Numerical Algorithms in Control

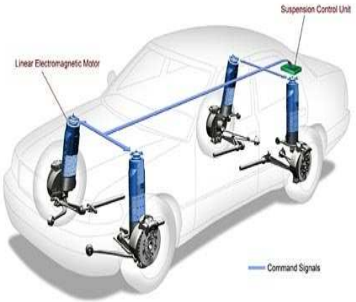
Zlatko Drmač

Department of Mathematics
University of Zagreb

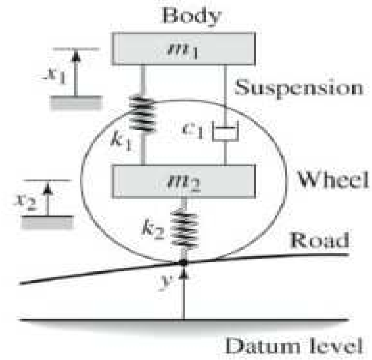
September 23.-27. 2013

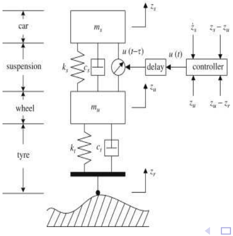
Outline

Example: Active car suspension



Physical model





Mathematical model

Newton equations

$$
\begin{aligned}
m_p\ddot{x}_p &= -k_p(x_p - x_s) - c_p(\dot{x}_p - \dot{x}_s) \\
m_s\ddot{x}_s &= -k_p(x_s - x_p) - c_p(\dot{x}_s - \dot{x}_p) - k_s(x_s - x_{us}) \\
&\quad -c_s(\dot{x}_s - \dot{x}_{us}) + f_a \\
m_{us}\ddot{x}_{us} &= -k_s(x_{us} - x_s) - c_s(\dot{x}_{us} - \dot{x}_s) - k_t(x_{us} - r) \\
&\quad -c_t(\dot{x}_{us} - \dot{r}) - f_a
\end{aligned}
$$

System of second order ODE's ; new variables:
$x_1 = x_p,\ x_2 = \dot{x}_1,\ x_3 = x_s,\ x_4 = \dot{x}_3,\ x_5 = x_{us},\ x_6 = \dot{x}_5.$

State space description

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_p}{m_p} & \frac{-c_p}{m_p} & \frac{k_p}{m_p} & \frac{c_p}{m_p} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_p}{m_s} & \frac{c_p}{m_s} & -\frac{k_s+k_p}{m_s} & -\frac{c_s+c_p}{m_s} & \frac{k_s}{m_s} & \frac{c_s}{m_s} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_s}{m_{us}} & \frac{c_s}{m_{us}} & -\frac{k_s+k_t}{m_{us}} & -\frac{c_s}{m_{us}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_s} & 0 \\ 0 & 0 \\ \frac{-1}{m_s} & \frac{k_t}{m_{us}} \end{pmatrix} \begin{pmatrix} f_a \\ r \end{pmatrix}$$

i.e.

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

$n = 5$ internal states $x_1, \ldots, x_5$;
state space matrix $A \in \mathbb{R}^{n\times n}$;
$m = 2$ inputs $u_1(t) = f_a(t)$, $u_2(t) = r(t)$;
input matrix $B \in \mathbb{R}^{n\times m}$

State space description

Not interested in (all) states, but only $x_1 \equiv x_p$, $\ddot{x}_p \equiv \dot{x}_2$:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \equiv \begin{pmatrix} x_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_p}{m_p} & \frac{-c_p}{m_p} & \frac{k_p}{m_p} & \frac{c_p}{m_p} & 0 & 0 \end{pmatrix} x(t)$$

i.e. interested in the output

$$\boxed{y = Cx + Du}$$

$p = 2$ outputs; $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$.
LTI system

$$\Sigma = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$ (In Matlab: `S=ss(A,B,C,D)`)

Task: passenger comfort

Write the LTI as

$$\dot{x} = Ax + \tilde{B}v + Gw, \quad \tilde{B} = B(:,1), \quad G = B(:,2), \quad v = f_a, \quad w = r$$

Determine the control input v to minimize

$$J = \int_0^\infty (x(t)^T Q x(t) + v^T(t) R v(t)) dt \longrightarrow \min \quad (Q \succeq 0, \quad R \succ 0)$$

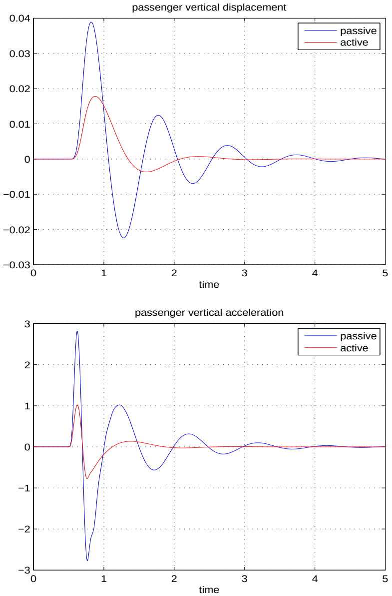
In our case, $y = Cx$, so choosing $Q = \gamma C^T C$ yields $x(t)^T Q x(t) = \gamma y(t)^T y(t)$ ($\gamma > 0$).
Assume $v(t) = -Kx(t)$ to obtain

$$\dot{x}(t) = Ax(t) - \tilde{B}Kx(t) + Gw(t) = (A - \tilde{B}K)x(t) + Gw(t)$$

Theorem: Optimal K is $K = R^{-1}\tilde{B}^T X$, where $X = X^T \succ 0$ solves the algebraic Riccati equation

$$XA + A^T X + Q - X\tilde{B}R^{-1}\tilde{B}^T X = 0$$

Simulation


passenger vertical displacement
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
0
1
2
3
4
5
time
passive
active
passenger vertical acceleration
3
2
1
0
−1
−2
−3
0
1
2
3
4
5
time
passive
active

LTI systems

Space station, CD player, vehicle suspension system, ...

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \quad E, A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m} \\ y(t) &= Cx(t) + Du(t), \quad C \in \mathbb{R}^{p\times n},\ D \in \mathbb{R}^{p\times m}. \end{aligned} \tag{1}$$

Given an initial $x_0 = x(t_0)$, the solution is

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

Assume $A = S\Lambda S^{-1}$ diagonalizable; $As_j = \lambda_j s_j$ then

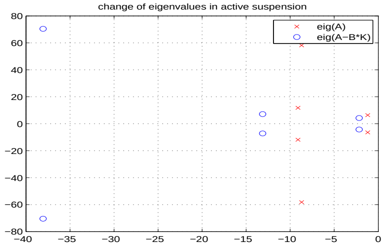
$$e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!}A^k = S \begin{pmatrix} e^{\lambda_1 t} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & e^{\lambda_n t} \end{pmatrix} S^{-1}$$

Hence (take $t_0 = 0$)

$$e^{At}x_0 = \sum_{j=1}^{n}(S^{-1}x_0)_j e^{\lambda_j t}s_j, \quad t \longrightarrow \infty$$

Stability, eigenvalue assignment

To ensure $\lim_{t\to\infty} e^{\lambda_j t} = 0$, need $\Re(\lambda_j) < 0$.
A is stable (Hurwitz) if $\Re(\lambda_j) < 0$, $j = 1, \ldots, n$.



If A is not stable, and $Bu = \tilde{B}v + Gw$, with v control input assumed as $v = -Kx$, feedback stabilization task is to find K such that $A - \tilde{B}K$ has prescribed (stable) eigenvalues.

Controllability

$$\text{LTI system } \mathcal{S} \quad :: \quad \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array}$$

is controllable if it can be driven with appropriate $u(t)$ from any initial state $x(0)$ to any state $x_1 = x(t_1)$ in finite time t_1.

Theorem: The following are equivalent:

1. $\mathcal{S}$ is controllable.
2. $C = \begin{pmatrix} B & AB & A^2B & \ldots & A^{n-1}B \end{pmatrix} \in \mathbb{R}^{n \times nm}$ has rank n.
3. $P(t) = \int_0^t e^{A\tau}BB^Te^{A^T\tau}d\tau \succ 0$ for any $t > 0$.
4. $x \neq 0$, $x^TA = \lambda x^T \Longrightarrow x^TB \neq 0$.
5. $\mathrm{rank}(\begin{pmatrix} A - \lambda I, & B \end{pmatrix}) = n$, λ any eigenvalue of A.
6. The eigenvalues of $A - BK$ can be arbitrarily placed with suitable K.

Task: Determine whether $\mathcal{S}$ is controllable.

Controllability

Theorem: Given x, the control $\tilde{u}(t) = B^T e^{A^T(t_1-t)} P(t_1)^{-1} x$ drives the system from $x(0) = 0$ into x in time t_1. For any other control u with same property,

$$\|u\|_2 \geq \|\tilde{u}\|_2 = \sqrt{x^T P(t_1)^{-1} x}.$$

Numerical procedure; single input ($m = 1, B = b$). Controller Hessenberg form:

$$Q^T A Q = H = \begin{pmatrix} * & * & * & * & * \\ \star & * & * & * & * \\ & \star & * & * & * \\ & & \star & * & * \\ & & & \star & * \end{pmatrix}, \quad Q^T b = \tilde{b} = \begin{pmatrix} \star \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Theorem: (A, b) is controllable iff

$$\tilde{b}_1 \prod_{i=1}^{n-1} H_{i+1,i} = \star \prod_{i=1}^{n-1} \star \neq 0$$

.

Observability

The LTI system $\mathcal{S}$ is observable if there exists t_1 such that $x(0)$ can be determined uniquely from $u(t)$, $y(t)$, $0 \leq t \leq t_1$.

Theorem: The following are equivalent to observability of $\mathcal{S}$:

1. $O = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}$ has rank n.
2. $Q(t) = \int_0^t e^{A^T\tau} C^T C e^{A\tau} d\tau \succ 0$ for any $t > 0$.
3. $\begin{pmatrix} \lambda I - A \\ C \end{pmatrix}$ has rank n for any eigenvalue λ of A
4. $Ay = \lambda y, y \neq 0 \Longrightarrow Cy \neq 0$.
5. The eigenvalues of $A - LC$ can be arbitrarily placed with suitable L.

Example: state estimation

$$\text{LTI system } \mathcal{S} \quad :: \quad \begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

Want an estimate $\hat{x}(t)$ of $x(t)$ such that $\epsilon(t) = x(t) - \hat{x}(t) \to 0$ as fast as possible for any initial $x(0)$ and every input $u(t)$.
Build a new system (L not yet specified)

$$\text{LTI system } \widehat{\mathcal{S}} \quad :: \quad \dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + L\overbrace{y(t)}^{Cx(t)} + Bu(t)$$

Note that

$$\dot{\epsilon}(t) = \dot{x}(t) - \dot{\hat{x}}(t) = (A - LC)e(t), \quad \text{i.e.} \quad \epsilon(t) = e^{(A-LC)t}\epsilon(0)$$

If (A, C) is observable, can choose L to make $A - LC$ stable and thus

$$\lim_{t\to\infty} e(t) = 0, \quad \text{for any } e(0).$$

Grammians

Controllability Grammian

$$P = \int_0^\infty e^{At} BB^T e^{A^T t} dt, \quad AP + PA^T + BB^T = 0$$

and observability Grammian

$$Q = \int_0^\infty e^{A^T t} C^T C e^{At} dt, \quad QA + A^T Q + C^T C = 0$$

- directions of eigenvectors of small eigenvalues of P hard to control
- directions of eigenvectors of small eigenvalues of Q hard to observe
- Lyapunov equations hard to solve for large n.

Transfer function

LTI system $\mathcal{S}$ :: $\begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array}$, $x(0) = x_0$

Laplace transform

$$\begin{array}{l} s\hat{x}(s) - x_0 = A\hat{x}(s) + B\hat{u}(s) \\ \hat{y}(s) = C\hat{x}(s) + D\hat{u}(s) \end{array}$$

Let $G(s) = C(sI - A)^{-1}B + D$. $G()$ is called transfer function. Then $\hat{y}(s) = G(s)\hat{u}(s) + C(sI - A)^{-1}x(0)$. If $x(0) = 0$,

$$\hat{y}(s) = G(s)\hat{u}(s)$$

$$G(s) = \begin{pmatrix} G_{11}(s) & \dots & G_{1m}(s) \\ \vdots & \dots & \vdots \\ G_{p1}(s) & \cdots & G_{pm}(s) \end{pmatrix}$$ rational matrix function

$G(\zeta) = \infty$ – ζ is a pole of G; $G(\infty) = const.$ – G is proper
$G(\infty) = 0$ – G is strictly proper;

Frequency response

Let $G(s) = C(sI - A)^{-1}B + D$ be the TF.
For $\omega \in \mathbb{R}$ (frequency), the matrix

$$G(i\omega) = C(i\omega I - A)^{-1}B + D$$

is called frequency response.
Example (SISO): Let $u(t) = A\sin(\omega t + \vartheta)$. Then

$$\hat{u}(s) = \frac{A\omega\cos\vartheta}{s^2+\omega^2} + \frac{A\sin\vartheta - s}{s^2+\omega^2}$$

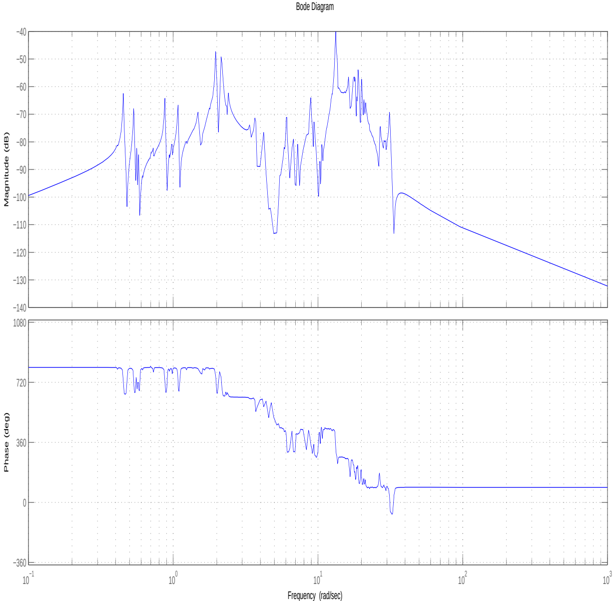
and at $t \longrightarrow \infty$

$$y(t) \approx A|G(\mathrm{i}\omega)|\sin(\omega t + \vartheta + \psi(\omega)), \quad G(\mathrm{i}\omega) = |G(\mathrm{i}\omega)|e^{\mathrm{i}\psi(\omega)}$$

Hence, important is

$$\sup_{\omega\in\mathbb{R}} |G(\mathrm{i}\omega)|, \quad \sup_{\omega\in\mathbb{R}} \|G(\mathrm{i}\omega)\|_2 \equiv \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(\mathrm{i}\omega)).$$

Bode plot: $G(i\omega) = |G(i\omega)| e^{i\psi(\omega)}$



$\mathcal{H}_\infty$

Consider complex functions f, analytic in

$$\mathbb{C}_+ = \{s \equiv x + \imath y \in \mathbb{C} \;:\; x > 0\}$$

and $|f(z)| \leq b$, $z \in \mathbb{C}_+$. Then

$$\|f\|_\infty = \sup\{|f(z)| \;:\; z \in \mathbb{C}_+\} < \infty; \;\; \mathcal{H}_\infty = \{f\}.$$

