Mathe für Informatiker II

Mitschrift der Vorlesung von Christian Bender Sommersemester 17

Lukas Koschorke

10. Mai 2017

Vorwort

Hallo zusammen, ich versuche mein Skript immer aktuell zu halten und wenn ich es eh schon abtippe, kann ich es auch noch zusätzlich auf StudyDrive hochladen. Ich kann leider keine Garantie geben, dass es immer den kompletten Stoff aller Vorlesungen beinhaltet, auch wenn ich mein Bestes dafür gebe. Wenn euch Fehler auffallen, haut sie einfach in die Kommentare von StudyDrive.

Änderungen

28.04.2017:

- Vorlesung vom 28.04.2017 hinzugefügt.
- Darstellung der Brüche verbessert.
- Section Änderungen eingefügt
- (L) verbessert

02.05.17

- Aufzählungen verändert

05.05.17

- Vorlesung 05.05.17 hinzugefügt

10.05.17

- Vorlesung 10.05.17 hinzugefügt
- kleine Verbesserungen

Inhaltsverzeichnis

1 Lineare Gleichungssysteme und der Gaußalgorithmus

1.1 Beispiel (PageRank)

Ein Nutzer surft auf den vorhandenen Seiten des Internets S_1,...,S_N. Er beginnt auf irgendeiner Seite und folgt typischerweise einem der Links. Er kann aber auch auf eine beliebige Seite springen.
Zur Modellierung sei $0 < d < 1$ (Damping-Faktor, typischerweise $d = 0,85$). Auf einer Seite angekommen, folgt der Nutzer mit Wahrscheinlichkeit d einem rein zufällig ausgewähltem Link, mit Wahrscheinlichkeit $1 - d$ springt er auf eine rein zufällig gewählte Seite. (Konvention: Falls die Seite keinen Outlink hat, so wählt man rein zufällig eine Seite aus)
„Auf lange Sicht“ (d.h. wenn die Anzahl der Surfschritte gegen unenlich strebt, ist die Wahrscheinlichkeit p_i, sich auf der Seite S_i zu befinden, beschrieben durch

$$p_i = \frac{1-d}{N} + d\sum_{j=1}^{N} a_{ij}p_j, i = 1, ..., N \tag{1.1}$$

wobei:

$$a_{ij} = \begin{cases} \frac{1}{c_j} & \text{, falls Seite } S_j \text{ auf Seite } S_i \text{ verlinkt} \\ \frac{1}{N} & \text{, falls } S_j \text{ keinen Outlink hat} \\ 0 & \text{, sonst} \end{cases}$$

und c_j die Anzahl an Seiten angibt, auf die S_j verlinkt (siehe dazu das „Das Kapital der Markovketten“ in der MFI3).
p_i wird <u>PageRank</u> der Seite S_i genannt.

1.1 ist ein System von N linearen Gleichungen zu N Unbekannten.

In einem Netz mit 3 Seiten sei die Verlinkung schematisch dargestellt durch:

$$S_1 \leftrightarrows S_3, S_2 \to S_1, S_3 \to S_2$$

Hier ist also $c_3 = 2, c_1 = c_2 = 1$, d.h. 1.1 wird im Spezialfall zu:

$$\begin{array}{rcrcrcl}
p_1 & - & dp_2 & - & \frac{d}{2}p_3 & = & \frac{1-d}{3} \quad (I) \\
 & & p_2 & - & \frac{d}{2}p_3 & = & \frac{1-d}{3} \quad (II) \\
- dp_1 & & & + & p_3 & = & \frac{1-d}{3} \quad (III)
\end{array}$$

Indem man (I) durch $(\tilde{I}) = (I) + d(II) + \frac{1}{d}(III)$ersetzt, erhält man die äquivalente Form:

$$\begin{array}{rcrcl}
- dp_1 & + & p_3 & = & \frac{1-d}{3} \quad (III) \\
p_2 & - & \frac{d}{2}p_3 & = & \frac{1-d}{3} \quad (II) \\
 & & (\frac{1}{d} - \frac{d^2}{2} - \frac{d}{2})p_3 & = & \frac{1-d}{3}(1 + \frac{1}{d} + d) \quad (\tilde{I})
\end{array}$$

Nun haben wir das System in „Zeilenstufenform“ und man kann die Lösung direkt ablesen $(0 < d < 1)$:

$$\begin{aligned}
p_3 &= \frac{1-d}{3} \frac{d^2 + d + 1}{1 - \frac{d^3}{2} - \frac{d^2}{2}} \\
p_2 &= \frac{1-d}{3} + \frac{d}{2}p_3 \\
p_1 &= \frac{1}{d}(p_3 - \frac{1-d}{3})
\end{aligned}$$

Da $p_3 > \frac{1-d}{3}$ folgt $p_1 > 0$ und $p_2 > 0$.
Da ferner $p_1 + p_2 + p_3 = 1$ (Nachrechnen), kann der PageRank tatsächlich als Wahrscheinlichkeit interpretiert werden.
Für $d = 0,85$ ist:
$p_1 \approx 0,397$
$p_2 \approx 0,215$
$p_3 \approx 0,388$
Betrachten wir den „Grenzfall“ $d = 1$, in dem nur Links zum Surfen verwendet werden,

dann:

$$\begin{array}{rrrrrrrrl} & p_1 & - & p_2 & - & \frac{1}{2}p_3 & = & 0 & (I) \\ & & & p_2 & - & \frac{1}{2}p_3 & = & 0 & (II) \\ - & p_1 & & & + & p_3 & = & 0 & (III) \end{array}$$

und wie zuvor:

$$\begin{array}{rrrrrrrrl} - & p_1 & & & + & p_3 & = & 0 & (III) \\ & & & p_2 & - & \frac{1}{2}p_3 & = & 0 & (II) \\ & & & & & 0 & = & 0 & (\tilde{I}) \end{array}$$

Hier lösen alle p_1, p_2, p_3 der Form

$$p_1 = p_3, p_2 = \frac{1}{2}p_3,\ p_3 \in \mathbb{R}$$

das System 1.1. Fordert man zusätzlich die Wahrscheinlichkeitsbedingung $p_1+p_2+p_3 = 1$, so ist die eindeutige Lösung:

$$p_3 = p_1 = 0,4,\ p_2 = 0,2$$

Zur Bedeutung des Damping-Faktor:

a) Die Wahl $0 \leq d < 1$ sichert, das 1.1 genau eine Lösung hat und, dass diese Lösung positive Einträge hat.
Dies werden wir im Zusammenhang mit „Eigenwertproblemen“ besser verstehen.

b) Im realistischen Problem ist N sehr groß ($N > 20 Mio.$). Dann müssen nummerische Verfahren zur Approximation des PageRank herangezogen werden. Wir werden sehen, dass die Wahl von d eine wichtige Rolle bei der Konvergenzgeschwindigkeit derartiger Verfahren spielt.

