Vorlesung 401-1652-10L : Numerische Mathematik I

Frühlingssemester 2014 : Prof. Ralf Hiptmair (SAM, D-MATH)

[Vorlesungsniederschrift, Rohformat, URL: http://www.math.ethz.ch/education/bachelor/lectures/fs2014/math/nm]

Kapitel 2 : Direkte Lösungsverfahren für lineare Gleichungssysteme (LGS)

LGS : $\underline{A}\underline{x} = \underline{b}$, $\underline{b} \in \mathbb{C}^n$, $\underline{A} \in \mathbb{C}^{n,n}$, A regulär (= invertierbar) $\Rightarrow \boxed{\underline{x} = \underline{A}^{-1}\underline{b}}$

Grundsatz : Berechne nie numerisch die Inverse einer Matrix

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ \vdots & \vdots & \underline{A} & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \underline{x} \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \underline{b} \\ \vdots \\ b_n \end{pmatrix}$$

Lösungsalgorithmus : Gausselimination

$$\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ -3 \end{pmatrix} \quad \longleftrightarrow \quad \begin{array}{rcl} x_1 + x_2 & = & 4 \\ 2x_1 + x_2 - x_3 & = & 1 \\ 3x_1 - x_2 - x_3 & = & -3 \end{array} .$$

$$\hookrightarrow \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \\ -3 \end{pmatrix} \rightarrow \hookrightarrow \begin{pmatrix} \mathbf{1} & 1 & 0 \\ 0 & -1 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ -7 \\ -3 \end{pmatrix} \rightarrow \hookrightarrow \begin{pmatrix} \mathbf{1} & 1 & 0 \\ 0 & -1 & -1 \\ 0 & -4 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ -7 \\ -15 \end{pmatrix}$$

▭ ≙ Pivotzeile

$$\rightarrow \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & \mathbf{-1} & -1 \\ 0 & 0 & 3 \end{pmatrix}}_{\text{Obere Dreiecksmatrix}} \begin{pmatrix} 4 \\ -7 \\ 13 \end{pmatrix}$$

≙ gestaffeltes Gleichungssystem, äquivalent (= gleiche Lösungsmenge) zum Ausgangs-LGS

Dann Rücksubstitution $x_3 \rightarrow x_2 \rightarrow x_1$

```
1  function x = GEnopivot(A,b)
2  % Raw Gaussian elimination without pivoting for
3  % linear system Ax=b: "Linear algebra version", UNSTABLE!
4  [m,n] = size(A); A = [A,b];   <- rechte Seite -> letzte Spalte
5  % Forward elimination
6  for i=1:n-1
7    if (A(i,i) == 0.0), error('Zero pivot'); end
8    for j=i+1:n
9      fac = A(j,i)/A(i,i);
10     for k=i+1:n+1
11       A(j,k) = A(j,k) - fac*A(i,k);
12     end
13   end
14 end
15 % Backward substitution
16 x = [zeros(n-1,1);A(n,n+1)/A(n,n)];
17 for i=n-1:-1:1
18   x(i) = A(i,n+1);
19   for k=i+1:n
20     x(i) = x(i) - A(i,k)*x(k);
21   end
22   x(i) = x(i)/A(i,i);
23 end
```

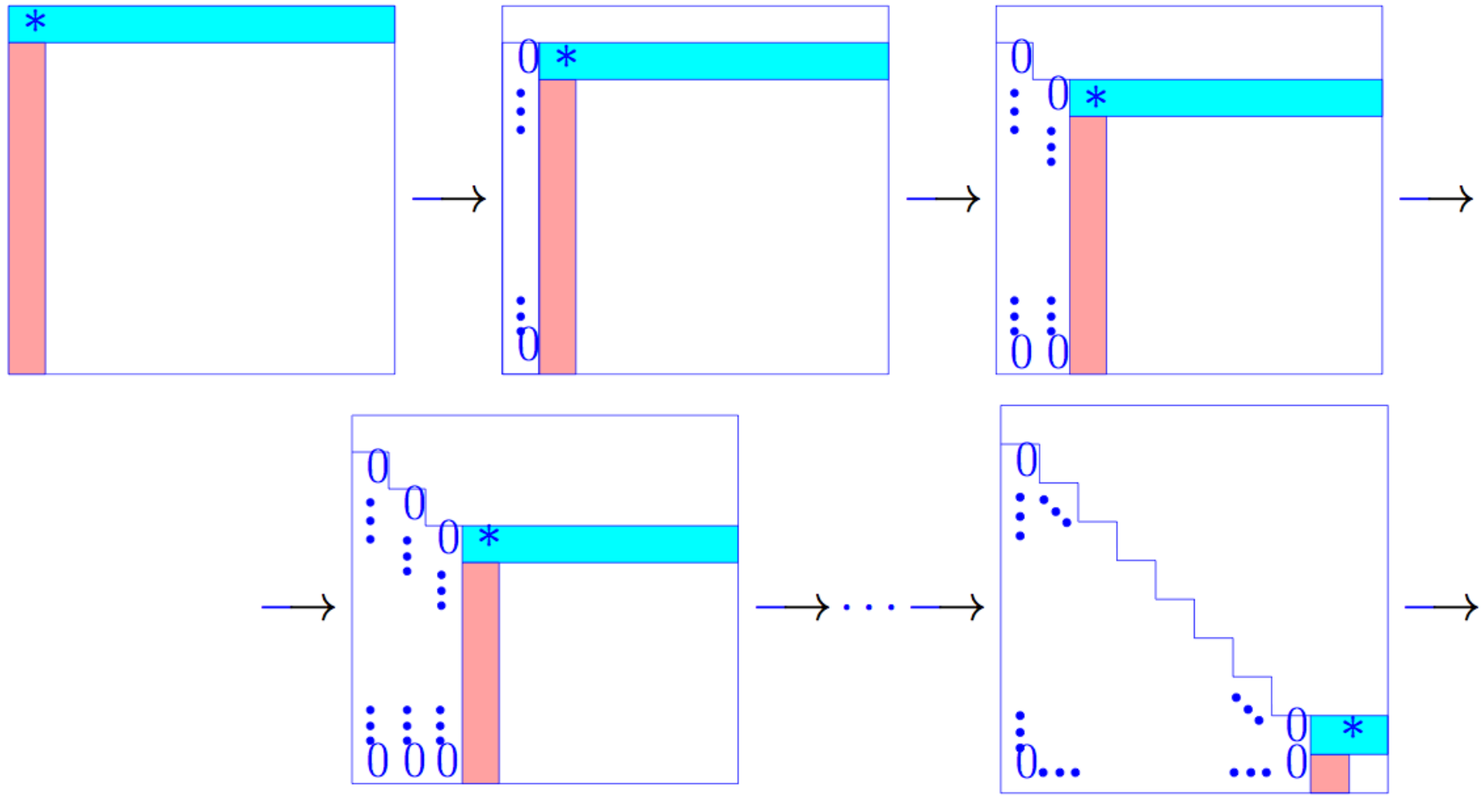
Numerische Sünde! wg. Rundungsfehlern

Anstatt (x == 0) frage ab

abs(x) ≤ eps * abs(x_{ref})

↳ Hier A(1:n,1:n) ≙ obere Dreiecksmatrix (bei exakter Arithmetik)

Vorwärts elimination:



$\cong$ obere Dreiecksmatrix

2.1. Gestaffelte LGS

$$\boldsymbol{L}\underline{x} = \underbrace{\begin{pmatrix} \ell_{11} & 0 & \cdots & \cdots & 0 \\ \ell_{21} & \ell_{22} & 0 & \cdots & \\ \ell_{31} & \ell_{32} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn-1} & \ell_{nn} \end{pmatrix}}_{\text{untere Dreiecksmatrix, } \ell_{ii} \neq 0 \Rightarrow L \text{ regulär}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

Algorithmus 2.1 Vorwärtseinsetzen

Input: Reguläre untere Dreiecksmatrix $\boldsymbol{L} \in \mathbb{R}^{n\times n}$, $\underline{b} \in \mathbb{R}^n$.
Output: Lösung $\underline{x}$ von $\boldsymbol{L}\underline{x} = \underline{b}$.

for $i = 1, 2, \ldots, n$ **do**

$$x_i := \frac{1}{\ell_{ii}} \left(b_i - \sum_{k=1}^{i-1} \ell_{ik} x_k \right)$$

end for

$x_1 \to x_2 \to \ldots \to x_n$

$$\boldsymbol{U}\underline{x} = \begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & \cdots & u_{2n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

, $u_{ii} \neq 0$

Algorithmus 2.2 Rückwärtseinsetzen

Input: Reguläre obere Dreiecksmatrix $\boldsymbol{U} \in \mathbb{R}^{n\times n}$, $\underline{b} \in \mathbb{R}^n$.
Output: Lösung $\underline{x}$ von $\boldsymbol{U}\underline{x} = \underline{b}$.

for $i = n, n-1, \ldots, 1$ **do**

$$x_i := \frac{1}{u_{ii}} \left(b_i - \sum_{k=i+1}^{n} u_{ik} x_k \right)$$

end for

Rechenaufwand: # der für einen Algorithmus erforderlichen elementaren Operationen (flops)

Alg. 2.2: R.A. $= \sum_{i=1}^{n} 1 + 1 + (n-i) + (n-i-1) = 2n + 2n^2 - n - n(n+1) = n^2$

DIV SUB MUL ADD

[evtl. negative Terme $= 0$ setzen]

↕ korreliert mit Rechenzeit

Einfluss von Rundungsfehlern

Lemma 2.4. Die von `forwardsubst` (oder Algorithmus 2.2 für $\boldsymbol{U}$ anstelle von $\boldsymbol{L}$) in Gleitkommaarithmetik $\mathbb{F}$ berechnete Lösung $\widehat{\underline{x}}$ erfüllt

$$(\boldsymbol{L}+\Delta\boldsymbol{L})\widehat{\underline{x}} = \underline{b}, \quad \text{für ein} \quad \Delta\boldsymbol{L} \quad \text{mit} \quad |\Delta\boldsymbol{L}| \leq \gamma_n(\mathbb{F})|\boldsymbol{L}| \,,$$

wobei $\gamma_n(\mathbb{F}) := \frac{nu(\mathbb{F})}{1-nu(\mathbb{F})}$ (wie in Lemma 1.15).