$\mathbb{R}\mathcal{H}_\infty \subset \mathcal{H}_\infty$, proper real rational functions
By the maximum principle

$$\|f\|_\infty = \sup\{|f(z)| \;:\; \Re(z) > 0\} = \sup\{|f(\imath\omega)| \;:\; \omega \in \mathbb{R}\}$$

Define (SISO) $\|G\|_\infty = \sup_{\omega\in\mathbb{R}} |G(\imath\omega)|$. In general (MIMO)

$$\|G\|_\infty = \sup_{\omega\in\mathbb{R}} \|G(\imath\omega)\|_2 \equiv \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(\imath\omega))$$

Hardy space $\mathcal{H}_2$

The space $\mathcal{H}_2^{p\times m}$ contains functions $H(s) = (h_{ij}(s)) \in \mathbb{C}^{p\times m}$, with all $h_{ij}(s)$ analytic in the open right half plane $\mathbb{C}_+$, and such that

$$\sup_{x>0} \int_{-\infty}^{+\infty} \|H(x+\mathrm{i}y)\|_F^2 dy < \infty.$$

$\mathcal{H}_2^{p\times m}$ is endowed with the Hilbert space structure with the inner product

$$\langle G, H\rangle = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathrm{trace}(H^*(\mathrm{i}\omega)G(\mathrm{i}\omega))d\omega, \quad \|G\|_2 = \sqrt{\langle G, G\rangle}. \tag{2}$$

Strictly proper transfer functions G, H of real stable LTI systems belong to $\mathcal{H}_2^{p\times m}$, and we have

$$\langle G, H\rangle = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathrm{trace}(H^T(-\mathrm{i}\omega)G(\mathrm{i}\omega))d\omega = \langle H, G\rangle$$

Internal balancing

$$\text{LTI system } \mathcal{S} \;::\; \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array}$$

Write the state in new coordinate system : $x(t) = T\tilde{x}(t)$.

$$\text{LTI system } \mathcal{S} \;::\; \begin{array}{l} \dot{\tilde{x}}(t) = T^{-1}AT\tilde{x}(t) + T^{-1}Bu(t) \\ y(t) = CT\tilde{x}(t) + Du(t) \end{array}$$

New Grammians; $\tilde{P} = T^{-1}PT^{-T}$; $\tilde{Q} = T^TQT$.
Let $P = L_cL_c^T$, $Q = L_oL_o^T$ be Cholesky factorizations, and $L_o^TL_c = U\Sigma V^T$ the SVD. Set $T = L_cV\Sigma^{-1/2}$. Then

$$\begin{aligned} \tilde{P} &= \Sigma^{1/2}V^TL_c^{-1}L_cL_c^TL_c^{-T}V\Sigma^{1/2} = \Sigma \\ \tilde{Q} &= \ldots = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n); \end{aligned}$$

$\sigma_1 \geq \cdots \geq \sigma_n$ are the **Hankel singular values.**
$(T^{-1}AT, T^{-1}B, CT)$ is balanced realization of $\mathcal{S}$.

Balanced truncation

$$\text{LTI system } \mathcal{S} \quad :: \quad \begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

Suppose the dimension n is big. Want to approximate $\mathcal{S}$ with $\tilde{\mathcal{S}}$ of dimension $r \ll n$.

- $\tilde{\mathcal{S}}$ must mimic the input-output behaviour of $\mathcal{S}$.
- Reducing dimension = reducing the number of states.
- Which states can be ignored without affecting the input-output relation too much?
- Consider the states hard to control and hard to observe. Suppose we had a way to determine them.
- If we somehow get rid of them, the new description of the system will not departure from the original system too much.

Balanced truncation

Theorem: Consider stable internally balanced LTIS

$$G(s) = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & 0 \end{array}\right], \quad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}$$

$\Sigma_1 = \mathrm{diag}(\sigma_1 I_{k_1}, \ldots, \sigma_\ell I_{k_\ell})$, $\Sigma_2 = \mathrm{diag}(\sigma_{\ell+1} I_{k_{\ell+1}}, \ldots, \sigma_q I_{k_q})$
$\sigma_1 > \ldots > \sigma_\ell > \sigma_{\ell+1} > \ldots > \sigma_q$. Then

$$G_r(s) = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & 0 \end{array}\right]$$

is balanced and stable, and

$$\|G - G_r\|_\infty \leq 2(\sigma_{\ell+1} + \cdots + \sigma_q)$$

Algorithm

- Input:

  LTI system $\mathcal{S} :: \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array}, \quad x(0) = x_0$

- Solve the Lyapunov equations

$$AP + PA^T + BB^T = 0\,; \;\; A^TQ + QA + C^TT = 0.$$

- Compute the Cholesky factors $P = L_c L_c^T$, $Q = L_o L_o^T$
- Compute the SVD $L_o^T L_c = U\Sigma V^T$
- $T = L_c V \Sigma^{-1/2}$
- $\tilde{A} = T^{-1}AT = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix}$; $\tilde{B} = T^{-1}B = \begin{pmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{pmatrix}$;

  $\tilde{C} = CT = \begin{pmatrix} \tilde{C}_1 & \tilde{C}_2 \end{pmatrix}$.

- Output: $(\tilde{A}_{11}, \tilde{B}_1, \tilde{C}_1)$

Algorithm: details

$$\tilde{A} = T^{-1}AT = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix}; \tilde{B} = T^{-1}B = \begin{pmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{pmatrix};$$

$$\tilde{C} = CT = \begin{pmatrix} \tilde{C}_1 & \tilde{C}_2 \end{pmatrix}; T = L_c V \Sigma^{-1/2}$$

$$\tilde{A} = T^{-1}AT = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix} \begin{pmatrix} \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix}, U = (U_1, U_2), V = (V_1, V_2)$$

From $L_o^T L_c = U \Sigma V^T$, $L_c^{-1} = V \Sigma^{-1} U^T L_o^T$.

Set $Z = L_c V_1 \Sigma_1^{-1/2}$, $S = L_o U_1 \Sigma_1^{-1/2}$.

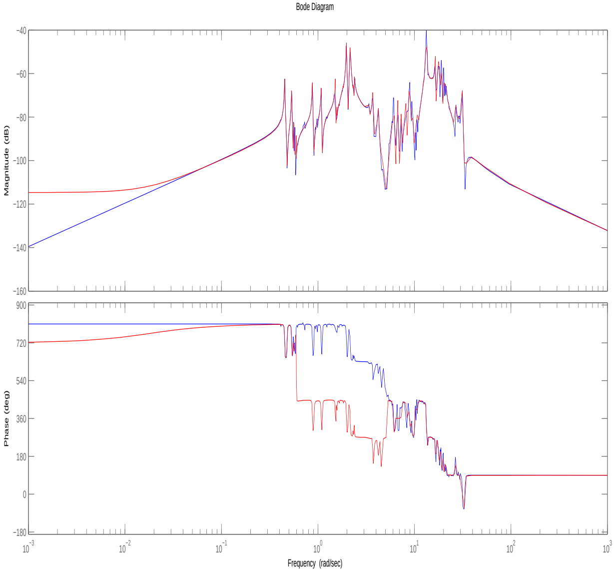
Then $\tilde{A}_{11} = S^T A Z$, $\tilde{B}_1 = S^T B$, $\tilde{C}_1 = CZ$.

Example: ISS

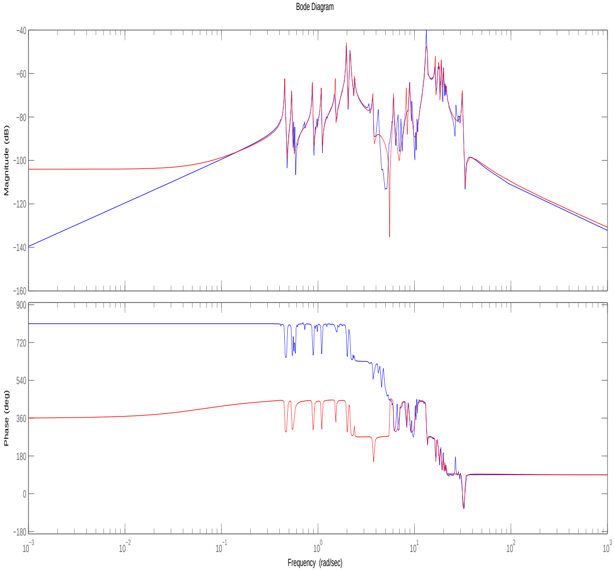
International Space Station, Russian module

Reduce from $n = 1412$ to $r = 100$, $r = 50$

ISS, $r = 100$


Bode Diagram
Magnitude (dB)
-40
-60
-80
-100
-120
-140
-160
Phase (deg)
900
720
540
360
180
0
-180
Frequency (rad/sec)

ISS, $r = 50$


Bode Diagram
Magnitude (dB)
-40
-60
-80
-100
-120
-140
-160
Phase (deg)
900
720
540
360
180
0
-180
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
Frequency (rad/sec)

Schur approach to balancing

$P = L_c L_c^T$, $Q = L_o L_o^T$, $L_o^T L_c = U\Sigma V^T$, $T = L_c V \Sigma^{-1/2}$.

$$\tilde{A} = T^{-1}AT = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix} \begin{pmatrix} \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{pmatrix}$$

$$PQT = L_c \underbrace{L_c^T L_o}_{V\Sigma U^T} \overbrace{L_o^T L_c}^{U\Sigma V^T} V\Sigma^{-1/2} = (L_c V \Sigma^{-1/2})\Sigma = T\Sigma$$

Hence $(PQ)T = T\Sigma$; $T^{-1}(PQ) = \Sigma T^{-1}$. Hence, the needed parts of T and T^{-1} span the dominant left and right spectral subspaces of PQ, and

$$\begin{pmatrix} * & * & * & * \\ * & * & * & * \end{pmatrix} \begin{pmatrix} \star & \star \\ \star & \star \\ \star & \star \\ \star & \star \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_r .$$

Idea: Use the Schur form to compute those subspaces.

Schur approach to balancing

Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the Hankel SV.

- Compute the ordered Schur forms

$$U^T(PQ)U = \begin{pmatrix} \lambda_1 & * & * \\ & \ddots & * \\ & & \lambda_n \end{pmatrix}, \; V^T(PQ)V = \begin{pmatrix} \lambda_n & * & * \\ & \ddots & * \\ & & \lambda_1 \end{pmatrix}$$

(In Matlab, use `schur` and `ordschur`.)

- Partition $U = (U_1, U_2)$, $V = (V_1, V_2)$.
- Compute the SVD $U_2^T V_1 = Q \Sigma R^T$
- Set $S_1 = U_2 Q \Sigma^{-1/2}$, $S_2 = V_1 R \Sigma^{-1/2}$
- $\tilde{A}_{11} = S_1^T A S_2$, $\tilde{B}_1 = S_1^T B$, $\tilde{C}_1 = C S_2$

$(\tilde{A}_{11}, \tilde{B}_1, \tilde{C}_1)$ gives the same TF as before. It is not balanced.

Rational LS approximation

Consider rational approximation of the transfer function $H(\zeta) = c^T(\zeta I - A)^{-1}b$ of a SISO linear time invariant dynamical system. If H is sampled/measured at the nodes σ_i, and we want to interpolate those values with the rational function in barycentric form

$$H_r(\zeta) = \frac{\sum_{i=1}^{r} \frac{\phi_i}{\zeta - \lambda_i}}{\sum_{i=1}^{r} \frac{\varphi_i}{\zeta - \lambda_i} + 1}, \quad H_r(\sigma_i) = H(\sigma_i), \quad i = 1, \ldots, \ell, \qquad (3)$$

then, for given poles λ_i and the nodes $(\sigma_i, H(\sigma_i))$, the parameters ϕ_i, φ_i are obtained by solving the least squares problem $\|Ax - h\|_2 \to \min$, where

$$\begin{aligned} h &= \begin{pmatrix} H(\sigma_1) & H(\sigma_2) & \cdots & H(\sigma_\ell) \end{pmatrix}^T \qquad (4) \\ x &= \begin{pmatrix} \phi_1 & \phi_2 & \cdots & \phi_r & \varphi_1 & \varphi_2 & \cdots & \varphi_r \end{pmatrix}^T . \end{aligned}$$

Vector fitting

$$A = \begin{pmatrix} \frac{1}{\sigma_1-\lambda_1} & \frac{1}{\sigma_1-\lambda_2} & \cdots & \frac{1}{\sigma_1-\lambda_r} & \frac{-H(\sigma_1)}{\sigma_1-\lambda_1} & \frac{-H(\sigma_1)}{\sigma_1-\lambda_2} & \cdots & \frac{-H(\sigma_1)}{\sigma_1-\lambda_r} \\ \frac{1}{\sigma_2-\lambda_1} & \frac{1}{\sigma_2-\lambda_2} & \cdots & \frac{1}{\sigma_2-\lambda_r} & \frac{-H(\sigma_2)}{\sigma_2-\lambda_1} & \frac{-H(\sigma_2)}{\sigma_2-\lambda_2} & \cdots & \frac{-H(\sigma_2)}{\sigma_2-\lambda_r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\sigma_\ell-\lambda_1} & \frac{1}{\sigma_\ell-\lambda_2} & \cdots & \frac{1}{\sigma_\ell-\lambda_r} & \frac{-H(\sigma_\ell)}{\sigma_\ell-\lambda_1} & \frac{-H(\sigma_\ell)}{\sigma_\ell-\lambda_2} & \cdots & \frac{-H(\sigma_\ell)}{\sigma_\ell-\lambda_r} \end{pmatrix}$$

Note that the matrix A has a paired Cauchy structure,

$$A = \begin{pmatrix} \mathcal{C} & D_\sigma \mathcal{C} \end{pmatrix}, \quad \mathcal{C} = \left(\frac{1}{\sigma_i - \lambda_j} \right)_{i,j=1}^{\ell,r}, \quad D_\sigma = -\operatorname{diag}(H(\sigma_i))_{i=1}^{\ell} \tag{5}$$

Numerical tasks

Generate many interesting and challenging problems.

- Simple questions, difficult answers: Compute the transfer function $G(\zeta) = C(\zeta E - A)^{-1}B$ for many complex values of ζ. Here n can be large. Reduce to generalized upper Hessenberg form

$$Q^T E Z = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare \\ & & \blacksquare & \blacksquare \\ & & & \blacksquare \end{pmatrix},\; Q^T A Z = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare & \blacksquare \\ & & \blacksquare & \blacksquare \end{pmatrix},\; Q^T b = \begin{pmatrix} \blacksquare \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

  and work on $(E, A, b, c) \equiv (Q^T EZ, Q^T AZ, Q^T b, cZ)$ is efficient. Simpler if $E = I$, $Z = Q$.
- By changing the state space coordinates, $x(t) = T\hat{x}(t)$, the new representation is, e.g. for $E = I$, given with $(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (T^{-1}AT, T^{-1}B, CT, D)$. Find T such that the new representation reveals structural properties of the system. Various canonical forms.
- Solve Lyapunov equation $AH + HA^T + BB^T = 0$. Solve Riccati eqn: $XA + A^T X + Q - XSX = 0$.

