada is a trademark of the U.S. Department of Defense.

[17] J. Dongarra *et al.*, *LINPACK Users' Guide.* Philadelphia, PA: SIAM, 1979.

[18] J. Dorrenbacker, D. Paddock, D. Wisneski, and L. D. Fosdick, "POLISH, A Fortran program to edit Fortran programs," Dep. Comput. Sci., Univ. Colorado, Boulder, 1974, Rep. CU-CS-050-74.

[19] J. C. Doyle and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 4–16, 1981.

[20] A. Emami-Naeini and P. VanDooren, "Computation of zeros of linear multivariable systems," *Automatica*, vol. 18, pp. 415–430, 1982.

[21] W. Enright, "On the efficient and reliable numerical solution of large linear systems of ODE's," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 905–908, 1979.

[22] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1967.

[23] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations.* Englewood Cliffs, NJ: Prentice-Hall, 1977.

[24] L. D. Fosdick, Ed., *Performance Evaluation of Numerical Software.* Amsterdam, The Netherlands: North-Holland, 1979.

[25] J. G. F. Francis, "The QR transformation I," *Comput. J.*, vol. 4, pp. 265–271, 1961; and "The QR transformation II," *Comput. J.*, vol. 4, pp. 332–345, 1962.

[26] D. Gannon, "A note on pipelining a mesh connected microprocessor for finite element problems by nested dissection," in *Proc. 1980 Int. Conf. Parallel Proc.*, 1980, pp. 197–204.

[27] B. S. Garbow *et al.*, *Matrix Eigensystem Routines—EISPACK Guide Extension* (Lecture Notes in Comput. Sci., Vol. 51). New York: Springer-Verlag, 1977.

[28] M. Gentleman and H. T. Kung, "Matrix triangularization by systolic arrays," *Proc. SPIE*, vol. 298, 1981.

[29] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials.* New York: Academic, 1982.

[30] G. H. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *SIAM J. Numer. Anal.*, vol. 2, pp. 205–224, 1965.

[31] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math*, vol. 14, pp. 403–420, 1970.

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

[33] G. H. Golub, V. C. Klema, and G. W. Stewart, "Rank degeneracy and least squares problems," Dep. Comput. Sci., Stanford Univ., Tech. Rep. STAN-CS-76-559, Aug. 1976 (also issued as NBER Rep. and Univ. Maryland, Rep.)

[34] G. H. Golub, S. Nash, and C. F. Van Loan, "A Hessenberg-Schur method for the problem $AX + XB = C$," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 909–913, 1979.

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

[36] D. E. Heller and I. C. F. Ipsen, "Systolic networks for orthogonal equivalence transformations and their applications," in *Proc. Conf. Advanced Research in VLSI*, P. Penfield, Ed., Mass. Inst. Technol., Cambridge, Jan. 1982, pp. 113–122.

[37] M. A. Hennel and L. M. Delves, Eds., *Production and Assessment of Numerical Software.* New York: Academic, 1980.

[38] P. Henrici, *Essentials of Numerical Analysis.* New York: Wiley, 1982.

[39] C. J. Herget and A. J. Laub, "Editorial," *Contr. Syst. Mag.*, vol. 2, Dec. 1982.

[40] A. S. Householder, *The Theory of Matrices in Numerical Analysis.* New York: Balisdell, 1964.

[41] R. L. Johnston, *Numerical Methods: A Software Approach.* New York: Wiley, 1982.

[42] B. Kågström, "Bounds and perturbation bounds for the matrix exponential," *BIT*, vol. 17, pp. 39–57, 1977.

[43] B. Kågström and A. Ruhe, Eds., *Matrix Pencils* (Lecture Notes in Math. 973) New York: Springer-Verlag, 1983.

[44] V. C. Klema and A. J. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 164–176, 1980.

[45] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems.* New York: Wiley, 1972.

[46] A. J. Laub, "A Schur method for solving algebraic Riccati equations," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 913–921, 1979.

[47] ——, "ALGORITHM: Efficient calculation of frequency response matrices from state space models," *ACM Trans. Math. Software*, June 1982.

[48] A. J. Laub and B. C. Moore, "Calculation of transmission zeros using *QZ* techniques," *Automatica*, vol. 14, pp. 557–566, 1978.

[49] A. J. Laub, "Efficient multivariable frequency response calculations," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 407–408, 1981.

