Matrizen - Algorithmen

Ac 2017-2020

Matrizen sind Tabellen mit ze Zeilen und sp Spalten .

Man kann mit ihnen Operationen durchführen, die in verschiedenen Bereichen benötigt werden (z.B. Lösen von Linearen Gleichungssystemen).

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1;sp} \\ a_{21} & a_{22} & \dots & a_{2;sp} \\ \dots & & & \\ a_{ze;1} & a_{ze;2} & \dots & a_{ze;sp} \end{pmatrix} \text{ mit } a_{ik} \in \mathbb{R} \text{ ist eine (ze;sp)-Matrix.} \qquad \text{Schreibweise: } A = (a_{ik})$$

Die Addition A+B :

Bei der Addition addiert man jedes a_{ik} zu dem entsprechenden b_{ik} .

Beispiel:

$$\begin{pmatrix} -4 & 3 & 11 \\ 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 & -3 \\ 5 & -1 & 4 \end{pmatrix} = \begin{pmatrix} -4+2 & 3+0 & 11-3 \\ 1+5 & 0-1 & 1+4 \end{pmatrix} = \begin{pmatrix} -2 & 3 & 8 \\ 6 & -1 & 5 \end{pmatrix}$$

Die Multiplikation A·B :

A ist eine (m, n) - Matrix, d.h. sie hat m Zeilen und n Spalten .
B ist eine (n, p) - Matrix, d.h. sie hat n Zeilen und p Spalten .
Die Produktmatrix A · B sei C . Es wird sich zeigen, daß C eine (m, p) - Matrix ist !

Multiplikationsregel :

1. Lege die i-te Zeile von A auf die k-te Spalte von B.
2. Multipliziere aufeinanderliegende Elemente miteinander.
3. Die Summe dieser Produkte („Skalarprodukte“) ist dann das Element c_{ik} .
4. Wiederhole die Schritte 1 bis 3 für i von 1 bis m und für k von 1 bis p !

Beispiel:

$$\begin{pmatrix} -4 & 3 & 11 \\ 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -2 & 1 \\ 4 & -3 \end{pmatrix} = \begin{pmatrix} -4\cdot 1+3\cdot(-2)+11\cdot 4 & -4\cdot 0+3\cdot 1+11\cdot(-3) \\ 1\cdot 1+0\cdot(-2)+1\cdot 4 & 1\cdot 0+0\cdot 1+1\cdot(-3) \end{pmatrix} = \begin{pmatrix} 34 & -30 \\ 5 & -3 \end{pmatrix}$$

Die skalare Multiplikation r ·A (r reell) :

Bei der skalaren Multiplikation mutipliziert man jedes a_{ik} mit r .

Beispiel:

$$3 \cdot \begin{pmatrix} -4 & 3 & 11 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3\cdot(-4) & 3\cdot 3 & 3\cdot 11 \\ 3\cdot 1 & 3\cdot 0 & 3\cdot 1 \end{pmatrix} = \begin{pmatrix} -12 & 9 & 33 \\ 3 & 0 & 3 \end{pmatrix}$$

Die Einheitsmatrix E ist das **neutrale Element der Multiplikation** ; E muss quadratisch sein !

$$E = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Es gilt: **A · E = E* · A = A**

A muss nicht quadratisch sein, aber die Dimensionen von E und E* müssen so gewählt werden, dass in beiden Fällen eine Multiplikation möglich ist.

Beispiel: $A = \begin{pmatrix} 2 & 1 & 0 \\ -3 & 4 & -2 \end{pmatrix}$ $E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $E^* = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Die Nullmatrix N ist das **neutrale Element der Addition** ; N muss nicht quadratisch sein !

$$N = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Es gilt: **A + N = N + A = A**

A muss nicht quadratisch sein, aber die gleiche Dimension wie N haben, damit in beiden Fällen eine Addition möglich ist .

Anmerkung: Bei Matrizen gibt es sog. „**Nullteiler**“, d.h. A·B = N mit A ≠ N und B ≠ N .

Ein einfaches Beispiel: $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

Gäbe es Nullteiler bei reellen Zahlen a, b , so wäre a·b = 0 mit a ≠ 0 und b ≠ 0 möglich !!

Die Transponierte A^T entsteht durch die Vertauschung von Zeilen und Spalten der Matrix A .

Beispiel: $A = \begin{pmatrix} 3 & -1 & 2 \\ 0 & 4 & 1 \end{pmatrix}$. Dann ist $A^T = \begin{pmatrix} 3 & 0 \\ -1 & 4 \\ 2 & 1 \end{pmatrix}$. Man spiegelt also an der Hauptdiagonale !

Die Spur sp(A) ist die Summe der Hauptdiagonalenelemente der quadratischen Matrix A .

Beispiel: $A = \begin{pmatrix} -5 & 4 & 2 \\ 3 & 3 & 5 \\ -4 & -2 & 1 \end{pmatrix}$. Dann ist $sp(A) = -5 + 3 + 1 = -1$

Der **Rang rg(A) einer Matrix A** ist die Anzahl der linear unabhängigen Zeilen- bzw. Spaltenvektoren von A. Dies spielt z.B. eine Rolle bei der Existenz von Lösungen bei Linearen Gleichungssystemen.

Z.B. ist der Rang von $A = \begin{pmatrix} 1 & 1 \\ 4 & 3 \\ 3 & 4 \end{pmatrix}$ gleich **2** , denn für z.B. den untersten Zeilenvektor [3 ; 4] gilt:

[3 ; 4] = 7·[1 ; 1] - [4 ; 3] = [7 ; 7] - [4 ; 3] ; daher ist er von den ersten beiden Zeilenvektoren linear abhängig !

Der Rang lässt sich bequem mit dem Gaußschen Eliminationsverfahren (s. unten) berechnen .

<u>Lineare Gleichungssysteme (LGSe) :</u>

Beispiel: Zu lösen ist das LGS $\begin{vmatrix} 1x_1 & -5x_2 & +7x_3 & = 2 \\ 4x_1 & -19x_2 & +27x_3 & = 0 \\ -2x_1 & +13x_2 & -16x_3 & = -1 \end{vmatrix}$.