... algorithms, software

- Solve eigenvalue and singular value problems.
- Find invariant subspace that corresponds to specified eigenvalues.
- Given A with eigenvalues $\lambda_1, \ldots, \lambda_n$ and B, find K such that $A - BK$ has prescribed eigenvalues $\alpha_1, \ldots, \alpha_n$.
- Pressure from applications to deliver accurate solutions quickly. Computing environments changing rapidly.
- Users from applied sciences and engineering – usually not interested in math details, just solutions, software.
- Pure mathematicians not interested because the problems are "trivial", non–fundamental or just too messy.
- And we have high performance computers. So, why is this difficult?

Yes, have computer, but ..

Machine (floating–point) numbers $\mathbb{F} \subset \mathbb{Q}$.

$$f = \pm m \cdot 2^e, \quad e = -126 : 127, \quad m = 1.z_1 \ldots z_{23}.$$

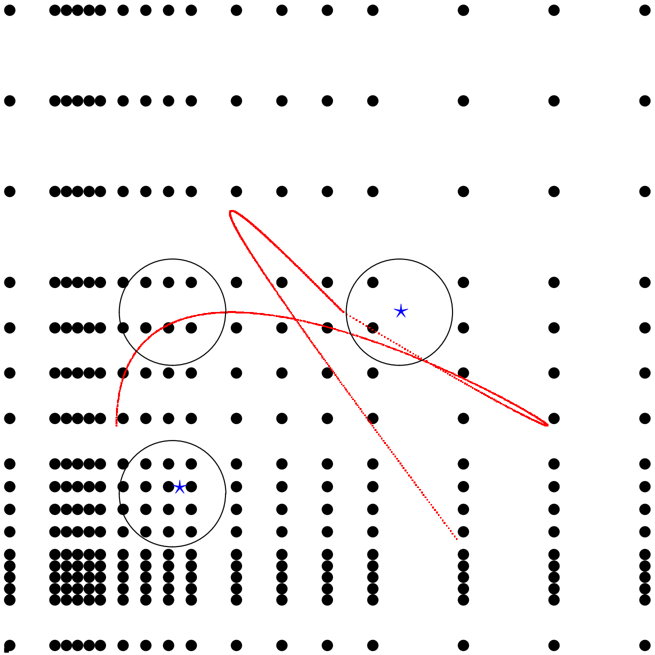
$$\overline{\mathbb{F}} = \mathbb{F} \bigcup \{+\texttt{Infinity}, -\texttt{Infinity}, \texttt{NaN}\}$$

Machine arithmetic $\oplus$, $\ominus$, $\odot$, $\oslash$.

- $\overline{\mathbb{F}}$ finite, 2^{32} (single), 2^{64} (double); $0.1 \not\in \mathbb{F}$;
- $a \oplus b \equiv \mathbf{FL}(a+b) = (a+b)(1+\epsilon_{a,b})$,

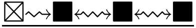
$$|\epsilon_{a,b}| \leq \mathbf{u} \equiv \texttt{eps} = \textbf{round-off} \approx 10^{-8}.$$

- In general, $(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$, $(a \odot b) \odot c \neq a \odot (b \odot c)$; $x \oplus y \oplus z =$??
- $1 \oplus 10^{-9} = 1$; $x = y \not\Leftrightarrow x - y = 0$; $10^{-30} \odot 10^{-30} = 0$;
- Finite speed, finite memory.
- Faster $\Longrightarrow$ more mess per second.



$(0, 0)$

Early loss of definiteness

The stiffness matrix of a mass spring system with 3 masses
 with spring constants $k_1 = k_3 = 1$, $k_2 = \varepsilon/2$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{pmatrix}, \quad \lambda_{\min}(K) \approx \varepsilon/4.$$

The true and the computed assembled matrix are

$$K = \begin{pmatrix} 1 + \frac{\varepsilon}{2} & -\frac{\varepsilon}{2} & 0 \\ -\frac{\varepsilon}{2} & 1 + \frac{\varepsilon}{2} & -1 \\ 0 & -1 & 1 \end{pmatrix}, \quad \tilde{K} = \begin{pmatrix} 1 & -\frac{\varepsilon}{2} & 0 \\ -\frac{\varepsilon}{2} & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

$\tilde{K}$ is component–wise relative perturbation of K with

$$\max_{i,j} \frac{|\tilde{K}_{ij} - K_{ij}|}{|K_{ij}|} = \frac{\varepsilon}{(2+\varepsilon)} < \varepsilon/2.$$

$\tilde{K}$ is indefinite with $\lambda_{\min}(\tilde{K}) \approx -\varepsilon^2/8$. Too late for $\lambda_{\min}(K)$. :(

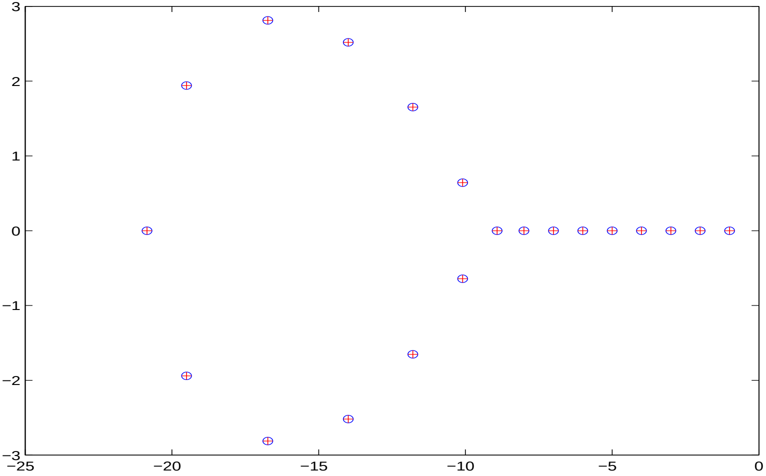
Consequences

Almost never have exactly given data. Have $A \in \mathbb{F}^{m\times n}$ as approximation of an ideal, not accessible A_0, $A = A_0 + E$. Do not have E, but know that $\|E\|/\|A\| \approx f(m,n)\mathbf{u}$ is small. $A \in \mathcal{B} = \{A_0 + E,\ \|E\| \leq \varepsilon\}$ and any $X \in \mathcal{B}$ is just as good as A.

- Full rank matrices dense in $\mathbf{M}_{m\times n}$. What is then the rank of $A_0 = A - E$? Rank of A? Any technique will fail over $\mathbb{F}$.
- Chance to compute zero exactly is exactly zero.
- Matrices with simple eigenvalues dense in $\mathbf{M}_{n\times n}$. Jordan form? Diagonalizability?
- Is A +definite? Invertible? Orthogonal? Stable ($\mathrm{Re}(\lambda(A) < 0)$)? $A^{-1} =?$ $A^{\dagger} =?$
- In 1950 Goldstine and von Neuman concluded that solving linear systems with $n > 15$ with guaranteed accuracy would be nearly impossible!

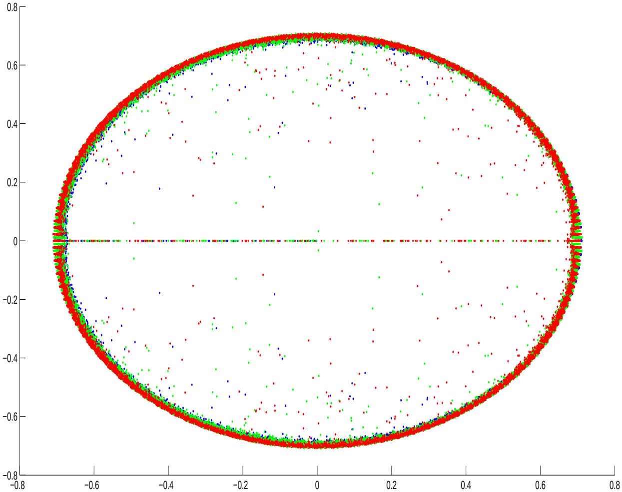
Roots of polynomials

Wilkinson's example: change 210 to $210 + 2^{-23}$ in $p(x) = \prod_{i=1}^{20}(x+i) = x^{20} + 210x^{19} + \cdots + 20!$



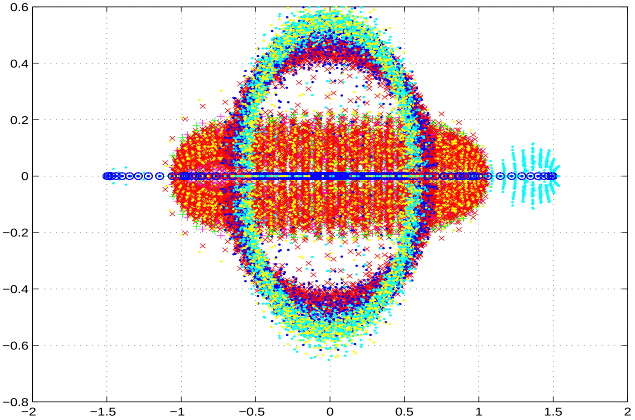
Eigenvalues of Jordan block

Matlab: `eig(J + eps*randn(100,100))`; $J = J_{100}(0)$.



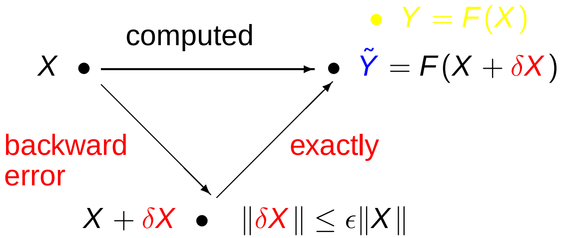
Eigenvalue assignment

$\alpha_1, \ldots, \alpha_n$ given. Find K such that the spectrum of $A + BK^T$ is $\{\alpha_1, \ldots, \alpha_n\}$. Try many B's and methods to hit $\odot$:



Placing plenty of poles is pretty preposterous (Chunyang, Laub, Mehrmann)

Error? Distance to what?!?



Backward stability: solve exactly a problem close to X

Not preserved under composition of mappings

$\longrightarrow \|\delta X\| \leq \epsilon \|X\|, \quad \|\delta X(:, i)\| \leq \epsilon \|X(:, i)\|$

$\longrightarrow |\delta X_{ij}| \leq \epsilon |X_{ij}|, \quad |\delta X_{ij}| \leq \epsilon \sqrt{|X_{ii} X_{jj}|}$

$\longrightarrow X + \delta X$ same structure as X

Perturbation theory: $\|\tilde{Y} - Y\| \leq K \cdot \|\delta X\|$

Von Neumann, Turing, Givens, Wilkinson

Example: dot product

Consider $s = y^T x = \sum_{i=1}^m x_i y_i$, $x_i, y_i \in \mathbb{F}$, computed as

$$\begin{aligned}
\tilde{s} &= x_1 \odot y_1; \quad \text{for } \; i = 2, \ldots, m \; \{\tilde{s} = \tilde{s} \oplus x_i \odot y_i .\} \\
\tilde{s} &= ((((x_1 \odot y_1) \oplus x_2 \odot y_2) \oplus x_3 \odot y_3) \oplus x_4 \odot y_4) \\
&= (((x_1 y_1 (1+\epsilon_1) + x_2 y_2 (1+\epsilon_2))(1+\xi_2) \\
&+ x_3 y_3 (1+\epsilon_3))(1+\xi_3) + x_4 y_4 (1+\epsilon_4))(1+\xi_4) \\
&= x_1 y_1 \underbrace{(1+\epsilon_1)(1+\xi_2)(1+\xi_3)(1+\xi_4)}_{1+\zeta_1} \\
&+ x_2 y_2 \underbrace{(1+\epsilon_2)(1+\xi_2)(1+\xi_3)(1+\xi_4)}_{1+\zeta_2} \\
&+ x_3 y_3 \underbrace{(1+\epsilon_3)(1+\xi_3)(1+\xi_4)}_{1+\zeta_3} + x_4 y_4 \underbrace{(1+\epsilon_4)(1+\xi_4)}_{1+\zeta_4} \\
&= \sum_{i=1}^{m=4} x_i y_i (1+\zeta_i).
\end{aligned}$$

Example

In general, for $m\varepsilon < 1$,

$$\tilde{s} = \sum_{i=1}^{m} x_i y_i (1+\zeta_i), \quad |\zeta_i| \leq \frac{m\varepsilon}{1-m\varepsilon}, \quad i = 1, 2, \ldots, m.$$

In this example, we have $\tilde{s} = s + \delta s$, $\delta s = \sum_{i=1}^{m} \zeta_i x_i y_i$ and

$$\frac{|\tilde{s}-s|}{|s|} \leq \frac{m\varepsilon}{1-m\varepsilon} \frac{\sum_{i=1}^{m} |x_i||y_i|}{|y^T x|} \leq \frac{m\varepsilon}{1-m\varepsilon} \frac{\|x\|_2 \|y\|_2}{|y^T x|} = \frac{1}{|\cos\angle(x,y)|} \frac{m\varepsilon}{1-m\varepsilon}.$$

This implies that tiny relative backward errors in the input can be amplified to a large relative errors in the computed inner product if the vectors x and y are nearly orthogonal. The amplification factor (in this case $1/|\cos\angle(x,y)|$) is called *condition number*.

Catastrophic cancelation

In some cases, catastrophic loss of accuracy occurs in a single operation. Suppose we need to compute $z = x - y$, but x and y are given with small relative errors as $\tilde{x} = x(1+\epsilon_x)$, $\tilde{y} = y(1+\epsilon_y)$. Then the actually computed difference $\tilde{z} = \tilde{x} \ominus \tilde{y}$ satisfies

$$\begin{aligned} \tilde{z} &= (x(1+\epsilon_x) - y(1+\epsilon_y))(1+\epsilon_-) = x(1+\epsilon_1) - y(1+\epsilon_2) \\ &= (x-y)(1+\underbrace{\frac{x\epsilon_1 - y\epsilon_2}{x-y}}_{\epsilon_z}), \quad |\epsilon_z| \leq \frac{\max\{|\epsilon_1|,|\epsilon_2|\}}{\frac{|x-y|}{|x|+|y|}}. \end{aligned}$$

The above operation is clearly backward stable, the difference $\tilde{x} - \tilde{y}$ is computed to full machine precision, but the relative error in $\tilde{z}$ can be large.
Such catastrophic loss of accuracy (called *catastrophic cancelation*) occurs if we subtract two relatively close values that are already contaminated by errors.

