6.2.3 Das Gaußsche Eliminationsverfahren

lin. Gleich Syst.

m Gleichungen

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases}$$

rechte Seite

n Variable

Kurzschreibweise: $Ax = b$ $\quad A = \begin{pmatrix} a_{11} & \dots & \\ & \ddots & \\ \dots & & a_{mn} \end{pmatrix}$ $\quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$

$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$

Lösung: Ein Vektor x der $Ax = b$ erfüllt

Lösungsraum = Menge aller Lösungen

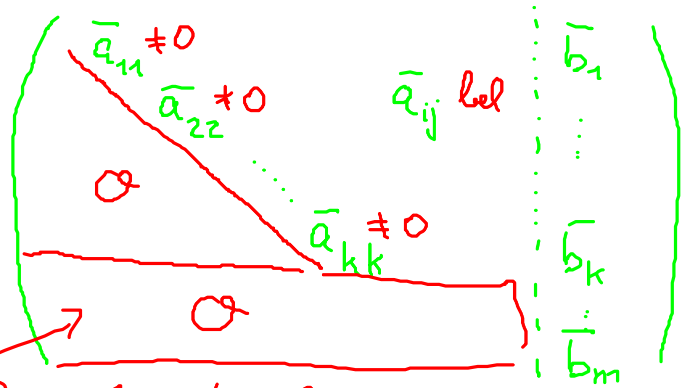
Der Lösungsraum ändert sich nicht wenn

elementare Umformungen
- man eine Gleichung mit Zahl $\neq 0$ multipliziert
- man ein Vielfaches einer Gleichung zu einer anderen addiert

Vertauschungen
- man Zeilen (Gleichungen) vertauscht oder Spalten vertauscht

ohne Beweis (siehe Skript, VL lineare Algebra)

SATZ: Jedes lin. Gleich Syst $Ax=b$ kann durch elem. Umformungen und Vertauschungen auf <u>Dreiecksform</u> gebracht werden, d.h. die <u>erweiterte Matrix</u> $(A \vdots b)$ hat dann folgende Form:



falls hier noch Zeilen übrig bleiben (also $k < m$)

<u>Bsp</u>: 1. $m \le n$

$$x_1 + x_2 - x_3 = 2$$
$$x_1 \qquad + x_3 = -1$$

$$(A \vdots b) = \left(\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 1 & 0 & 1 & -1 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & -1 & 2 & -3 \end{array}\right)$$

1. Gleich. von 2. abziehen

Dreiecksform mit $k = 2 = m$

$k = 2 < n \quad x_3 = 0$

$-x_2 + 0 = -3 \Rightarrow x_2 = 3$

$x_1 + 3 + 0 = 2 \Rightarrow x_1 = -3$

2.

$$(A \vdots b) = \left(\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 1 & 1 & 1 & -1 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 0 & 2 & -3 \end{array}\right)$$

$\longrightarrow$ Vertauschung Spalte 2 und 3

$$\begin{array}{c} \\ \\ \\ \end{array} \left(\begin{array}{ccc|c} \overset{\neq 0}{1} & -1 & 1 & 2 \\ 0 & \overset{\neq 0}{2} & 0 & -3 \end{array}\right)$$

$x_1 \quad x_3 \quad x_2$ Dreiecksform

3. $m > n$:

$$x_1 + x_2 = 1$$
$$x_1 - x_2 = 0$$
$$x_1 + 2x_2 = 2$$

$k = 2 < n = 3$

wähle $x_2 = c$ (letzte Spalte)

$2x_3 + 0x_2 = -3$

$\Rightarrow x_3 = -\frac{3}{2}$

$\Rightarrow x_1 + \frac{3}{2} + c = 2 \Rightarrow x_1 = \frac{1}{2} - c$

$$(A \vdots b) = \left(\begin{array}{cc|c} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 2 & 2 \end{array}\right) \longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 1 & 1 \end{array}\right)$$

1. Gleich von anderen abziehen

$\longrightarrow$

$\frac{1}{2}$ faches von Gleich 2 zu Gleich 3 addieren

$$\left(\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & -2 & -1 \\ \boxed{0} & \boxed{0} & \frac{1}{2} \end{array}\right)$$

