4. Eliminationsverfahren zur Lösung linearer Gleichungssysteme

4.1 Dreieckssysteme (auch: gestaffelte Systeme)

Sei $A \in \mathbb{R}^{n \times n}$ (eigentlich geht auch alles für $\mathbb{K} = \mathbb{C}$) von oberer Dreiecksgestalt. Mithin ist zu lösen:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots &+ a_{1n}x_n = b_1 \\ a_{22}x_2 + \ldots &+ a_{2n}x_n = b_2 \\ &\vdots \\ a_{nn}x_n &= b_n \end{aligned}$$

Dieses System ist regulär genau dann wenn alle $a_{ii} \neq 0$, $i = 1, \ldots, n$.

eine Möglichkeit: (Übung), noch einfacher: Determinante nach unterster Zeile entwickeln.

$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$. A regulär $\Leftrightarrow$ A_{11} und A_{12} regulär

Man rechne $A^{-1} = \begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12}A_{22}^{-1} \\ 0 & A_{22}^{-1} \end{bmatrix}$

Lösen mittels rückwärts einsetzen:

$$x_n = b_n / a_n;$$

$$x_i = \left(b_i - \sum_{k=i+1}^{n} a_{ik} x_k \right) / a_{ii};$$

Die Anzahl der benötigten Rechenoperationen beträgt:

$$N_{\text{Dreieck}}(n) = \sum_{i=0}^{n-1} (\underbrace{2i}_{\Sigma} + \underbrace{1}_{\text{teilen durch } a_{ii}}) = n^2$$

(Alle Operationen dauern gleich lange).

Analog: Untere Dreiecksgestalt und vorwärts einsetzen.

4.2 Gauß Elimination

Sei nun $A \in \mathbb{R}^{n \times n}$ regulär gegeben.

Das Gleichungssystem $Ax = b$ soll durch äquivalente Umformungen auf (obere) Dreiecksgestalt gebracht werden.

Dazu benutzt man:

(i) Vertauschen zweier Gleichungen,

(ii) Addition eines Vielfachen einer Gleichung zu einer anderen.

Wir schreiben

$$[A, b] = \left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ a_{12} & \cdots & a_{2n} & \\ \vdots & & & \\ a_{n1} & \cdots & a_{nn} & b_n \end{array}\right] \in \mathbb{R}^{n \times n+1}.$$

Das Gaußsche Eliminationsverfahren erzeugt eine Folge von Matrizen

$$[A^{(0)}, b^{(0)}] = [A, b], [A^{(1)}, b^{(1)}], \ldots, [A^{(n-1)}, b^{(n-1)}]$$

so dass nach <u>n-1</u> Schritten $A^{(n-1)}$ obere Dreiecksgestalt hat.

<u>Schritt 1 : $[A^{(0)}, b^{(0)}] \to [A^{(1)}, b^{(1)}]$:</u> $(k=1)$

a) $[A^{(0)}, b^{(0)}] \to [\tilde{A}^{(0)}, \tilde{b}^{(0)}]$ (nach Zeilenvertauschung)

Bestimme ein $r \in \{1, \ldots, n\}$ so dass $a_{r1}^{(0)} \neq 0$. Dies ex. wg. <u>Regularität</u>, Vertausche Zeilen 1 und r womit $\tilde{a}_{11}^{(0)} \neq 0$

b) $[\tilde{A}^{(0)}, \tilde{b}^{(0)}] \to [A^{(1)}, b^{(1)}]$

Für $i \in \{2, \ldots, n\}$:

- Setze $q_{i1} = \dfrac{\tilde{a}_{i1}^{(0)}}{\tilde{a}_{11}^{(0)}}$; $\tilde{a}_{11}^{(0)}$ heißt <u>Pivotelement</u>.