Problem:

Gegeben sei ein System von $m \in \mathbb{N}$ Gleichungen mit $n \in \mathbb{N}$ Unbekannten:

$$\left.\begin{array}{ccccccc} a_{11}x_1 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ \vdots & & \ddots & & \vdots & & \vdots \\ a_{m1}x_1 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array}\right\} (L)$$

Hierbei seien die Koeffizienten $a_{ij} \in \mathbb{R}$ (mit doppelter Indexschreibweise) für $i = 1, ..., m$, $j = 1, ..., n$, sowie die rechte Seite $b_i \in \mathbb{R}, i = 1, ..., m$, gegeben.
Gesucht sind alle n-Tupel $(x_1, ..., x_n)$ von reellen Zahlen, die das System (L) lösen.
Zur übersichtlichen Notation schreibt man die Koeffizienten als Rechteckschema, das $m \times n$-Matrix genannt wird

$$A := (a_{ij})_{i=1,...,m,j=1,...,n} := \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

und die $b_i's$ und $x_j's$ als **Spaltenvektor**

$$x := \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, b := \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

Definiert man nun ein Produkt zwischen $m \times n$-Matrix und Spaltenvektor der Höhe n, das einen Spaltenvektor der Höhe m ergibt, durch:

$$Ax := \begin{pmatrix} \sum_{j=1}^{n} a_{1j}x_1 \\ \vdots \\ \sum_{j=1}^{n} a_{mj}x_j \end{pmatrix}$$

dann kann (L) umgeschrieben werden in die Form

$$Ax = b \text{ (L')}$$

als Gleichung von Spaltenvektoren der Höhe m.
Wir bezeichnen die Menge aller Spaltenvektoren der Höhe m mit reellen Einträgen als $\mathbb{R}^n$ und die Menge aller $m \times n$-Matrizen mit reellen Einträgen als $\mathbb{R}^{m \times n}$.Dann ist die Lösungsmenge von (L) gegeben durch:

$$\text{Lös}(A, b) := \{x \in \mathbb{R}^n | Ax = b\}$$

1.2 Definition Zeilenstufenform

Wir sagen, eine $m \times n$-Matrix$A = (a_{ij})_{i=1,...,m,j=1,...,n}$ hat **Zeilenstufenform**, falls:

1. Es gibt ein $0 \leq r \leq m$, so dass in den Zeilen mit Index 1 bis r nicht alle Einträge gleich 0 sind und in der Zeile(r+1) bis m alle Einträge gleich 0 sind
und

2. es gelte

$$j_1 < ... < j_r$$

wobei für jedes $1 \leq i \leq r$
$$j_i = min\{j | a_{ij} \neq 0\}$$
der niedrigste Spaltenindex angibt, in dem in der i-ten Zeile ein von 0 verschiedener Eintrag steht.
Die Zahl r wird **Zeilenrang** von A genannt.
Die Einträge $a_{i,j_i}, i = 1, ..., r$, werden **Pivots** genannt.
Offenbar gilt: $r \leq min\{m, n\}$

$$A = \begin{pmatrix} (*) & & & \\ 0 & (*) & & \\ 0 & 0 & \ddots & \\ 0 & 0 & 0 & (*) \end{pmatrix}$$

Matrix in Zeilenstufenform: Die mit $(*)$ gekennzeichneten Einträge sind die von 0 verschiedenen Pivots, die Einträge unterhalb der „Stufenlinie" sind gleich 0, die übrigen Einträge sind beliebig.
Im Beispiel 1.1 hat die Koeffizientenmatrix zum System $(III), (II), (\tilde{I})$ Zeilenstufenform, nämlich:

$$\begin{pmatrix} -d & 0 & 1 \\ 0 & 1 & -\frac{d}{2} \\ 0 & 0 & (\frac{1}{2} - \frac{d^2}{2} - \frac{d}{2}) \end{pmatrix}$$

Fall 1:

Löse $Ax = b$, wobei A Zeilenstufenform hat.

a) Durch Umordnung der Spalten und damit Umnummerierung der Unbekannten, kann man stets voraussetzen, dass die Pivots in den ersten r Spalten stehen, d.h $j_i = i$ für $1 \leq i \leq r$.
Beispiel:

$$\underbrace{\begin{pmatrix} 0 & 2 & 0 & 4 & 0 & 5 \\ 0 & 0 & 1 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{j_1=2, j_2=3, j_3=5} \rightsquigarrow \begin{pmatrix} 2 & 0 & 0 & 5 & 0 & 4 \\ 0 & 1 & 1 & 0 & 0 & 3 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

b) Wir betrachten die **erweiterte Koeffizientenmatrix** zu $Ax = b$ (A beliebige $m \times n$-Matrix).

$$(A, b) = \begin{pmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{pmatrix}$$

, bei der die rechte Seite b als (n+1)-te Spalte an die Koeffizientenmatrix angehängt wird. Ist A in Zeilenstufenform mit Pivots in den ersten r Spalten, so

$$(A,b) = \begin{pmatrix} a_{11} & & & b_1 \\ & \ddots & & \vdots \\ & & a_{rr} & b_r \\ & & & b_{r+1} \\ & & & \vdots \\ & & & b_m \end{pmatrix}$$

1.3 Algorithmus (Rückwärtseinsetzen)

Input: $m \times n$-Matrix A in Zeilenstufenform mit Zeilenrang r $(0 \leq r \leq m)$ und Pivots in den ersten r Spalten, $b \in \mathbb{R}^m$.

Output: „es existiert keine Lösung“ oder eine bijektive Abbildung:

$$\Phi_{A,b} : \mathbb{R}^{n-r} \to \text{Lös}(A,b)$$

1. Falls $b_i \neq 0$ für ein $i = r+1, ..., m$, so gebe „keine Lösung“ als Output aus.

2.1 Andernfalls setze in Abhängigkeit von $\lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{n-r} \end{pmatrix}$

$$x_{r+1}(\lambda) := \lambda_1, ..., x_n(\lambda) := \lambda_{n-r}$$

und für $i = r, ..., 1$ rekursiv

$$x_i := \frac{1}{a_{ij}}(b_i - \sum_{j=i+1}^{n} a_{ij}x_j(\lambda))$$

2.2 Gebe $\Phi_{A,b} : \mathbb{R}^{n-r} \to \text{Lös}(A,b), \lambda \mapsto x(\lambda) := \begin{pmatrix} x_1(\lambda) \\ \vdots \\ x_n(\lambda) \end{pmatrix}$ aus.