[50] ——, "On computing "balancing" transformations," in *Proc. 1980 JACC*, San Francisco, CA, 1980, pp. FA8-E.

[51] ——, "Robust stability of linear systems—Some computational considerations," in *Information Linkage Between Applied Mathematics and Industry II*, A. L. Schoenstadt *et al.*, Eds. New York: Academic, 1980, pp. 57–84.

[52] ——, "Schur techniques for Riccati differential equations," in *Feedback Control of Linear and Nonlinear Systems*, D. Hinrichsen and A. Isidori, Eds. New York: Springer-Verlag, 1982, pp. 165–174.

[53] ——, "Schur techniques in invariant imbedding methods for solving two-point boundary value problems," in *Proc. 21st Conf. Decision Contr.*, Orlando, FL, Dec. 1982, pp. 56–61.

[54] ——, "Survey of computational methods in control theory," in *Electric Power Problems: The Mathematical Challenge*, A. M. Erisman, *et al.*, Eds. Philadelphia, PA: SIAM, 1980, pp. 231–260.

[55] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice-Hall. 1974.

[56] K. H. Lee, "Generalized eigenproblem structures and solution methods for Riccati equations," Ph.D. dissertation, Dep. Elec. Eng.-Syst., Univ. Southern California, Jan. 1983.

[57] R. E. Lord, J. S. Kowalik, and S. P. Kumar, "Solving linear algebraic equations on a MIMD computer," in *Proc. 1980 Int. Conf. on Parallel Proc.*, 1980, pp. 205–210.

[58] R. S. Martin and J. H. Wilkinson, "Similarity reduction of a general matrix to Hessenberg form," *Numer. Math.*, vol. 12, pp. 349–368, 1968.

[59] P. Messina and A. Murli, Eds., *Problems and Methodologies in Mathematical Software Production* (Lecture Notes in Comput. Sci., No. 142). New York: Springer-Verlag, 1982.

[60] W. Miller and C. Wrathall, *Software for Roundoff Analysis of Matrix Algorithms.* New York: Academic, 1980.

[61] G. S. Miminis and C. C. Paige, "An algorithm for pole assignment of time-invariant multi-input linear systems," in *Proc. 21st Conf. Decision Contr.*, Orlando, FL, Dec. 1982, pp. 62–67.

[62] C. B. Moler, "Matrix computations with Fortran and paging," *Comm. Ass. Comput. Mach.*, vol. 15, pp. 268–270, 1972.

[63] C. B. Moler and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems," *SIAM J. Numer. Anal.*, vol. 10, pp. 241–256, 1973.

[64] C. B. Moler and C. F. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Rev.*, vol. 20, pp. 801–837, 1978.

[65] B. C. Moore and A. J. Laub, "Computation of supremal (A, B)-invariant and controllability subspaces," *IEEE Trans. Automat. Contr.*, vol. AC-23, pp. 783–792, 1978.

[66] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 17-32, 1981.

[67] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *User Guide for MINPACK-1*, Argonne National Lab., Aug. 1980, Rep. ANL-80-74.

[68] A. M. Ostrowski, "On the spectrum of a one-parametric family of matrices," *Journal für die reine und angewandte Mathematik*, band 193, heft 3/4, pp. 143–160, 1954.

[69] C. C. Paige and C. F. Van Loan, "A Schur decomposition for Hamiltonian matrices," *Linear Algebra and Its Appl.*, vol. 41, pp. 11–32, 1981.

[70] C. C. Paige, "Properties of numerical algorithms related to computing controllability," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 130–138, 1981.

[71] T. Pappas, A. J. Laub, and N. R. Sandell, "On the numerical solution of the discrete-time algebraic Riccati equation," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 631–641, 1980.

[72] B. N. Parlett, *The Symmetric Eigenvalue Problem.* Englewood Cliffs, NJ: Prentice-Hall, 1980.

[73] R. V. Patel, "Computational algorithms for pole assignment in linear multivariable systems," in *Proc. 2nd IFAC Symp. on Comp.-Aided Design of Multivariable Tech. Syst.*, Purdue Univ., West Lafayette, IN, Sept. 1982, pp. 79–89.

[74] P. Hr. Petkov, N. D. Christov, and M. M. Konstantinov, "A computational algorithm for pole assignment of linear single-input systems," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 1045–1048, Nov. 1983.