- Subtrahiere das q_{i1}-fache der Zeile 1 von Zeile i :

$$\forall j \in \{1, \ldots, n\}: a_{ij}^{(1)} = \tilde{a}_{ij}^{(0)} - q_{i1} \tilde{a}_{1j}^{(0)}; \quad b_i^{(1)} = \tilde{b}_i^{(0)} - q_{i1} \tilde{b}_1^{(0)}.$$

nächste Zeile!

wegen $a_{i1}^{(1)} = \tilde{a}_{i1}^{(0)} - \frac{\tilde{a}_{i1}}{\tilde{a}_{11}} \cdot \tilde{a}_{11} = 0$ $(j=1)$ für $i \geq 2$ gilt:

$$[A^{(1)}, b^{(1)}] = \left[\begin{array}{c|ccc|c} \tilde{a}_{11}^{(0)} & \tilde{a}_{12}^{(0)} & \cdots & \tilde{a}_{1n}^{(0)} & \tilde{b}_1^{(0)} \\ \hline 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right].$$

Dies führt man nun rekursiv für die $(n-1) \times n$ Untermatrix fort.

<u>Schritt k : $[A^{(k-1)}, b^{(k-1)}] \to [A^{(k)}, b^{(k)}]$</u> $\quad 1 \leq k < n$

Ausgangssituation:

$$[A^{(k-1)}, b^{(k-1)}] = \begin{array}{r} \\ k-1 \\ \\ n-k+1 \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \overbrace{\qquad\quad}^{k-1} & \overbrace{\qquad\qquad\quad}^{n-k+1} & \sim \end{array} \\ \left[\begin{array}{cc|ccc|c} a_{11}^{(1)} & * & & * & & * \\ 0 & \tilde{a}_{k-1,k-1}^{(k-1)} & & & & \\ \hline & & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\ & 0 & \vdots & & \vdots & \\ & & a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} & b_n^{(k-1)} \end{array}\right] \end{array}$$

a) $[A^{(k-1)}, b^{(k-1)}] \to [\tilde{A}^{(k-1)}, \tilde{b}^{(k-1)}]$

Bestimme $r \in \{k, \dots, n\}$ so dass $a_{rk} \neq 0$. Ex. wg. Regularität

Tausche Zeilen k und r. Somit ist $\tilde{a}_{kk}^{(k-1)} \neq 0$.

b) $[\tilde{A}^{(k-1)}, \tilde{b}^{(k-1)}] \to [A^{(k)}, b^{(k)}]$

Für $i \in \{k+1, \dots, n\}$:

- Setze $q_{ik} = \frac{\tilde{a}_{ik}^{(k-1)}}{\tilde{a}_{kk}^{(k-1)}}$
- Subtrahiere das q_{ik}-fache der Zeile k von Zeile i:

$$\forall j \in \{k, \dots, n\}: \quad a_{ik}^{(k)} = \tilde{a}_{ik}^{(k-1)} - q_{ik}\, \tilde{a}_{kj}^{(k-1)};$$

$$b_i^{(k)} = \tilde{b}_i^{(k-1)} - q_{ik}\, \tilde{b}_k^{(k-1)};$$

Nach $n-1$ Schritten hat $A^{(n-1)}$ obere Dreiecksgestalt

→ Verstreichen

Algorithmus 4.1
Beispiel 4.2 !

Lemma 4.3 Der Aufwand zur Transformation von A auf obere Dreiecksgestalt beträgt

$$N_{Gauß}(n) = \frac{2}{3} n^3 + O(n^2).$$

Beweis:

$$N_{Gauß}(n) = \sum_{k=1}^{n-1} \Big\{ \underbrace{n-k}_{\text{Ber. } q_{ik}} + \underbrace{(n-k)}_{\#\text{Zeilen}} \Big[\underbrace{(n-k)}_{\#\text{Spalten in } A} \cdot \underset{\text{je 1 Mult/Subtr.}}{2} + \underset{b}{2} \Big] \Big\}$$

$$= 2 \sum_{k=1}^{n-1} (n-k)^2 + \sum_{k=1}^{n-1} 3(n-k)$$

$$= \frac{2}{3} n^3 + O(n^2) \qquad \boxtimes$$

~~Formulierung als Algorithmus fehlt!~~

Beispiel 4.2 [Beispiel 10.9 aus Neunstoch Skript kopieren] → Beamer

---

Programmieren Ü

a) Gauß-Elimination in C++

wichtig: A und b sind Referenzargumente und werden überschrieben

b) Rückwärtssubstitution → L ist const-Argument?

c) Laufzeit messen?