Hierfür kann man mittels Matrizen folgendes schreiben: $\begin{pmatrix} 1 & -5 & 7 \\ 4 & -19 & 27 \\ -2 & 13 & -16 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}$

Kurzschreibweise: $A \cdot \vec{x} = \vec{b}$ oder noch kürzer $A \mid \vec{b}$

$A = \begin{pmatrix} 1 & -5 & 7 \\ 4 & -19 & 27 \\ -2 & 13 & -16 \end{pmatrix}$ nennt man <u>Koeffizientenmatrix</u> und

$A \mid \vec{b} = \begin{pmatrix} 1 & -5 & 7 & | & 2 \\ 4 & -19 & 27 & | & 0 \\ -2 & 13 & -16 & | & -1 \end{pmatrix}$ <u>erweiterte Koeffizientenmatrix</u> .

Man löst ein solches LGS mit dem **<u>Gauß-Algorithmus</u>** bzw. **Gauß-Jordan-Algorithmus**:

Ziel ist dabei <u>zunächst</u> die Erzeugung einer sog. **Zeilen-Stufenform** bzw. **row echelon form („ref “) .**

1.Schritt: Subtrahiere das 4-fache der 1. Zeile von der 2. Zeile.
Subtrahiere das (-2)-fache der 1. Zeile von der 3. Zeile.

Ergebnis: $\begin{pmatrix} 1 & -5 & 7 & | & 2 \\ 0 & 1 & -1 & | & -8 \\ 0 & 3 & -2 & | & 3 \end{pmatrix}$. In der ersten Spalte stehen in 2 Zeilen Nullen.

Jetzt muss in der letzten Zeile aus $a_{31} = 3$ eine 0 gemacht werden.
Dazu subtrahiert man das 3-fache der 2.Zeile von der 3.Zeile .

Ergebnis: $\begin{pmatrix} 1 & -5 & 7 & | & 2 \\ 0 & 1 & -1 & | & -8 \\ 0 & 0 & 1 & | & 27 \end{pmatrix}$. Dies ist die gewünschte **Zeilen-Stufenform (ref) ; Gauß-Alg.** .

Aus der 3.Zeile liest man sofort die Lösung **x_3 = 27** ab.

Durch „Rückwärtseinsetzen“ von unten nach oben ermittelt man die restlichen beiden Lösungen.

Insgesamt erhält man so die **reduzierte Zeilen-Stufenform** bzw. **reduced row echelon form („rref “)**
Der Algorithmus zum Erreichen der „rref“ heißt <u>Gauß-Jordan-Algorithmus</u> !

Addition der 3.Zeile zur 2.Zeile liefert $\begin{pmatrix} 1 & -5 & 7 & | & 2 \\ 0 & 1 & 0 & | & 19 \\ 0 & 0 & 1 & | & 27 \end{pmatrix}$. Also ist **x_2 = 19** .

In der 1.Zeile müssen noch die beiden Elemente -5 und 7 zu 0 gemacht werden.

Dies erreicht man durch 2 Schritte:

Subtraktion des (-5)-fachen der 2.Zeile von der 1.Zeile: $\begin{pmatrix} 1 & 0 & 7 & | & 97 \\ 0 & 1 & 0 & | & 19 \\ 0 & 0 & 1 & | & 27 \end{pmatrix}$

Subtraktion des 7-fachen der 3.Zeile von der 1.Zeile: $\begin{pmatrix} 1 & 0 & 0 & | & -92 \\ 0 & 1 & 0 & | & 19 \\ 0 & 0 & 1 & | & 27 \end{pmatrix}$ **reduzierte Zeilen-Stufenform (rref)**

Schließlich erhält man noch **$x_1 = -92$** .

Spaltenpivotisierung:

Das Verfahren der Zeilenstufenformerzeugung muss erweitert werden, denn das Diagonalelement, durch das dividiert wird, hat nicht immer einen von 0 verschiedenen Wert . Z.B. kann gelten: $a_{11} = 0$.

Man sucht dann in der zugehörigen Spalte (hier: Spalte 1) ein Element ungleich 0. Findet man dies in der i-ten Zeile, so vertauscht man die beiden Zeilen (hier: 1 und i). Dabei sucht man aus Gründen der numerischen Stabilität das betragsgrößte Element der betreffenden Spalte heraus (sog. „Pivot").
Pivot: franz.: Drehpunkt; Angelpunkt; das Element, welches als erstes gewählt wird

Beispiel: $\begin{pmatrix} 0 & 1 & 2 & | & 8 \\ 1 & 0 & 5 & | & 16 \\ 2 & 4 & 0 & | & 10 \end{pmatrix}$ In Spalte1 ist 2 das betragsgrößte Element; also Tausch der Zeilen 1 und 3.

$\begin{pmatrix} 2 & 4 & 0 & | & 10 \\ 1 & 0 & 5 & | & 16 \\ 0 & 1 & 2 & | & 8 \end{pmatrix}$. Anschließend werden in der 1. Spalte unterhalb der 2 Nullen erzeugt.

$\begin{pmatrix} 2 & 4 & 0 & | & 10 \\ 0 & -2 & 5 & | & 11 \\ 0 & 1 & 2 & | & 8 \end{pmatrix}$ Schließlich muss noch die 1 in Zeile3 zu Null gemacht werden.

Man erhält durch entsprechende Umformungen die Stufenform : $\boxed{\begin{pmatrix} 2 & 4 & 0 & | & 10 \\ 0 & -2 & 5 & | & 11 \\ 0 & 0 & 4,5 & | & 13,5 \end{pmatrix}}$

Weitere Vorgehensweise zur Lösung des LGS:

Man dividiert die letzte Zeile durch 4,5 (die Lösung $x_3 = 3$ kann man hier schon erkennen) .
Nun subtrahiert man das 5-fache dieser Zeile von Zeile2.

Zwischenergebnis: $\begin{pmatrix} 2 & 4 & 0 & | & 10 \\ 0 & -2 & 0 & | & -4 \\ 0 & 0 & 1 & | & 3 \end{pmatrix}$

Zum Schluss subtrahiert man das (-2)-fache der 2.Zeile von Zeile1.

$\begin{pmatrix} 2 & 0 & 0 & | & 2 \\ 0 & -2 & 0 & | & -4 \\ 0 & 0 & 1 & | & 3 \end{pmatrix}$. Durch entsprechende Divisionen erhält man: $\begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 0 & 1 & 0 & | & 2 \\ 0 & 0 & 1 & | & 3 \end{pmatrix}$

Die Lösungen sind also **$x_1 = 1$; $x_2 = 2$; $x_3 = 3$** .