Example: triangular systems

Consider solving triangular system of equations $Tx = b$ with an $n \times n$ real upper triangular matrix T. Since in the backward substitutions each x_i is computed as

$$x_i = (1/T_{ii})(b_i - \sum_{j=i+1}^{n} T_{ij}x_j),$$

we can use similar analysis and "blame" all rounding errors in computing $\tilde{x}_i \approx x_i$ to the entries in the i-th row of T. This proves hat there exists an upper triangular δT such that the computed solution $\tilde{x}$ satisfies exactly the triangular system

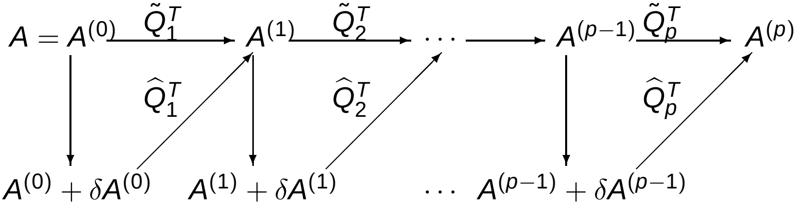
$$(T + \delta T)\tilde{x} = b, \quad \text{where } |\delta T_{ij}| \leq \frac{n\varepsilon}{1 - n\varepsilon}|T_{ij}|, \quad 1 \leq i, j \leq n.$$

QR factorization

$$Q_p^T Q_{p-1}^T \cdots Q_2^T Q_1^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad Q = Q_1 Q_2 \cdots Q_p .$$

$$A^{(1)} = Q_1^T A = \begin{pmatrix} \star & \star & \star \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{pmatrix}, \quad A^{(2)} = Q_2^T A^{(1)} = \begin{pmatrix} \star & \star & \star \\ 0 & \star & \star \\ 0 & 0 & \times \\ 0 & 0 & \times \end{pmatrix},$$

$$A^{(3)} = Q_3^T A^{(2)} = \begin{pmatrix} \star & \star & \star \\ 0 & \star & \star \\ 0 & 0 & \star \\ 0 & 0 & 0 \end{pmatrix} \equiv A^{(p)} = \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix},$$



$$\tilde{A}^{(i)} = \tilde{Q}_i^T * A^{(i-1)} = \widehat{Q}_i^T (\tilde{A}^{(i-1)} + \delta \tilde{A}^{(i-1)}), \; \|\widehat{Q}_i - \tilde{Q}_i\|_2 = O(\varepsilon)$$

Analysis

Reading the diagram backward, we have

$$\begin{aligned} A^{(p)} &= \widehat{Q}_p^T(\widehat{Q}_{p-1}^T(A^{(p-2)} + \delta A^{(p-2)}) + \delta A^{(p-1)}) \\ &= \widehat{Q}_p^T\widehat{Q}_{p-1}^T(A^{(p-2)} + \delta A^{(p-2)} + \widehat{Q}_{p-1}\delta A^{(p-1)}) \\ &= \widehat{Q}_p^T\widehat{Q}_{p-1}^T(\widehat{Q}_{p-2}^T(A^{(p-3)} + \delta A^{(p-3)}) + \delta A^{(p-2)} + \widehat{Q}_{p-1}\delta A^{(p-1)}) \\ &= \underbrace{\widehat{Q}_p^T\cdots\widehat{Q}_1^T}_{\widehat{Q}^T}(A + \underbrace{\delta A^{(0)} + \widehat{Q}_1\delta A^{(1)} + \cdots + \widehat{Q}_1\cdots\widehat{Q}_{p-1}\delta A^{(p-1)}}_{\delta A}), \end{aligned}$$

and the (by construction upper triangular) $A^{(p)}$ satisfies

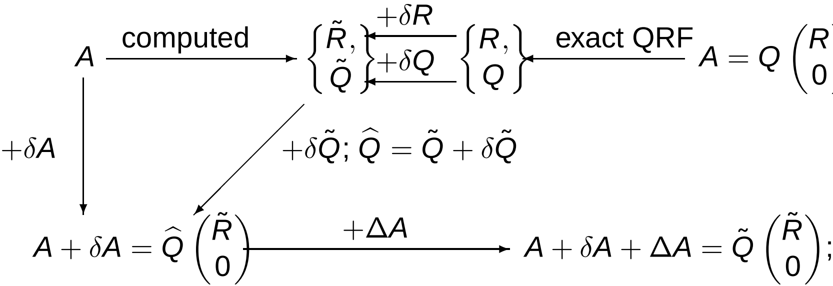
$$A^{(p)} = \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} = \widehat{Q}^T(A + \delta A), \quad (6)$$

where (note that $\|A^{(k)}\|_F \leq (1 + O(\varepsilon))^k\|A\|_F$)

$$\|\delta A\|_F \leq \sum_{k=0}^{p-1}\|\delta A^{(k)}\|_F \leq [(1 + O(\varepsilon))^p - 1]\|A\|_F. \quad (7)$$

QR analysis

$$\|\delta A(:,j)\|_2 \leq \sum_{k=0}^{p-1} \|\delta A^{(k)}(:,j)\|_2 \leq \underbrace{[(1+O(\varepsilon))^p - 1]}_{\equiv \zeta} \|A(:,j)\|_2 .$$



$$\Delta A = -\delta\tilde{Q}\begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}$$

Ill–conditioned = close to ill–posed

Relative condition number

$$\kappa(\mathcal{F}, X) = \limsup_{\Delta X \to 0} \frac{\dfrac{\|\mathcal{F}(X+\Delta X) - \mathcal{F}(X)\|}{\|\mathcal{F}(X)\|}}{\|\Delta X\|/\|X\|} = \frac{\|D\mathcal{F}(X)\|\|X\|}{\|\mathcal{F}(X)\|}$$

For $A \mapsto A^{-1}$, $\kappa(A) = \|A\| \cdot \|A^{-1}\|$, and the bad set is the variety of singular matrices.

$$\frac{\mathrm{distance}(A, bad)}{\|A\|} = \frac{1}{\kappa(A)}, \quad bad = \det^{-1}(\{0\}).$$

$$A_{ij} \mapsto A_{ij} + \epsilon|A_{ij}|$$

$$\inf\{|\epsilon| : \det(A + \epsilon E) = 0\} = \frac{1}{\rho_0(A^{-1}E)}$$

Probability of being too close to bad set. Algebraic and geometric properties of bad sets.

Example using MATLAB, eps $\approx 2.2 \cdot 10^{-16}$

$$X = (x \quad y) \in \mathbf{R}^{m\times 2}, \quad \begin{pmatrix} a & c \\ c & b \end{pmatrix} = computed(X^T X),$$

Let $\quad x = 5{\cdot}10^{153} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad y = 10 \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \quad (\cos \angle(x,y) = \dfrac{3}{\sqrt{10}}).$

Test the orthogonality of x and y,

$$\cos \angle(x,y) \equiv \frac{c}{\sqrt{ab}} \le \epsilon$$

$$\begin{aligned}
\texttt{(c / sqrt(a*b) <= eps)} &= 1,\\
\texttt{((c / sqrt(a)) / sqrt(b) <= eps)} &= 0,\\
\texttt{(c <= sqrt(a*b) * eps)} &= 1,\\
\texttt{(c <= sqrt(a)*sqrt(b) * eps)} &= 0.
\end{aligned}$$

Example using MATLAB, eps $\approx 2.2 \cdot 10^{-16}$

Let $x = 5 \cdot 10^{-153} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad y = 10^{-16} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Then

```
( c / sqrt(a*b) <= eps )                 = 0,
( (c / sqrt(a)) / sqrt(b) <= eps )       = 1,
( c <= sqrt(a*b) * eps )                 = 0,
( c <= sqrt(a)*sqrt(b) * eps )           = 1.
```

Another example

$$A = \begin{pmatrix} 1 & 1 & \boxed{1} \\ 0 & 1 & \xi \\ 0 & -1 & \xi \end{pmatrix}, \text{ where } \xi = 10/\text{eps}.\ \xi \approx 4.5\text{e+}016$$

Givens rotation kills A_{13}: $\tilde{A}^{(1)} = \begin{pmatrix} 1 & \alpha & 0 \\ 0 & \beta & \beta \\ 0 & \boxed{\beta} & \beta \end{pmatrix}$;

$\alpha \approx \sqrt{2}$, $\beta = 3.184525836262886\text{e+}016$.

	svd(A)	svd(A^T)
σ_1	6.369051672525773e+16	6.369051672525772e+16
σ_2	5.747279316501105e+00	3.004066501831585e+00
σ_3	9.842664568695829e-01	4.220776043599739e-01

$$\tilde{A}^{(1)} = \begin{pmatrix} 1 & \alpha & 0 \\ 0 & \beta & \beta \\ 0 & \boxed{\beta} & \beta \end{pmatrix}, \quad \tilde{A}^{(2)} = \begin{pmatrix} 1 & \alpha & 0 \\ 0 & \gamma & \gamma \\ 0 & 0 & 0 \end{pmatrix},$$

$A = BD$, $\kappa_2(B) < 2$.

A 2×2 example

Take in MATLAB

$$A = \begin{pmatrix} 1.0e250 & 0 \\ 0 & 1.0e\text{–}201 \end{pmatrix},$$

$d = \mathrm{diag}(A)$, $\sigma = \mathrm{svd}(A)$. A is (bi)diagonal, and its singular values are on the diagonal. However,

$$\begin{aligned} d &= \mathrm{diag}(A) = \begin{pmatrix} 9.999999999999999e+249 \\ 1.000000000000000e-201 \end{pmatrix}, \\ \sigma &= \mathtt{svd}(A) = \begin{pmatrix} 9.999999999999999e+249 \\ 1.0000000000\underline{16167}e-201 \end{pmatrix}. \\ \lambda &= \mathtt{eig}(A) = \begin{pmatrix} 9.999999999999999e+249 \\ 1.000000000000000e-201 \end{pmatrix} \end{aligned}$$

The 2×2 example

LAPACK's driver routine xGESVD computes $\alpha = \max_{i,j} |A_{ij}|$ and scales the input matrix A with $(1/\alpha)\sqrt{\nu}/\varepsilon$ (if $\alpha < \sqrt{\nu}/\varepsilon$) or with $(1/\alpha)\varepsilon\sqrt{\omega}$ (if $\alpha > \varepsilon\sqrt{\omega}$). Here ε, ν and ω denote the round–off unit, underflow and overflow thresholds, respectively.

Let $\alpha = \max_{i,j} |A_{ij}|$, $\varepsilon = eps/2$, $\omega = realmax$, $\nu = realmin$, $s = \varepsilon\sqrt{\omega}/\alpha$, and scale A with s. The singular values of sA are on its diagonal; scaling the diagonal of sA with $1/s$ changes the $(2,2)$ entry precisely to $1.0000000000\mathbf{16167}e{-201}$. Five digits in the second singular value of a 2×2 diagonal matrix are lost due to scaling $\sigma = (1/s) * (s * d)$. (In MATLAB, $\omega \approx 1.79 \cdot 10^{308}$, $\nu \approx 2.22 \cdot 10^{-308}$.) The problem is not removed if s is changed to the closest integer power of two.

Note that this scaling is designed to avoid overflow in the implicit use of $A^T A$.

What is the spectrum of H?

$$H = \begin{pmatrix} 10^{40} & 10^{29} & 10^{19} \\ 10^{29} & 10^{20} & 10^{9} \\ 10^{19} & 10^{9} & 1 \end{pmatrix};$$

use MATLAB, eps $\approx 2.22 \cdot 10^{-16}$

$$eig(H) = \begin{array}{r} 1.000000000000000e+040 \\ -8.100009764062724e+019 \\ -3.966787845610502e+023 \end{array}$$

L=chol(H)' ($H = LL^T$)

$$L = \begin{pmatrix} 1.0000000e+20 & 0 & 0 \\ 9.9999999e+8 & 9.9498743e+9 & 0 \\ 9.9999999e-2 & 9.0453403e-2 & 9.9086738e-1 \end{pmatrix}$$

Is H positive definite?

What is the spectrum of H now?

	eig(H)	eig($P^T HP$), $P \simeq (2,1,3)$
λ_1	1.000000000000000e+40	1.000000000000000e+40
λ_2	-8.100009764062724e+19	9.900000000000000e+19
λ_3	-3.966787845610502e+23	9.818181818181818e-01

	1./eig(inv(H))	eig(inv(inv(H)))
λ_1	1.000000000000000e+40	1.000000000000000e+40
λ_2	9.900000000000000e+19	9.900000000000000e+19
λ_3	9.818181818181817e-01	9.818181818181817e-01

	eig($H+E_1$)	eig($H+E_2$)
λ_1	1.000000000000000e+40	1.000000000000000e+40
λ_2	-8.100009764062724e+19	1.208844819952007e+24
λ_3	-3.966787845610502e+23	9.899993299416013e-01

E_1: $H_{22} = 10^{20} \longrightarrow -10^{20}$, E_2: $H_{13}, H_{31} \longrightarrow H_{13} * (1 + \texttt{eps})$, $\texttt{eps} \approx 2.22 \cdot 10^{-16}$; All numbers correct!?

Backward stability: eig()

$H = H^T$, $n \times n$ symmetric.

$$Hu_i = \lambda_i u_i, \quad H = U \Lambda U^T, \quad \Lambda = \mathrm{diag}(\lambda_i)_{i=1}^n$$

Symm. EigenValue Problem perfect $\bigstar\ \bigstar\ \bigstar$:

$\star$ eigenvalues real, eigenvectors orthogonal

$\star$ algorithms use orthogonal transformations

$\star$ Weyl: If $H \rightsquigarrow H + \delta H$, then $\lambda \rightsquigarrow \lambda + \delta\lambda$, with

$$\max_{\lambda} |\delta\lambda| \leq \|\delta H\|$$

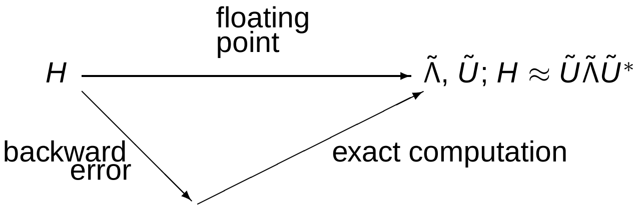
$$\cdots U_k^T \cdots (U_2^T (U_1^T H \underbrace{U_1) U_2) \cdots U_k \cdots}_{U} \longrightarrow \Lambda$$

Computed (finite prec., $O(n^3)$) $\tilde{U} \approx U$, $\tilde{\Lambda} \approx \Lambda$.

Backward stability:

$$\tilde{U}^T (H + \delta H) \tilde{U} \approx \tilde{\Lambda}, \quad \frac{\|\delta H\|}{\|H\|} \leq \epsilon \approx 10^{-16} \ \textit{small}.$$

Forward error



$H + \delta H$, $\|\delta H\| \leq \epsilon \|H\|$, ϵ small

Weyl: $|\delta\lambda_i| \leq \|\delta H\|$, $i = 1, \ldots, n$. Bad news for small λ_i's:

$$\frac{|\delta\lambda_i|}{|\lambda_i|} \leq \epsilon \frac{\|H\|}{|\lambda_i|}$$

Let $\kappa_2(H) = \|H\| \|H^{-1}\|$. Then

$$\max_i \left| \frac{\delta\lambda_i}{\lambda_i} \right| \leq \kappa_2(H) \frac{\|\delta H\|}{\|H\|}.$$

Want better accuracy for better inputs.

Error in the eigenvalues

Let $H = LL^T \succ 0$ and $\tilde{L}\tilde{L}^T = H + \delta H \succ 0$, $|\delta H_{ij}| \leq \eta_C \sqrt{H_{ii}H_{jj}}$.
Compare the eigenvalues of H and $\tilde{H} = H + \delta H = \tilde{L}\tilde{L}^T$:

- $H = LL^T$ is similar to $L^T L$, $H \sim L^T L$.
- Let $Y = \sqrt{I + L^{-1}\delta H L^{-T}}$. Then

$$H + \delta H = L(I + L^{-1}\delta H L^{-T})L^T = LYY^T L^T \sim Y^T L^T LY.$$

Compare $\lambda_i(L^T L) = \lambda_i(H)$ and $\lambda_i(Y^T L^T LY) = \lambda_i(H + \delta H)$.