Bemerkung:

Ist $r = n$, so ist $\mathbb{R}^{n-r} = \mathbb{R}^0 := \{0\}$ eineindeutig und das Rückwärtseinsetzen liefert genau eine Lösung für $Ax = b$

Beweis der Korrektheit des Algorithmus:

Ist $b_i \neq 0$ für ein $i = r+1, ..., m$, so lautet die i-te Gleichung

$$0 * x_1 + ... + 0 * x_n = b_i \neq 0$$

und kann also für kein $x \in \mathbb{R}^n$ erfüllt sein. Andernfalls sind die Gleichungen r+1 bis m immer erfüllt (0=0).
Für $i = 1, ..., r$ lautet die i-te Gleichung

$$a_{ii}x_i + a_{i,i+1}x_{i+1} + ... + a_{in}x_n = b_i$$

und ist wegen$a_{ii} \neq 0$ äquivalent zu:

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

Also: $x(\lambda) \in \text{Lös}(A, b)$ für alle $\lambda \in \mathbb{R}^{n-r}$
Zur Bijektivität:

Injektiv ist klar, da $x(\lambda) = \begin{pmatrix} x_1(\lambda) \\ \vdots \\ x_r(\lambda) \\ \lambda_1 \\ \vdots \\ \lambda_{n-r} \end{pmatrix}$
und also $x(\lambda) \neq x(\tilde{\lambda})$ für $\lambda \neq \tilde{\lambda}$.
Surjektiv: Sei $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \text{Lös}(A, b)$. Setze $\lambda = \begin{pmatrix} x_{i+1} \\ \vdots \\ x_n \end{pmatrix}$. Dann:

$$x_{r+i}(\lambda) = \lambda_i = x_{r+i} \forall i = 1, ..., n-r$$

Für $i = r, ..., 1$ folgt induktiv (Induktionsannahme: $x_j(\lambda) = x_j \forall j \geq i+1$):

$$x_1(\lambda) = \frac{1}{a_{ii}}(b_i - \sum_{j=i+1}^{n} a_{ij}x_j(\lambda)) = \frac{1}{a_{ij}}(b_i - \sum_{j=i+1}^{n} a_{ij}x_j) = x_i \square$$

Fall 2:

Löse $Ax = b$, wobei A in allgemeiner Form.
Strategie: Finde ein Äquivalentes System $\tilde{A}x = \tilde{b}$ (also mit $\text{Lös}(A, b)$= $\text{Lös}(\tilde{A}, \tilde{b})$), wobei $\tilde{A}$ Zeilenstufenform hat.
Dazu: Unter **elementaren Zeilenumformungen** einer Matrix verstehen wir (vorläufig):

1) Das Vertauschen von zwei Zeilen

2) Die Addition des λ-fachen der i-ten Zeile zur j-ten Zeile($i \neq j, \lambda \in \mathbb{R} \backslash \{0\}$)

1.4 Satz

Sei (A,b) die erweiterte Koeffizientenmatrix zu $Ax = b$ und $(\tilde{A}, \tilde{b})$ gehe aus(A,b) durch endlich viele Zeilenumformungen hervor. Dann:

$$\text{Lös}(A, b) = \text{Lös}(\tilde{A}, \tilde{b})$$

Beweis:

Es reicht zu zeigen, dass sich die Lösungsmenge bei einer elementaren Zeilenumformung nicht ändert.

Typ1: Da alle Gleichungen in (L) simultan erfüllt sein müssen, ist die Reihenfolge der Gleichungen egal.

Typ2: Da nur die Zahlen i und j betroffen sind, reicht es zu zeigen, dass die Systeme

$$\left.\begin{array}{ccccccc} a_{i1}x_1 & + & \cdots & + & a_{in}x_n & = & b_i \\ a_{j1}x_1 & + & \cdots & + & a_{jn}x_n & = & b_j \end{array}\right\} (*)$$

und

$$\left.\begin{array}{ccccccc} a_{i1}x_1 & + & \cdots & + & a_{in}x_n & = & b_i \\ (a_{j1} + \lambda a_{i1})x_1 & + & \cdots & + & (a_{jn} + \lambda a_{in})x_n & = & b_j + \lambda b_i \end{array}\right\} (**)$$

Löst $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$ das System $(*)$, so auch die erste Gleichung in $(**)$ und durch Addition des λ-fachen der ersten Gleichung in $(*)$ zur zweiten auch die zweite Gleichung in $(**)$.
Umgekehrt schließt man analog durch Subtraktion des λ-fachen der 1.Gleichung in $(**)$ von der zweiten. $\square$

1.5 Satz

Jede $m \times n$-Matrix kann durch endlich viele elementare Zeilenumformungen in Zeilenstufenform Überführt werden. Zum Beweis geben wir einen Algorithmus an, der das Geforderte leistet:

1.6 Algorithmus

Input: $m \times n$-Matrix A

Output: $m \times n$-Matrix in Zeilenstufenform

1. setze $A_1 = A$, $p_1 = m, q_1 = n$

2. Wende so lange die Funktion „Zeilenstufenschritt“ an, d.h. setzte $(A_k, p_k, q_k) :=$ Zeilenstufenschritt$(A_{k-1}, p_{k-1}, q_{k-1})$ bis $\min(p_k, q_k) = 0$

3. Nenne den Abbruchindex k_0 und gebeA_{k_0} aus.

Hierbei ist:

Zeilenstufenschritt

$$(\mathbb{R}^{m\times n} \times \{0, \cdots, m\} \times \{0, \cdots, n\} \to \mathbb{R}^{m\times n} \times \{0, \cdots, m\} \times \{0, \cdots, n\})$$

Wie folgt definiert:

Zeilenstufenschritt$(A, p, q) := (\tilde{A}, \tilde{p}, \tilde{q})$, wobei:

1. Sei $B = (b_{ij})_{i=1,\cdots,p,j=1,\cdots,q}$ mit $b_{ij} := a_{m-p+i,n-q+j}$
Veranschaulichung:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1,n-p} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m-q,1} & \cdots & a_{m-q,n-p} & \cdots & a_{m-q,n} \\ \vdots & \ddots & \vdots & & B \\ a_{m1} & \cdots & a_{m,n-p} & & \end{pmatrix}$$

2. Ist $b_{ij} = 0$ für alle $i = 1, \cdots, p$ und $j = 1, \cdots, q$, setze $\tilde{p} = \tilde{q} = 0$ und $\tilde{A} = A$

3.1 Andernfalls seien: $j_1 = min\{j | b_{ij} \neq 0$ für ein $i = 1, \cdots, p\}$ (die Spalte von B mit minimalem Index, in der ein von 0 verschiedener Eintrag ist.)
und $i_1 = min\{i | b_{ij} \neq 0\}$ (die Zeile mit minimalem Index, in der j_1-ten Spalte ein von 0 verschiedener Eintrag steht)