Gauß-Jordan-Algorithmus für eine Koeffizientenmatrix (a_{ik}) mit ze Zeilen und sp Spalten

PSEUDOCODE:

(1) Erzeugung der Zeilenstufenform:

```
für spalte von 1 bis ze tue
   falls aik[spalte,spalte] = 0
      suche betragsgrößtes Element a[zmax,spalte] in der Spalte  //Maximumsuche
      falls a[zmax,spalte] = 0
         dann Abbruch (Matrix singulär)
      vertausche Zeilen mit den Nummern zeile und zmax

   für zeile von spalte+1 bis ze tue
      faktor = aik[zeile,spalte] / aik[spalte,spalte]
      für j von spalte+1 bis sp tue
         aik[zeile,j] = aik[zeile,j] - aik[spalte,j] * faktor
      aik[zeile,spalte] = 0
```

(2) Rückwärtssubstitution:

```
// erst alle Diagonalelemente auf 1 bringen
für zeile von 1 bis ze tue
   für spalte von sp ab bis zeile tue
      aik[zeile,spalte] = aik[zeile,spalte] / aik[zeile/zeile]

// dann rücksubstituieren
für spalte von ze ab bis 2 tue
   für zeile von 1 bis spalte-1 tue
      für j von sp ab bis spalte tue
         aik[zeile,j] = aik[zeile,j] - aik[spalte,j] * aik[zeile,spalte]
```

Warum wählt man das betragsgrößte Element einer Spalte als Pivot ?

Wenn das Pivotelement sehr klein wäre, könnten große numerische Fehler entstehen, wie das folgende Beispiel zeigt. Zunächst wird die exakte Lösung ermittelt:

$$\begin{pmatrix} \varepsilon & 1 & | & 1 \\ 1 & 1 & | & 2 \end{pmatrix} \to \begin{pmatrix} 1 & 1/\varepsilon & | & 1/\varepsilon \\ 1 & 1 & | & 2 \end{pmatrix} \to \begin{pmatrix} 0 & 1-1/\varepsilon & | & 2-1/\varepsilon \\ 1 & 1 & | & 2 \end{pmatrix} \to \begin{pmatrix} 0 & 1-1/\varepsilon & | & 2-1/\varepsilon \\ 1-1/\varepsilon & 1-1/\varepsilon & | & 2-2/\varepsilon \end{pmatrix}$$

$$\to \begin{pmatrix} 0 & 1-1/\varepsilon & | & 2-1/\varepsilon \\ 1-1/\varepsilon & 0 & | & -1/\varepsilon \end{pmatrix} \to \begin{pmatrix} 0 & 1 & | & (2-1/\varepsilon)/(1-1/\varepsilon) \\ 1 & 0 & | & -1/\varepsilon/(1-1/\varepsilon) \end{pmatrix} \to \boxed{\begin{pmatrix} 1 & 0 & | & 1/(1-\varepsilon) \\ 0 & 1 & | & (1-2\varepsilon)/(1-\varepsilon) \end{pmatrix}}$$

Für $\begin{pmatrix} 10^{-16} & 1 & | & 1 \\ 1 & 1 & | & 2 \end{pmatrix}$ müsste man bei Verwendung von 15 Stellen folgende Ergebnisse erhalten:

$x_1 = 1/(1-10^{-16})$ = 10000000000000000 / 9999999999999999 = 1,000000000000000100...

$x_2 = (1-2\cdot10^{-16}) / (1-10^{-16})$ = 9999999999999998 / 9999999999999999 = 0,99999999999999899... .

Mit Spaltenpivotisierung liefert Java7 die Ergebnisse: 1.0000000000000009 und 0.999999999999999.

Ohne Spaltenpivotisierung: 2.0 (falsch !) und 0.9999999999999998

Weitere Beispiele für Lineare Gleichungssysteme (mit Lösungen) :

$$\begin{pmatrix} 2 & 5 & 3 & | & 21 \\ 4 & 3 & 5 & | & 25 \\ 10 & 5 & 1 & | & 23 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 0 & 1 & 0 & | & 2 \\ 0 & 0 & 1 & | & 3 \end{pmatrix}$$

Die Lösungen sind $x_1 = 1$; $x_2 = 2$; $x_3 = 3$.

$$\begin{pmatrix} 1 & 1 & 2 & | & 6 \\ 1 & 1 & 2 & | & 9 \\ 1 & 2 & 1 & | & 8 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 3 & | & 0 \\ 0 & 1 & -1 & | & 0 \\ 0 & 0 & 0 & | & 1 \end{pmatrix}$$

Widerspruch 0=1 in der letzten Zeile ! Die Lösungsmenge ist leer.

$$\begin{pmatrix} 1 & -1 & 2 & | & 6 \\ -3 & 3 & -6 & | & -18 \\ 1 & 0 & 3 & | & 8 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 3 & | & 8 \\ 0 & 1 & 1 & | & 2 \\ 0 & 0 & 0 & | & 0 \end{pmatrix}$$

In der letzten Zeile stehen ausschließlich Nullen.

Es entsteht die Identität 0 = 0 . Dies bedeutet, dass 2 Zeilen des Systems linear abhängig sind.
Das LGS besitzt daher unendlich viele Lösungen ($x_1 + 3x_3 = 8$ und $x_2 + x_3 = 2$) !

Unterbestimmte Systeme (weniger Gleichungen als Unbekannte):

Im Allgemeinen existiert keine eindeutige Lösung, sondern es gibt Abhängigkeiten zwischen den Variablen.

$$\begin{pmatrix} 4 & 1 & 2 \\ 1 & 3 & 5 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2550 \\ 3750 \end{pmatrix} \Rightarrow \begin{pmatrix} 4 & 1 & 2 & | & 2550 \\ 1 & 3 & 5 & | & 3750 \end{pmatrix} \Rightarrow \begin{pmatrix} 11 & 0 & 1 & | & 3900 \\ 0 & 11 & 18 & | & 12450 \end{pmatrix}$$

Hier gibt es unendlich viele Lösungen mit: $11x_1 = 3900 - x_3$ und $11x_2 = 12450 - 18x_3$

Überbestimmte Systeme (mehr Gleichungen als Unbekannte):

Im Allgemeinen existiert keine eindeutige Lösung, sondern es entsteht ein Widerspruch, wie das folgende Beispiel zeigt:

Beispiel:

$$\begin{pmatrix} 3 & 2 \\ 1 & 2 \\ 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 24 \\ 17 \\ 35 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 3x_1 + 2x_2 \\ x_1 + 2x_2 \\ 4x_2 \end{pmatrix} = \begin{pmatrix} 24 \\ 17 \\ 35 \end{pmatrix} \Leftrightarrow \begin{array}{ccc} \cdot x_1 & \cdot x_2 & \cdot 1 \\ \hline 3 & 2 & 24 \\ 1 & 2 & 17 \\ 0 & 4 & 35 \end{array} \; rref \Rightarrow \begin{array}{ccc} \cdot x_1 & \cdot x_2 & \cdot 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

In der letzten Zeile ergibt sich der Widerspruch (0 = 1) . Also gibt es keine Lösung !