- Ostrowski: $\tilde{M} = Y^T M Y$, then, for all i, $\lambda_i(\tilde{M}) = \lambda_i(M)\xi_i$, $\lambda_{\min}(Y^T Y) \leq \xi_i \leq \lambda_{\max}(Y^T Y)$. Here $Y^T Y = I + L^{-1}\delta H L^{-T}$.
- Hence $|\lambda_i(H) - \lambda_i(\tilde{H})| \leq \lambda_i(H)\|L^{-1}\delta H L^{-T}\|_2$,

$$\begin{aligned}
\|L^{-1}\delta H L^{-T}\|_2 = \|L^{-1}DD^{-1}\delta H D^{-1}DL^{-T}\|_2 &= \|L^{-1}D(\delta H_s)DL^{-T}\|_2 \\
&\leq \|L^{-1}D\|_2^2 \|\delta H_s\|_2 = \|DL^{-T}L^{-1}D\|_2 \|\delta H_s\|_2 \\
&= \|(D^{-1}HD^{-1})^{-1}\|_2 \|\delta H_s\|_2 = \|H_s^{-1}\|_2 \|\delta H_s\|_2
\end{aligned}$$

Error in the eigenvalues

Since $\delta H_s = (\delta H_{ij}/\sqrt{H_{ii}H_{jj}})$

$$\max_i \left|\frac{\delta\lambda_i}{\lambda_i}\right| \leq \|H_s^{-1}\|_2 \underbrace{\left\|\left[\frac{\delta H_{ij}}{\sqrt{H_{ii}H_{jj}}}\right]\right\|_2}_{\leq n\eta_C}$$

Compare with $\max_i \left|\frac{\delta\lambda_i}{\lambda_i}\right| \leq \kappa_2(H)\frac{\|\delta H\|_2}{\|H\|_2}$

Van der Sluis: $\|H_s^{-1}\|_2 \leq \kappa_2(H_s) \leq n \min_{D=diag} \kappa_2(DHD)$.
Our 3 × 3 example: $H = DH_sD$, $D = \mathrm{diag}(10^{20}, 10^{10}, 1)$,

$$\begin{pmatrix} 10^{40} & 10^{29} & 10^{19} \\ 10^{29} & 10^{20} & 10^{9} \\ 10^{19} & 10^{9} & 1 \end{pmatrix} = DH_sD = D \begin{pmatrix} 1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{pmatrix} D,$$

$\kappa_2(H) > 10^{40}$, $\kappa_2(H_s) < 1.4$, $\|H_s^{-1}\|_2 < 1.2$.

Diagonalizing the Grammians

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t),\; x(0) = x_0 \\ y(t) &= Cx(t) \end{aligned}$$

Grammians $H = L_H L_H^T$, $M = L_M L_M^T$ via Lyapunov equations:

$$AH + HA^T = -BB^T, \;\; A^T M + MA = -C^T C.$$

Hankel SV, $\sigma_i = \sqrt{\lambda_i(HM)}$, $HM \mapsto T^{-1}HMT = \Sigma^2$.
Different scaling (change of units, x may contain quantities of different physical nature) $x(t) = D\hat{x}(t)$; $A \mapsto D^{-1}AD$, $B \mapsto D^{-1}B$, $C \mapsto CD$;

$$H \mapsto \hat{H} = D^{-1}HD^{-T}, \;\; M \mapsto \hat{M} = D^T MD$$

Change of units (scaling) changes classical condition numbers $\kappa_2(H)$, $\kappa_2(M)$ thus making an algorithm numerically inaccurate/unstable, while the underlying problem is the same. Is this acceptable?!?

Rank Revealing Decomposition

In a +60 pages LAA paper Demmel, Drmač, Gu, Eisenstat, Slapničar, Veselić (DGESVD paper) noted that some classes of matrices allow so called Rank Revealing Decomposition (RRD),

$$P_1AP_2 = LDU, \quad P_1, P_2 \text{ permutations},$$

where D is diagonal, and L and U are well conditioned. Moreover, L, D, U can be computed in a forward stable way. An example of a RRD of A is obtained by non–standard Gaussian eliminations using certain structural properties of A. More examples by Demmel and Koev.
Then, we can use the accurate PSVD algorithm and get the SVD of LDU.
Example: Cauchy matrix $C_{ij} = 1/(x_i + y_j)$
(displacement rank one, $XC + CY = d_1d_2^T$)

Cauchy matrices

$$\det(C) = \frac{\prod_{i<j}(x_j - x_i)(y_j - y_i)}{\prod_{i,j}(x_i + y_j)}$$

Can get accurate *LDU* at high cost, $O(n^5)$. Then Demmel reduced it to the usual $O(n^3)$ using the recursive structure of the Schur complement.

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ C_{21}C_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ 0 & S^{(k)} \end{pmatrix}$$

$$S_{ij}^{(k)} = S_{ij}^{(k-1)} \frac{(x_i - x_k)(y_j - y_k)}{(x_k + y_j)(x_i + y_k)}$$

Straightforward extension to Cauchy–like matrices $D_1 C D_2$, D_i diagonal. Simplified for symmetric positive definite cases.

An illustration

After computing $C = LDU$, one applies accurate Jacobi PSVD to the product $(LD)U$. All forward stable, but the spectrum is ill-conditioned!

An illustration of the power of this algorithm is the example of 100×100 Hilbert matrix H_{100}. Computation done by Demmel:

- The singular values of H_{100} range over 150 orders of magnitude and are computed using the package Mathematica with 200–decimal digit software floating point arithmetic. The computed singular values are rounded to 16 digits and used as reference values.
- The singular values computed in IEEE double precision floating–point ($\varepsilon \approx 10^{-16}$) by the Jacobi PSVD agree with the reference values with relative error less than $34 \cdot \varepsilon$.

Rational approximation

New highly accurate NLA algorithms open new possibilities in other computational tasks.
For instance, Haut and Beylkin (2011) used Adamyan–Arov–Krein theory to show that nearly L^∞–optimal rational approximation on $|z| = 1$ of

$$f(z) = \sum_{i=1}^{n} \frac{\alpha_i}{z-\gamma_i} + \sum_{i=1}^{n} \frac{\overline{\alpha_i} z}{1-\overline{\gamma_i} z} + \alpha_0$$

with $\max_{|z|=1} |f(z) - r(z)| \rightsquigarrow \min$,

$$r(z) = \sum_{i=1}^{m} \frac{\beta_i}{z-\eta_i} + \sum_{i=1}^{m} \frac{\overline{\beta_i} z}{1-\overline{\eta_i} z} + \alpha_0$$

is numerically feasible if one can compute the con–eigenvalues and con–eigenvectors

$$Cu = \lambda \overline{u}, \quad C_{ij} = \frac{\sqrt{\alpha_i}\sqrt{\overline{\alpha_j}}}{\gamma_i^{-1} - \overline{\gamma_j}} \leftrightsquigarrow \frac{\alpha_i \overline{\alpha_j}}{1 - \gamma_i \overline{\gamma_j}}$$

Con–eigenvalues

Here $C = (\frac{\sqrt{\alpha_i}\sqrt{\overline{\alpha_j}}}{\gamma_i^{-1} - \overline{\gamma_j}})$ is positive definite Cauchy matrix C.
The con–eigenvalue problem $Cu = \lambda \overline{u}$ is equivalent to solving

$$\overline{C} C u = |\lambda|^2 u,$$

where C is factored as $C = XD^2X^*$. The problem reduces to computing the SVD of the product $G = DX^T XD$. Accurate SVD via the PSVD based on the Jacobi SVD. Haut and Beylkin tested the accuracy with $\kappa_2(C) > 10^{200}$ and using Mathematica with 300 hundred digits for reference values. Over 500 test examples of size 120, the maximal error in IEEE 16 digit arithmetic ($\varepsilon \approx 2.2 \cdot 10^{-16}$) was

$$\frac{|\tilde{\lambda}_i - \lambda_i|}{|\lambda_i|} < 5.2 \cdot 10^{-12}, \quad \frac{\|\tilde{u}_i - u_i\|_2}{\|u_i\|_2} < 5.4 \cdot 10^{-12}.$$

Case study

Given $A \in \mathbb{C}^{m\times n}$, determine whether for some small δA, the matrix $A + \delta A$ is of rank $\rho < \mathrm{rank}(A)$.

- needed and useful if A is close to matrices of lower rank (i.e. ill–conditioned)
- in the case of ill–conditioning, one does not expect much and any bad result is attributed to ill–conditioning;
- condition number can be ill–conditioned
- numerical instability in a software implementation of a basic numerical linear algebra decomposition (QR factorization with column pivoting) for almost 40 years hidden in all major numerical software packages

Eckart–Young–Mirsky–Schmidt

A $m \times n$ real of rank r

$$A = U\Sigma V^T = \sum_{k=1}^{r} \sigma_i u_i v_i^T$$

$$\sigma_1 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_{\min(m,n)}$$

Let $\ell < r$ and $A_\ell = \sum_{i=1}^{\ell} \sigma_i u_i v_i^T$ Then

$$\min_{rank(X) \leq \ell} \|A - X\|_F = \|A - A_\ell\|_F = \sqrt{\sum_{i=\ell+1}^{r} \sigma_i^2}$$

$$\min_{rank(X) \leq \ell} \|A - X\|_2 = \|A - A_\ell\|_2 = \sigma_{\ell+1}$$

Example: imagesvd('Osijek.jpg')

QRCP with Businger–Golub pivoting

$$\underbrace{A}_{m\times n}\overbrace{P}^{\text{permutation}} = Q\begin{pmatrix} R \\ 0 \end{pmatrix},\quad R = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \hline 0 & 0 & \color{red}{\blacksquare} & \color{blue}{\bullet} & \color{blue}{\blacksquare} & \color{blue}{\blacklozenge} \\ 0 & 0 & 0 & \color{blue}{\bullet} & \color{blue}{\blacksquare} & \color{blue}{\blacklozenge} \\ 0 & 0 & 0 & 0 & \color{blue}{\blacksquare} & \color{blue}{\blacklozenge} \\ 0 & 0 & 0 & 0 & 0 & \color{blue}{\blacklozenge} \end{pmatrix}$$

$$Q^*Q = I_m.$$

$$|R_{ii}| \geq \sqrt{\sum_{k=i}^{j} |R_{kj}|^2}, \quad \text{for all } 1 \leq i \leq j \leq n. \qquad (8)$$

$$|R_{11}| \geq |R_{22}| \geq \cdots \geq |R_{\rho\rho}| \gg |R_{\rho+1,\rho+1}| \geq \cdots \geq |R_{nn}| \qquad (9)$$

The structure (8), (9) may not be rank revealing but it must be guaranteed by the software (e.g. LAPACK, Matlab)

QRCP as preconditioner

Let $AP = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$; $A_c = A \cdot \mathrm{diag}(\frac{1}{\|A(:,1)\|_2}, \ldots, \frac{1}{\|A(:,n)\|_2})$;

$$R_c = R \cdot \mathrm{diag}(\frac{1}{\|R(:,1)\|_2}, \ldots, \frac{1}{\|R(:,n)\|_2}) = \begin{pmatrix} \downarrow & \downarrow & \downarrow \\ 0 & \downarrow & \downarrow \\ 0 & 0 & \downarrow \end{pmatrix};$$

$$R_r = \mathrm{diag}(\frac{1}{\|R(1,:)\|_2}, \ldots, \frac{1}{\|R(n,:)\|_2}) \cdot R = \begin{pmatrix} \rightarrow & \rightarrow & \rightarrow \\ 0 & \rightarrow & \rightarrow \\ 0 & 0 & \rightarrow \end{pmatrix}.$$

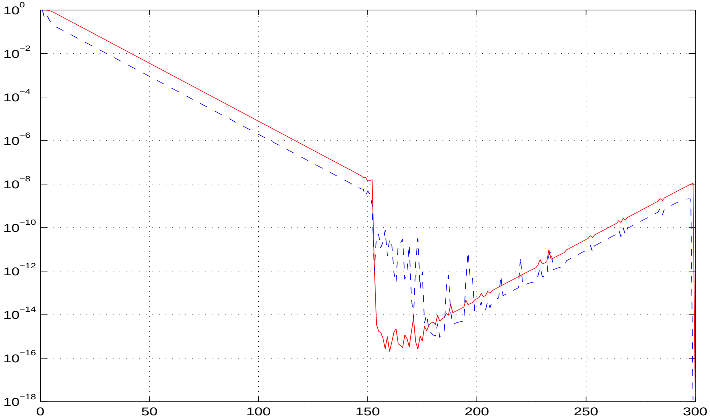
Proposition:

Let $AP = QR$, where $|R_{ii}| \geq \sqrt{\sum_{k=i}^{j} |R_{kj}|^2}$, $1 \leq i \leq j \leq n$.
Then $\| \, |R_r^{-1}| \, \|_2 \leq \sqrt{n} \| \, |R_c^{-1}| \, \|_2$, $\kappa_2(R_r) \leq n^{3/2} \kappa_2(A_c)$.
Moreover, $\|R_r^{-1}\|_2$ is bounded by $O(2^n)$, independent of A.
With exception of rare pathological cases, $\|R_r^{-1}\|_2$ is below $O(n)$ for any A. RR^* is *more diagonal* than R^*R.

Example:

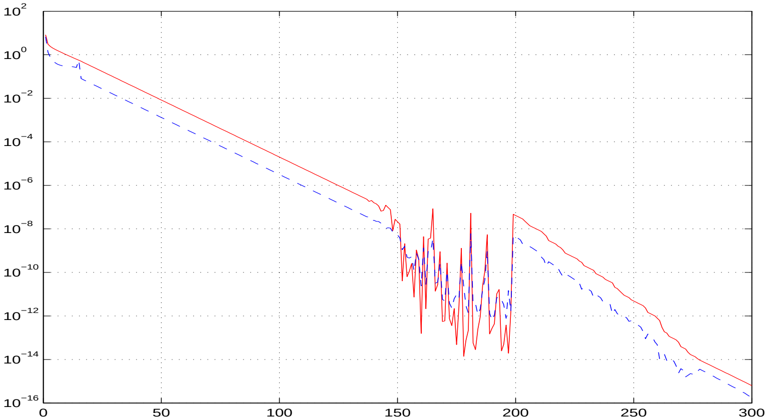
Let $A = \mathit{Hilbert}(100)$. $\kappa_2(A) > 10^{150} \gg \mathrm{cond}(A) \approx 3.6\mathrm{e}19$
$\kappa_2(A_c) = \kappa_2(R_c) > 10^{19}$, $\kappa_2(R_r) \approx 48.31$. Repeat with $A \leftarrow R^T$, $P = I$, to get new $\kappa_2(R_r) \approx 3.22$.