↑ komponentenweise Abschätzung

```
1  function x = forwardsubst(L,b)
2  % Solution of lower triangular linear system of equations
3  % by means of forward substitution
4  [n,m] = size(L); x = zeros(n,1);
5  for i=1:n
6    for j=1:i-1, x(i) = x(i) + L(i,j)*x(j); end    (*)
7    x(i) = (b(i) - x(i))/L(i,i);
8  end
```

⇓

numerisch stabil

$\Delta L \triangleq$ „Rückwärtsfehler" : Lemma 2.4 ⇒

Beweis : (→ siehe Rundungsfehleranalyse für Euklidisches Skalarprodukt, Kap. 1, Seite 10)

$$\widetilde{x}_i = \Big(b_i - \sum_{j=1}^{i-1} l_{ij}(1+\Theta_j)\widetilde{x}_j\Big)(1+\delta)/l_{ii}(1+\delta) \,, \qquad |\Theta_j| \leq \gamma_j(\mathbb{F}) \leq \gamma_n(\mathbb{F})$$

$$\Rightarrow \underbrace{l_{ii}(1+\Theta_2)}_{=:\widetilde{l}_{ii}}\widetilde{x}_i = b_i - \sum_{j=1}^{i-1} \underbrace{l_{ij}(1+\Theta_j)}_{=:\widetilde{l}_{ij}}\widetilde{x}_j \qquad \frac{1}{(1+\delta)(1+\delta)} = 1+\Theta_2$$

$$\Rightarrow \quad \widetilde{L}\widetilde{\underline{x}} = \underline{b} \,, \quad \widetilde{L} = L + \Delta L \,, \quad (\Delta L)_{ij} = l_{ij}\cdot\Theta_j$$

□

2.2. LU-Zerlegung und Gausselimination

Neue Perspektive : $A\underline{x} = \underline{b}$, $A \in \mathbb{C}^{n\times n}$ regulär, $\underline{b} \in \mathbb{C}^n$

Es sei bekannt : $A = L\cdot U$, $L, U \in \mathbb{C}^{n\times n}$,

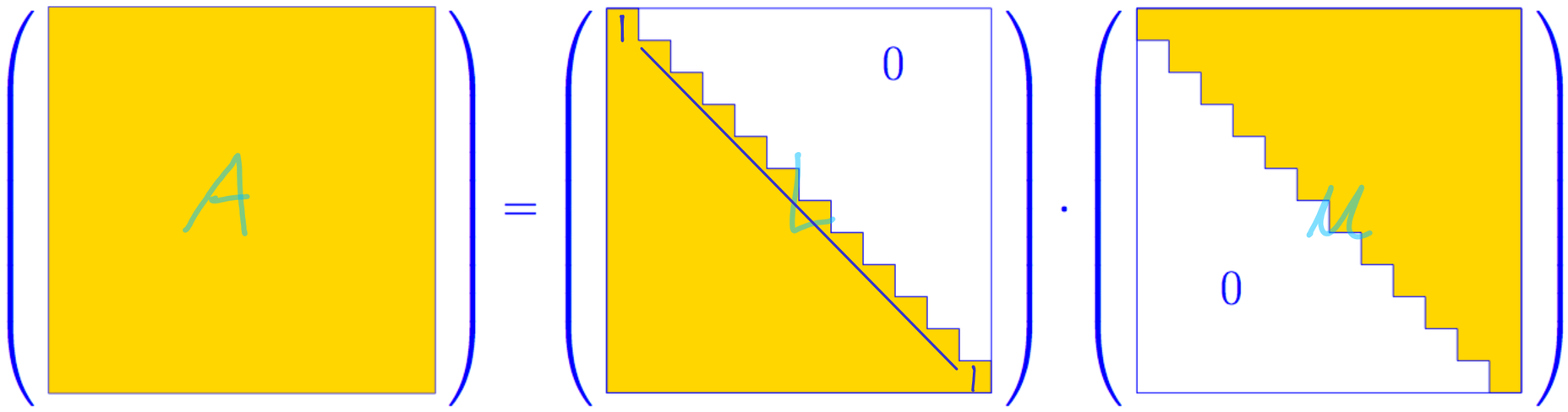
$L \triangleq$ untere Dreiecksmatrix
$U \triangleq$ obere Dreiecksmatrix
} invertierbar

$$L\underline{z} = \underline{b} \quad \Rightarrow \quad \underline{z} = L^{-1}\underline{b} \,, \quad \underline{x} = U^{-1}\underline{z}$$

gestaffelte Gleichungssysteme

[MATLAB : $\underline{x} = U\backslash(L\backslash\underline{b})$]

"Lösen von $Ax = b$ in MATLAB: $x = A \backslash b$ 'backslash'



Definition 2.5 (LU-Zerlegung)

Sei $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. Dann besitzt $\boldsymbol{A}$ eine LU-Zerlegung, falls es eine obere Dreiecksmatrix $\boldsymbol{U}$ und eine untere Dreiecksmatrix $\boldsymbol{L}$ mit $\ell_{11} = \cdots = \ell_{nn} = 1$ gibt, so dass $\boldsymbol{A} = \boldsymbol{LU}$.

Satz 2.6. (Eindeutigkeit der LU-Zerlegung)

Sei $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ regulär und habe eine LU-Zerlegung $\boldsymbol{LU} = \boldsymbol{A}$. Dann ist $(\boldsymbol{U})_{ii} \neq 0$ für $i = 1, \ldots, n$, und die Zerlegung ist eindeutig.

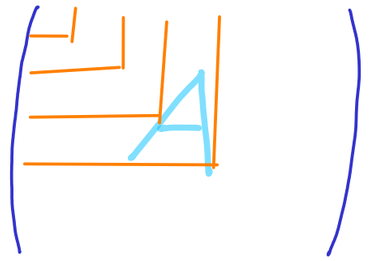
Beweis: L, U sind regulär, $A = L_1 M_1 = L_2 M_2 \Rightarrow M_1 M_2^{-1} = L_1^{-1} L_2 = I$

($M_1 M_2^{-1}$: obere Δ-Mat; $L_1^{-1} L_2$: untere Δ-Mat (mit Einheitsdiagonalen))

Lemma Die invertierbaren oberen und unteren Dreiecksmatrizen bilden eine Gruppe mit der Matrixmultiplikation als Verknüpfung.

$\Rightarrow M_1 = M_2 \wedge L_1 = L_2$ (wg. Eindeutigkeit der Inversen) □

Theorem 2.9. (Existenz einer LU-Zerlegung)
Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ regulär. Dann besitzt $\boldsymbol{A}$ eine LU-Zerlegung genau dann, wenn alle Hauptminoren $\boldsymbol{A}_k$, $k = 1,\ldots,n$ regulär sind. Hierbei ist $\boldsymbol{A}_k \in \mathbb{R}^{k\times k}$ gegeben durch $(\boldsymbol{A}_k)_{ij} = (\boldsymbol{A})_{ij}$, i, $j = 1,\ldots,k$. [MATLAB: A_k = A(1:k,1:k)]



Beweis: Induktion: $n = 1$ ✓

$n-1 \to n$:

Ansatz (Block-Matrix-Perspektive)

$$\left(\begin{array}{c|c} A_{n-1} & \underline{b} \\ \hline \underline{a}^T & \alpha \end{array}\right) = \left(\begin{array}{c|c} L_{n-1} & 0 \\ \hline \underline{x}^T & 1 \end{array}\right)\left(\begin{array}{c|c} U_{n-1} & \underline{y} \\ \hline 0 & \xi \end{array}\right) \Rightarrow \begin{array}{l} L_{n-1}\underline{y} = \underline{b} \Rightarrow \underline{y} = L_{n-1}^{-1}\underline{b} \\ \underline{x}^T U_{n-1} = \underline{a}^T \Rightarrow \underline{x} = (U_{n-1}^{-1})^T\underline{a} \end{array}$$

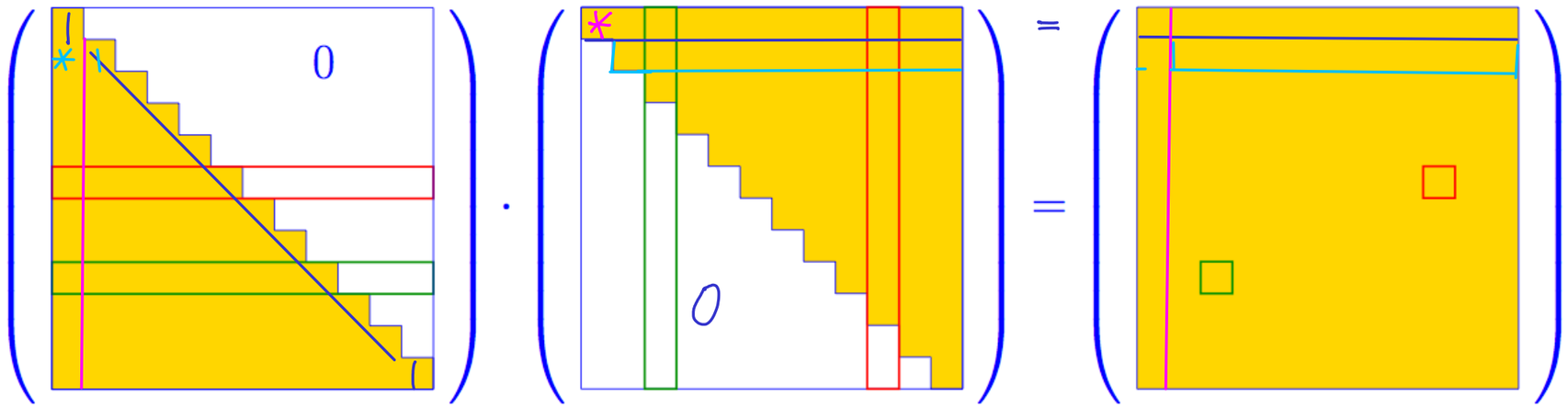
(A_{n-1}: regulär; $L_{n-1}\cdot U_{n-1}$: LU-Zerlegung von A_n)

Induktionsannahme: $A_{n-1} = L_{n-1}\cdot U_{n-1}$ (LU-Zerlegung) (L_{n-1}, U_{n-1}: regulär) $\quad \xi = \alpha - \underline{x}^T\underline{y}$

□

▷ Rekursiver Algorithmus zur Berechnung der LU-Zerlegung

Praxis: Parkettierungsalgorithmus



$$\mathbf{LU} = \mathbf{A} \quad \Rightarrow \quad a_{ik} = \sum_{j=1}^{\min\{i,k\}} l_{ij}u_{jk} = \begin{cases} \sum_{j=1}^{i-1} l_{ij}u_{jk} + 1\cdot u_{ik} & \text{, if } i \le k\,, \\ \sum_{j=1}^{k-1} l_{ij}u_{jk} + l_{ik}u_{kk} & \text{, if } i > k\,. \end{cases}$$

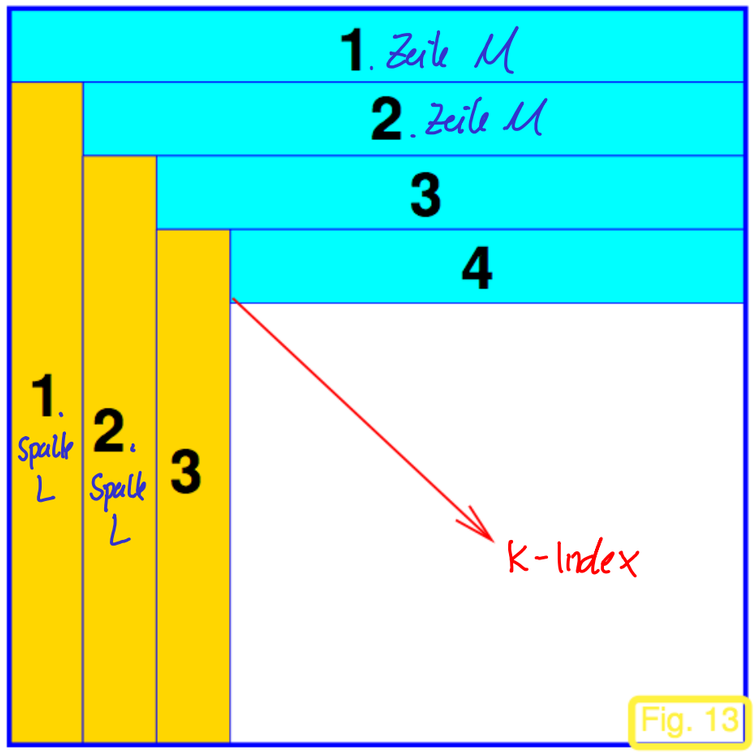


Fig. 13

```
1   function [L,U] = lufak(A)
2   % Algorithm of Crout: LU-factorization of A :  Instabil !
3   n = size(A,1); if (size(A,2) ~= n), error('n ~= m'); end
4   L = eye(n); U = zeros(n,n);
5   for k=1:n
6     % Compute row of U
7     for j=k:n, U(k,j) = A(k,j) - L(k,1:k-1)*U(1:k-1,j); end
8     % Compute column of L
9     for i=k+1:n, L(i,k) = (A(i,k) - L(i,1:k-1)*U(1:k-1,k)) /U(k,k); end
10  end
11
```