Algorithmus 4.1

Im Skript Indizes immer ab 1! 16.11.09

Input: $A \in \mathbb{R}^{n \times n}$ (wird überschrieben)
$b \in \mathbb{R}^n$ (wird überschrieben)
Output $x \in \mathbb{R}^n$

```
for (k=1; k<n; k=k+1) {
    Finde r ∈ {k,...,n} so dass a_rk ≠ 0; sonst Fehler;
    if (r≠k) { //tausche Zeile k mit Zeile r
        for (j=k; j≤n; j=j+1) {
            t = a_kj; a_kj = a_rj; a_rj = t;
        }
        t = b_k; b_k = b_r; b_r = t;
    }
    if

    for (i=k+1; i≤n; i=i+1) {
        q_ik = a_ij / a_kk;
        for (j=k+1; j≤n; j=j+1)
            a_ij = a_ij - q_ik a_kj;
        b_i = b_i - q_ik b_k;
    }
}
```

$b_i = b_i - q_{ik} b_k;$
  end for
end for

Bemerkung 10.8. Elemente von $A^{(k)}$ werden jeweils mit denen von $A^{(k+1)}$ überschrieben. Das ursprüngliche A und b stehen somit *nicht* mehr zur Verfügung.

Der gegebene Algorithmus ist nicht numerisch stabil gegenüber Rundungsfehlern. Dazu nächstes Mal mehr. □

Für den Aufwand erhält man:

$$
\begin{aligned}
F_{\text{Gauß}}(n) &= \sum_{k=1}^{n-1} \left\{ \underbrace{n-k}_{\text{Multiplikatoren } q_{ik}} + (n-k)[2 + 2(n-k)] \right\} \\
&= 2 \sum_{k=1}^{n-1} (n-k)^2 + O(n^2) \\
&= \frac{2}{3} n^3 + O(n^2) \quad .
\end{aligned}
$$

Der oben angegebene naive Algorithmus nutzt den Cache in heutigen Prozessoren für große n nicht gut aus.

Es gibt jedoch cache-optimale Implementierungen, die Tatsache, dass $O(n^3)$ Operationen auf $O(n^2)$ Daten (Speicher für A, b) ausgeführt werden ausnutzen können.

Die gesamte Prozedur zur Lösung von $Ax = b$ besteht somit aus:

(i) Bringe A auf obere Dreiecksgestalt.

(ii) Löse Dreieckssystem durch Rückwärtseinsetzen.

57 **Beispiel 10.9.** Wir geben ein Beispiel zur Gauß-Elimination. Hier sind keine Zeilenvertauschungen notwendig. Das Pivotelement ist jeweils durch einen Kasten gekennzeichnet.

$$
\left[\begin{array}{cccc|c} \boxed{2} & 4 & 6 & 8 & 40 \\ 16 & 33 & 50 & 67 & 330 \\ 4 & 15 & 31 & 44 & 167 \\ 10 & 29 & 63 & 97 & 350 \end{array}\right]
\rightarrow
\left[\begin{array}{cccc|c} 2 & 4 & 6 & 8 & 40 \\ 0 & \boxed{1} & 2 & 3 & 10 \\ 0 & 7 & 19 & 28 & 87 \\ 0 & 9 & 33 & 57 & 150 \end{array}\right]
$$

$$
\rightarrow
\left[\begin{array}{cccc|c} 2 & 4 & 6 & 8 & 40 \\ 0 & 1 & 2 & 3 & 10 \\ 0 & 0 & \boxed{5} & 7 & 17 \\ 0 & 0 & 15 & 30 & 60 \end{array}\right]
\rightarrow
\left[\begin{array}{cccc|c} 2 & 4 & 6 & 8 & 40 \\ 0 & 1 & 2 & 3 & 10 \\ 0 & 0 & 5 & 7 & 17 \\ 0 & 0 & 0 & 9 & 9 \end{array}\right]
$$

auf → pdf

Schließlich liefert Rückwärtseinsetzen:

$x_4 = 9/9 = \boxed{1}$, $\quad x_3 = (17 - 7 \cdot 1)/5 = \boxed{2}$,

$x_2 = (10 - 2 \cdot 2 - 3 \cdot 1)/1 = \boxed{3}$, $\quad x_1 = (40 - 4 \cdot 3 - 6 \cdot 2 - 8 \cdot 1)/2 = \boxed{4}$.