[75] J. K. Reid, Ed., *The Relationship Between Numerical Computation and Programming Languages.* Amsterdam, The Netherlands: North-Holland, 1982.

[76] J. R. Rice, "A theory of condition," *SIAM J. Numer. Anal.*, vol. 3, pp. 287–310, 1966.

[77] ——, *Matrix Computations and Mathematical Software.* New York: McGraw-Hill, 1981.

[78] ——, *Numerical Methods, Software, and Analysis.* New York: McGraw-Hill, 1983.

[79] B. G. Ryder, *The PFORT Verifier: User's Guide*, Bell Labs, 1975 CS. Tech. Rep. 12; and *Software Practice and Experience*, vol. 4, pp. 359–377, 1974.

[80] M. G. Safonov, A. J. Laub, and G. L. Hartman, "Feedback properties of multivariable systems: The role and use of the return difference matrix," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 47–65, 1981.

[81] A. H. Sameh and D. J. Kuck, "On stable parallel linear system solvers," *J. Ass. Comput. Mach.*, vol. 25, pp. 81–91, 1978.

[82] B. T. Smith *et al.*, *Matrix Eigensystem Routines—EISPACK Guide*, 2nd ed. (Lecture Notes in Comput. Sci., Vol. 6). New York: Springer-Verlag, 1976.

[83] G. W. Stewart, *Introduction to Matrix Computations.* New York: Academic, 1973.

[84] ——, "On the perturbation of pseudo-inverses, projections, and linear least squares," *SIAM Rev.*, vol. 19, pp. 634–662, 1977.

[85] ——, "The economic storage of plane rotations," *Numer. Math.*, vol. 25, pp. 137–138, 1976.

[86] ——, "Rank degeneracy," *SIAM J. Sci. Stat. Comput.*, vol. 5, pp. 403–413, 1984.

[87] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis.* New York: Springer-Verlag, 1980.

[88] P. Van Dooren, "A generalized eigenvalue approach for solving Riccati equations," *SIAM J. Sci. Stat. Comput.*, vol. 2, pp. 121–135, 1981.

[89] ——, "Numerical linear algebra: An increasing interest in linear system theory," in *Proc. Europe Conf. Cir. Theory and Design 81*, The Hague, The Netherlands, Aug. 1981, pp. 243–251.

[90] ——, "The generalized eigenstructure problem. Applications in linear system theory," Ph.D. dissertation, Katholieke Univ. Leuven, Leuven, Belgium, 1979.

[91] ——, "The generalized eigenstructure problem in linear system theory," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 111–129, 1981.

[92] C. F. Van Loan, "The sensitivity of the matrix exponential," *SIAM J. Numer. Anal.*, vol. 14, pp. 971–981, 1977.

[93] A. Varga, "A Schur method for pole assignment," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 517–519, 1981.

[94] R. C. Ward, "Balancing the generalized eigenvalue problem," *SIAM J. Sci. Stat. Comput.*, vol. 2, pp. 141–152, 1981.

[95] J. H. Wilkinson and C. Reinsch, *Handbook for Automatic Computation, Vol. II, Linear Algebra.* New York: Springer-Verlag, 1971.

[96] J. H. Wilkinson, *Rounding Errors in Algebraic Processes.* Englewood Cliffs, NJ: Prentice-Hall, 1963.

[97] ——, *The Algebraic Eigenvalue Problem.* London, England: Oxford University Press, 1965.

[98] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 2nd ed. New York: Springer-Verlag, 1979.

Alan J. Laub (M'75–SM'82) received the B.Sc. degree in mathematics from the University of British Columbia in 1969, the M.S. degree in mathematics from the University of Minnesota in 1972, and the Ph.D. degree in control sciences from the University of Minnesota in 1974.

From 1974 to 1983 he taught at Case Western Reserve University, Cleveland, OH, the University of Toronto, Toronto, Ont., Canada, the Massachusetts Institute of Technology, Cambridge, and the University of Southern California. His research interests are in numerical analysis, mathematical software, computer-aided control system design, and linear and large-scale systems theory. He has published numerous technical papers and has given several short courses in these areas. His algorithms and software are now used extensively and he has been influential in promulgating the use of numerically stable algorithms and reliable mathematical software in the control and systems community. He also serves as Consultant to several major companies and laboratories.

Dr. Laub has been an Associate Editor for the *International Journal of Control* and was Associate Editor for Computational Methods and Discrete Systems for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1979 to 1981.