↓ = 0 ist möglich

⇔ Nullpivot in GE

Zusammenhang mit GE :

```
1   function x = GEnopivot(A,b)
2   % Raw Gaussian elimination without pivoting for
3   % linear system Ax=b: "Linear algebra version", UNSTABLE!
4   [m,n] = size(A); A = [A,b];
5   % Forward elimination
6   for i=1:n-1
7     if (A(i,i) == 0.0), error('Zero pivot'); end
8     for j=i+1:n
9       fac = A(j,i)/A(i,i);
10      for k=i+1:n+1
11        A(j,k) = A(j,k) - fac*A(i,k);
12      end
13    end
14  end
15  % Backward substitution
16  x = [zeros(n-1,1);A(n,n+1)/A(n,n)];
17  for i=n-1:-1:1
18    x(i) = A(i,n+1);
19    for k=i+1:n
20      x(i) = x(i) - A(i,k)*x(k);
21    end
22    x(i) = x(i)/A(i,i);
23  end
```

```
1   function [L,U] = gelu(A)
2   % LU-factorization based on Gaussian elimination
3   % No pivoting: UNSTABLE!
4   [m,n] = size(A); L = eye(n);
5   % Forward elimination
6   for i=1:n-1
7     if (A(i,i) == 0.0), error('Zero pivot'); end
8     for j=i+1:n
9       fac = A(j,i)/A(i,i); L(j,i) = fac;
10      for k=i+1:n
11        A(j,k) = A(j,k) - fac*A(i,k);
12      end
13    end
14  end
15  U = triu(A);
```

|| numerisch "äquivalent

```
1   function [L,U] = lufak(A)
2   % Algorithm of Crout: LU-factorization of A
3   n = size(A,1); if (size(A,2) ~= n), error('n ~= m'); end
4   L = eye(n); U = zeros(n,n);
5   for k=1:n                                  Skalarprodukt
6     % Compute row of U                          ↓
7     for j=k:n, U(k,j) = A(k,j) - L(k,1:k-1)*U(1:k-1,j); end
8     % Compute column of L
9     for i=k+1:n, L(i,k) = (A(i,k) - L(i,1:k-1)*U(1:k-1,k)) /U(k,k); end
10  end
11
```

Rechenaufwand für 'lufak':

$$\#flops = \sum_{k=1}^{n}\Big(\sum_{j=k}^{n}\{\underbrace{(k-1)}_{\text{MUL}}+\underbrace{(k-1)}_{\text{ADD}}\}+\sum_{i=k+1}^{n}\{1+2(k-1)\}\Big) = \sum_{k=1}^{n}(n-k+1)2(k-1)+(n-k)(1+2(k-1)) =$$

$$= \sum_{k=1}^{n} 2k^2 + \ldots\ldots = 2\tfrac{1}{6}(2n^3+3n^2+n)+\ldots\ldots$$

$$= \tfrac{2}{3}n^3 + O(n^2) \quad \text{für} \quad n \to \infty$$

↑ 3 geschachtelte Schleifen

2.3. Rundungsfehleranalyse der Gausselimination

Algorithmus: lufak + Vorwärtsrechnung + Rücksubstitution ('backwardsubst')

Theorem 2.15. (Rückwärtsfehler beim Lösen eines LGS mit Gausselimination)
Sei die LU-Zerlegung mit `lufak` in Gleitkommaarithmetik $\mathbb{F}$ durchführbar.* Dann erfüllt die in $\mathbb{F}$ berechnete Lösung $\widehat{x}$ des linearen Gleichungssystems $\boldsymbol{A}\underline{x} = \underline{b}$

$$(\boldsymbol{A}+\triangle\boldsymbol{A})\widetilde{\underline{x}} = \underline{b}, \quad \text{für ein} \quad \triangle\boldsymbol{A} \quad \text{mit} \quad |\triangle\boldsymbol{A}| \le (3\gamma_n+\gamma_n^2)|\widehat{\boldsymbol{L}}||\widehat{\boldsymbol{U}}|,$$

wobei $\widehat{\boldsymbol{L}}$ und $\widehat{\boldsymbol{U}}$ die von Algorithmus `lufak` in $\mathbb{F}$ berechneten Faktoren der LU-Zerlegung sind und $\gamma_n(\mathbb{F}) := \frac{nu(\mathbb{F})}{1-nu(\mathbb{F})}$ (wie in Lemma 1.15).

* keine Division durch 0

▷ stabil, falls $|\hat{L}||\hat{U}| \approx |A|$

Beweis: ① LU-Zerlegung

```
1   function [L,U] = lufak(A)
2    % Algorithm of Crout: LU-factorization of A
3    n = size(A,1); if (size(A,2) ~= n), error('n ~= m'); end
4    L = eye(n); U = zeros(n,n);
5    for k=1:n
6      % Compute row of U
7      for j=k:n, U(k,j) = A(k,j) - L(k,1:k-1)*U(1:k-1,j); end   → Skalarprodukt (*)
8      % Compute column of L
9      for i=k+1:n, L(i,k) = (A(i,k) - L(i,1:k-1)*U(1:k-1,k)) /U(k,k); end   (**)
10   end
11
```

$[(1+\theta_2)\,U(k,k)L(i,k) = A(i,k) - L(i,1:k-1)*U(1:k-1,k)]$

Technik wie bei R.F.A. für Euklidisches Skalarprodukt

(*) $\left| 1\cdot\tilde{u}_{kj} + \sum_{i=1}^{k-1} \tilde{l}_{ki}\tilde{u}_{ij} - a_{kj} \right|$ (*)$\leq$ $\gamma_k(\mathbb{F}) \sum_{i=1}^{k} |\tilde{l}_{ki}||\tilde{u}_{ij}|$

Rechenfehler im Skalarprodukt

⌜ Erinnerung: Abschnitt 1.2
$s_n = \sum_{j=1}^{n} x_j y_j$
$\Rightarrow |\tilde{s}_n - s_n| \leq \gamma_n(\mathbb{F}) \sum_{j=1}^{n} |x_j||y_j|$ ⌟(*)

(**) $\left| \sum_{j=1}^{k} \tilde{l}_{ij}\tilde{u}_{jk} - a_{ik} \right|$ (*)$\leq$ $\gamma_k(\mathbb{F}) \sum_{j=1}^{k} |\tilde{l}_{ij}||\tilde{u}_{jk}|$

$\Downarrow$ $\qquad\qquad$ $\Downarrow$

$|\tilde{L}\tilde{U} - A| \leq \gamma_n(\mathbb{F})\,|\tilde{L}||\tilde{U}| \Rightarrow \tilde{L}\tilde{U} = A + \Delta\tilde{A}$, $|\Delta\tilde{A}| \leq \gamma_n(\mathbb{F})|\tilde{U}||\tilde{L}|$

② Vorwartsrechnung: $\tilde{L}\tilde{y} = \underline{b}$

```
1  function x = forwardsubst(L,b)
2  % Solution of lower triangular linear system of equations
3  % by means of forward substitution
4  [n,m] = size(L); x = zeros(n,1);
5  for i=1:n
6    for j=1:i-1, x(i) = x(i) + L(i,j)*x(j); end   -> Skalarprodukt
7    x(i) = (b(i) - x(i))/L(i,i);
8  end
```
($L(i,i) = 1$)

$\tilde{x}_i = b_i - \sum_{j=1}^{i-1} \tilde{l}_{ij}\tilde{x}_j$ ↑ Skalarprodukt

⌜→ Anwendung von (*) auf
$\left| \sum_{j=1}^{i-1} \tilde{l}_{ij}\tilde{x}_j + 1\cdot\tilde{x}_i - b_i \right| \leq \gamma_i(\mathbb{F}) \sum_{j=1}^{i} |\tilde{l}_{ij}||\tilde{x}_j|$
$[\tilde{l}_{ii} = 1]$

▷ Aus der "Rückwartsfehleranalyse des Skalarprodukts

$\tilde{x}_i = b_i - \sum \tilde{\tilde{l}}_{ij}\tilde{x}_j$ mit $|\tilde{\tilde{l}}_{ij} - \tilde{l}_{ij}| \leq \gamma_i(\mathbb{F})|\tilde{l}_{ij}|$

▷ Berechnete Lsg. $\tilde{y}$ erfüllt: $(\tilde{L} + \Delta L)\tilde{y} = \underline{b}$ mit $|\Delta L| \leq \gamma_n(\mathbb{F})|\tilde{L}|$

③ "Rückwärtsrechnung": $\tilde{U}\underline{x} = \tilde{y}$ → Berechnetes $\tilde{x}$ erfüllt: $(\tilde{U} + \Delta U)\tilde{x} = \tilde{y}$ mit $|\Delta U| \leq \gamma_n(\mathbb{F})|\tilde{U}|$

$\Rightarrow \quad (\tilde{L} + \Delta L)(\tilde{U} + \Delta U)\tilde{x} = \underline{b}$

$\Leftrightarrow \quad (A + \underbrace{\Delta\tilde{A} + \Delta L\cdot\tilde{U} + \tilde{L}\cdot\Delta U + \Delta L\cdot\Delta U}_{=\text{ Störung } \Delta A})\tilde{x} = \underline{b}$

$$|\Delta A| \le |\Delta\tilde{A}| + |\Delta L||\hat{U}| + |\hat{L}||\Delta U| + |\Delta L|\cdot|\Delta U| \le \gamma_n(F)|\hat{L}||\hat{U}| + \gamma_n|\hat{L}||\hat{U}| + \gamma_n|\hat{U}||\hat{L}| + \gamma_n^2|\hat{U}||\hat{L}|$$

□

Normbasierte Formulierung:

Satz 2.15 $\Rightarrow$

$$\max_i \sum_j |(\Delta A)_{ij}| \le (3\gamma_n + \gamma_n^2) \underbrace{\max_i \sum_j (|\hat{L}||\hat{U}|)_{ij}}_{\| |\hat{L}||\hat{U}| \|_\infty}$$

$$\| \Delta A \|_\infty \le (\ \text{“}\ ) \ \| |\hat{L}||\hat{U}| \|_\infty$$

In $\|\cdot\|_2$: Verwende $\frac{1}{\sqrt{n}}\|M\|_\infty \le \|M\|_2 \le \sqrt{n}\,\|M\|_\infty$

$$\Rightarrow \quad \|\Delta A\|_2 \le \underbrace{n(3\gamma_n + \gamma_n^2)}_{\sim n^2 u(F)} \| |\hat{L}||\hat{U}| \|_2$$