□

4.3 LR-Zerlegung

Vorüberlegung:
Wir wollen zunächst die Gauß-Elimination abstrakter mit Matrixmultiplikationen schreiben.

Für den Schritt (1) definieren wir die sog. Permutationsmatrizen $P_{rs} \in \mathbb{R}^{n\times n}$, $1 \le r,s \le n$, $r \neq s$, mit

$$(P_{rs})_{ij} = \begin{cases} 1 & i \neq r,s \wedge i = j \\ 1 & (i = r \wedge j = s) \vee (i = s \wedge j = r) \\ 0 & \text{sonst.} \end{cases}$$

d.h.

$$P_{rs} = \begin{array}{c} \\ \\ r \\ \\ s \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & r & & s & & \end{array} \\ \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & 1 & & \\ & & & \ddots & & & \\ & & 1 & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix} \end{array}$$

~~($P_{rs} = P_{sr}$ braucht man das)~~

Hilfssatz 4.4 Die Matrizen P_{rs} haben folgende Eigenschaften

i) $\tilde{A} = P_{rs} A$ ist A mit den Zeilen r und s vertauscht.

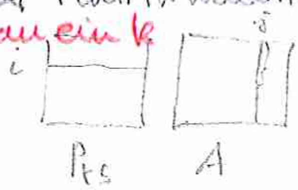
ii) $\tilde{A} = A P_{rs}$ ist A mit den Spalten r und s vertauscht.

iii) $P_{rs} = P_{rs}^T$

iv) $P_{rs}^{-1} = P_{rs}$ $\Rightarrow P_{rs}^{-1} = P_{rs}^T$ „orthogonal"

Beweis: durch Anwenden der Definition der Matrixmultiplikation:

i) $\tilde{a}_{ij} = \sum_{k=1}^{n} (P_{rs})_{ik} a_{kj}$ $(P_{rs})_{ik} \neq 0$ für genau ein k



Sei $i \neq r \wedge i \neq s$ dann ist $\tilde{a}_{ij} \overset{k=i}{=} a_{ij}$

$i = r \rightsquigarrow k = s$ $\tilde{a}_{rj} = a_{sj}$ $\forall j$

$i = s \rightsquigarrow k = r$ $\tilde{a}_{sj} = a_{rj}$ $\forall j$

ii) analog.

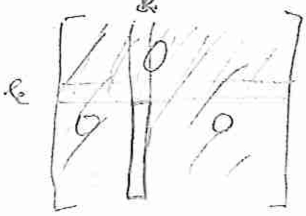
iii) folgt aus Def. von P_{rs}

iv) $P_{rs} P_{rs} = I$ wg i) also $P_{rs}^{-1} = P_{rs}$ ▨

Wir definieren (für Schritt ②) sog. <u>Frobeniusmatrizen</u> welche folgende Struktur haben:

$$G_k \in \mathbb{R}^{n\times n}, \quad 1 \le k < n$$

$$G_k = I + G_k' \quad \text{mit} \quad (G_k')_{ij} = 0 \quad \text{für } (j \neq k) \vee (i \le k),$$



d.h.

$$G_k = \begin{array}{cc} & \begin{matrix} \quad & \quad & k & \quad \end{matrix} \\ k & \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & g_{k+1,k} & \ddots & \\ & & \vdots & & \\ & & g_{n,k} & & 1 \end{pmatrix} \end{array}.$$

<u>Hilfssatz 4.5</u> Die Frobeniusmatrizen haben folgende Eigenschaften:

i) Für $\tilde{A} = G_k A$ gilt

$$\tilde{a}_{ij} = \begin{cases} a_{ij} & i \le k \\ a_{ij} + g_{ik} a_{kj} & i > k \end{cases}$$

(Dies ist gerade der Eliminationsschritt in der Gauß-Elimination wenn $g_{ik} := -q_{ik}$)

ii) $G_k' G_k' = 0$

iii) $G_k^{-1} = (I - G_k')$

iv) $G_1 G_2 \cdots G_k = I + \sum_{j=1}^{k} G_k'$, analog: $G_1^{-1} \cdots G_k^{-1} = I - \sum_{j=1}^{k} G_j'$

v) $\forall\, k < \alpha \le \beta$: $\quad P_{\alpha\beta} G_k = \underbrace{(I + P_{\alpha\beta} G_k')}\, P_{\alpha\beta}$ $\quad I + P_{\alpha\beta} G_k'$ ist eine geänderte Frobeniusmatrix!