Ein besonders interessantes überbestimmtes System für 2 Größen x_1 , x_2 ist das folgende :

$$\begin{pmatrix} 48 & 60 & | & 2220 \\ 16 & 21 & | & 765 \\ 29 & 38 & | & 1385 \\ 23 & 30 & | & 1095 \end{pmatrix} \rightarrow \begin{pmatrix} 48 & 60 & | & 2220 \\ 0 & 1 & | & 25 \\ 0 & 1{,}75 & | & 43{,}75 \\ 0 & 1{,}25 & | & 31{,}25 \end{pmatrix} \rightarrow \begin{pmatrix} 48 & 60 & | & 2220 \\ 0 & 1{,}75 & | & 43{,}75 \\ 0 & 1 & | & 25 \\ 0 & 1{,}25 & | & 31{,}25 \end{pmatrix} \rightarrow \begin{pmatrix} 48 & 60 & | & 2220 \\ 0 & 1{,}75 & | & 43{,}75 \\ 0 & 0 & | & 0 \\ 0 & 0 & | & 0 \end{pmatrix}$$

Obwohl sogar 3 Zeilen linear abhängig sind, gibt es hier eine eindeutige Lösung.
Die 2.Zeile liefert $x_2 = 25$ und aus der 1.Zeile ermittelt man durch Einsetzen $x_1 = 15$.

JAVA-Implementierung des Gaußschen Algorithmus: „ref“ bzw. „rref“ bzw. „LR-Zerlegung(siehe unten)“

```java
public static double[][] ref(double[][] matUr, boolean LR, boolean pivot, boolean rueckSubst) {
	int zeM = matUr.length; // Zeilenzahl von matUr
	int spM = matUr[0].length; // Spaltenzahl der 0. Zeile von matUr
	if (LR && zeM != spM)
		throw new IllegalArgumentException("Matrix muss quadratisch sein !");

	// deep copy;  matUr soll nicht verändert werden !
	double [][] mat = new double[zeM][spM];
	for (int ze = 0; ze < zeM; ze++)
		for (int sp = 0; sp < spM; sp++)
			mat[ze][sp] = matUr[ze][sp];

	double faktor, tmp, merke;

	for (int sp = 0; sp < zeM; sp++) {
		// suche betragsgrößtes Element(Pivot) in Spalte sp ab Zeile Nr. sp
		double max = Math.abs(mat[sp][sp]);
		int zeMax = sp;
		if (pivot || max == 0.0) {
			for (int ze = sp + 1; ze < zeM; ze++) {
				tmp = Math.abs(mat[ze][sp]);
				if (tmp > max) {
					max = tmp;
					zeMax = ze;
				}
			} // for ze
		}

		if (mat[zeMax][sp] == 0.0)
			if (zeM != spM)
				return mat;
			else
				throw new NumberFormatException("Matrix ist singulär !");

		if (sp != zeMax) { // tausche Zeilen sp und zeMax
			for (int j = 0; j < spM; j++) {
				merke = mat[zeMax][j];
				mat[zeMax][j] = mat[sp][j];
				mat[sp][j] = merke;
			}
		}

		// für alle Zeilen unterhalb des Pivotelements:
		for (int ze = sp + 1; ze < zeM; ze++) {
			// für alle verbleibenden Elemente in der Zeile
			faktor = mat[ze][sp] / mat[sp][sp];

			for (int j = sp + 1; j < spM; j++)
				mat[ze][j] = mat[ze][j] - mat[sp][j] * faktor;

			// setze das am linken Rand stehende Element = 0 (bzw. = faktor)
			if (LR)
				mat[ze][sp] = faktor;
			else
				mat[ze][sp] = 0.0;
		} // for ze
	} // for sp

	if (rueckSubst) { // Rücksubstitution
		// erst alle Diagonalelemente auf 1 bringen
		for (int ze = 0; ze < zeM; ze++)
			for (int sp = spM - 1; sp >= ze; sp--)
				mat[ze][sp] = mat[ze][sp] / mat[ze][ze];
		// dann rückwärtsrechnen
		for (int sp = zeM - 1; sp >= 1; sp--)
		    for (int ze = 0; ze <= sp - 1; ze++)
		        for (int vglSp = spM - 1; vglSp >= sp; vglSp--)
				mat[ze][vglSp] = mat[ze][vglSp] - mat[sp][vglSp] * mat[ze][sp];
	}
	return mat;
}  // Ende ref
```

<u>Die Determinante det (A) :</u>

Die Determinante det (A) = | A | ist eine <u>reelle Zahl</u>, mit der man feststellen kann, ob das LGS $A \cdot \vec{x} = \vec{b}$ eindeutig lösbar ist. Dies ist genau dann der Fall, wenn det(A) ≠ 0 gilt . **A muss quadratisch sein !**

Für eine (2,2)-Matrix gilt folgende Formel : $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$

Für (n,n)-Matrizen mit n>2 kann man die Determinante durch **Entwicklung** nach der <u>k-ten Zeile</u> berechnen :

$$\det(A) = \sum_{i=1}^{n} (-1)^{i+k} \cdot a_{ik} \cdot \det(A_{ik})$$ **<u>Laplacescher Entwicklungssatz</u>**

Eine (3,3)-Matrix lässt sich somit auf (2,2)-Matrizen zurückführen (z.B. Entwicklung nach der 1.Zeile):