$\ulcorner \|v\|_2^2 = \sum_{i=1}^n |v_i|^2 \le n \max_i |v_i|^2 = n\|v\|_\infty^2$

$\|v\|_\infty^2 = \max_i |v_i|^2 \le \sum_{i=1}^n |v_i|^2 = \|v\|_2^2$

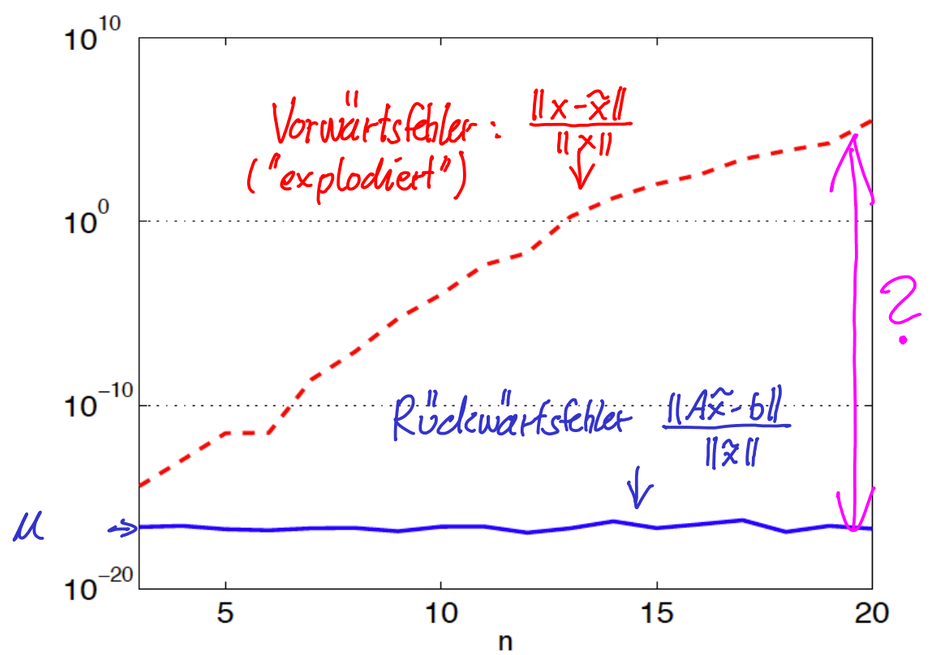
$\|M\|_2 = \sup_{x\neq 0} \frac{\|Mx\|_2}{\|x\|_2} \le$

$\le \sup_{x\neq 0} \frac{\sqrt{n}\,\|Mx\|_\infty}{\|x\|_\infty} = \sqrt{n}\,\|M\|_\infty \ \lrcorner$

2.4. Sensitivität und Kondition

Bsp: Hilbertmatrix $A \in \mathbb{R}^{n,n}$, $a_{ij} := \frac{1}{i+j-1}$, $1 \le i,j \le n$

```
1  digits(100); rw = []; vw = [];
2  for n = 3:20,
3    A = hilb(n); b = ones(n,1);
4    x = A\b; % Solve LSE with std precision  -> Gausselimination
5    rw = [rw;n,norm(A*x-b)/norm(x)];
6    xe = vpa(A)\vpa(b); % high precision solve  -> "exakte Lösung"
7    vw = [vw;n,norm(double(x-xe))/norm(double(xe))];  -> relativer Fehler
8  end
9  figure('name','Gaussian elimination for Hilbert matrix');
10 plot(rw(:,1),rw(:,2),'b-',vw(:,1),vw(:,2),'r--');
11 xlabel('n','fontsize',14);
```



Warum betrachten wir $\frac{\|A\hat{x}-b\|}{\|\hat{x}\|}$?

Theorem 2.17. (Schranke für Rückwärtsfehler für Lösung eines LGS)
Sei $\hat{x} \neq 0$ die berechnete Lösung des linearen Gleichungssystems $\boldsymbol{A}x = b$. Dann gilt

$$\min\{\|\Delta \boldsymbol{A}\|_2 : (\boldsymbol{A}+\Delta\boldsymbol{A})\hat{x} = b\} = \frac{\|r\|_2}{\|\hat{x}\|_2},$$

◁ Normen verfügbarer Vektoren

↑ „Rückwärtsfehler"

wobei $r = b - \boldsymbol{A}\hat{x}$ das Residuum zu $\hat{x}$ ist.

Beweis: (i) $\Delta A \cdot \hat{x} = r \Rightarrow \|\Delta A\|_2 \|\hat{x}\|_2 \geq \|\Delta A \cdot \hat{x}\|_2 = \|r\|_2 \Rightarrow \|\Delta A\|_2 \geq \frac{\|r\|}{\|\hat{x}\|}$

(i) Mit $\Delta A = \frac{1}{\|\hat{x}\|_2^2} r \hat{x}^H \in \mathbb{C}^{n\times n}$: $(A+\Delta A)\hat{x} = \left(A + \frac{1}{\|\hat{x}\|_2^2} r\hat{x}^H\right)\hat{x} = A\hat{x} + r = b$

$$\|\Delta A\|_2 = \frac{1}{\|\hat{x}\|_2^2}\|r\hat{x}\|_2 = \frac{1}{\|\hat{x}\|_2^2}\|r\|_2\|\hat{x}\|_2 = \frac{\|r\|_2}{\|\hat{x}\|_2}$$

2.4.1. Kondition einer Abbildung

$X, Y \triangleq$ normierte Vektorräume, Problemabbildung: $F : \mathcal{D}(F) \subset X \to Y$

↑ Datenraum ↑ Ergebnisraum

Wie wirkt sich eine (infinitesimal) kleine Störung von $x \in \mathcal{D}(F)$ auf $F(x)$ aus?

Def. [Absolute Kondition(szahl)]
Für eine Abbildung $F : \mathcal{D}(F) \subset X \to Y$ zwischen normierten Vektorräumen X und Y ist die absolute Kondition(szahl) von $F(x)$ für $x \in \mathcal{D}(F)$ gegeben durch

$$\mathrm{cond}_{\mathrm{abs}}(F(x)) = \lim_{\delta\to 0}\sup\left\{\frac{\|F(x+\Delta x)-F(x)\|_Y}{\|\Delta x\|_X} : 0 < \|\Delta x\|_X < \delta,\ x+\Delta x \in \mathcal{D}(F)\right\}.$$

▷ cond_{abs} abhängig von $\|\cdot\|_X$, $\|\cdot\|_Y$!

→ misst Verstärkung kleiner absoluter Datenfehler

Differenzielle Konditionsanalyse: F 2x stetig diff.-bar

$X = \mathbb{R}^m, Y = \mathbb{R}^n$: $F(x+\Delta x) = F(x) + DF(x)\Delta x + O(\|\Delta x\|^2)$ [Mehrdim. Taylorformel]

↑ Jacobimatrix $\left(\frac{\partial F_i}{\partial x_j}\right)_{\substack{i=1,\dots n \\ j=1,\dots,m}}$

$\triangleright$ $\text{cond}_{abs}(F(x)) = \lim_{\|\Delta x\| \to 0} \frac{\|F(x)+DF(x)\Delta x - F(x)\| + O(\|\Delta x\|^2)}{\|\Delta x\|} \leq \|DF(x)\|$

Def. 2.19 [Relative Kondition(szahl)]

Für eine Abbildung $F : \mathcal{D}(F) \subset X \to Y$ zwischen normierten Vektorräumen X und Y ist die relative Kondition(szahl) von $F(x)$ für $x \in \mathcal{D}(F)$ mit $F(x) \neq 0$ gegeben durch

$$\text{cond}_{\text{rel}}(F(x)) = \lim_{\delta \to 0} \sup\left\{ \frac{\|F(x+\Delta x) - F(x)\|_Y}{\|F(x)\|_Y} \frac{\|x\|_X}{\|\Delta x\|_X} : 0 < \|\Delta x\|_X < \delta,\ x + \Delta x \in \mathcal{D}(F) \right\}$$

→ misst Verstärkung kleiner relativer Datenfehler

Falls F 2x stetig diff.-bar:

$$\text{cond}_{rel}(F(x)) = \lim_{\Delta x \to 0} \frac{\|DF(x)\Delta x\|_Y + O(\|\Delta x\|^2)}{\|\Delta x\|_X} \cdot \frac{\|x\|_X}{\|F(x)\|_Y} \leq \|DF(x)\| \frac{\|x\|_X}{\|F(x)\|_Y}$$

Lemma. [Relative Konditionszahl differenzierbarer Funktionen]

Ist $F : \mathcal{D}(F) \subset X \to Y$ stetig differenzierbar in einem inneren Punkt $x \in \mathcal{D}(F)$ mit $F(x) \neq 0$, so gilt

$$\text{cond}_{\text{rel}}(F(x)) = \frac{\|DF(x)\|_{\mathcal{L}(X,Y)}}{\|F(x)\|_Y} \|x\|_X .$$

Bsp: Rel. Kondition der Subtraktion: $F(a,b) = a-b$, $X = \mathbb{R}^2$, $Y = \mathbb{R}$, Eukl. Norm

$$DF(a,b) = \left(\frac{\partial F}{\partial a}(a,b), \frac{\partial F}{\partial b}(a,b)\right) = (1,-1) \Rightarrow \|DF(a,b)\|_2 = \|(1,-1)^T\|_2 = \sqrt{2}$$

$$\text{cond}_{rel}(F(a,b)) = \sqrt{2}\,\frac{\sqrt{a^2+b^2}}{|a-b|} \longrightarrow \infty \quad \text{für } b \to a \quad \text{vgl. Auslöschung, Sect. 1.5}$$

Kondition – Vorwärtsfehler – Rückwärtsfehler

$$\frac{\|\overbrace{F(x+\Delta x)}^{=\tilde{F}(x)} - F(x)\|}{\|F(x)\|} \overset{\text{für } \|\Delta x\| \approx 0}{\lesssim} \text{cond}_{rel}(F(x))\,\frac{\|\Delta x\|}{\|x\|}$$

↳ relativer Vorwärtsfehler unter Rundungsfehlereinfluss

↳ relativer Rückwärtsfehler

↳ $\tilde{F}(x) = F(x+\Delta x)$

↳ durch Rundungsfehler gestörte Abbildung, realisiert durch Algorithmus

▷ Falls $\text{cond}_{rel}(F(x)) \gg 1$ ⇒ winzige Rückwärtsfehler trotz riesiger Vorwärtsfehler

↳ numerisch gutes Ergebnis

"numerisch harmlos"