Das ist genau die Situation in der GEM.

Beweis: Wird alles mit der Nullstruktur von G begründet:

i) $$\tilde{a}_{ij} = \sum_{s=1}^{n} (G_k)_{is}\, a_{sj} = \begin{cases} a_{ij} \\ \underbrace{g_{ik} a_{kj}}_{s=k} + a_{ij} \end{cases}$$

$i \le k$ $(\to (G_k)_{ii} = 1$ also $s = i$

sonst.

$(G_k)_{ik} = g_{ik}$

$(G_k)_{ii} = 1$

9

29.10.09

ii)

$$G_k' = \begin{pmatrix} 0 & 0 \\ 0 & B_k \end{pmatrix} \begin{array}{l} \}\, k-1 \\ \}\, n-k+1 \end{array}$$

with column widths $k-1$ and $n-k+1$.

$$\begin{pmatrix} 0 & 0 \\ 0 & B_k \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & B_k \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} 0 & 0 \\ g_{k+1,k} & \\ \vdots & 0 \\ g_{n,k} & \end{pmatrix}}_{B_k} \underbrace{\begin{pmatrix} 0 & 0 \\ g_{k+1,k} & \\ \vdots & 0 \\ g_{n,k} & \end{pmatrix}}_{B_k}$$

Es genügt $B_k B_k$ zu betrachten, das ist Null

iii) $\underbrace{(I + G_k')}_{G_k}(I - G_k') = I - G_k' + G_k' - \underbrace{G_k' G_k'}_{=0 \text{ wg ii)}} = I$

iv) Per Induktion.

$k=1$: $G_1 = I + G_1'$ hat die geforderte Gestalt

$k-1 \to k$: Es sei also $G_1 \dots G_{k-1} = I + \sum_{j=1}^{k-1} G_j'$ bereits gezeigt.

$$G_1 \dots G_k = \left(I + \sum_{j=1}^{k-1} G_j'\right)(I + G_k') = I + G_k' + \sum_{j=1}^{k-1} G_j' + \underbrace{\left(\sum_{j=1}^{k-1} G_j'\right) G_k'}_{=0 \text{ ist zu zeigen}}$$

$$\left(\sum_{j=1}^{k-1} G_j'\right) G_k' = \begin{pmatrix} * \; 0 & 0 \\ * & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 \; \begin{matrix} 0 \\ * \\ \vdots \\ * \end{matrix} & 0 \end{pmatrix} = 0$$

(row blocks $k-1$, $n-k+1$; column blocks $k-1$, $n-k+1$)

da einer der beiden Faktoren immer Null ist

G_k, G_k^{-1} unterscheiden sich nur durch Vorzeichen. Beweis hängt nur an Nullstellen

v) $\boxed{k < \alpha \leq \beta}$: $P_{\alpha\beta} G_k = P_{\alpha\beta}(I + G_k') = P_{\alpha\beta} + P_{\alpha\beta} G_k'$

$$= P_{\alpha\beta} + \underbrace{P_{\alpha\beta} G_k'}_{\substack{\text{nur Spalte} \\ k \text{ besetzt}}} \underbrace{P_{\alpha\beta}}_{\substack{\text{tauscht zwei Spalten } \alpha,\beta > k, \\ \text{diese sind Null!}}}$$

$$= (I + P_{\alpha\beta} G_k') P_{\alpha\beta}$$

$$P_{\alpha\beta} G_k' = \begin{pmatrix} & & \\ \alpha \leftrightarrow \beta & * & \\ \alpha,\beta > k & * & \end{pmatrix}$$

(k-te Spalte)

Somit ist $(I + P_{\alpha\beta} G_k')$ wieder Frob. Matrix

Wir definieren das Produkt von Matrizen:

$$\prod_{i=a}^{b} B_i = B_b \cdots B_{a+1} B_a .$$

Beachte die Reihenfolge, da die Matrixmultiplikation nicht kommutativ ist.