3.2 Setze $\tilde{B} = (\tilde{b}_{ij})_{i=1,\cdots,p,j=1,\cdots,q}$, wobei $\tilde{b}_{1j} := b_{i_1,j} \forall j = 1, \cdots, q$ und für $i = 2, \cdots, p$ und $j = 1, \cdots, q$:

$$\tilde{b}_{ij} = \begin{cases} -\frac{b_{ij_1}}{\tilde{b}_{1j_1}} & \tilde{b}_{1j} & + & b_{ij} & i \neq i_1 \\ -\frac{b_{1j_1}}{\tilde{b}_{1,j_1}} & \tilde{b}_{1j} & + & b_{1j} & i = i_1 \end{cases}$$

(Hier wird erst die i_1-te mit der ersten Zeile vertauscht und dann ein geeignetes Vielfaches der neuen ersten Zeile zu den anderen Zeilen addiert. $\tilde{B}$ geht also durch elementare Zeilenumformungen aus B hervor.)

3.3 Setze $\tilde{p} = p - 1, q = \tilde{q} - j_1$

$$\tilde{A} = \begin{pmatrix} a_{11} & \cdots & a_{1,n-p} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m-q,1} & \cdots & a_{m-q,n-p} & \cdots & a_{m-q,n} \\ \vdots & \ddots & \vdots & & \tilde{B} \\ a_{m1} & \cdots & a_{m,n-p} & & \end{pmatrix}$$

Beachte, die Matrix $\tilde{B}$ hat die Form

$$\tilde{B} = \begin{pmatrix} 0 & \cdots & 0 & b_{i_1,j_1} & * & \cdots & * \\ \vdots & \ddots & \vdots & 0 & \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & * & \cdots & * \end{pmatrix} * : \text{belibige Einträge, } b_{i_1,j_1} \neq 0$$

1.7 Beispiel

$$A = \begin{pmatrix} 0 & 0 & 1 & 2 & 9 \\ 0 & 3 & 4 & 5 & 9 \\ 0 & 6 & 7 & 8 & 9 \\ 0 & 9 & 9 & 9 & 9 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 & 3 & 4 & 5 & 9 \\ 0 & 0 & 1 & 2 & 9 \\ 0 & 6 & 7 & 8 & 9 \\ 0 & 9 & 9 & 9 & 9 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 & 3 & 4 & 5 & 9 \\ 0 & 0 & 1 & 2 & 9 \\ 0 & 0 & -1 & -2 & -9 \\ 0 & 0 & -3 & -6 & -18 \end{pmatrix} = A_2$$

$$\rightsquigarrow \begin{pmatrix} 0 & 3 & 4 & 5 & 9 \\ 0 & 0 & 1 & 2 & 9 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{pmatrix} = A_3 \rightsquigarrow \begin{pmatrix} 0 & 3 & 4 & 5 & 9 \\ 0 & 0 & 1 & 2 & 9 \\ 0 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = A_4$$

Wir zeigen: Algorithmus 1.6 führt jede $m \times n$-Matrix A mit endlich vielen elementaren Zeilenumformungen in Zeilenstufenform über.

Beweis: Da $min(p_k, q_k) < min(p_{k-1}, q_{k-1})$, bricht der Algorithmus nach endlich vielen Schritten ab. Ist $A_1 = A$ die Nullmatrix, so ist A bereits in Zeilenstufenform und der Algorithmus liefert A als Output. Andernfalls entsteht A_2 aus A_1 durch endlich viele elementare Zeilenumformungen und hat die Form

$$A_2 = \begin{pmatrix} 0 & \cdots & 0 & (*) & * & \cdots & * \\ \vdots & \ddots & \vdots & 0 & & & \\ \vdots & \ddots & \vdots & \vdots & & B_2 & \\ 0 & \cdots & 0 & 0 & & & \end{pmatrix}$$

Der Eintrag $(*) \neq 0$ steht in der $m - p_2 = 1$-ten Zeile und $n - q_2$-ten Spalte. Also ist B_2 eine $p_2 \times q_2$-Matrix.
Ist B_2 de Nullmatrix, so ist A_2 in Zeilenstufenform und der Algorithmus gibt A_2 aus.

Andernfalls wird B_2 im nächsten Schritt mit endlich vielen Elementaren Zeilenumformungen auf die Form (1.1) gebracht. Dehnt man diese Umformungen auf die Zeilen 2 bis m von A_2 aus, ändern sich die ersten $n - q_2$ Spalten von A_2 nicht, da dort nur Nullen stehen.
A_2 wird also in diesem Schritt mit endlich vielen elementaren Zeilenumformungen auf die Form

$$A_3 = \begin{pmatrix} 0 & \cdots & 0 & (*) & * & \cdots & \cdots & \cdots & \cdots & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & 0 & \cdots & 0 & (*) & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots & 0 & & & \\ \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & & B_3 & \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & & & \end{pmatrix}$$

gebracht. Induktiv überzeugt man sich leicht, dass der Algorithmus das Geforderte leistet. $\square$

Diese Überlegungen führen zu:

1.8 Algorithmus (Gaußsche Eliminationsverfahren zur Lösung von $Ax = b$)

Input: $m \times n$-Matrix A, $b \in \mathbb{R}^m$

Output: „es ex. keine Lösung“ od.($0 \leq r \leq n$ und eine Bijektion $\Phi_{A,b} : \mathbb{R}^{n-r} \to$ Lös(A, b))

1. Bringe die erweiterte Koeffizientenmatrix (A, b) in Zeilenstufenform $(\tilde{A}, \tilde{b})$ mit Algorithmus 1.6

2. Bestimme den Zeilenrang r von $\tilde{A}$(m minus die Anzahl der Nullstellen von $\tilde{A}$)

3. Tausche die Spalten von $\tilde{A}$(falls nötig) so, dass die Pivots in den ersten r Spalten stehen. (Wir bezeichnen die entstehende Matrix weiterhin mit $\tilde{A}$)

4. Verwende Algorithmus 1.3 mit Input $\tilde{A}, \tilde{b}, r$ und setzten im Existenzfall $\Phi_{(A,b)} := \Phi_{(\tilde{A},\tilde{b})}$ (wobei ggf. die Umnummerierung der Unbekannten aus Schritt 3 zu beachten ist).
Die Korrektheit des Algorithmus folgt aus Satz 1.4.

1.9 Folgerung

Die Lös(A, b) für $Ax = b$ ist entweder leer oder einelementig oder überabzählbar unendlich.