2.4.2. Kondition einer Matrix

Bsp: Kondition Matrix × Vektor $F(\underline{x}) = A\underline{x}$, $A \in \mathbb{R}^{n\times n}$ regulär, $X = \mathbb{R}^n$, $Y = \mathbb{R}^n$

$$DF(x) = A \overset{[\text{Def 2.19}]}{\Rightarrow} \text{cond}_{rel}(F(x)) = \|A\|\frac{\|x\|}{\|Ax\|} = \|A\|\frac{\|A^{-1}Ax\|}{\|Ax\|} \leq \|A\|\|A^{-1}\|, \quad [x \neq 0]$$

$\|\cdot\| \triangleq$ Vektor- oder assoz. Matrixnorm

Bsp: Matrixinversion: $F(X) = X^{-1}$, $X \in GL(n) := \{M \in \mathbb{R}^{n\times n}, M \text{ invertierbar}\} \subset \mathbb{R}^{n\times n}$ mit $\|\cdot\|$

Hilfsmittel: Neumannsche Reihe ("geometrische Reihe für Matrizen"): $M \in \mathbb{R}^{n\times n}$: $\|M\| < 1 \Rightarrow (I-M)^{-1} = \sum_{j=0}^{\infty} M^j$

$$(X+\Delta X)^{-1} - X^{-1} = X^{-1}\left[(I-\Delta X\cdot X^{-1})^{-1} - I\right] \overset{N.R.}{=} X^{-1}\sum_{j=1}^{\infty}(-\Delta X\cdot X^{-1})^j \quad \text{, falls } \|\Delta X\cdot X^{-1}\| < 1$$

$$= -X^{-1}\cdot\Delta X\cdot X^{-1} + O(\|\Delta X\|^2) \qquad (*)$$

$$\text{cond}_{rel}(F(X)) = \lim_{\Delta X\to 0} \frac{\|-X^{-1}\cdot\Delta X\cdot X^{-1}\| + O(\|\Delta X\|^2)}{\|X^{-1}\|}\cdot\frac{\|X\|}{\|\Delta X\|} \leq \|X^{-1}\|\,\|X\|$$

$\|\cdot\|$ submultiplikativ

Nichtasymptotische Analyse:

Proposition 2.21. [Relative Kondition der Matrixinversion]
Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ regulär und $\Delta\boldsymbol{A} \in \mathbb{R}^{n\times n}$ Störung von $\boldsymbol{A}$ mit $\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\| < 1$ für eine submultiplikative Matrixnorm $\|\cdot\|$. Dann gilt

$$\underbrace{\frac{\|(\boldsymbol{A}+\Delta\boldsymbol{A})^{-1}-\boldsymbol{A}^{-1}\|}{\|\boldsymbol{A}^{-1}\|}}_{\text{Rel. Fehler in } \boldsymbol{A}^{-1}} \leq \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\underbrace{\frac{\|\Delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}}_{\text{Rel. Fehler in } \boldsymbol{A}} \left(1+O(\|\Delta\boldsymbol{A}\|)\right).$$

Beweis: (i) N.R. (wie oben): $A+\Delta A$ invertierbar: $A+\Delta A = A\underbrace{(I+A^{-1}\Delta A)}_{\in GL(n) \text{ nach Voraussetzung. [wird gezeigt mit N.R.]}}$

(ii) $\|(A+\Delta A)^{-1}-A^{-1}\| = \|-A^{-1}\cdot\Delta A\,(A+\Delta A)^{-1}\| \leq$
$\leq \|A^{-1}\cdot\Delta A\|\,\|(A+\Delta A)^{-1}-A^{-1}+A^{-1}\| \leq$
$\leq \underbrace{\|A^{-1}\Delta A\|}_{<1}\left(\|(A+\Delta A)^{-1}-A^{-1}\| + \|A^{-1}\|\right)$

$\Rightarrow \underbrace{(1-\|A^{-1}\cdot\Delta A\|)}_{1+O(\|\Delta A\|)}\|(A+\Delta A)^{-1}-A^{-1}\| \leq \|A^{-1}\|^2\|\Delta A\| = \|A^{-1}\|\,(\|A\|\,\|A^{-1}\|)\frac{\|\Delta A\|}{\|A\|}$

□

Def. [Matrixkondition]

Für eine Matrixnorm $\|\cdot\|$ und eine invertierbare Matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ heisst

$$\kappa(\boldsymbol{A}) := \|\boldsymbol{A}\| \left\|\boldsymbol{A}^{-1}\right\|$$

die Kondition der Matrix $\boldsymbol{A}$.

→ Abhängig von der Matrixnorm $\|\cdot\|$

MATLAB: cond(A)

2.4.3 Kondition der Lösung von LGS

$X = GL(n)\times\mathbb{R}^n \subset \mathbb{R}^{n\times n}\times\mathbb{R}^n$, $Y=\mathbb{R}^n$: $F(A,b) = A^{-1}b$

Sensitivität bzgl. Störung in b : $\text{cond}_{rel}(F(\cdot,b)) = \|A^{-1}\|\|A\| = \kappa(A)$

Sensitivität bzgl. Störung in A (asymptotisch): $\frac{\|(A+\Delta A)^{-1}b - A^{-1}b\|}{\|A^{-1}b\|}\frac{\|A\|}{\|\Delta A\|} = \frac{\|A^{-1}\cdot\Delta A\cdot A^{-1}b\| + O(\|\Delta A\|_2^2)}{\|A^{-1}b\|}\frac{\|A\|}{\|\Delta A\|}$

$= \|A^{-1}\|\|A\| + O(\|\Delta A\| = \kappa(A) + O(\|\Delta A\|)$

Nichtasymptotische Analyse:

Satz 2.22 [Sensitivität für lineare Gleichungssysteme]

Sei $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ invertierbar und sei $\Delta\boldsymbol{A} \in \mathbb{C}^{n\times n}$ so dass

$$\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\| < 1$$

für eine submultiplikative Matrixnorm $\|\cdot\|$. Dann ist

$$(\boldsymbol{A}+\Delta\boldsymbol{A})\hat{\underline{x}} = \underline{b} + \Delta\underline{b}$$

eindeutig lösbar und es gilt mit $\underline{x} = \boldsymbol{A}^{-1}\underline{b}$

$$\frac{\|\underline{x}-\hat{\underline{x}}\|}{\|\underline{x}\|} \le \frac{\|\boldsymbol{A}^{-1}\|\|\boldsymbol{A}\|}{1-\|\boldsymbol{A}^{-1}\Delta\boldsymbol{A}\|}\left(\frac{\|\Delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}+\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\right).$$

Beweis: $\left.\begin{array}{l} A\underline{x} = \underline{b} \\ (A+\Delta A)\hat{\underline{x}} = \underline{b}+\Delta\underline{b}\end{array}\right\} \Rightarrow$

$\underline{x}-\hat{\underline{x}} = (A+\Delta A)^{-1}(\Delta A\,\underline{x} - \Delta\underline{b})$

Mit Störungslemma: $\|(A+\Delta A)^{-1}\| \le \|A^{-1}\|\frac{1}{1-\|A^{-1}\Delta A\|}$

$\|\underline{x}-\hat{\underline{x}}\| \le \|A^{-1}\|\frac{1}{1-\|A^{-1}\Delta A\|}(\|\Delta A\|\|\underline{x}\| + \|\Delta\underline{b}\|)$

$\Rightarrow \frac{\|x-\hat{x}\|}{\|x\|} = \frac{\|A^{-1}\|\|A\|}{1-\|A^{-1}\Delta A\|}\left(\frac{\|\Delta A\|}{\|A\|}+\frac{\|\Delta b\|}{\|A\|\|x\|}\right)$

□

Hilfsmittel :

Lemma [Störungslemma für lineare Abbildungen im $\mathbb{C}^{n\times n}$]

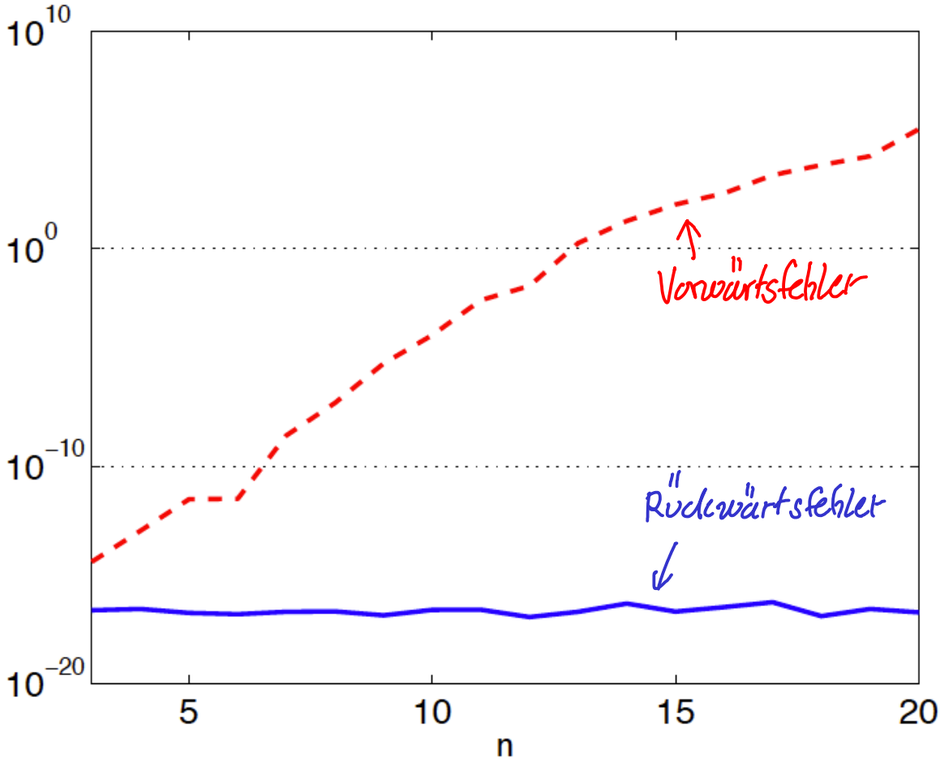
$$\boldsymbol{B}\in\mathbb{C}^{n\times n},\ \|\boldsymbol{B}\|<1 \ \Rightarrow\ \boldsymbol{I}+\boldsymbol{B} \text{ regulär} \ ,\ \|(\boldsymbol{I}+\boldsymbol{B})^{-1}\|\le\frac{1}{1-\|\boldsymbol{B}\|}\,.$$

Beweis: (i) N.R.: $(I+B)^{-1}=\sum_{j=0}^{\infty}(-B)^j$, da $\|B\|<1$ $\Rightarrow$ $\|(I+B)^{-1}\|\overset{\Delta\text{-ungl.}}{\le}\sum_{j=0}^{\infty}\|B^j\|\overset{\text{submult.}}{\le}\sum_{j=0}^{\infty}\|B\|^j=\frac{1}{1-\|B\|}$