Satz 4.6 (LR-Zerlegung)

Sei $A \in \mathbb{R}^{n\times n}$ regulär, dann gibt es eine Zerlegung

$$PA = LR$$

wobei

$$L = \begin{pmatrix} 1 & & 0 \\ l_{21} & \ddots & \\ \vdots & & \ddots \\ l_{n1} & \cdots & l_{n,n-1} \ 1 \end{pmatrix}, \quad R = \begin{pmatrix} r_{11} & & r_{1n} \\ & \ddots & \\ 0 & & r_{nn} \end{pmatrix},$$

und $P = \prod_{k=1}^{n-1} P_{k,r_k}$ ein Produkt von Permutationsmatrizen mit $r_k \geq k$.

Im Fall $P = I$ ist die Zerlegung eindeutig.

Beweis a) Wir behandeln zunächst den Fall $P = I$, d.h. ohne Zeilentausch.

Das Gaußsche Eliminationsverfahren lässt sich schreiben als

$$Ax = b$$

Schritt 1: $G_1 A x = G_1 b$ mit Frobeniusmatrix $(G_1)_{i1} = -q_{i1} = -\dfrac{\tilde{a}^{(1)}_{i1}}{\tilde{a}^{(1)}_{11}}$

Schritt 2: $G_2 G_1 A x = G_2 G_1 b$

$\vdots$

nach n-1 Schritten $\underbrace{G_{n-1} \cdots G_1 A}\, x = G_{n-1} \cdots G_1 b$ mit $(G_k)_{ik} = -q_{ik}$ aus GEM

Ergebnis der GEM: $G_{n-1} \cdots G_1 A = R$ (rechte obere Dreiecksmatrix).

Nun nutze Hilfssatz 4.5 ii.

$$A = G_1^{-1} G_2^{-1} \cdots G_{n-1}^{-1} R$$
$$\overset{(iii)}{=} (I - G_1') \cdot (I - G_2') \cdots (I - G_{n-1}') R$$
$$\overset{(iv)}{=} \underbrace{\left(I - \sum_{j=1}^{n-1} G_j'\right)}_{=: L} R$$

L hat die geforderte Gestalt.

b) Nun mit den Zeilenvertauschungen. GEM liefert

$$G_{n-1} P_{n-1, r_{n-1}} \cdots \underset{\alpha\ \beta}{P_{2, r_2}} \underset{k}{G_1} \underset{k < \alpha \leq \beta}{P_{1, r_1}} A = R$$

r_k ist der im Schritt k bestimmte Vertauschungsindex $r_k \geq k$.

Wende sukzessive HS 4.5 v) an

$$G_{n-1} P_{n-1, r_{n-1}} \cdots P_{2, r_2} G_1 \cdot P_{1, r_1} A$$
$$= G_{n-1} P_{n-1, r_{n-1}} \cdots P_{3, r_3} G_2 \underbrace{\left(I + P_{2, r_2} G_1'\right)}_{\text{wieder Frobeniusm.}} P_{2, r_2} P_{1, r_1} A$$
$$= G_{n-1} P_{n-1, r_{n-1}} P_{4, r_4} G_3 \left(I + P_{3, r_3} G_2'\right)\left(I + P_{3, r_3} P_{2, r_2} G_1'\right) P_{3, r_3} P_{2, r_2} P_{1, r_1} A$$
$$\vdots$$
$$= \underbrace{\prod_{k=1}^{n-1} \left(I + \left(\prod_{\alpha = k+1}^{n-1} P_{\alpha, r_\alpha}\right) G_k'\right)}_{\text{nach rechts bringen}} \underbrace{\prod_{k=1}^{n-1} P_{k, r_k}}_{=: P} A = R$$

alle „späteren" Tauschoperationen.

$$\Longleftrightarrow \quad PA = \underbrace{\prod_{k=n-1}^{1} \left(I - \left(\prod_{\alpha = k+1}^{n-1} P_{\alpha, r_\alpha}\right) G_k'\right)}_{=: L} R$$

wg Inverse

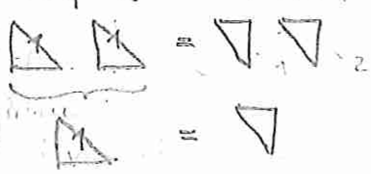
c) Eindeutigkeit. Es ist $P = I$, also $A = LR$.