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \cdot \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \cdot \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \cdot \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Beispiel: $\begin{vmatrix} \boxed{-4} & \boxed{3} & \boxed{5} \\ 2 & -4 & -3 \\ 5 & -2 & -7 \end{vmatrix} = -4 \cdot \begin{vmatrix} -4 & -3 \\ -2 & -7 \end{vmatrix} - 3 \cdot \begin{vmatrix} 2 & -3 \\ 5 & -7 \end{vmatrix} + 5 \cdot \begin{vmatrix} 2 & -4 \\ 5 & -2 \end{vmatrix} =$

$$-4 \cdot (-4 \cdot (-7) - (-3) \cdot (-2)) - 3 \cdot (2 \cdot (-7) - (-3) \cdot 5) + 5 \cdot (2 \cdot (-2) - (-4) \cdot 5) =$$

$$-4 \cdot (28 - 6) - 3 \cdot (-14 + 15) + 5 \cdot (-4 + 20) = -4 \cdot 22 - 3 + 5 \cdot 16 = -88 - 3 + 80 = -11$$

Auf diese Weise kann man (n,n)-Matrizen auf (n-1,n-1)-Matrizen zurückführen und somit bis auf (2,2)-Matrizen „herunterrechnen“ . Jedoch ist der <u>Rechenaufwand für große n sehr hoch</u>.

Besser geeignet ist die **<u>Methode der Verdichtung</u>** , die an einem (3,3)-Beispiel verdeutlicht werden soll.

Folgende Regeln werden angewendet:

- Multiplikation einer Zeile mit r bedeutet eine Multiplikation von det(A) mit r .
- Vertauschen zweier Zeilen bedeutet eine Multiplikation von det(A) mit -1 .
- Differenz oder Summe von Zeilen hat keine Auswirkungen auf det(A) .

Beispiel: $\begin{vmatrix} \boxed{-4} & 3 & 5 \\ 2 & -4 & -3 \\ 5 & -2 & -7 \end{vmatrix} \begin{matrix} \\ \cdot(-4) \\ \cdot(-4) \end{matrix} = \frac{1}{(-4)^2} \cdot \begin{vmatrix} -4 & 3 & 5 \\ -8 & 16 & 12 \\ -20 & 8 & 28 \end{vmatrix} = \frac{1}{(-4)^2} \cdot \begin{vmatrix} -4 & 3 & 5 \\ 0 & 10 & 2 \\ 0 & -7 & 3 \end{vmatrix} = \frac{1}{(-4)^2} \cdot (-4) \cdot \begin{vmatrix} 10 & 2 \\ -7 & 3 \end{vmatrix} =$

$$-0,25 \cdot (10 \cdot 3 - 2 \cdot (-7)) = -0,25 \cdot 44 = -11$$

Das Verfahren funktioniert wie der Gaußsche Algorithmus, und auch hier muss mit Pivots gearbeitet werden.

Man sieht, dass in Spalte1 alle Elemente außer des ersten auf 0 gebracht werden.
Nach dem Laplaceschen Entwicklungssatz (s.o.) sind dann 2 der 3 Unterdeterminanten 0,
so dass man zur Lösung nur die restliche Unterdeterminante benötigt .

Die Inverse Matrix A^{-1} bzw. (besser !) A^{I} :

Für die **Inverse** A^I einer Matrix A gilt der Ansatz: $A \cdot A^I = A^I \cdot A = E$ (E = Einheitsmatrix; s.u.). Nur für quadratische Matrizen A kann eine Inverse existieren, andernfalls wären die o.a. Multiplikationen unmöglich !

Ist die Determinante von A = 0 , so ist A nicht invertierbar.

Invertierbare Matrizen heißen **regulär**, nichtinvertierbare **singulär** .

Beispiel 1: $A = \begin{pmatrix} 2 & 1 & 1 \\ 3 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix}$. Gesucht ist $A^I = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$

Ansatz: $\begin{pmatrix} 2 & 1 & 1 \\ 3 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

Die reduzierte Koeffizientenmatrix ist dann: $\begin{pmatrix} 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 \end{pmatrix}$

Wendet man den Gauß-Algorithmus an, so erhält man $\begin{pmatrix} 1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 1 & 0 & -4 & 3 & -1 \\ 0 & 0 & 1 & 1 & -1 & 1 \end{pmatrix}$.

Es gilt daher $\boxed{A^I = \begin{pmatrix} 2 & -1 & 0 \\ -4 & 3 & -1 \\ 1 & -1 & 1 \end{pmatrix}}$ (die gesuchte Inverse von A)

Man kann überprüfen, dass $A \cdot A^I = A^I \cdot A = E$ gilt .

Beispiel 2: $\mathrm{A} = \begin{pmatrix} 48 & 60 & 2220 \\ 16 & 21 & 765 \\ 29 & 38 & 1385 \end{pmatrix}$ Wegen $\mathrm{A} \cdot \mathrm{A}^{\mathrm{I}} = \mathrm{E}$ folgt $\begin{pmatrix} 48 & 60 & 2220 & 1 & 0 & 0 \\ 16 & 21 & 765 & 0 & 1 & 0 \\ 29 & 38 & 1385 & 0 & 0 & 1 \end{pmatrix}$

Der Gauß-Algorithmus liefert dann: $\begin{pmatrix} 1 & 0 & 15 & 0 & -38 & 21 \\ 0 & 1 & 25 & 0 & 29 & -16 \\ 0 & 0 & 0 & 1 & 84 & -48 \end{pmatrix}$

Hier entsteht in der linken Hälfte der erweiterten Koeffizientenmatrix nicht die gewünschte Einheitsmatrix E .

Daher gibt es keine Inverse für A , A ist singulär !

Anmerkung:

Die Singularität von A kann man auch durch Berechnung der Determinante feststellen, da det(A) = 0 gilt !

Die LR (LU) - Zerlegung :

Jede <u>reguläre</u> (n,n)-Matrix A kann in eine linke (untere) Dreiecksmatrix L (mit Einsen auf der Hauptdiagonale) und eine rechte (obere) Dreiecksmatrix R (alle Diagonalelemente ≠ 0) zerlegt werden.
Es gilt dann: **A = L · R** bzw. **P · A = L · R** bei Zeilenvertauschung (P = Permutationsmatrix; s.u.)

Anmerkung: Die englische Bezeichnung ist LU , wobei L = Lower und U = Upper bedeuten .