$m = 3$

$k = 2$

$\bar{b}_3 \neq 0$

$\Rightarrow$ unlösbar

letzte Gleichung sagt $\underbrace{0 \cdot x_1 + 0 \cdot x_2 = \frac{1}{2}}_{\text{nicht erfüllbar}}$

<u>Beweis:</u> Angabe eines Algo, der dies leistet

Algo im Struktogramm

Setze $(\tilde{A} \vdots \tilde{b}) := (A \vdots b)$ und $i := 1$

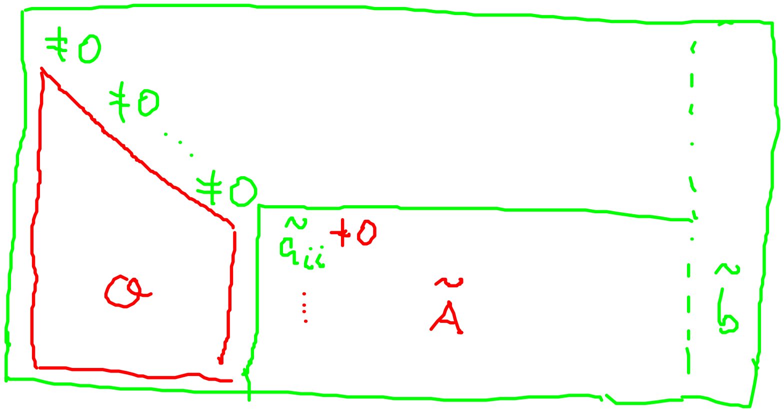
$\tilde{A} \neq 0$ (Nullmatrix) und $i \leq \min\{m, n\}$

> Ändere $(\tilde{A} \vdots \tilde{b})$ durch Vertauschungen bis $\tilde{a}_{ii} \neq 0$. {Dieses $\tilde{a}_{ii}$ heißt <u>Pivotelement</u>}
>
> Addiere zu allen Zeilen l mit $\tilde{a}_{li} \neq 0$ ein geeignetes Vielfaches c von Zeile i so dass $\tilde{a}_{li} + c \cdot \tilde{a}_{ii} = 0$
> {Dies nennt man <u>Pivotisierung</u>
>
> $i := i+1$;
>
> $i \leq \min\{n, m\}$
>
> true
>
> Lasse von $(\tilde{A} \vdots \tilde{b})$ die Zeile und Spalte mit Index i weg und bezeichne die entstehende Matrix wieder mit $(\tilde{A}, \tilde{b})$

$\tilde{A}$ ist die verbleibende Restmatrix

Alg. arbeitet korrekt wegen folgender Schleifeninvariante:

Bei jedem Eintritt in die Schleife hat die erweiterte Matrix folgende Form:



□

SATZ: Lösungskriterien für lin. Gleich. Syst.

Sei $Ax=b$ gegeben und seien $(\bar A \vdots \bar b)$ und k die durch das Gaußsche Verfahren gegebenen Größen. Dann gilt

a) $Ax=b$ hat genau dann (mindestens) eine Lösung wenn im Falle $\underbrace{k<m \quad \bar b_i = 0 \text{ für } i=k+1,\dots,m}_{(*)}$

b) In diesem Fall (also wenn (*) gilt) erhält man alle Lösungen, indem man (für $k<n$) $x_{k+1}, x_{k+2},\dots,x_n$ beliebig wählt und die anderen Variablen $x_1\, x_2 \dots x_k$ aus der Dreiecksform von unten nach oben berechnet

$$\bar a_{11}x_1 + \underbrace{\bar a_{12}x_2+\dots+\bar a_{1n}x_n}_{\text{bekannt}} = \bar b_1 - (\bar a_{1,k+1}x_{k+1}+\dots+\bar a_{1,n}x_n)$$

oben

$$\bar a_{22}x_2 \underbrace{+\dots+\bar a_{2k}x_k}_{\text{bekannt}} = \bar b_2 - (\bar a_{2,k+1}x_{k+1}+\dots+\bar a_{k,n}x_n)$$

$$\ddots$$

$$\bar a_{kk}x_k = \bar b_k - (\bar a_{k,k+1}\underset{\uparrow}{x_{k+1}}+\dots+\bar a_{kn}\underset{\uparrow}{x_n})$$

unten

Werte beliebig gewählt

Beweis: zu a)