Ang. es gebe zwei Zerlegungen $A = L_1 R_1 = L_2 R_2$ der geforderten Gestalt.

$A^{-1} = R_1^{-1} L_1^{-1} = R_2^{-1} L_2^{-1}$, also

$$I = AA^{-1} = \underbrace{L_1 R_1}_{A} \underbrace{R_2^{-1} L_2^{-1}}_{A^{-1}} = L_1 R_1 R_2^{-1} L_2^{-1} \iff L_1^{-1} L_2 = R_1 R_2^{-1}$$

- Inverse ist wieder unter (oben Dreiecksm) (mit 1 auf der HDiag.)
- Produkt ist wieder



geht nur für $L_1^{-1} L_2 = R_1 R_2^{-1} = I$ ■

Wo ist L?

Aus dem Beweis sehen wir, dass

$$L = \prod_{k=n-1}^{1} \underbrace{\left(I - \left(\prod_{\alpha=k+1}^{n-1} P_{\alpha, r_\alpha} \right) G_k' \right)}_{\text{Frobeniusmatrix}}$$

HS 4.5 (iv)
$$= I - \sum_{k=1}^{n+1} \left(\prod_{\alpha=k+1}^{n-1} P_{\alpha, r_\alpha} \right) G_k' \qquad \text{mit } (G_k')_{ik} = -q_{ik} \text{ aus GEM.}$$

- Nichtnullelemente in G_k' nehmen gerade den Platz der eliminierten Einträge $\tilde{a}_{ik}^{(k)}$ ein → kein zusätzlicher Speicher notwendig (A wird ohnehin überschrieben).
- Die G_k' nehmen an den späteren Zeilenvertauschen ebenso teil wie der Rest $A^{(k+1)}$, ⇒ ganze Zeile tauschen!

Wozu das Ganze?

Lösen von $Ax = b$ mittels LR-Zerlegung:

$$Ax = b$$
$$\iff PAx = Pb$$
$$\iff L\underbrace{Rx}_{=:y} = Pb$$
$$\iff \begin{cases} Ly = Pb & \text{vorwärts einsetzen} \\ Rx = y & \text{rückwärts einsetzen.} \end{cases}$$

also
1. Berechne LR-Zerl. von A und P
2. Für geg. b berechne $b' = Pb$
3. Löse $Ly = b'$
4. Löse $Rx = y$

Aufwand

Beobachtung: Transformation auf obere Dreiecksgestalt mittels GEM äquivalent zu LR-Zerlegung berechnen + $Ly = Pb$ lösen.

Somit: Der Aufwand zur Berechnung der LR-Zerlegung ist

$$N_{LR}(n) = \frac{2}{3}n^3 + O(n^2).$$

LR-Zerlegung ist insbesondere interessant, wenn $Ax = b_i$ zu mehreren rechten Seiten b_i zu lösen ist.

Zur Permutation

$$P = P_{n-1,r_{n-1}} \cdots P_{2,r_2} \, P_{1,r_1} \quad \text{mit } r_k \geq k$$

speichert man nicht als Matrix sondern man speichert nur die Zahlen $r_1, \dots, r_{n-1}$ in einem Vektor (Feld).

4.7 Algorithmus zur LR-Zerlegung

Input: $A \in \mathbb{R}^{n\times n}$ (wird überschrieben)

Output: $L \in \mathbb{R}^{n\times n}$ in $a_{ij}, j<i$, $l_{ii}=1$ implizit

$R \in \mathbb{R}^{n\times n}$ in $a_{ij}, j\geq i$

$p: \{0,\dots,n-1\} \to \{0,\dots,n-1\}$

```
for (k=1; k≤n; k=k+1) {
    Finde r ∈ {k,...,n} : so dass a_rk ≠ 0; sonst Fehler;
    if (r≠k)    // tausche Zeilen
        for (j=1; j≤n; j=j+1) {
            t = a_kj; a_kj = a_rj; a_rj = t;
        }
    p[k] = r;    // merke Permutation

    for (i=k+1; i≤n; i=i+1) {
        a_ik = a_ik / a_kk;
        for (j=k+1; j≤n; j=j+1)
            a_ij = a_ij - a_ik a_kj;
    }
}
```