Beispiel (ohne Zeilenvertauschung): $A = \begin{pmatrix} 2 & 1 & 1 \\ 3 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix}$ Gesucht: $L \cdot R = \begin{pmatrix} 1 & 0 & 0 \\ d & 1 & 0 \\ g & h & 1 \end{pmatrix} \cdot \begin{pmatrix} a & b & c \\ 0 & e & f \\ 0 & 0 & i \end{pmatrix}$

Man geht vor wie beim Gauß-Algorithmus zur Bestimmung der Zeilenstufenform. Allerdings werden hier in der linken Dreiecksmatrix keine Nullen erzeugt, sondern es werden die <u>Faktoren</u> gespeichert, die man benötigt, um (beim Gauß-Algorithmus) die 0 an der betreffenden Stelle zu erzeugen.

Konkret: Im ersten Schritt betrachtet man die Zahl 2 in Spalte1;Zeile1 . Um in der 2.Zeile die 3 zur 0 zu machen, benötigt man den Faktor 1,5. Subtrahiert man nämlich das 1,5-fache der 1.Zeile von der 2.Zeile, dann würde aus der 3 eine 0. Also speichert man anstelle der 3 eine 1,5. Der Rest der Zeile wird wie beim Gauß-Algorithmus verändert. Entsprechend muss man in der Spalte1;Zeile3 aus der 1 eine 0,5 machen, denn es muss das 0,5-fache der 1.Zeile von der 2.Zeile subtrahiert werden.

Zwischenergebnis: $\begin{pmatrix} 2 & 1 & 1 \\ 1{,}5 & 0{,}5 & 0{,}5 \\ 0{,}5 & 0{,}5 & 1{,}5 \end{pmatrix}$

Genauso wie beim Gauß-Algorithmus betrachtet man anschließend nur noch den (2,2)-Matrixanteil rechts unten. Ziel wäre, die 0,5 in Spalte2;Zeile3 zu 0 zu machen. Dies gelingt mit dem Faktor 1, den wir anstelle der 0,5 speichern. Die letzte verbleibende Zahl 1,5 wird ersetzt durch 1,5-1·0,5 = 1 .

Das Endergebnis der Umformungen ist: $\begin{pmatrix} 2 & 1 & 1 \\ 1{,}5 & 0{,}5 & 0{,}5 \\ 0{,}5 & 1 & 1 \end{pmatrix}$

In dieser Matrix steckt sowohl L als auch R. Bei L muss berücksichtigt werden, dass alle Diagonalelemente 1 sein sollen (obwohl dies hier nicht zu sehen ist). Damit erhält man als

Lösung: $L = \begin{pmatrix} 1 & 0 & 0 \\ 1{,}5 & 1 & 0 \\ 0{,}5 & 1 & 1 \end{pmatrix}$ und $R = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 0{,}5 & 0{,}5 \\ 0 & 0 & 1 \end{pmatrix}$

Man sieht, dass R die gleiche Matrix ist, die bei der normalen Zeilenstufenform entsteht !
L wird zusätzlich erzeugt durch Speichern der o.a. Faktoren anstelle der Nullen.

Ein positiver Nebeneffekt ist, dass man mit der LR-Zerlegung in einfacher Weise auch die Determinante der Matrix A berechnen kann (<u>Vorsicht</u>: Bei Pivotisierung/Zeilentausch kann sich das Vorzeichen ändern !):

Es gilt: <u>det(A) ist das Produkt der Diagonalenelemente von R (ohne Zeilenvertauschungen !)</u>
(Bei einer ungeraden Anzahl von Zeilenvertauschungen ändert sich das Vorzeichen)

Für das obige Beispiel ist dann det(A) = 2·0,5·1 = 1 .

Der Vorteil der LR-Zerlegung ist, dass man simultan LGSe lösen kann :

$$A \cdot \vec{x} = \vec{b} \Rightarrow (L \cdot R) \cdot \vec{x} = \vec{b} \Rightarrow L \cdot (R \cdot \vec{x}) = \vec{b} \Rightarrow L \cdot \vec{y} = \vec{b} \;\; mit \;\; \vec{y} = R \cdot \vec{x} \; .$$

Man löst daher zuerst $L \cdot \vec{y} = \vec{b}$ und anschließend $R \cdot \vec{x} = \vec{y}$ (beides sind einfache Berechnungen) .

Sei $\vec{b} = \begin{pmatrix} 4 \\ 2 \\ 3 \end{pmatrix}$ gesetzt. Dann verläuft die Rechnung für $L \cdot \vec{y} = \vec{b}$ folgendermaßen:

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 4 \\ 1{,}5 & 1 & 0 & 2 \\ 0{,}5 & 1 & 1 & 3 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -4 \\ 0 & 1 & 1 & 1 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 5 \end{array}\right) \Rightarrow \vec{y} = \begin{pmatrix} 4 \\ -4 \\ 5 \end{pmatrix}$$

Dann noch die Lösung für $R \cdot \vec{x} = \vec{y}$:

$$\left(\begin{array}{ccc|c} 2 & 1 & 1 & 4 \\ 0 & 0{,}5 & 0{,}5 & -4 \\ 0 & 0 & 1 & 5 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 2 & 0 & 0 & 12 \\ 0 & 1 & 1 & -8 \\ 0 & 0 & 1 & 5 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & -13 \\ 0 & 0 & 1 & 5 \end{array}\right) \Rightarrow \vec{x} = \begin{pmatrix} 6 \\ -13 \\ 5 \end{pmatrix}$$

Weitere Beispiele:

$$\underbrace{\begin{pmatrix} 2 & -1 & -3 \\ 6 & 1 & -10 \\ -2 & -7 & 8 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & -2 & 1 \end{pmatrix}}_{L} \cdot \underbrace{\begin{pmatrix} 2 & -1 & -3 \\ 0 & 4 & -1 \\ 0 & 0 & 3 \end{pmatrix}}_{R}$$

det(A) = 2·4·3 = 24

$$\underbrace{\begin{pmatrix} 2 & 3 & 5 \\ 6 & 10 & 17 \\ 8 & 14 & 28 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 4 & 2 & 1 \end{pmatrix}}_{L} \cdot \underbrace{\begin{pmatrix} 2 & 3 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 4 \end{pmatrix}}_{R}$$

det(A) = 2·1·4 = 8 .