Examples of failure (Matlab)



$$|R_{ii}|,\ \max_{j\geq i}\sqrt{\sum_{k=i}^{j}|R_{kj}|^2},\ R = \begin{pmatrix} ■ & ■ & ■ & ■ & ■ & ■ \\ 0 & ■ & ■ & ■ & ■ & ■ \\ 0 & 0 & ■ & ● & ■ & ♦ \\ 0 & 0 & 0 & ● & ■ & ♦ \\ 0 & 0 & 0 & 0 & ■ & ♦ \\ 0 & 0 & 0 & 0 & 0 & ♦ \end{pmatrix}$$

Examples of failure (Matlab)

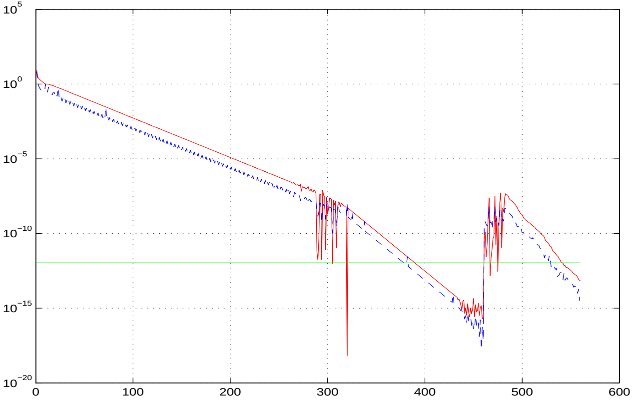


$$|R_{ii}|, \max_{j\geq i}\sqrt{\sum_{k=i}^{j}|R_{kj}|^2},\ R = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & \blacksquare & \bullet & \blacksquare & \blacklozenge \\ 0 & 0 & 0 & \bullet & \blacksquare & \blacklozenge \\ 0 & 0 & 0 & 0 & \blacksquare & \blacklozenge \\ 0 & 0 & 0 & 0 & 0 & \blacklozenge \end{pmatrix}$$

Consequences (Matlab)

$\|Ax - d\|_2 \to \min$; $x = A\backslash d$ (Matlab LS solution)

`Warning: Rank deficient, rank = 304 tol = 1.0994e-012.` $R = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & \blacksquare & \bullet & \blacksquare \\ 0 & 0 & 0 & \bullet & \blacksquare \\ 0 & 0 & 0 & 0 & \blacksquare \end{pmatrix}$



`rank(A,1.0994e-12)` returns 466

Consequences

Any routine based on xQRDC (LINPACK) or xGEQPF, xGEQP3 (LAPACK) can catastrophically fail.

- xGEQPX (TOMS ♯ 782, rank revealing QRF)
- xGELSX and xGELSY in LAPACK ($\|Ax - b\|_2 \to \min$)
- xGGSVP in LAPACK (GSVD of (A, B))

$$U^T AQ = \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \\ 0 & 0 & 0 \end{pmatrix}, \quad V^T BQ = \begin{pmatrix} 0 & 0 & B_{13} \\ 0 & 0 & 0 \end{pmatrix}.$$

- ... and many others ... long list. Need a new xGEQP3.

Resolved by Drmač and Bujanović (ACM TOMS, 2008) and included in LAPACK.
In control, included in SLICOT in 2010.

SLICOT

The SLICOT (Subroutine Library In COntrol Theory)

- is used as computational layer in sophisticated CACSD packages such as EASY5 (since 2002. MSC.Software, initially developed in the Boeing Company), Matlab (The MathWorks) and Scilab (INRIA).
- Since its initial release, SLICOT has been growing at an impressive rate, from 90 user–callable subroutines in 1997., 200 subroutines in 2004., 470 subroutines in 2009., ...
- Efficiency an reliability based on BLAS, LAPACK and state of the art numerical linear algebra

The problem illustrated in the previous examples of QRCP failure affects SLICOT and thus many other control theory libraries.

SLICOT

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx + Du. \end{aligned} \qquad (10)$$

- Strategically placed "`WRITE(*,*) variable`" statements in the affected subroutines can completely change the computed properties of (10).
- Substantial variations of the output can also be caused by changing the compiler and optimizer options.
- This is undesired behavior, even if the computation is backward stable, and even if it is doomed to fail, due to ill–conditioning.

The problem occurs only at certain distance to singularity, and the rank revealing task itself is usually performed if the matrix is close to singularity. Since many things can happen close to singularity, any ill–behavior is usually attributed to ill–conditioning and the true cause remains inconspicuous.

SLICOT Example: MB03OY

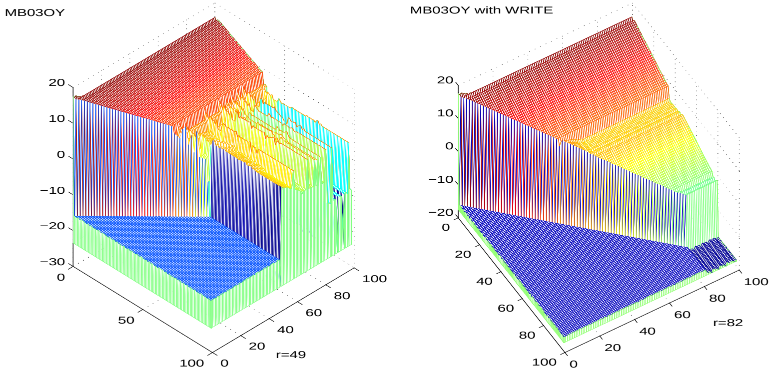


Figure: Left: The matrix R computed by MB03OY, shown by meshz(log10(abs(R))). The computed rank is 49. Right: The matrix R computed with MB03OY, with "`WRITE(*,*) TEMP2`" statement added after the line 339 in MB03OY.f. The computed rank is 82.

Affected SLICOT routines

1. MB03OY $\twoheadleftarrow$:

- **1.1** AB01ND $\twoheadleftarrow$ **1.1.1** AB01OD
- **1.2** AB08NX $\twoheadleftarrow$:
  - **1.2.1** AB08ND
  - **1.2.2** AB08MD $\twoheadleftarrow$ **1.2.2.1** AB09JD
- **1.3** AG08BY $\twoheadleftarrow$ **1.3.1** AG08BD
- **1.4** MB02QD $\twoheadleftarrow$ **1.4.1** SB01DD
- **1.5** TB01UD $\twoheadleftarrow$:
  - **1.5.1** TB01PD $\twoheadleftarrow$:
    - **1.5.1.1** TD04AD $\twoheadleftarrow$ **1.5.1.1.1** SB10ZP $\twoheadleftarrow$ **1.5.1.1.1.1** SB10YD $\twoheadleftarrow$ **1.5.1.1.1.1.1** SB10MD
    - **1.5.1.2** AB09ID
  - **1.5.2** TB03AD $\twoheadleftarrow$ **1.5.2.1** TD03AD
  - **1.5.3** TB04AY $\twoheadleftarrow$ **1.5.3.1** TB04AD
- **1.6** TG01FD $\twoheadleftarrow$ **1.6.1** AG08BD

... affected SLICOT routines

2. MB04GD $\twoheadleftarrow$ **2.1** MB03PD

3. MB03OD $\twoheadleftarrow$:

3.1 IB01ND $\twoheadleftarrow$ **3.1.1** IB01AD $\twoheadleftarrow$: **3.1.1.1** IB03AD, **3.1.1.2** IB03BD

3.2 IB01PD $\twoheadleftarrow$ **3.2.1** IB01BD $\twoheadleftarrow$: **3.2.1.1** IB03AD, **3.2.1.2** IB03BD

3.3 IB01PY $\twoheadleftarrow$ **3.3.1** IB01PD $\twoheadleftarrow$ **3.3.1.1** IB01BD $\twoheadleftarrow$: **3.3.1.1.1** IB03AD, **3.3.1.1.2** IB03BD

3.4 MB02QD $\twoheadleftarrow$ **3.4.1** SB01DD

3.5 *MB02YD $\twoheadleftarrow$:
- **3.5.1** NF01BQ $\twoheadleftarrow$ **3.5.1.1** NF01BP
- **3.5.2** MD03BY $\twoheadleftarrow$: **3.5.2.1** MD03BB, **3.5.2.2** NF01BP

3.6 *MD03BY $\twoheadleftarrow$:
- **3.6.1** MD03BB
- **3.6.2** NF01BP

3.7 *NF01BR $\twoheadleftarrow$:
- **3.7.1** NF01BP
- **3.7.2** NF01BQ $\twoheadleftarrow$ **3.7.2.1** NF01BP

60 out of 470 affected!

4. MB3OYZ ↞:

4.1 AB8NXZ ↞: **4.1.1** AB08MZ; **4.1.2** AB08NZ

4.2 AG8BYZ ↞ AG08BZ; **4.3** TG01FZ ↞ AG08BZ

5. MB03PY ↞ **5.1** AB08NX ↞:

5.1.1 AB08MD ↞ **5.1.1.1** AB09JD

5.1.2 AB08ND

6. MB3PYZ ↞ **6.1** AB8NXZ ↞: **6.1.1** AB08MZ; **6.1.2** AB08NZ

7. MB02CU ↞:

7.1. MB02GD; **7.2.** MB02HD; **7.3.** MB02ID

7.4. MB02JD; **7.5.** MB02JX

8. AG08BY ↞ **8.1** AG08BD

9. AG8BYZ ↞ **9.1** AG08BZ

10. TG01HX ↞: **10.1** TG01HD; **10.2** TG01ID; **10.3** TG01JD

Implementation: L(IN+A)PACK, 1970s, 1990s, ...

$$A^{(k)}\Pi_k = \begin{pmatrix} \cdot & \cdot & \odot & \cdot & \oplus & \cdot \\ & \cdot & \odot & \cdot & \oplus & \cdot \\ & & \blacksquare & \circledast & \circledast & \circledast \\ & & \circledcirc & * & * & * \\ & & \circledcirc & * & * & * \\ & & \circledcirc & * & * & * \end{pmatrix}, \quad \mathbf{a}_j^{(k)} = \begin{pmatrix} \oplus \\ \oplus \\ \circledast \\ * \\ * \\ * \end{pmatrix} \equiv \begin{pmatrix} \mathbf{x}_j^{(k)} \\ \mathbf{z}_j^{(k)} \end{pmatrix}$$

$$\mathrm{H}_k \mathbf{z}_k^{(k)} = \begin{pmatrix} R_{kk} \\ 0 \end{pmatrix}, \mathrm{H}_k \mathbf{z}_j^{(k)} = \begin{pmatrix} \beta_j^{(k+1)} \\ \mathbf{z}_j^{(k+1)} \end{pmatrix}, \quad \omega_j^{(k)} = \|\mathbf{z}_j^{(k)}\|$$

$$\|\mathbf{z}_j^{(k+1)}\| \equiv \omega_j^{(k+1)} = \sqrt{(\omega_j^{(k)})^2 - (\beta_j^{(k+1)})^2}, \quad \text{provided that}$$

$$\mathit{computed}\left(\left(1 - \left(\frac{\tilde{\beta}_j^{(k+1)}}{\tilde{\omega}_j^{(k)}} \right)^2 \right) \cdot \left(\frac{\tilde{\omega}_j^{(k)}}{\tilde{\nu}_j} \right)^2 \right) > \mathit{tol}, \quad \mathit{tol} \approx 20 \cdot \mathrm{eps},$$

L(IN+A)PACK update

```
      DO 30 J = I+1, N
         IF ( WORK( J ).NE.ZERO ) THEN
            TEMP = ONE - ( ABS( A( I, J ) ) / WORK( J ) )**2
            TEMP = MAX( TEMP, ZERO )
            TEMP2 = ONE + 0.05*TEMP*( WORK( J ) / WORK( N+J ) )**2
            WRITE(*,*) TEMP2
            IF( TEMP2.EQ.ONE ) THEN
               IF( M-I.GT.0 ) THEN
                  WORK( J ) = SNRM2( M-I, A( I+1, J ), 1 )
                  WORK( N+J ) = WORK( J )
               ELSE
                  WORK( J ) = ZERO
                  WORK( N+J ) = ZERO
               END IF
            ELSE
               WORK( J ) = WORK( J )*SQRT( TEMP )
            END IF
         END IF
30    CONTINUE
```

g77 -c -O -ffloat-store

Critical part in the column norm update. (For the full source see http://www.netlib.org/lapack/single/sgeqpf.f)

NEW update

$$A^{(k)}\Pi_k = \begin{pmatrix} \cdot & \cdot & \odot & \cdot & \oplus & \cdot \\ & \cdot & \odot & \cdot & \oplus & \cdot \\ & & \blacksquare & \circledast & \circledast & \circledast \\ & & \circledcirc & * & * & * \\ & & \circledcirc & * & * & * \\ & & \circledcirc & * & * & * \end{pmatrix}, \quad \mathbf{a}_j^{(k)} = \begin{pmatrix} \oplus \\ \oplus \\ \circledast \\ * \\ * \\ * \end{pmatrix} \equiv \begin{pmatrix} \mathbf{x}_j^{(k)} \\ \mathbf{z}_j^{(k)} \end{pmatrix} \tag{11}$$

$$\mathrm{H}_k \mathbf{z}_k^{(k)} = \begin{pmatrix} R_{kk} \\ 0 \end{pmatrix}, \mathrm{H}_k \mathbf{z}_j^{(k)} = \begin{pmatrix} \beta_j^{(k+1)} \\ \mathbf{z}_j^{(k+1)} \end{pmatrix}, \quad \omega_j^{(k)} = \|\mathbf{z}_j^{(k)}\| \tag{12}$$

$$\|\mathbf{a}_j^{(k)}\| = \alpha_j^{(k)} = \alpha_j^{(0)}; \quad \xi_j^{(k+1)} = \sqrt{(\xi_j^{(k)})^2 + (\beta_j^{(k+1)})^2}$$

$$\|\mathbf{z}_j^{(k+1)}\| \equiv \omega_j^{(k+1)} = \sqrt{(\alpha_j^{(0)})^2 - (\xi_j^{(k+1)})^2}$$

New update – conclusion

- Provably delivers Businger–Golub structured R (up to roundoff)
- For the computed $\tilde{R} = R + \delta R$, not only $\|\delta R\|/\|R\|$, but also $\|\delta R(:,i)\|/\|R(:,i)\|$ and $\|\delta R(i,:)\|/\|R(i,:)\|$ are small.
- Row scaled $\tilde{R}_r$ well conditioned. $\begin{pmatrix} \rightarrow & \rightarrow & \rightarrow \\ 0 & \rightarrow & \rightarrow \\ 0 & 0 & \rightarrow \end{pmatrix}$
- Same efficiency as original routines
- Makes many other solvers more robust and can prevent catastrophes in mission critical applications
- Included in LAPACK, SLICOT

$H - \lambda I,\ HM - \lambda I$

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t),\ x(0) = x_0 \\ y(t) &= Cx(t) \end{aligned}$$

Grammians $H = L_H L_H^T$, $M = L_M L_M^T$ via Lyapunov equations:

$$H = \int_0^\infty e^{tA} B B^T e^{tA^T} dt, \quad M = \int_0^\infty e^{tA^T} C^T C e^{tA} dt$$

$$AH + HA^T = -BB^T, \quad A^T M + MA = -C^T C.$$

Hankel singular values, $\sigma_i = \sqrt{\lambda_i(HM)}$. Need spectral decomposition of the product HM of positive definite matrices, $HM \mapsto T^{-1} HMT = \Sigma^2$. New state coo's $x(t) = T\hat{x}(t)$; $A \mapsto T^{-1}AT$, $B \mapsto T^{-1}B$, $C \mapsto CT$;

$$H \mapsto \hat{H} = T^{-1} H T^{-T} = \Sigma, \quad M \mapsto \hat{M} = T^T M T = \Sigma$$

Positive definiteness in floating–point

Demmel and Veselić

Let $H = DH_sD$, where $D = \mathrm{diag}(\sqrt{H_{ii}})_{i=1}^n$, and let $\lambda_{\min}(H_s)$ be the minimal eigenvalue of H_s.