1.10 Beispiel (Rundungsfehler im Gauß-Algorithmus)

Zu lösen ist:

$$\begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Der Gaußalgorithmus liefert:

$$\left(\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 0 & -9999 & -9998 \end{array}\right)$$

Also:

$$x_2 = \frac{9998}{9999} \approx 0,9999$$

$$x_1 = 10^4(1 - 1 * \frac{9998}{9999}) \approx 1,0001$$

Führen wir nun den Algorithmus mit der Rechengenauigkeit von 3 Dezimalstellen aus, so:

$$\left(\begin{array}{cc|c} 1,00 * 10^{-4} & 1,00 & 1,00 \\ 1,00 & 1,00 & 2,00 \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|c} 1,00 * 10^{-4} & 1,00 & 1,00 \\ 0 & 1,00 * 10^4 & 1,00 * 10^4 \end{array}\right)$$

Also: $x_2 = 1, x_1 = 10^4(1 - 1) = 0$, was stark von der exakten Lösung abweicht.

Problem:

Der erste Pivot 10^{-4}, durch den zu Beginn des Algorithmus geteilt wird, ist „sehr klein“ und führt zu einem „großen“ Rundungsfehler, durch im Lauf des Algorithmus fortsetzen kann.

Ausweg:

Vertausche die Zeilen im Verlauf des Gauß-Algorithmus so, dass in der jeweiligen Spalte immer ein betragsmäßig möglichst großer Pivot entsteht (Gauß-Algorithmus mit Teilpivotierung)
In dieser Form ist der Gauß-Algorithmus „stabiler“, also weniger anfällig gegen Rundungsfehler.

Im Beispiel mit Rundungen auf 3 Dezimalstellen:

$$\left(\begin{array}{cc|c} 1,00 * 10^{-4} & 1,00 & 1,00 \\ 1,00 & 1,00 & 2,00 \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|c} 1,00 & 1,00 & 2,00 \\ 1,00 * 10^{-4} & 1,00 & 1,00 \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|c} 1,00 & 1,00 & 2,00 \\ 0 & 1,00 & 1,00 \end{array}\right)$$

da $1 - 10^{-4} = 0,9999 und 1 - 2 * 10^{-4} = 0,9998$ auf 3 Stellen gerundet gleich 1 ist,
Also $x_2 = 1,00$ und $x_1 = 1,00$, was der Rundung der korrekten Lösung auf 3 Stellen entspricht.

Formal bedeutet die Teilpivotierung:
Verwende in „Zeilenstufenschritt“ (Alg. 1.6) in Schritt 3.1:

$$i_1 := min\{i | |b_{i,j_1}| \geq |b_{k,j_1}| \forall k = 1, ..., p\}$$

Rechenaufwand des Gauß-Algorithmus

Wir bestimmen die Anzahl der Multiplikationen, die im Verlauf der Gauß-Algorithmus durchgeführt werden, im Fall $m = n = r$.

- Umformen in Zeilenstufenform:
  Die erste Zeile von (A, b) auf Zeilenstufenform zu bringen, benötigt

  $$\underset{\text{Zeilen}}{\underset{\uparrow}{(n-1)}} * \underset{\text{Faktor}}{\underset{\uparrow}{1}} + \underset{\text{Zeilen}}{\underset{\uparrow}{(n-1)}} * \underset{\text{Spalten}}{\underset{\uparrow}{n}} = (n-1)*(n+1)$$

  Multiplikationen.
  Iteration führt zu:

  $$\sum_{k=1}^{n}(k-1)(k+1) = \sum_{k=1}^{n}(k^2-1) = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n$$

  Multiplikationen.

- Rückwärtseinsetzen:
  Da der Zeilenrang von $\tilde{A}$ gleich n ist, so werden zur Berechnung von x_k

  $$(n-k+1)$$

  Multiplikationen benötigt, also insgesamt:

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

  Multiplikationen.

- Gesamtaufwand: $\frac{1}{3}n^3 + O(n^2)$ Multiplikationen

Ist n sehr groß (s. Beispiel 1.1), so müssen approximative Lösungsverfahren mit niedrigerem Rechenaufwand zum Einsatz gebracht werden.

2 Vektorräume

Sei $A \in \mathbb{R}^m \times n$ und $b \in \mathbb{R}^m$. Um das System $Ax = b$ weiter zu studieren , bietet es sich an, die Funktion

$$f_A : \mathbb{R}^n \to \mathbb{R}^m, x \mapsto Ax$$

zu untersuchen. Dies werden im allgemeineren Rahmen von „Linearen Abbildungen“ zwischen „Vektorräumen“ vornehmen.
Zunächst einige Begriffe:

2.1 Definition

Eine Menge G zusammen mit einer Verknüpfung $*$, also einer Abbildung $* : G \times G \to G$ heißt **Gruppe**, falls

(G1) $\forall a, b, c \in G : (a * b) * c = a * (b * c)$ (assoziativ)

(G2) Es gibt:

- ein $e \in G$ mit: $\forall_{a \in G} : e * a = a$
- für alle $a \in G$ ein $a^{-1} \in G$ mit: $a^{-1} * a = e$ (e wird **neutrales Element genannt, a^{-1} das inverse Element zu a)**

Gilt zudem $a * b = b * a$ für alle $a, b \in G$, so wird G **abelsch** (oder **kommutativ**) genannt.

2.2 Beispiel

i) $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ sind jeweils mit der Addition eine abelsche Gruppe.$\mathbb{Q}\backslash\{0\}, \mathbb{R}\backslash\{0\}, \mathbb{C}\backslash\{0\}$ bilden mit der Multiplikation als Verknüpfung eine abelsche Gruppe.

ii) Die Menge aller bijektiven Funktionen von $\mathbb{R}$ nach $\mathbb{R}$ mit der Hintereinanderausführung $\circ$ als Verknüpfung ist eine Gruppe, aber nicht kommutativ (Erinnerung: $(f \circ g)(x) = f(g(x))$)

2.3 Satz

Sei $(G, *)$ eine Gruppe. Dann:

i) Das neutrale Element e ist eindeutig und $a * e = a \forall a \in G$

ii) Das inverse Element a^{-1} zu$a \in G$ ist eindeutig und $a * a^{-1} = e \forall a \in G$

iii) Für alle $a, b \in G$ gilt: $(a^{-1})^{-1} = a$ und $(a * b)^{-1} = b^{-1} * a^{-1}$

iv) Für $a, b, \tilde{b} \in G$ gelten die „Kürzungsregeln“:

$$a * b = a * \tilde{b} \Rightarrow b = \tilde{b}$$

$$b * a = \tilde{b} * a \Rightarrow b = \tilde{b}$$

Beweis:

Es seien $a \in G, a^{-1}$ ein inverses Element zu a und $(a^{-1})^{-1}$ ein inverses Element zu a^{-1}.
Dann:

$$a*a^{-1} = e*(a*a^{-1}) = ((a^{-1})^{-1}*a^{-1})*(a*a^{-1}) = (a^{-1})^{-1}*(a^{-1}*a)*a^{-1} = (a^{-1})^{-1}*(e*a^{-1})$$

$$(a^{-1})^{-1} * a^{-1} = e$$

Somit:

$$a * e = a * (a^{-1} * a) = (a * a^{-1}) * a = e * a = a$$

Sei $\tilde{e}$ ein weiteres neutrales Element. Dann:

$$\tilde{e} * e = \tilde{e} \text{ und } \tilde{e} * e = e$$

Also $e = \tilde{e}$
Ist a’ ein weiteres neutrales Element zu a. Dann:

$$a' = a' * e = a' * (a * a^{-1}) = (a' * a) * a^{-1} = e * a^{-1} = a^{-1}$$

Dies zeigt i) und ii). iii) und iv): Übung □
Erinnerung:

2.4 Definition

Eine Menge K mit Verknüpfungen $+ : K \times K \to K, * : K \times K \to K$ heißt **Körper**,falls

K1) $(K, +)$ ist eine abelsche Gruppe. Man bezeichnet das neutrale Element mit 0 und das inverse Element zu a mit $-a$.

K2) $(K \backslash \{0\}, *)$ ist eine abelsche Gruppe. Man bezeichnet das neutrale Element mit 1 und das inverse Element zu a mit a^{-1} oder $\frac{1}{a}$.

K3) Für alle $a, b, c \in K$ gilt:

$$(a + b) * c = a * c + b * c \text{ Distributivgesetz}$$

2.5 Beispiel

i) $(\mathbb{Q}, +, *), (\mathbb{R}, +, *), (\mathbb{C}, +, *)$ sind Körper, $(\mathbb{Z}, +, *)$ nicht.

ii) Ist p eine Primzahl, so bilden die Restkassen modulo p, versehen mit der modularen Addition und Multiplikation einen Körper, der mit $\mathbb{Z}_p$ oder $\mathbb{F}_p$ bezeichnet wird (siehe MFI1).
Erinnerung: $\mathbb{F}_2 = \{0, 1\}$ mit Addition und Multiplikation erklärt durch:

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} \qquad \begin{array}{c|cc} * & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

Notation: Sei K ein Körper und $n \in \mathbb{N}_0$. Dann schreiben wir $a^n := a * a * \cdots * a$, $a^{-n} := \left(\frac{p}{a}\right)$, $a^0 := 1$

2.6 Definiton

Sei K ein Körper. Eine Menge V zusammen mit Abbildungen:

$$+ : V \times V \to V \text{ (Addition)}$$

$$* : V \times V \to V \text{ (skalare Multiplikation)}$$

wird **K-Vektorraum** (oder **Vektorraum** über K) genannt, wenn:

V1) $(V, +)$ ist eine abelsche Gruppe (Das neutrale Element heißt Nullvektor und wird mit 0 bezeichnet, das inverse Element zu $v \in V$ mit $-v$).

V2) Die skalare Multiplikation ist verträglich mit den Übrigen Verknüpfungen, d.h. für alle $\lambda, \mu \in K$ und $u, v \in V$ gilt:

$$(\lambda + \mu) * v = \lambda * v + \mu * v, \lambda * (v + w) = \lambda * v + \lambda * w$$

$$(\lambda * \mu) * v = \lambda * (\mu * v), 1 * v = v (1 \in K)$$

Achtung

$+$ bezeichnet hier sowohl die Addition im Körper, als auch die Addition im Vektorraum. Aus dem Kontext wird klar, was gemeint ist,**zum Beispiel:**

$$(\lambda \underset{\substack{\uparrow \\ \text{Addition} \\ \text{im Körper}}}{+} \mu) * v = \lambda * v \underset{\substack{\uparrow \\ \text{Addition} \\ \text{im Vektorraum}}}{+} \mu * v$$

Analog wird $*$ für die Multiplikation im Körper und für die skalare Multiplikation verwendet und 0 kann für die neutrale Elemente bzgl. der Addition im Körper und im Vektorraum stehen.

2.7 Beispiel

Sei K ein Körper.

i) $K^n = \left\{ \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} | x_i \in K \forall i = 1, ..., n \right\}$

Also die Menge aller Spalten der Höhe n mit Einträgen in K mit

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} := \begin{pmatrix} x_1 & + & y_1 \\ & \vdots & \\ x_n & + & y_n \end{pmatrix}$$

,

$$\lambda * \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} := \begin{pmatrix} \lambda * x_1 \\ \vdots \\ \lambda * x_n \end{pmatrix}$$

ist ein **K-Vektorraum**.

ii) Für eine nichtleere Menge A sei $K^A := \{f : A \to K\}$ die Menge aller Funktionen von A nach K. Dann wird K^A zu einem K-Vektorraum mit $(f + g)(a) := f(a) + g(a)$, $(\lambda * f)(a) := \lambda * f(a)$, $a \in A$

2.8 Beispiel

Ein Datenblock aus n Bits kann als Vektor in $\mathbb{F}_2$ notiert werden, also Vektor der Länge n mit Einträgen$\{0, 1\}$. Aufgrund der Additionstabelle in (Beispiel 2.5) folgt:

$$\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} \text{ wobei } z_i = \begin{cases} 0 & , v_i = w_i \\ 1 & , v_i \neq w_i \end{cases}$$

2.9 Definition

Sei V ein K-Vektorraum und $U \subset V$ nicht leer. Dann wird U ein **Untervektorraum (UVR)** von V genannt, falls

1. $\forall u, v \in U : u + v \in U$

2. $\forall \lambda \in K, u \in U : \lambda * u \in U$

2.10 Bemerkung

Sei U ein UVR von V. Schränkt man die Addition und die skalare Multiplikation auf $U \times U$ bze. $K \times U$ ein, so ist $(U, +, *)$ ein Vektorraum. Denn : Für jeden Vektor $u \in U$ gilt: $-u = (-1) * u \in U$ und $0 = u + (-u) \in U$.

2.11 Beispiel

i) Sei $[a, b] \subset \mathbb{R}$ ein Intervall. Dann sind

$$\zeta([a.b]) = \{f : [a, b] \to \mathbb{R} | f \text{ stetig}\}$$
$$\zeta^1([a.b]) = \{f : [a, b] \to \mathbb{R} | f \text{ stetig, differenzierbar}\}$$
$$\zeta^\infty([a.b]) = \{f : [a, b] \to \mathbb{R} | f \text{ belibig oft differenzierbar}\}$$

ii) Sei K ein Körper. Die Menge der **Polynomfunktionen**

$$Pol(K) = \left\{f : K \to K | f(\lambda) = a_n\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_0\lambda^0, \text{ für ein } n \in \mathbb{N}_0 \text{ und } a_0, \cdots, a_n \in K\right\}$$

ist ein UVR des K^K.