<u>Achtung:</u>

Falls eine Pivotisierung notwendig sein sollte, so müssten die <u>Zeilenvertauschungen</u> berücksichtigt werden.

Dies kann durch Multiplikation mit <u>Permutationsmatrizen P</u> geschehen. Es gilt: $P \cdot A = L \cdot R$

Dies lässt sich umformen in $A = P^{-1} \cdot L \cdot R = (P^{-1} \cdot L) \cdot R = (P \cdot L) \cdot R$, da $P^{-1} = P$!

Für $A \cdot \vec{x} = \vec{b}$ folgt dann: $(P \cdot L) \cdot R \cdot \vec{x} = \vec{b}$ bzw. $(P \cdot L) \cdot \vec{y} = \vec{b}$ mit $R \cdot \vec{x} = \vec{y}$.

Im Gegensatz zu der obigen Rechnung verwendet man bei Zeilenvertauschungen also P · L statt L !

Dies wird später auch an einem Beispiel durchgerechnet.

Permutationsmatrizen P

Bei Permutationsmatrizen (**Vertauschungsmatrizen**) steht in jeder Zeile und jeder Spalte genau eine 1 , ansonsten Nullen.

(1) Multipliziert man eine Matrix A von <u>links</u> mit P, also PA, so tauscht man damit <u>Zeilen</u> .
p_{ik} = 1 bedeutet dabei, dass aus der k-ten Zeile die i-te Zeile wird. ze k → ze i

<u>Beispiele:</u>

$$P = E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix} \quad PA = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix} \text{ keine Veränderung durch P = E !}$$

Vertauscht man aber bei P die Zeilen 1 und 2, erzeugt man also $p_{12} = 1$:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix} \quad PA = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix} \text{ Vertauschung der Zeilen 1 und 2 bei A !}$$

(2) Multipliziert man eine Matrix A von <u>rechts</u> mit P, also AP, so tauscht man damit <u>Spalten</u> .
p_{ik} = 1 bedeutet dabei, dass aus der i-ten Spalte die k-te Spalte wird. sp k → sp i

<u>Beispiel:</u> Bei Verwendung der obigen Matrix P mit $p_{12} = 1$ entsteht folgendes:

$$\underbrace{\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{P} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 2 & 1 & 3 & 4 \\ 2 & 1 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix} .$$

Bei der LR-Zerlegung kann es zur Vertauschung <u>zweier Zeilen</u> von A kommen. Das nach der Vertauschung entstandene A berechnet man durch die Multiplikation **P · A** .

Für die LR-Zerlegung (mit Pivotisierung) des folgenden Beispiels $A = \begin{pmatrix} 2 & 1 & 1 \\ 3 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix}$ bedeutet das zunächst

eine Vertauschung der Zeilen 1 und 2.

Die entsprechende Permutationsmatrix ist $P = \begin{pmatrix} 0 & \boxed{1} & 0 \\ \boxed{1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Ansatz für die LR-Zerlegung der durch Zeilentausch erhaltenen Matrix PA :

$$\mathrm{PA} = \begin{pmatrix} \boxed{3} & 2 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \to \begin{pmatrix} 3 & 2 & 2 \\ 2/3 & \boxed{-1/3} & -1/3 \\ 1/3 & 1/3 & 4/3 \end{pmatrix} \to \underbrace{\begin{pmatrix} 3 & 2 & 2 \\ 2/3 & -1/3 & -1/3 \\ 1/3 & -1 & 1 \end{pmatrix}}_{L \setminus R}$$

Für A folgt dann:

$$A=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}\cdot\underset{L}{\begin{pmatrix}1&0&0\\2/3&1&0\\1/3&-1&1\end{pmatrix}}\cdot\underset{R}{\begin{pmatrix}3&2&2\\0&-1/3&-1/3\\0&0&1\end{pmatrix}}=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}\cdot\begin{pmatrix}3&2&2\\2&1&1\\1&1&2\end{pmatrix}=\begin{pmatrix}2&1&1\\3&2&2\\1&1&2\end{pmatrix}$$

Bei der Pivotisierung können unter Umständen mehrere Zeilenvertauschungen anfallen, so dass entsprechend viele Permutationsmatrizen $P_1, P_2, \ldots, P_n$ entstehen.
Die Gesamtpermutationsmatrix P ergibt sich dann zu $P = P_1 \cdot P_2 \cdot \ldots \cdot P_n$.

Ein Beispiel:

$$A=\begin{pmatrix}\boxed{24}&0&-12&-12\\6&6&18&0\\6&18&66&18\\-12&0&18&84\end{pmatrix}\rightarrow\begin{pmatrix}24&0&-12&-12\\1/4&\boxed{6}&21&3\\1/4&18&69&21\\-1/2&0&12&78\end{pmatrix}\quad P_1=E$$

Kein Tausch (P_1 überflüssig !)

$$\rightarrow\begin{pmatrix}24&0&-12&-12\\1/4&\boxed{18}&69&21\\1/4&6&21&3\\-1/2&0&12&78\end{pmatrix}\rightarrow\begin{pmatrix}24&0&-12&-12\\1/4&18&69&21\\1/4&1/3&\boxed{-2}&-4\\-1/2&0&12&78\end{pmatrix}\quad P_2=\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}$$

Tausch Zeilen 2 ; 3

$$\rightarrow\begin{pmatrix}24&0&-12&-12\\1/4&18&69&21\\-1/2&0&\boxed{12}&78\\1/4&1/3&-2&-4\end{pmatrix}\rightarrow\begin{pmatrix}24&0&-12&-12\\1/4&18&69&21\\-1/2&0&\boxed{12}&78\\1/4&1/3&-1/6&9\end{pmatrix}\quad P_3=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

Tausch Zeilen 3 ; 4

Die Zerlegungsmatrizen sind $L=\begin{pmatrix}1&0&0&0\\1/4&1&0&0\\-1/2&0&1&0\\1/4&1/3&-1/6&1\end{pmatrix}\quad R=\begin{pmatrix}24&0&-12&-12\\0&18&69&21\\0&0&12&78\\0&0&0&9\end{pmatrix}$

Die Determinante berechnet sich zu det(A) = 24·18·12·9 = 46656

<u>**Anmerkung: Bei einer ungeraden Anzahl von Zeilenvertauschungen kehrt sich das Vorzeichen von det um !**</u>