If δH is symmetric perturbation such that $H + \delta H$ is not positive definite, then

$$\max_{1\leq i,j\leq n} \frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \geq \frac{\lambda_{\min}(H_s)}{n} = \frac{1}{n\|H_s^{-1}\|_2}.$$

If $\delta H = -\lambda_{\min}(H_s)D^2$, then $\max_{i,j} \frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} = \lambda_{\min}(H_s)$ and $H + \delta H$ is singular.

If $\|H_s^{-1}\|_2$ is too big ($\gtrapprox 1/\varepsilon$) then H is entry–wise close to a non–definite matrix. Can say: H is numerically definite iff $\|H_s^{-1}\|_2 < 1/\varepsilon$.

Implicit definiteness

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{pmatrix}.$$

Note that $K = LL^T$, where

$$L = \begin{pmatrix} \sqrt{k_1} & 0 & 0 \\ -\sqrt{k_2} & \sqrt{k_2} & 0 \\ 0 & -\sqrt{k_3} & \sqrt{k_3} \end{pmatrix} = \begin{pmatrix} \sqrt{k_1} & 0 & 0 \\ 0 & \sqrt{k_2} & 0 \\ 0 & 0 & \sqrt{k_3} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

+definite $HMx = \lambda x$

Entry–wise backward stability also possible for $HMx = \lambda x$.

- Use contragredient scaling $H := DHD$, $M := D^{-1}MD^{-1}$ to get all $M_{ii} = 1$. Here $D = \mathrm{diag}(\sqrt{M_{ii}})_{i=1}^{n}$.
- Cholesky f. $H := P^T HP = L_H L_H^T$, $M = L_M L_M^T$
- $HM = PL_H L_H^T P^T L_M L_M^T = L_M^{-T}(L_M^T PL_H L_H^T P^T L_M)L_M^T$
- $HM = L_M^{-T}(AA^T)L_M$, $A = L_M^T PL_H$, $L_H = L_{H,s}D_H$
- Compute $A = (L_M^T P)L_H$. (No fast matrix–multiply allowed. Must pay $O(n^3)$.)
- Compute the SVD $A = U\Sigma V^T$ using the Jacobi SVD ($AV = U\Sigma$, $AA^T = U\Sigma^2 U^T$).
- Assemble: $T = DL_M^{-T}U\Sigma^{1/2}$.
- It holds $T^{-1}HT^{-T} = T^T MT = \Sigma$

Accurate solution

The algorithm solves

$$(H + \delta H)(M + \delta M)x = \tilde{\lambda} x$$

exactly, with symmetric δH, δM,

$$\frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \leq f(n)\cdot\varepsilon, \quad \frac{|\delta M_{ij}|}{\sqrt{M_{ii}M_{jj}}} \leq g(n, L_{H,s})\cdot\varepsilon, \quad 1 \leq i,j \leq n$$

$$\frac{|\delta\lambda|}{\lambda} \leq h(n)(\|H_s^{-1}\| + \|M_s^{-1}\|)\cdot\varepsilon, \quad \varepsilon = \mathrm{eps}.$$

$H_s = \mathrm{diag}(H)^{-1/2} H \mathrm{diag}(H)^{-1/2}$, $\kappa_2(H_s) \leq n \min\limits_{D=diag} \kappa_2(DHD)$

All λ's stable IFF $\|H_s^{-1}\|$ and $\|M_s^{-1}\|$ moderate.
We have optimal accuracy.

PSVD

Implicit diagonalization of HM is actually computing the SVD of a product of two matrices, $BC^T = U\Sigma V^T$.
$A = BC^T = U\Sigma V^T$, B, C full column rank

$$BC^T = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix} \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$$

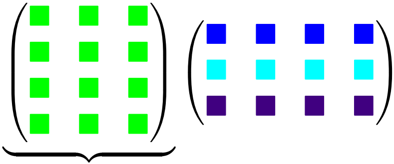
- A=yxGEMM(B,C^T) fastesssst matrix multiply
- CALL yxGESDD(A) fastesssst SVD
- $\begin{pmatrix} 1 & \epsilon \\ -1 & \epsilon \end{pmatrix} \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 2+2\epsilon & 2+\epsilon \\ -2+2\epsilon & -2+\epsilon \end{pmatrix} \approx \begin{pmatrix} 2 & 2 \\ -2 & -2 \end{pmatrix}$
- $U_2 B U_1^T U_1 C^T U_3 \rightsquigarrow \Sigma$, U_i orthogonal

PSVD

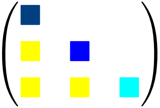
- $\begin{pmatrix} 1 & \epsilon \\ -1 & \epsilon \end{pmatrix} \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \approx \begin{pmatrix} 2 & 2 \\ -2 & -2 \end{pmatrix}$, ϵ not happy
- $\begin{pmatrix} 1 & \epsilon \\ -1 & \epsilon \end{pmatrix} U_1^T U_1 \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix}$, $U_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$
- $U_1 C^T = \begin{pmatrix} \sqrt{8} & \frac{\sqrt{18}}{2} \\ 0 & -\frac{\sqrt{2}}{2} \end{pmatrix}$
- $BU_1^T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1+\epsilon & -1+\epsilon \\ -1+\epsilon & 1+\epsilon \end{pmatrix} \approx \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$
- ϵ happy because U_1 is orthogonal ?!
- backward errors: $\|\delta B\| \lesssim \mathrm{eps}\|B\|$, $\|\delta C\| \lesssim \mathrm{eps}\|C\|$
- Is that the best we can do?

PSVD: SVD(BC^T)

How to compute the SVD of a product of two matrices, $BC^T = U\Sigma V^T$, accurately?

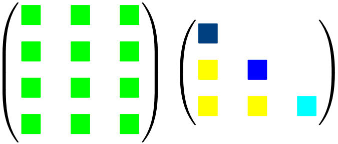
$B\Delta_B^{-1}\Delta_B C^T =$ 

unit columns

- $C\Delta_B P = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$; $BC^T = (B\Delta_B^{-1}P)\begin{pmatrix} R^T & 0 \end{pmatrix} Q^T$;
- $A = (B\Delta_B^{-1}P)R^T$; $R^T=$  $=$ well.cond $\times$ diag.
- $[U, \Sigma, V_1]$=SVD$(A)_{\text{Jacobi}}$; $V = Q \begin{pmatrix} V_1 & 0 \\ 0 & I_{n-p} \end{pmatrix}$

And what about BPR^T?

$BPT^T \equiv$



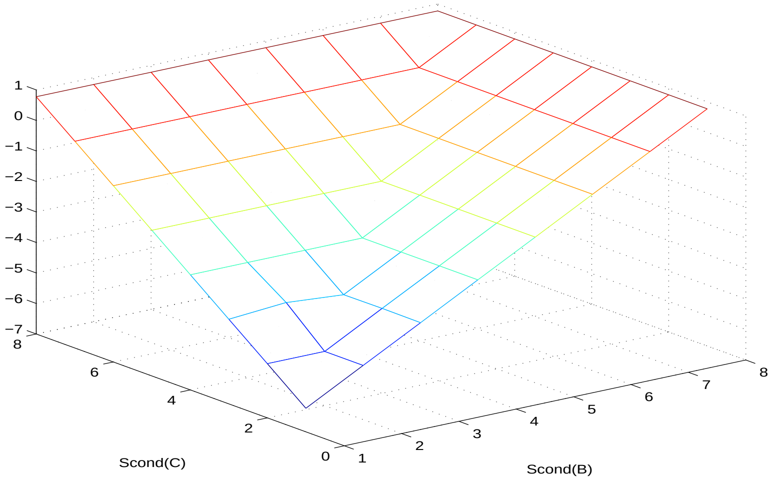
Consider $\begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix} \begin{pmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{pmatrix}$

$$\begin{aligned}
\tilde{a}_3 &= (b_3 + \delta b_3)\ell_{33} \\
\tilde{a}_2 &= (b_2 + \delta_2 b_2)\ell_{22} + (b_3 + \delta_2 b_3)\ell_{32} \\
&= (b_2 + \delta_2 b_2)\ell_{22} + (b_3 + \delta b_3 - \delta b_3 + \delta_2 b_3) \\
&= (b_2 + \delta_2 b_2 + (\delta_2 b_3 - \delta b_3)\frac{\ell_{32}}{\ell_{22}})\ell_{22} + (b_3 + \delta b_3)\ell_{33} \\
&= (b_2 + \delta b_2)\ell_{22} + (b_3 + \delta b_3)\ell_{33} \\
\begin{pmatrix} \tilde{a}_2 & \tilde{a}_3 \end{pmatrix} &= \begin{pmatrix} b_2 + \delta b_2 & b_3 + \delta b_3 \end{pmatrix} L, \quad L = \tilde{R}^T
\end{aligned}$$

Backward stability

- $C = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$;
  - $C + \delta C = \tilde{Q} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}$;
  - $\|\delta C(:, i)\| \leq \epsilon \|C(:, i)\|$, for all columns i
- $A = BR^T$;
  - $\tilde{A} = (B + \delta B)\tilde{R}^T$,
  - $\|\delta B(:, i)\| \leq \epsilon \|B(:, i)\|$, for all columns i
- $(B + \delta B)(C + \delta C)^T = (I + \delta B B^{\dagger}) B C^T (I + \delta C C^{\dagger})^T$
- $B = B_{scaled} D$, $scond(B) = cond(B_{scaled})$

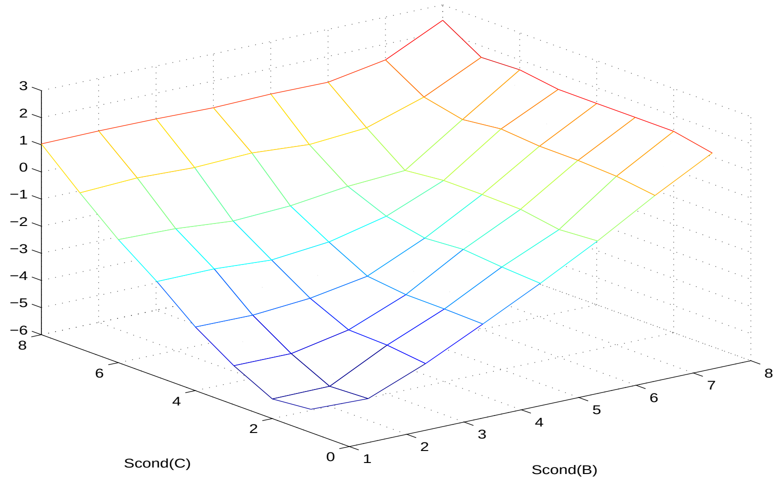
Theoretical accuracy



theory:

$$\max_i \frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq f(m,p,n) \cdot \mathfrak{e} \cdot \max\{scond(B), scond(C)\}$$

Measured accuracy



theory: measured $\max_i \frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i}$; in $(0.3, 46) \times$ theory

Jacobi, 1844, 1846

Ein leichtes Verfahren ...

$H = H^T$, $H^{(k+1)} = U_k^T H^{(k)} U_k \longrightarrow \Lambda = \mathrm{diag}(\lambda_i)$ $(k \longrightarrow \infty)$

Each U_k annihilates (p_k, q_k), (q_k, p_k) positions in $H^{(k)}$.

$$\cdots U_3^T U_2^T U_1^T \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix} U_1 U_2 U_3 \cdots = \begin{pmatrix} \bullet & \circledast & \otimes & 0 \\ \circledast & \bullet & \star & \bullet \\ \otimes & \star & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \end{pmatrix}$$

$$U_1 = \begin{pmatrix} \cos\psi_1 & \sin\psi_1 \\ -\sin\psi_1 & \cos\psi_1 \end{pmatrix} \bigoplus I_{n-2}, \quad U_2 = \cdots$$

$$\boxed{\text{Jacobi rotation}} \quad \cot 2\psi_k = \frac{H^{(k)}_{q_k q_k} - H^{(k)}_{p_k p_k}}{2H^{(k)}_{p_k q_k}},$$

$$\tan\psi_k = \frac{\mathrm{sign}(\cot 2\psi_k)}{|\cot 2\psi_k| + \sqrt{1 + \cot^2 2\psi_k}} \in (-\frac{\pi}{4}, \frac{\pi}{4}],$$

$(p, q) = \mathcal{P}(k)$ pivot strategy, $\mathcal{P} : \mathbb{N} \to \{(i, j) : i < j\}$

Convergent strategies

Jacobi: $|h_{pq}^{(k)}| = \max_{i \neq j} |h_{ij}^{(k)}|$, $\mathcal{P}(k) = (p, q)$.

Reading Jacobi's 1846. paper recommended.

Cyclic: $\mathcal{P}$ periodic, one full period called sweep.

Row–cyclic and column–cyclic:

$$\begin{pmatrix} \bullet & 1 \rightarrow & 2 \rightarrow & 3 \\ \bullet & \bullet & 4 \rightarrow & 5 \\ \bullet & \bullet & \bullet & 6 \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}, \quad \begin{pmatrix} \bullet & 1 \downarrow & 2 \downarrow & 4 \downarrow \\ \bullet & \bullet & 3 \downarrow & 5 \downarrow \\ \bullet & \bullet & \bullet & 6 \downarrow \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$$

$$\mathrm{Off}(H^{(k)}) = \sqrt{\sum_{i \neq j} (H^{(k)})_{ij}^2} \longrightarrow 0 \;\; (k \longrightarrow \infty)$$

$H^{(k)} \longrightarrow \Lambda$, $U_1 \cdots U_k \cdots \longrightarrow U$ as $(k \longrightarrow \infty)$; $U^T H U = \Lambda$

Asymptotically quadratic reduction of $\mathrm{Off}(H^{(k)})$.

Forsythe, Henrici, Wilkinson, Rutishauser, Hari, Veselić

Asymptotically cubic strategies exist.

One–sided Jacobi SVD

Hestenes used implicit Jacobi for SVD of $A \in \mathbb{R}^{m\times n}$:
Diagonalize $H = H^{(0)} = A^T A$; $A \equiv A_0$.
$H^{(1)} = V_0^T H^{(0)} V_0 = V_0^T A^T (AV_0) = A_1^T A_1$

$$\boxed{H^{(k+1)} = V_k^T H^{(k)} V_k} = A_{k+1}^T A_{k+1} \longrightarrow \Lambda = \mathrm{diag}(\lambda_i)$$

$\leftrightarrow \boxed{A_{k+1} = A_k V_k}$, where $H^{(k)} = A_k^T A_k$
V_k uses Jacobi rotation to diagonalize

$$\begin{pmatrix} h_{pp}^{(k)} & h_{pq}^{k} \\ h_{qp}^{(k)} & h_{qq}^{(k)} \end{pmatrix} \qquad \begin{aligned} h_{pp}^{(k)} &= \|A_k(1:m,p)\|^2 \\ h_{qq}^{(k)} &= \|A_k(1:m,q)\|^2 \\ h_{pq}^{(k)} &= A_k(1:m,p)^T A_k(1:m,q) \end{aligned}$$

$h_{pp}^{(k)}$, $h_{qq}^{(k)}$ scalar update; $h_{pq}^{(k)}$ BLAS1 SDOT
$A_k \longrightarrow U\Sigma$, $\Sigma = \mathrm{diag}(\sqrt{\lambda_i})$, $U^T U = I$
$V_1 \cdots V_k \cdots \longrightarrow V$, $V^T V = I$, $AV = U\Sigma$
$A = U\Sigma V^T$ the SVD of A.