1. $\boxed{\text{Bedingung } \bar{b}_i = 0 \text{ für } i > k \text{ ist notwendig für die Lösbarkeit}}$

denn: i-te Gleichung sagt $\underbrace{0x_1 + \dots + 0x_n}_{0} = \bar{b}_i$

lösbar $\Rightarrow \bar{b}_i = 0$

2. $\boxed{\text{Bedingung } \bar{b}_i = 0 \text{ für } i > k \text{ ist hinreichend für die Lösbarkeit}}$

Dann funktioniert das Ausrechnen von oben nach unten (ggf. durch beliebige Wahl von $x_{k+1} \dots x_n$ falls $k < n$), da Diagonalelemente $\bar{a}_{kk}, \bar{a}_{k-1,k-1}, \dots, \bar{a}_1$ alle $\neq 0$ sind und daher $x_k, x_{k-1}, \dots, x_1$ sukkzessive ausgerechnet werden können

3. $\boxed{\text{Sobald } x_{k+1}, \dots, x_n \text{ (falls } k < n) \text{ gewählt sind, ergeben sich die Werte von } x_k, x_{k-1}, \dots, x_1 \text{ eindeutig}}$

klar durch Rückwärtsausrechnen

4. $\boxed{\text{Durch beliebige Wahl von } x_{k+1} \dots x_n \text{ und Rückwärts ausrechnen bekommt man alle Lösungen}}$

Sei $\bar{x}$ eine Lösung. Setze $x_{k+1} := \bar{x}_{k+1}, \dots, x_n := \bar{x}_n$

3. $\Rightarrow$ die anderen Komponenten von x ergeben sich eindeutig, das gilt auch für $\bar{x}$

$$\Rightarrow \quad x_1 = \bar{x}_1,\ x_2 = \bar{x}_2 \ \dots\ x_k = \bar{x}_k \quad \Rightarrow \quad x = \bar{x}$$

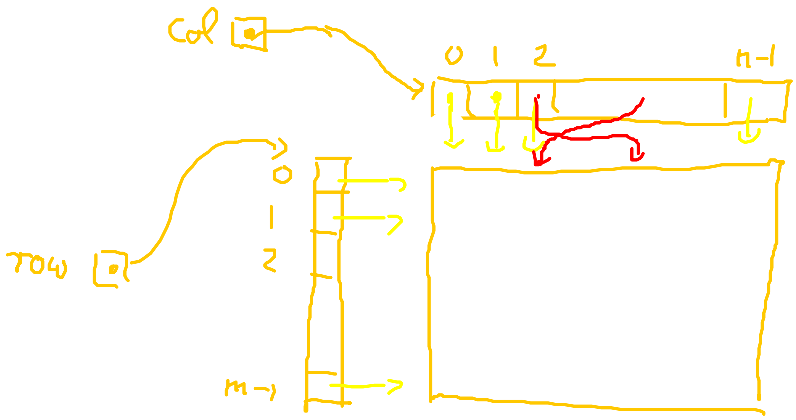
6.2.5

Java Klasse zur Lösung lin. Gleich Syst

Vertauschungen von Zeilen bzw. Spalten wird nicht durchgeführt (zu viel Aufwand)

Dafür: merken der Vertauschungen in zwei Hilfsarrays

```
int[] row;   // Zeilenindex
int[] col;   // Spaltenindex
```



← Vertauschung von 2 Spalten
col[i] = j   col[j] = i

double[][] a

anfangs   row[i] = i   col[j] = j

dann immer Zugriff auf aktuelle Element an Position i, j über   a[row[i]][col[j]]

Code der Klasse ansehen

speziell: - Pivotoperation

- von unten nach oben ausrechnen