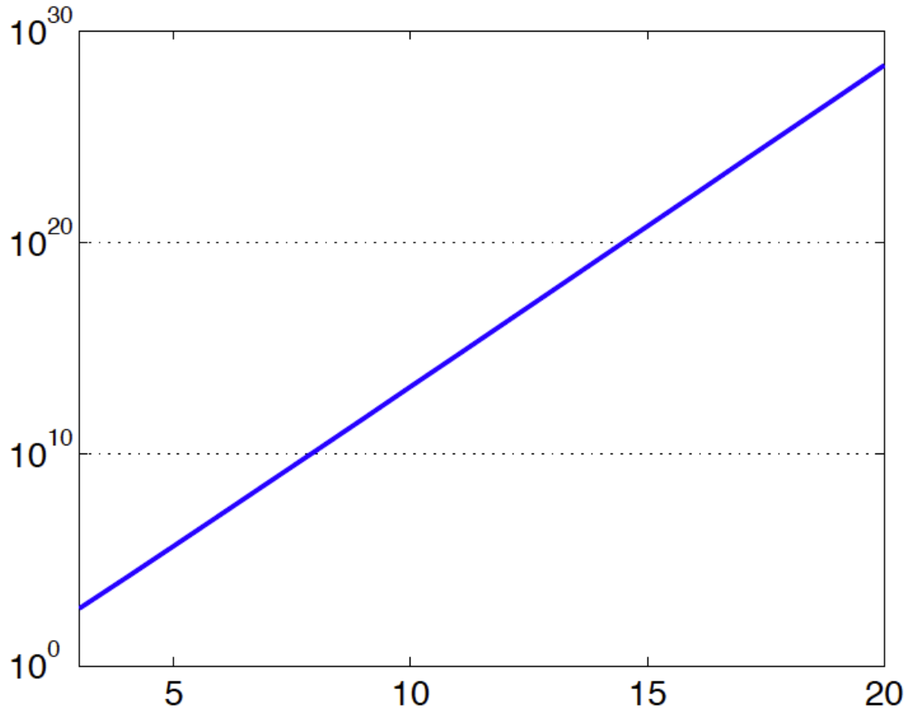
(ii) $v\in\mathbb{C}^n$: $\|(I+B)v\|\ge\|v\|-\|Bv\|\ge(1-\|B\|)\|v\|$

$\|v\|=\|(I+B)(I+B)^{-1}v\|\overset{\downarrow}{\ge}\underbrace{(1-\|B\|)}_{>0}\|(I+B)^{-1}v\|$ $\Rightarrow$ $\frac{\|(I+B)^{-1}v\|}{\|v\|}\le\frac{1}{1-\|B\|}$ [$v\neq0$] □

Zum Beispiel Hilbertmatrix :



$\sim$



2.5. Gausselimination mit Pivotsuche

```
1     % Instability of Gaussian
2     % elimination w/o pivoting
3 -   A = [2^(-55), 1; 1 , 1];
4 -   b = [1;2];
5 -   x1 = A\b;
6 -   x2 = GEnopivot(A,b);
7 -   [L,U] = lufak(A);
8 -   z = L\b; x3 = U\z;
9 -   format short; disp([x1,x2,x3]);
```

$\triangleleft \quad Ax = \underline{b}, \quad A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}, \quad \underline{b} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$\underline{x} = \begin{pmatrix} \frac{1}{1-\varepsilon} \\ \frac{1-2\varepsilon}{1-\varepsilon} \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \hat{\underline{x}}$ für $\varepsilon \ll 1$

$\begin{pmatrix} 1 \\ 2 \end{pmatrix} - A \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -\varepsilon \\ 0 \end{pmatrix} \hat{=}$ Residuum $\underline{r}$

$\frac{\|\underline{r}\|}{\|\underline{x}\|} \approx \varepsilon \rightarrow$ numerisch stabile Lösung, Satz 2.17

```
1     function [L,U] = lufak(A)
2     % Algorithm of Crout: LU-factorization of A
3 -   n = size(A,1); if (size(A,2) ~= n), error('n ~= m'); end
4 -   L = eye(n); U = zeros(n,n);
5 -   for k=1:n
6       % Compute row of U
7 -     for j=k:n, U(k,j) = A(k,j) - L(k,1:k-1)*U(1:k-1,j); end
8       % Compute column of L
9 -     for i=k+1:n, L(i,k) = (A(i,k) - L(i,1:k-1)*U(1:k-1,k)) /U(k,k); end
10 -  end
11
```

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

$= L \cdot U$

$\tilde{L} \qquad \tilde{U} \approx \begin{pmatrix} \varepsilon & 1 \\ 0 & -\varepsilon^{-1} \end{pmatrix}$

$\tilde{L}\tilde{U}\hat{\underline{x}} \approx \underline{b} \Rightarrow \hat{\underline{x}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow$ numerisch instabil

$|\tilde{L}||\tilde{U}| = \begin{pmatrix} \varepsilon & 1 \\ 1 & 2\varepsilon^{-1} \end{pmatrix} \gg |A|$, vgl. 2.14 $\Big\} \rightarrow$ riesiger Rückwärtsfehler

$A \approx A_0 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \Rightarrow$ keine zulässige Eingabe für 'lufak'

Faustregel: Numerische Instabilität schlägt zu bei Eingabedaten nahe bei unzulässigen Eingabedaten

⇒ Idee: Mache 'kritisch' A_0-tauglich → (LA) Zeilenvertauschung

$$A \rightsquigarrow A_P = \begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ \varepsilon & 1 \end{pmatrix}}_{\tilde{L}} \begin{pmatrix} 1 & 1 \\ 0 & 1-\varepsilon \end{pmatrix} \Rightarrow \tilde{U} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$\tilde{L}\tilde{U}\hat{\underline{x}} = \underline{b} \Rightarrow \hat{\underline{x}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

"Stabilisierte Gauss-Elimination"

```
1   function x = GELApivot(A,b)
2   % Raw Gaussian elimination with "theoretical" pivoting for
3   % linear system Ax=b: "Linear algebra version", UNSTABLE!
4   [m,n] = size(A); A = [A,b];
5   % Forward elimination
6   for i=1:n-1
7     l = find(A(i:n,i) ~= 0); % search nonzero pivot
8     A([i,l],:) = A([l,i],:); % swap rows → Zeilenvertauschung
9     for j=i+1:n
10      fac = A(j,i)/A(i,i);
11      for k=i+1:n+1
12        A(j,k) = A(j,k) - fac*A(i,k);
13      end
14    end
15  end
16  % Backward substitution
17  x = [zeros(n-1,1);A(n,n+1)/A(n,n)];
18  for i=n-1:-1:1
19    x(i) = A(i,n+1);
20    for k=i+1:n
21      x(i) = x(i) - A(i,k)*x(k);
22    end
23    x(i) = x(i)/A(i,i);
24  end
```

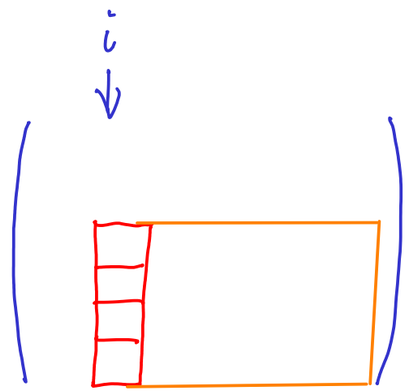
← Numerisch unzulässig !

Numerisch:

Abfrage auf relative Grösse $\approx \upsilon(\mathbb{F})$

"Gauß-Elimination mit Spaltenpivotsuche"

```
1  function x = GEpivot(A,b)
2  % Raw Gaussian elimination without pivoting for
3  % linear system Ax=b: "Linear algebra version", UNSTABLE!
4  [m,n] = size(A); A = [A,b];
5  % Forward elimination
6  for i=1:n-1
7    % Column pivoting (Spaltenpivotsuche)
8    rowmax = max(abs(A(i:n,i:n))')';
9    [pvt,l] = max(abs(A(i:n,i))./rowmax);
10   if (abs(A(l,i)) < eps*max(rowmax)), disp('A nearly singular');
11   A([i,l],:) = A([l,i],:); % swap rows
12   for j=i+1:n
13     fac = A(j,i)/A(i,i);
14     for k=i+1:n+1
15       A(j,k) = A(j,k) - fac*A(i,k);
16     end
17   end
18 end
19 % Backward substitution
20 x = [zeros(n-1,1);A(n,n+1)/A(n,n)];
21 for i=n-1:-1:1
22   x(i) = A(i,n+1);
23   for k=i+1:n
24     x(i) = x(i) - A(i,k)*x(k);
25   end
26   x(i) = x(i)/A(i,i);
27 end
```



$l \in \{i, .., n\}$ so dass $\frac{|a_{li}|}{\max_j |a_{lj}|} \to \max$

↑ relativ größtes Pivotelement

→ wegen Skalierungsinvarianz: gleiche numerische Lösung für $DA\underline{x} = D\underline{b}$ für beliebige reguläre Diagonalmatrix D.

'GEpivot' ist numerisch äquivalent zu

(i) Anwendung aller Zeilenvertauschungen auf A: $A_P = PA$
↳ Permutationsmatrix

(ii) LU-Zerlegung 'lufak': $A_P = LU$

(iii) $x = U \backslash (L \backslash (P\,b))$

Satz 2.31 (Existenz der LU-Zerlegung mit Spaltenpivotsuche)
Zu jeder regulären Matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ gibt es eine $n \times n$ Permutationsmatrix $\boldsymbol{P}$ so, dass $\boldsymbol{PA}$ eine LU-Zerlegung besitzt.

▷ GEpivot immer durchführbar (algebraisch)

Beweis: [Induktion]

A regulär $\Rightarrow$ $A(:, 1:n-1)$ max. Rang $n-1$ $\Rightarrow$ $\exists$ $n-1$ l.u. Zeilen von $A(:, n-1)$

P_{n-1} so, dass diese Zeilen oben stehen Zeilenrang = Spaltenrang

$\Rightarrow$ $A_p = P_{n-1} A$ erfüllt: $A_{p,n-1} = A_p(1:n-1, 1:n-1)$ ist regulär $\overset{\text{I.V.}}{\Rightarrow}$ $P_{n-2} A_{p,n-1}$ hat reguläre Hauptminoren mit geeigneter Permutation P_{n-2}

Thm 2.9: LU-Zerl. ex. $\Leftrightarrow$ Hauptminoren regulär

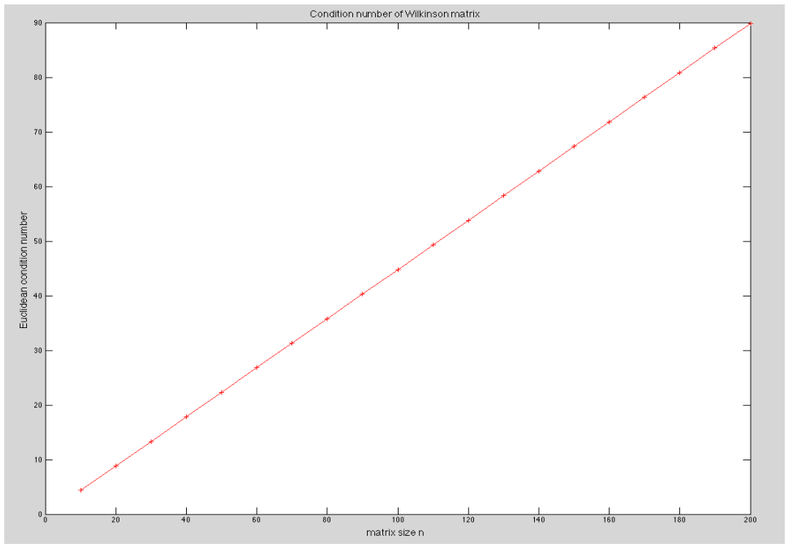
$P = P_{n-2} P_{n-1}$ $\Rightarrow$ $PA = \left(\begin{array}{c|c} P_{n-2} A_p & * \\ \hline * & * \end{array}\right)$ hat reguläre Hauptminoren $\rightarrow$ Thm. 2.9 $\square$