One–sided rotation

$d_p = \|A_k(1:m,p)\|^2$, $d_q = \|A_k(1:m,q)\|^2$,
$\xi = A_k(1:m,p)^T A_k(1:m,q)$;

ROTATE$(A_{1:m,p}, A_{1:m,q}, d_p, d_q, \xi, [V_{1:m,p}, V_{1:m,p}])$

1: $\vartheta = \dfrac{d_q - d_p}{2\cdot\xi}$; $t = \dfrac{\mathrm{sign}(\vartheta)}{|\vartheta| + \sqrt{1+\vartheta^2}}$;

$c = \dfrac{1}{\sqrt{1+t^2}}$; $s = t\cdot c$;

2: $\begin{pmatrix} A_{1:m,p} & A_{1:m,q}\end{pmatrix} = \begin{pmatrix} A_{1:m,p} & A_{1:m,q}\end{pmatrix}\begin{pmatrix} c & s \\ -s & c\end{pmatrix}$;

3: $d_p = d_p - t\cdot\xi$; $d_q = d_q + t\cdot\xi$;

4: **if** V is wanted **then**

5: $\begin{pmatrix} V_{1:n,p} & V_{1:n,q}\end{pmatrix} = \begin{pmatrix} V_{1:n,p} & V_{1:n,q}\end{pmatrix}\begin{pmatrix} c & s \\ -s & c\end{pmatrix}$

6: **end if**

Can avoid squared norms. Can use fast rotations. Unit stride memory access. Vectorizable. Parallelizable.

Jacobi SVD

$\hat{p} = n(n-1)/2$; $s = 0$; *convergence* = **false** ;
if V is wanted **then initialize** $V = I_n$ **end if**
for $i = 1$ **to** n **do** $d_i = A^T_{1:m,i} A_{1:m,i}$ **end for**;
repeat
$s = s + 1$; $p = 0$;
for $i = 1$ **to** $n - 1$ **do**
  for $j = i + 1$ **to** n **do**
    $\xi = A^T_{1:m,i} A_{1:m,j}$;
    if $|\xi| > m\varepsilon\sqrt{d_i d_j}$ **then**
      call $\mathrm{ROTATE}(A_{1:m,i}, A_{1:m,j}, d_i, d_j, \xi, [V_{1:m,i}, V_{1:m,j}])$;
    else $p = p + 1$ **end if**
  end for
end for
if $p = \hat{p}$ **then** convergence=**true**; **go to ▶** **end if**
until $s > 30$

▶ **if** *convergence* **then** $\Sigma_{ii} = \sqrt{d_i}$, $U_{1:m,i} = A_{1:m,i}\Sigma^{-1}_{ii}$, $i = 1 : n$;
else Error: *Did not converge in 30 sweeps.* **end if**

Jacobi in floating–point

Breakthrough: Jacobi method is more accurate than QR!

Demmel and Veselić: Let $\tilde{H}^{(k)}$, denote the computed matrices. Then, in the positive definite case, one step of Jacobi in floating–point arithmetic reads

$$\tilde{H}^{(k+1)} = \hat{U}_k^T(\tilde{H}^{(k)} + \delta\tilde{H}^{(k)})\hat{U}_k$$

where $\hat{U}_k$ is exactly orthogonal and ε–close to the actually used Jacobi rotation $\tilde{U}_k$, and $\delta\tilde{H}^{(k)}$ is sparse with

$$\mathbf{e}_k = \max_{i,j} \frac{|(\delta\tilde{H}^{(k)})_{ij}|}{\sqrt{(\tilde{H}^{(k)})_{ii}(\tilde{H}^{(k)})_{jj}}} \leq \epsilon$$

Relative perturbation of eigenvalues in the k–step bounded by $n\mathbf{e}_k\|(\tilde{H}_s^{(k)})^{-1}\|_2$, $\tilde{H}_s^{(k)}$ scaled to have unit diagonal.

IMPORTANT: Stop when $\max_{i\neq j}|(\tilde{H}_s^{(k)})_{ij}| \leq \epsilon$

The accuracy depends on $\max_k \|(\tilde{H}_s^{(k)})^{-1}\|_2$

Jacobi in floating–point

If the entries of the initial H are given with relative uncertainty ε, then:

- The spectrum is determined up to relative error of order of $n\varepsilon\|H_s^{-1}\|$ (H_s diagonally scaled H to have unit diagonal)
- The symmetric Jacobi method introduces perturbation of the order of $n\mathbf{e}_k \max_k \|(\tilde{H}_s^{(k)})^{-1}\|_2$

Numerical evidence: $\max_k \|(\tilde{H}_s^{(k)})^{-1}\|_2$ behaves well.
Theoretical (still open) problem: Bound

$$\max_{k\geq 1} \frac{\|(H_s^{(k)})^{-1}\|_2}{\|H_s^{-1}\|_2} \quad \text{or} \quad \max_{k\geq 1} \frac{\kappa_2(H_s^{(k)})}{\kappa_2(H_s)}$$

Demmel, Veselić, Slapničar, Mascarenhas, Drmač

Provable accuracy

Let $H = LL^T \succ 0$, L Cholesky factor.
Use Veselić–Hari trick:

- If we apply Jacobi SVD to L, $LV = U\Sigma$, where V is the product of Jacobi rotations, then $H = U\Sigma^2 U^T$.
- So, can apply Jacobi and get eigenvectors without accumulation of Jacobi rotations! This reduces flop count, memory requirements and memory traffic!
- This implicitly diagonalizes $L^T L$, which is similar to $H = LL^T$, and it is actually one step of the Rutishauser's LR method. If L is computed with pivoting, then $L^T L$ is 'more diagonal' than H.
- The cost of Cholesky ($n^3/3$) much less than one sweep of Jacobi ($2n^3$ with fast rotations).
- In floating point

$$\tilde{L}\tilde{L}^T = H + \delta H, \quad \max_{i,j} \frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \leq \eta_C \lessapprox n\varepsilon$$

Provable accuracy

Now to the SVD of $\tilde{L}$:
One sided Jacobi SVD $\tilde{L} V_1 V_2 \cdots V_k \cdots V_\ell \to \tilde{U}\tilde{\Sigma}$
In floating point

- $\tilde{L} \leftarrow (((\tilde{L}_1 + \delta\tilde{L}_1)\hat{V}_1 + \delta\tilde{L}_2)\hat{V}_2 + \delta\tilde{L}_3)\hat{V}_3 + \cdots$

- If $y = xV$, V rotation, x, y row vectors, then $\tilde{y} = (x + \delta x)\hat{V}$, $\hat{V}$ orthogonal, $\|\delta x\| \leq 6\varepsilon\|x\|$.
- Hence, each row of $\delta\tilde{L}_i$ is ε small relative to the corresponding row of $\tilde{L}_i$. The $\hat{V}_j$ with $j \neq i$ do not change the row norms of $\delta\tilde{L}_i$.
- At convergence, $\tilde{U}\tilde{\Sigma} = (\tilde{L} + \delta\tilde{L})\hat{V}$, with $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_i)$, $\|\delta\tilde{L}(i,:)\| \leq O(n)\varepsilon\|\tilde{L}(i,:)\|$ for all i.
- $\tilde{\lambda}_i = \tilde{\sigma}_i^2$ are the eigenvalues of $(\tilde{L} + \delta\tilde{L})(\tilde{L} + \delta\tilde{L})^T$

Provable accuracy

$$(\tilde{L} + \delta\tilde{L})(\tilde{L} + \delta\tilde{L})^T = \tilde{L}\tilde{L}^T + \underbrace{\tilde{L}\delta\tilde{L}^T + \delta\tilde{L}\tilde{L}^T + \delta\tilde{L} + \delta\tilde{L}^T}_{E}$$

By Cauchy–Schwarz,

$$\begin{aligned} |E_{ij}| &\leq 2O(n\varepsilon)\|\tilde{L}(i,:)\|\|\tilde{L}(j,:)\| + O(\varepsilon^2)\|\tilde{L}(i,:)\|\|\tilde{L}(j,:)\| \\ &\approx (O(n\varepsilon) + O(\varepsilon^2))\sqrt{(\tilde{L}\tilde{L}^T)_{ii}(\tilde{L}\tilde{L}^T)_{jj}} \\ &\approx (O(n\varepsilon) + O(\varepsilon^2))\sqrt{H_{ii}H_{jj}}, \\ &\quad \text{since } \tilde{L}\tilde{L}^T = H + \delta H, \ \max_{i,j} \frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \leq \eta_C \lessapprox n\varepsilon \end{aligned}$$

So, we have the eigenvalues of

$$\tilde{L}\tilde{L}^T + E = H + \delta H + E = H + \Delta H, \ \max_{i,j} \frac{|\Delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \leq O(n\varepsilon)$$

Provable accuracy–conclusion

If $H \succ 0$ then

- The algorithm:
  1. Compute the Cholesky factorization $H = LL^T$;
  2. Compute $L = U\Sigma V^T$ using one–sided Jacobi SVD;
  3. Output: Set $\Lambda = \Sigma^2$; $H = U\Lambda U^T$

  computes the eigenvalues and eigenvectors of H with entry–wise small backward error $\max_{i,j} \frac{|\Delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \leq O(n\varepsilon)$.
- The forward error is $\max_i |\delta\lambda_i|/\lambda_i \leq O(n^2\varepsilon)\|H_s^{-1}\|_2$.
- Most of the forward error comes from Step 1. Step 2. in floating point is as good as exact SVD.
- If Cholesky in Step 1 fails to compute L, then the matrix is entry–wise close to a non–definite matrix, and smallest eigenvalue can be lost due to symmetric tiny entry–wise perturbations.
- All computations in one $n \times n$ array.

SVD perturbation theory

Let $\mathrm{rank}(A) = n \leq m$, $D = \mathrm{diag}(\|A(:,i)\|)$, and

$$A \mapsto A + \delta A \implies \sigma_j \mapsto \sigma_j + \delta\sigma_j .$$

$$A + \delta A = (I + \delta A A^\dagger)A \Longrightarrow \max_j \frac{|\tilde{\sigma}_j - \sigma_j|}{\sigma_j} \leq \|\delta A A^\dagger\|,$$

$$\|\delta A A^\dagger\| \leq \begin{cases} \dfrac{\|\delta A\|}{\|A\|}(\|A^\dagger\|\|A\|) = \epsilon \cdot \kappa(A), \\ \|\delta A D^{-1}\|\|(AD^{-1})^\dagger\|. \end{cases}$$

$\|\delta A D^{-1}\| \leq \sqrt{n}\max_j \frac{\|\delta A(:,j)\|}{\|A(:,j)\|} \leq \sqrt{n}\varepsilon$;

$\|(AD^{-1})^\dagger\| \equiv \|A_s^\dagger\| \leq \sqrt{n}\min_{\Delta=diag}\kappa(A\Delta)$

Possible: $\|A_s^\dagger\| \ll \kappa(A)$; always $\|A_s^\dagger\| \leq \sqrt{n}\kappa(A)$.

Jacobi SVD: $\|A_s^\dagger\| \longrightarrow$ more accurate .

bidiagonal SVD: $\kappa(A) \longrightarrow$ less accurate ,

bidiagonalization provokes $\kappa(A)$.

Jacobi++ SVD: $A = D_1 C D_2 \to D_1(C + \delta C)D_2$.

QRF preprocessor for Jacobi

$A = QR$; $[\tilde{Q}, \tilde{R}] = \mathtt{qr}(A)$, $\tilde{Q}$, $\tilde{R}$ computed.
Backward error analysis:

$$(\exists \delta A) \ \ (\exists \hat{Q}, \ \hat{Q}^T \hat{Q} = I) \ \ A + \delta A = \hat{Q}\tilde{R},$$

$$\|\delta A(:,i)\| \leq \epsilon_1 \|A(:,i)\|, \quad i = 1, \ldots, n.$$

Perturbation analysis: $\sigma_i(\tilde{R}) = \sigma_i((I + \delta A A^\dagger)A)$

$$1 - \|\delta A A^\dagger\| \leq \frac{\sigma_i(\tilde{R})}{\sigma_i(A)} \leq 1 + \|\delta A A^\dagger\|, \quad \text{for all } i.$$

Let $A = A_s D$, $D = \mathrm{diag}(\|A(:,i)\|)$.

$$\|\delta A A^\dagger\| = \|\delta A D^{-1}(AD^{-1})^\dagger\| \leq \sqrt{n} \max_i \frac{\|\delta A(:,i)\|}{\|A(:,i)\|} \|A_s^\dagger\|$$

$$\|A_s^\dagger\| \leq \kappa(A_s) \leq \sqrt{n} \min_{\Delta = \mathrm{diag}} \kappa(A\Delta)$$

If $\kappa(A_s)$ is moderate, then $SVD(\tilde{R})$ is OK for the $SVD(A)$.

Strong backward stability

Jacobi SVD($\tilde{R}$): $\tilde{R}^T V = U\Sigma$. Computed: $[\tilde{U}, \tilde{V}, \tilde{\Sigma}] = \texttt{JacobiSVD}(\tilde{R}^T)$. Jacobi rotations $\tilde{V}$ such that

$$\max_{i \neq j} \left| \cos \angle((\tilde{U}\tilde{\Sigma})e_i, (\tilde{U}\tilde{\Sigma})e_j) \right| \leq O(n)\mathbf{u}$$

Error analysis:

$$(\exists \delta\tilde{R}) \;\; (\exists \hat{V},\; \hat{V}^T\hat{V} = I) \;\; (\tilde{R} + \delta\tilde{R})^T \hat{V} = (\tilde{U}\tilde{\Sigma})$$

$$\|\delta\tilde{R}(:,i)\| \leq \epsilon_2 \|\tilde{R}(:,i)\|, \quad i = 1, \ldots, n.$$

Finally,

$$\begin{aligned} \tilde{R} + \delta\tilde{R} &= \hat{Q}^T(A + \delta A) + \hat{Q}^T\hat{Q}\delta\tilde{R} \\ &= \hat{Q}^T(A + \underbrace{\delta A + \hat{Q}\delta\tilde{R}}_{\Delta A}) \end{aligned}$$

where $\|\Delta A(:,i)\| \leq (\epsilon_1 + \epsilon_2(1 + \epsilon_1))\|A(:,i)\|$ for all i, and the SVD is $(A + \Delta A)^T \hat{Q}\hat{V} = \tilde{U}\tilde{\Sigma}$. Very nice and simple. <u>Accurate</u>.