Achtung:

Ist etwa $K = \mathbb{F}_2$, so gilt für die Polynomfunktion

$$f : \mathbb{F}_2 \to \mathbb{F}_2, \lambda \mapsto \lambda + \lambda^2 :$$

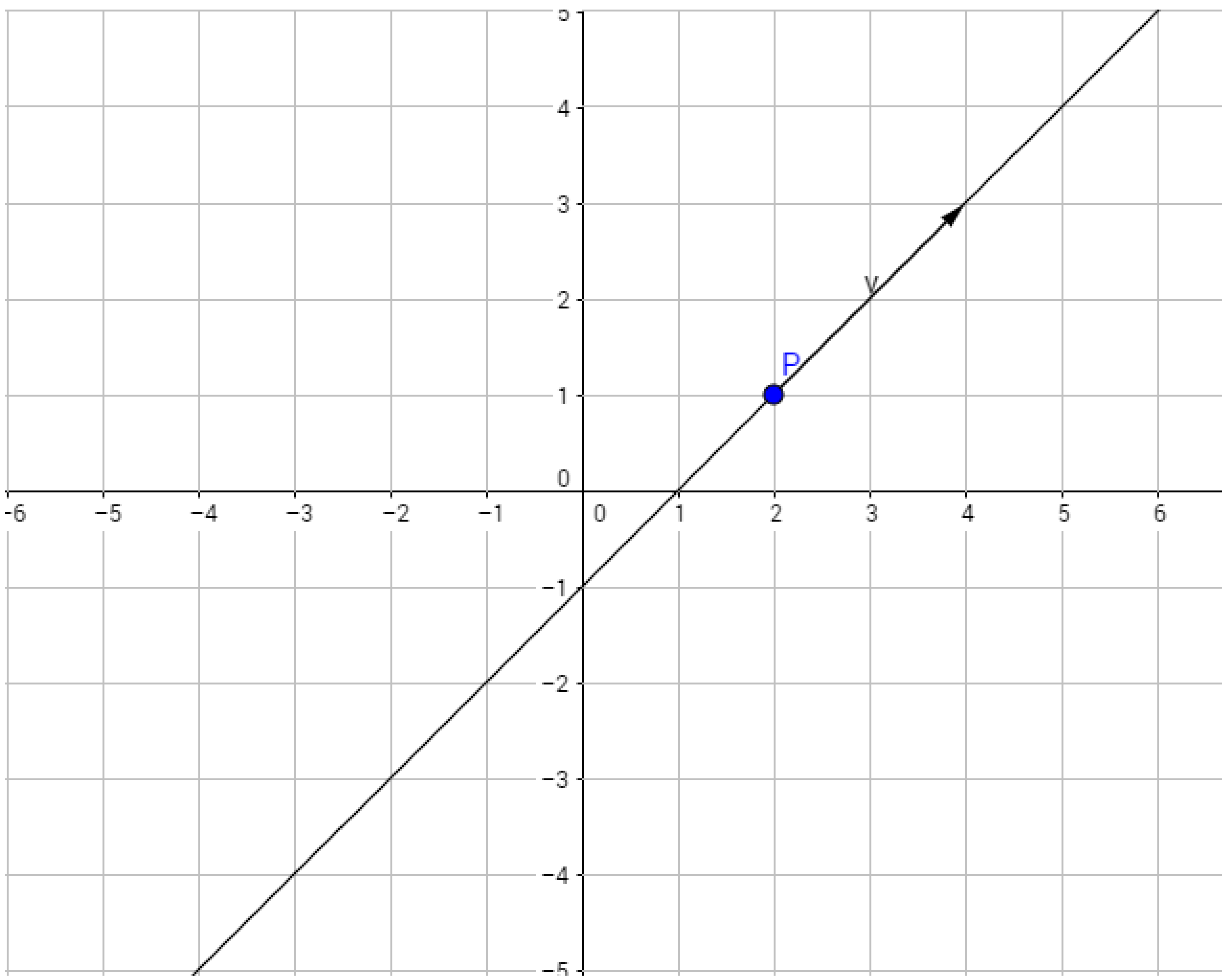
$f(0) = 0 + 0 = 0$
$f(1) = 1 + 1 = 0$ in $\mathbb{F}_2$
Also ist $f : \mathbb{F}_2 \to \mathbb{F}_2$ die Nullfunktion. Polynome mit verschiedenen Koeffizienten können in endlichen Körpern die gleiche Funktion darstellen.

2.12 Beispiel

Im $\mathbb{R}^n$ seinen Vektoren p und $v \neq 0$ gegeben. Eine Menge der Form

$$L(p, v) := \{p + \lambda * v | \lambda \in \mathbb{R}\}$$

heißt **Gerade** mit **Aufpunkt** p und **Richtungsvektor** v.
Eine Gerade ist genau dann ein UVR des $\mathbb{R}^n$ falls $0 \in l(p, v)$.


P
v
-6
−5
−4
−3
−2
−1
0
1
2
3
4
5
6
−1
−2
−3
−4
−5

3 Lineare Unabhängigkeit und Basen

3.1 Definition

Sei V ein K-Vektorraum und $(v_i)_{i\in I}$ eine Familie von Vektoren aus V.(d.h. $I \neq \emptyset$ ist eine Menge und für alle $i \in I$ gilt: $v_i \in V$)

i) v heißt **Linearkombination** von $(v_i)_{i\in I}$, falls es ein $r \in \mathbb{N}$, $i_1, \cdots, i_r \in I$ und $\lambda_1, \cdots, \lambda_r \in K$ gibt mit

$$v = \sum_{k=1}^{r} \lambda_k v_{i_k}.$$

Die Menge aller Linearkombinationen von $(v_i)_{i\in I}$ wird mit $span((v_i)_{i\in I})$ notiert und heißt **Spann** von $(v_i)_{i\in I}$.
Konvention: $span(\emptyset) = \{0\}$

ii) $(v_i)_{i\in I}$ wird **Erzeugendensystem** von V genannt, falls $span((v_i)_{i\in I}) = V$

iii) V heißt **endlich erzeugt**, falls es ein $r \in \mathbb{N}_0$ und $v_1, \cdots v_r \in V$so, dass $(v_1, \cdots, v_r)$ ein Erzeugendensystem von V ist.