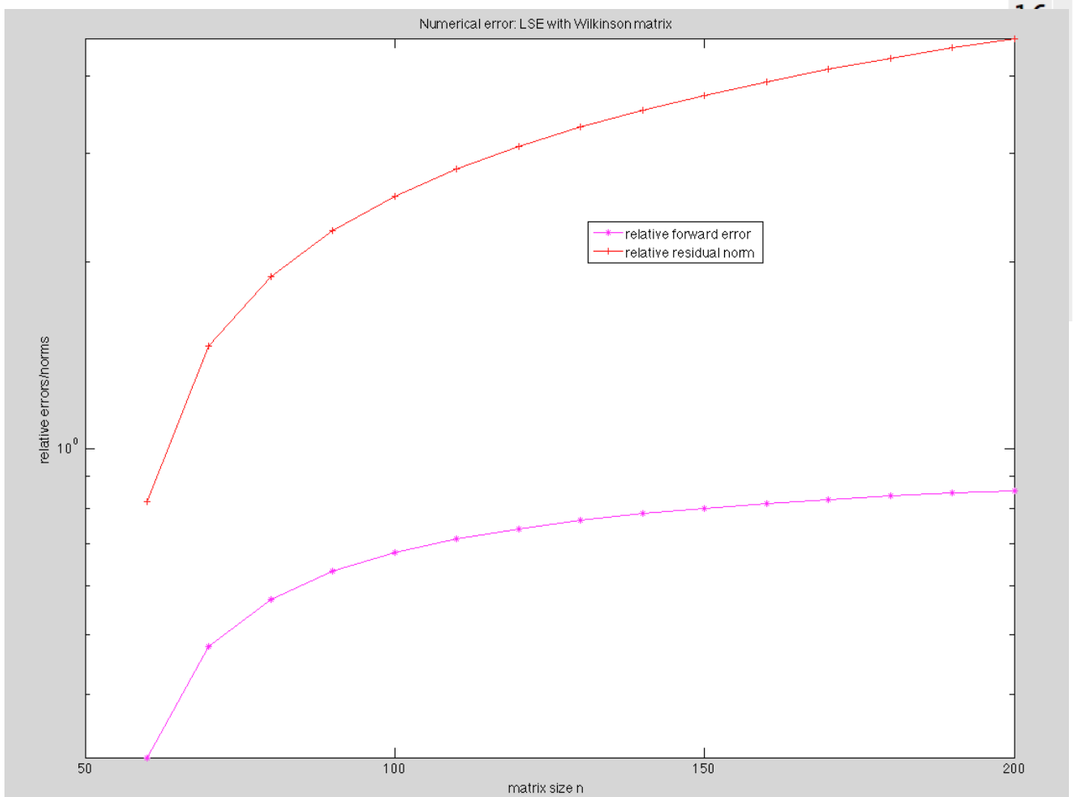
'GEpivot' nicht immer stabil

Wilkinsons Beispiel:

$$a_{ij} = \begin{cases} 1 & \text{, if } i = j \vee j = n \text{ ,} \\ -1 & \text{, if } i > j \text{ ,} \\ 0 & \text{else.} \end{cases} \quad \text{n=10:} \quad \boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}$$

Exakte LU-Zerlegung





$$\blacktriangleright \quad \boldsymbol{A} = \boldsymbol{LU}\,, \quad l_{ij} = \begin{cases} 1 & \text{, if } i = j\,, \\ -1 & \text{, if } i > j\,, \\ 0 & \text{else} \end{cases} \qquad u_{ij} = \begin{cases} 1 & \text{, if } i = j\,, \\ 2^{i-1} & \text{, if } j = n\,, \\ 0 & \text{else.} \end{cases}$$

◁ A gut konditioniert

```
% Wilkinson's counterexample: instability of Gaussian elimination
% despite column pivoting
res = [];
for n=10:10:200
  % Build Wilkinson matrix
  A = [tril(-ones(n,n-1))+2*[eye(n-1);zeros(1,n-1)],ones(n,1)];
  % imposed solution
  x = ((-1).^(1:n))'; b = A*x;
  % Numerical solution
  xh = A\b;
  % Residual
  r = b-A*xh;
  % Relative forward error
  relerr = norm(xh-x)/norm(x);
  % Relative norm of residual
  relresnorm = norm(r)/norm(xh);
  res = [res; n relerr relresnorm];
end
semilogy(res(:,1),res(:,2),'m-*',res(:,1),res(:,3),'r-+');
set(gca,'fontsize',14);
title('Numerical error: LSE with Wilkinson matrix');
xlabel('matrix size n','fontsize',14);
ylabel('relative errors/norms','fontsize',14);
legend('relative forward error','relative residual norm',...
       'location','best');
```

→ G.E. mit Spaltenpivotsuche

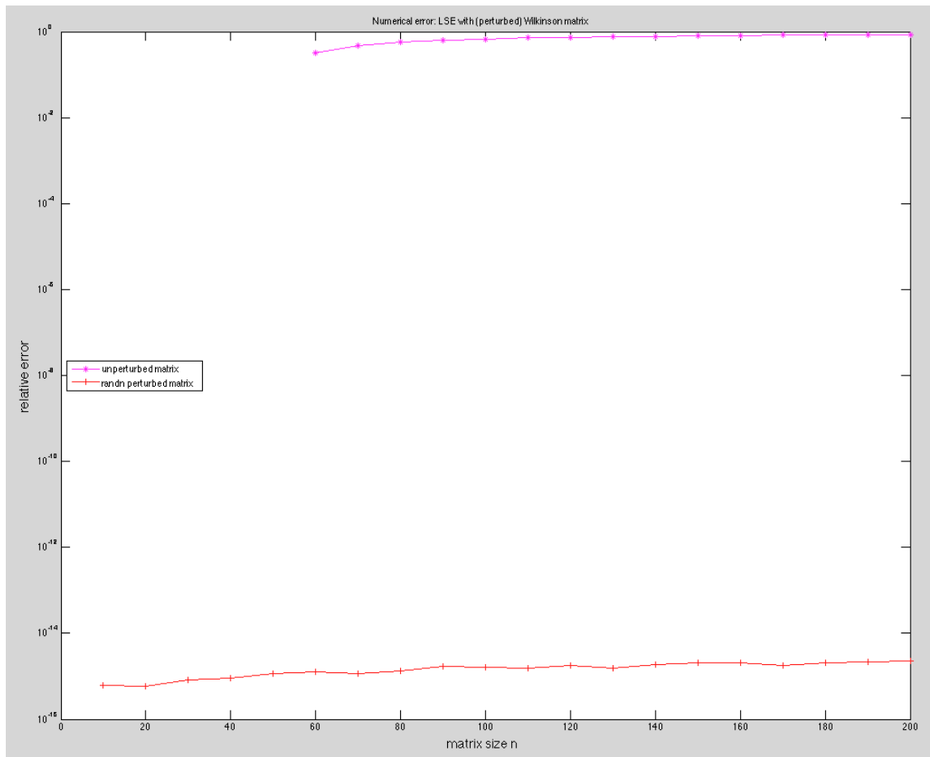
→ Vorwärtsfehler

→ Rückwärtsfehler

◁ Rückwärtsfehler / Vorwärtsfehler ≈ 1 (relativ)

'GEpivot' funktioniert in der Praxis immer (= numerisch stabil)

```
% Curing Wilkinson's counterexample by random perturbation
% Theory: Spielman and Teng
res = [];
for n=10:10:200
  % Build Wilkinson matrix
  A = [tril(-ones(n,n-1))+2*[eye(n-1);zeros(1,n-1)],ones(n,1)];
  % imposed solution
  x = ((-1).^(1:n))';
  relerr = norm(A\(A*x)-x)/norm(x);
  % Randomly perturbed Wilkinson matrix by matrix with iid
  % $N(0,{\mathrm{eps}})$ distributed entries
  Ap = A + eps*randn(size(A));
  relerrp = norm(Ap\(A*x)-x)/norm(x);
  res = [res; n relerr relerrp];
end
semilogy(res(:,1),res(:,2),'m-*',res(:,1),res(:,3),'r-+');
title('Numerical error: LSE with (perturbed) Wilkinson matrix');
xlabel('matrix size n','fontsize',14);
ylabel('relative error','fontsize',14);
legend('unperturbed matrix','randn perturbed matrix','location','best');
```



Bem. (Algorithmische Relevanz der LU-Zerlegung)

ungeschickt !

```
% Setting: N ≫ 1, large matrix A ∈ ℝ^{n,n}
for j=1:N
    x = A\b;   → G.E.
    b = some_function(x);
end
```

Rechenaufwand $O(Nn^3)$

schlau !

```
% Setting: N ≫ 1, large matrix A
[L,U] = lu(A);
for j=1:N
    x = U\(L\b);
    b = some_function(x);
end
```

Rechenaufwand $O(n^3 + Nn^2)$

MATLAB

[L, U] = lu(A)

↳ permutierte untere Δ-Matrix

[L, U, P] = lu(A)

↳ Permutationsmatrix

Bei mehrfachen rechten Seiten $\underline{b}_1, \ldots, \underline{b}_m$: MATLAB : `X = A\[b1, b2, ..., bm];`

2.7. Cholesky-Zerlegung für SPD Matrizen

Def. 2.34. (Symmetrisch positiv definite Matrizen)
Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst

1. **positiv definit**, falls $\underline{x}^{\mathsf{T}}\boldsymbol{A}\underline{x} > 0 \qquad \forall \underline{x} \in \mathbb{R}^n \backslash \{\underline{0}\}$;
2. **positiv semi-definit**, falls $\underline{x}^{\mathsf{T}}\boldsymbol{A}\underline{x} \geq 0 \qquad \forall \underline{x} \in \mathbb{R}^n$.

Eine symmetrische, positiv definite Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst kurz **SPD**.