Multipliziert man L mit R, so erhält man $L\cdot R=\begin{pmatrix}24&0&-12&-12\\6&18&66&18\\-12&0&18&84\\6&6&18&0\end{pmatrix}$

Um zur ursprünglichen Matrix A zurückzukehren, müsste man erst die Zeilen 3;4 tauschen und dann 2;3 .
Die oben durchgeführten Vertauschungen werden also rückgängig gemacht.
Ebenso kann man aber $(P_2 \cdot P_3) \cdot (L \cdot R)$ berechnen, denn: $P = P_3P_2 \Rightarrow P_{-1} = P_2P_3$. Dies ergibt :

$$(P_2\cdot P_3)\cdot(L\cdot R)=\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\cdot\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\cdot\begin{pmatrix}24&0&-12&-12\\6&18&66&18\\-12&0&18&84\\6&6&18&0\end{pmatrix}=$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 24 & 0 & -12 & -12 \\ 6 & 18 & 66 & 18 \\ -12 & 0 & 18 & 84 \\ 6 & 6 & 18 & 0 \end{pmatrix} = \begin{pmatrix} 24 & 0 & -12 & -12 \\ 6 & 6 & 18 & 0 \\ 6 & 18 & 66 & 18 \\ -12 & 0 & 18 & 84 \end{pmatrix} = A$$

Will man nun mit dieser LR-Zerlegung das Problem $A \cdot \vec{x} = \vec{b}$ lösen, so berechnet man erst (wie bereits weiter oben bewiesen) $\vec{y}$ mittels $(P^{-1} \cdot L) \cdot \vec{y} = \vec{b} \Rightarrow (P_2 \cdot P_3 \cdot L) \cdot \vec{y} = \vec{b}$ und dann $R \cdot \vec{x} = \vec{y}$.

Beispiel: $\vec{b} = \begin{pmatrix} 36 \\ 18 \\ 24 \\ -96 \end{pmatrix}$. Mit $P^{-1} \cdot L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ 1/4 & 1/3 & -1/6 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1/3 & -1/6 & 1 \\ 1/4 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \end{pmatrix}$

und $(P^{-1} \cdot L) \cdot \vec{y} = \vec{b}$: $\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 36 \\ 1/4 & 1/3 & -1/6 & 1 & 18 \\ 1/4 & 1 & 0 & 0 & 24 \\ -1/2 & 0 & 1 & 0 & -96 \end{array} \right) \to \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 36 \\ 0 & 1 & 0 & 0 & 15 \\ 0 & 0 & 1 & 0 & -78 \\ 0 & 0 & 0 & 1 & -9 \end{array} \right) \Rightarrow \vec{y} = \begin{pmatrix} 36 \\ 15 \\ -78 \\ -9 \end{pmatrix}$

Aus $R \cdot \vec{x} = \vec{y}$ folgt: $\left(\begin{array}{cccc|c} 24 & 0 & -12 & -12 & 36 \\ 0 & 18 & 69 & 21 & 15 \\ 0 & 0 & 12 & 78 & -78 \\ 0 & 0 & 0 & 9 & -9 \end{array} \right) \to \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{array} \right)$.

Die Lösung des LGS $\begin{pmatrix} 24 & 0 & -12 & -12 \\ 6 & 6 & 18 & 0 \\ 6 & 18 & 66 & 18 \\ -12 & 0 & 18 & 84 \end{pmatrix} \cdot \vec{x} = \begin{pmatrix} 36 \\ 18 \\ 24 \\ -96 \end{pmatrix}$ ist demnach $\vec{x} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ -1 \end{pmatrix}$

<u>Weiteres Beispiel (Lösung mit LR-Zerlegung; mit Zeilentausch):</u>

$A = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 5 & 4 \end{pmatrix}$ $\vec{b} = \begin{pmatrix} 0 \\ 4 \\ 15 \end{pmatrix}$. Die Rechnung liefert $L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/4 & 3/4 & 1 \end{pmatrix}$ $R = \begin{pmatrix} 4 & 1 & 0 \\ 0 & 5 & 4 \\ 0 & 0 & -2 \end{pmatrix}$ $P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

Aus $(P^{-1} \cdot L) \cdot \vec{y} = \vec{b}$ folgt: $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/4 & 3/4 & 1 \end{pmatrix} \cdot \vec{y} = \begin{pmatrix} 0 \\ 4 \\ 15 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 3/4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \vec{y} = \begin{pmatrix} 0 \\ 4 \\ 15 \end{pmatrix}$

$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 1/4 & 3/4 & 1 & 4 \\ 0 & 1 & 0 & 15 \end{array} \right) \to \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 15 \\ 0 & 1 & 1 & -29/4 \end{array} \right)$, also $\vec{y} = \begin{pmatrix} 0 \\ 15 \\ -29/4 \end{pmatrix}$.

Aus $R \cdot \vec{x} = \vec{y}$ folgt: $\left(\begin{array}{ccc|c} 4 & 1 & 0 & 0 \\ 0 & 5 & 4 & 15 \\ 0 & 0 & -2 & -29/4 \end{array} \right) \to \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1/40 \\ 0 & 1 & 0 & 1/10 \\ 0 & 0 & 1 & 29/8 \end{array} \right)$. Lösung: $\vec{x} = \begin{pmatrix} -1/40 \\ 1/10 \\ 29/8 \end{pmatrix}$

<u>Besondere Matrizen:</u>

1) Eine Matrix mit einer **<u>0-Spalte</u>** soll zeigen, dass eine Zeilenstufenform zuweilen schwer zu ermitteln ist:

```
0  0  0  0  1  2          0  0  0  0  1  2
0  2  3  0  5  6          0  2  3  0  5  6
0  0  0  0  0  0          0  0  1  0  3  4
0  0  1  0  3  4          0  0  0  0  0  0
```

Der einfache Gauß-Algorithmus (ref) versagt hier, weil die erste Zeile (mit 4 Nullen) unberücksichtigt bleibt.
Es entsteht somit keine Zeilenstufenform !
Ordnet man jedoch die Zeilen der Ausgangsmatrix nach den am weitesten links stehenden Nicht-Nullen, so ergibt sich die gewünschte Zeilenstufenform !

```
0  2  3  0  5  6
0  0  1  0  3  4
0  0  0  0  1  2
0  0  0  0  0  0
```