→ $\begin{cases} \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \\ (\underline{x}, \underline{y}) \to \underline{x}^T A \underline{y} \end{cases}$ ist ein Skalarprodukt, wenn $A = A^T$

Satz 2.35 (Eigenschaften von SPD Matrizen)
Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ SPD. Dann gilt:

1. $\boldsymbol{A}$ ist regulär (invertierbar);
2. $a_{ii} > 0$ für alle $i \in \{1, \ldots, n\}$;
3. $|a_{ij}| < \frac{1}{2}\left(a_{ii} + a_{jj}\right)$ für $i \neq j$ und damit $\max_{ij} |a_{ij}| = \max_i a_{ii}$;
4. ist $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ invertierbar, so ist $\boldsymbol{X}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{X}$ wieder SPD;
5. ist $\boldsymbol{A}$ eine Blockdiagonalmatrix $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{A}_1, \boldsymbol{A}_2)$, so sind sowohl $\boldsymbol{A}_1$ als auch $\boldsymbol{A}_2$ SPD.

zu 3.

$$\begin{pmatrix} * & * \end{pmatrix}\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}\begin{pmatrix} * \\ * \end{pmatrix}$$

$$\underline{x}^T (X^T A X)\underline{x} = (X\underline{x})^T A (X\underline{x})$$

6. Alle EW von A sind positiv. $\Rightarrow$ $\det(A) > 0$

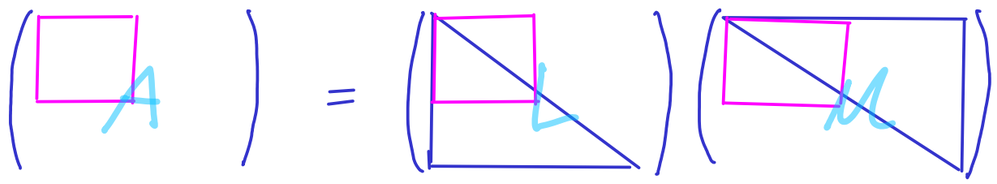
Satz (LU-Zerlegung von SPD Matrizen)
Jede SPD Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ besitzt eine LU-Zerlegung $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$, wobei $\boldsymbol{U}$ positive Diagonalelemente hat.

Beweis:
(i)

$$\begin{pmatrix} * \\ \vdots \\ * \\ 0 \end{pmatrix}^T \begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix}\begin{pmatrix} * \\ \vdots \\ * \\ 0 \end{pmatrix}$$

▷ Hauptminoren {SPD ⇒ regulär} : Satz 2.9.

(ii) $A = LM \Rightarrow 0 < \det(A) = \underbrace{\det(L)}_{=1}\det(M) = \prod_{i=1}^{n} (M)_{ii}$



: i-te Hauptminoren von L/M liefern LU-Zerlegung des i-ten Hauptminors von A ⇒ Induktion □

Satz 2.36 (Cholesky-Zerlegung von SPD Matrizen)
Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ SPD. Dann existiert eine eindeutige obere Dreiecksmatrix $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ mit positiven Diagonalelementen, so dass $\boldsymbol{A} = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}$.

Beweis: $A = L\tilde{M} = A^T = \tilde{M}^T L^T = \hat{M}^T D L^T = \hat{M}^T (D L^T)$ ≙ wieder LU-Zerlegung

[$\tilde{M} = D\hat{M}$, $D = \mathrm{diag}(\underbrace{M_{11}}_{>0}, \dots, \underbrace{M_{nn}}_{>0}) \Rightarrow (\hat{M})_{ii} = 1$

Eindeutigkeit LU-Zerl. ⇒ $L = \hat{M}^T \Rightarrow A = \hat{M}^T D \hat{M} = \underbrace{\hat{M}^T \sqrt{D}}_{M^T} \underbrace{\sqrt{D}\hat{M}}_{M}$ □

Satz 2.39 (Gausselimination ohne Pivotsuche stabil für SPD Matrizen)
Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ SPD und $\widehat{\underline{x}}$ die mittels Cholesky-Zerlegung und Vorwärts-/Rückwartseinsetzen berechnete Lösung von $\boldsymbol{A}\underline{x} = \underline{b}$. Dann gilt

$$(\boldsymbol{A} + \triangle\boldsymbol{A})\widehat{\underline{x}} = \underline{b}, \quad \|\triangle\boldsymbol{A}\|_2 \le 4n^2(3n+1)u\|\boldsymbol{A}\|_2 .$$

↑ ~ Rechenaufwand

Beweisskizze:
(i) Wie im Beweis zu Thm 2.19: $|(\triangle A)_{ij}| \le 4(3n+1)u(\mathbb{F})\,|M|^T|M|$ ↑ Cholesky-Faktor von A

(ii) $\Rightarrow \quad \|\Delta A\|_2 \leq 4n(3n+1)\,u(\mathbb{F})\,\| |M|^T|M| \|_2 \leq 4n^2(3n+1)\,u(\mathbb{F})\,\|A\|_2$

$$\| |M^T||M| \|_2 = \sup_{x\neq 0} \frac{x^T|M|^T|M|x}{\|x\|_2^2} = \| |M| \|_2^2 \leq \| |M| \|_F^2 = \|M\|_F^2 \leq n\|M\|_2^2 = n\|M^TM\|_2 = n\|A\|_2$$

$$\left[\; M = M^T \Rightarrow \|M\|_2 = \sup_{x\neq 0} \frac{x^TMx}{\|x\|_2^2} \quad \text{mit Spektralsatz} \;\right]$$

$$\left[\; \|M\|_F^2 = \sum_{ij} (M)_{ij}^2 = \sum_j \sigma_j(M)^2 \leq n\,\sigma_{max}(M)^2 = n\|M\|_2^2 \;\right]$$

$\|M\|_F^2$: Frobeniusnorm (siehe Abschnitt 0.7)

$\sigma_j(M)$: Singulärwert (siehe Abschnitt 0.10)

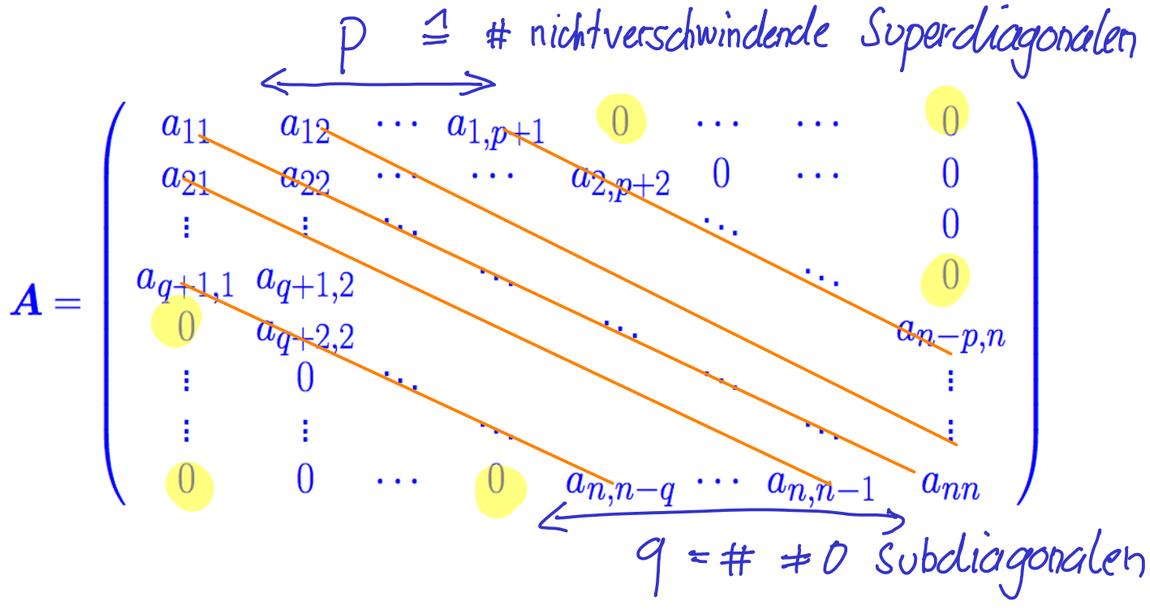
$\sigma_{max}(M)^2$: Satz: $\|M\|_2 = \max_j \sigma_j(M)$ □

2.8. LU-Zerlegung

Def. 2.40 (Bandmatrix)
Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst **Bandmatrix** mit **Bandbreite** $p+q+1$, falls es p, $q \in \mathbb{N}\cup\{0\}$ gibt, so dass

$$a_{ij} = 0 \quad \text{für} \quad j > i+p \quad \text{oder} \quad i > j+q\,.$$

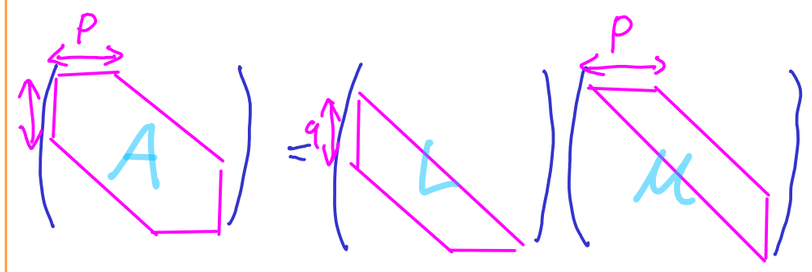
Die Zahl p heisst die obere Bandbreite und q die untere Bandbreite.



Satz 2.41 (Band-LU-Zerlegung)

Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ eine Bandmatrix mit oberer Bandbreite p und unterer Bandbreite q und existiere die LU-Zerlegung $\boldsymbol{LU} = \boldsymbol{A}$. Dann haben $\boldsymbol{L}$, $\boldsymbol{U}$ Bandstruktur, und es gilt:

$$l_{ij} = 0, \text{ falls } j > i \text{ oder } j < i - q\,,$$
$$u_{ij} = 0, \text{ falls } j < i \text{ oder } j > i + p\,.$$



Beweis: (Induktion, Blockmatrixperspektive, vgl. Beweis zu Satz 2.9)

$$\left(\begin{array}{c|c} \boldsymbol{A}_{n-1} & \underline{b} \\ \hline \underline{a}^\top & \alpha \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{L}_{n-1} & \underline{0} \\ \hline \underline{x}^\top & 1 \end{array}\right) \left(\begin{array}{c|c} \boldsymbol{U}_{n-1} & \underline{y} \\ \hline \underline{0}^\top & \xi \end{array}\right) \Rightarrow L_{n-1}\underline{y} = \underline{b} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ * \\ \vdots \\ * \end{pmatrix} \updownarrow p \Rightarrow \underline{y} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ * \\ * \end{pmatrix} \updownarrow p \Rightarrow U \text{ Bandmatrix, obere Bandbreite } p$$

$A = L U$

L_{n-1}: untere Δ-Matrix

[U_{n-1} Bandmatrix nach Induktionsvor.]

Analog: $U_{n-1}^\top \underline{x} = \underline{a} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ * \\ \vdots \\ * \end{pmatrix} \updownarrow q$ .....

$U_{n-1}^\top$: untere Δ-Matrix

```
1   function [L,U] = luband(A,p,q)
2   % Algorithm of Crout: LU-factorization of band matrix A
3   % p = upper bandwidth, q = lower bandwidth
4   n = size(A,1); if (size(A,2) ~= n), error('n ~= m'); end
5   L = eye(n); U = zeros(n,n);
6   for k=1:n
7     % Compute row of U (1+q nonzero entries only)
8     for j=k:min(n,k+q)
9         U(k,j) = A(k,j) - L(k,1:k-1)*U(1:k-1,j); end
10    % Compute column of L (p nonzero off-diagonal entries only)
11    for i=k+1:min(n,k+p)
12        L(i,k) = (A(i,k) - L(i,1:k-1)*U(1:k-1,k)) /U(k,k); end
13  end
```

Δ Verkürzte Schleifen und Skalarprodukte

Rechenaufwand: #flops $= O(npq)$

} Ineffizient implementiert

Hier: Keine Pivotsuche $\Rightarrow$ i.A. instabil (aber bedenkenlos durchführbar für A SPD, Satz 2.39)

↳ kann Bandstruktur zerstören