3.2 Bemerkung

Sei $U := span((v_i)_{i\in I})$. Dann ist U ein UVR von V. Denn sind $u_1, u_2 \in U$, so existiert $r \in \mathbb{N}, i_1, \cdots, i_r \in I$ und $\lambda_k^{(1)}, \lambda_k^{(2)} \in K (k = 1, \cdots, r)$ mit:

$$u_1 = \sum_{k=1}^{r} \lambda_k^{(1)} v_{i_k}, u_2 = \sum_{k=1}^{r} \lambda_k^{(2)} v_{i_k}$$

Somit folgt für alle $\mu_1, \mu_2 \in K$:

$$\mu_1 * u_1 + \mu_2 * u_2 = \sum_{k=1}^{r} \underbrace{(\mu_1 \lambda_k^{(1)} + \mu_2 \lambda_k^{(2)})}_{\in K} * v_{i_k} \in U$$

3.3 Beispiel

i) Für einen Körper K ist K^n endlich erzeugt. Sei:

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{i-te Position,}$$

i-te Einheitsvektor im K^n(i=1,...,n). Dann ist $(e_1, ..., e_n)$ ein Erzeugendensystem des K^n Denn für alle $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in K^n$ gilt:

$$x = \sum_{i=1}^{n} x_i * e_i$$

ii) Der $\mathbb{R}$-VR der Polynomfunktion Pol($\mathbb{R}$) ist nicht endlich erzeugt. Denn seinen $p_1, ..., p_n \in Pol(\mathbb{R})$ und $\lambda_1, ..., \lambda_n$ reelle Zahlen. Sei k der maximale Grad dieser n-Polynome. Dann hat

$$p := \sum_{i=1}^{n} \lambda_i p_i$$

höchstens Grad k. Also ist $(p_1, ..., p_n)$ kein Erzeugendensystem von Pol($\mathbb{R}$).

3.4 Definiton

Sei V ein K-VR.

i) Eine endliche Familie von Vektoren $(v_1, ..., v_n)$ wird **Linear unabhängig** genannt, falls für alle $\lambda_1, ..., \lambda_n \in K$ gilt:

$$\sum_{i=1}^{n} \lambda_i v_i = 0 \in V \Rightarrow \forall_{i=1,...,n} \lambda_i = 0 \in K$$

ii) Eine beliebige Familie von Vektoren $(v_i)_{i \in I}$ wird **linear unabhängig** genannt, falls jede endliche Teilfamilie linear unabhängig ist (d.h. für alle $J \subset i$ endlich gilt: $(v_j)_{j \in J}$ ist linear unabhängig)
Andernfalls heißt $(v_i)_{i \in I}$ **linear abhängig**.

iii) Unter einer **Basis** von V verstehen wir ein linear unabhängiges Erzeugendensystem von V.

3.5 Bemerkung

Lineare Abhängigkeit bedeutet also, dass man aus der Familie $(v_i)_{i \in I}$ den Nullvektor durch eine nicht-triviale Linearkombination darstellen kann (d.h. durch eine Linearkombination, bei der nicht alle Koeffizienten gleich 0 sind).
Die Familie

$$\left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} \right) \text{ im } \mathbb{R}^3$$

ist also linear abhängig, denn

$$-2 * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + (-2) * \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + 1 * \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

3.6 Beispiele

i) Die Familie $(e_i)_{i=1,...,n}$ im K^n ist eine Basis. Denn seien $\lambda_1, ..., \lambda_n \in K$ mit

$$\sum_{i=1}^{n} \lambda_i e_i = 0 \in K^n$$

Da $\sum_{i=1}^{n} \lambda_i e_i = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}$, folgt $\lambda_i = 0 \in K$ für alle i=1,...,n. Also ist $(e_i, ..., e_n)$ linear unabhängig. Nach Beispiel 3.3 ist $(e_i, ..., e_n)$ auch ein Erzeugendensystem, also eine Basis.
$(e_i, ..., e_n)$ wird **kanonische Basis** in K^n genannt.

ii) Die Familie $\left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \right)$ ist eine Basis im $\mathbb{R}^3$
Lineare Unabhängigkeit:

$$\lambda_1 * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \lambda_2 * \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + \lambda_3 * \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow 3 * \lambda_3 = 0 \Rightarrow \lambda_3 = 0$$

$$\Rightarrow \lambda_1 * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \lambda_2 * \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \lambda_2 = 0 \Rightarrow \lambda_1 = 0.$$

Erzeugendensystem: Sei $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathbb{R}^3$. Dann $\lambda_1 * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \lambda_2 * \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + \lambda_3 * \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ für $\lambda_3 = \frac{x_3}{3}, \lambda_2 = \frac{x_2}{2} - \frac{x_3}{3}, \lambda_1 = x_1 - \frac{x_2}{2}$.

iii) Seien $q_n(t) = t^n, n \in \mathbb{N}_0$. Dann ist $(q_n)_{n\in\mathbb{N}_0}$ eine Basis von Pol($\mathbb{R}$).
Lineare Unabhängigkeit: Sei $n \in \mathbb{N}$ beliebig. Ist $\lambda_i \neq 0$ für ein $i = 0, ..., n$, dann hat

$$\left(\sum_{i=0}^{n} q_i\right)(t) = \sum_{i=0}^{n} \lambda_i t^i$$

höchstens n Nullstellen. Also ist $\sum_{i=0}^{n} \lambda_i q_i \neq 0 \in Pol(\mathbb{R})$.
Erzeugendensystem: Sei $p \in Pol(\mathbb{R})$. Dann

$$p(t) = \sum_{k=0}^{n} \lambda_k t^k \text{für ein } n \in \mathbb{N}_0, \lambda_1, ..., \lambda_n \in \mathbb{R}.$$

$$\text{Also } p = \sum_{k=0}^{n} \lambda_k q_k.$$

Zur Charakterisierung von Basen:

3.7 Satz

Sei $B = (v_1, ..., v_n)$ eine Familie von Vektoren eines K-VR V. Dann sind äquivalent:

i) B ist Basis

ii) B ist ein unverkürzbares Erzeugendensystem , dass heißt für jedes $r \in \{1, ..., n\}$ ist $(v_1, ..., v_{r-1}, v_{r+1}, ..., v_n)$ kein Erzeugendensystem, aber B ist ERzeugendensystem.

iii) Zu jedem Vektor $v \in V$ ex. eindeutig bestimmte $\lambda_1, ..., \lambda_n \in K$ mit: $v = \sum_{i=0}^{n} \lambda_i v_i$

iv) B ist unverlängerbar linear unabhängig, d.h. B ist linear unabhängig und für alle $v \in V$ ist $(v_1, ..., v_n, v)$ linear abhängig.

Beweis:

i) $\Rightarrow$ ii): Sei B ein ERzeugendensystem. Ist B verkürzbar, so gilt:

$$v_r = \lambda_1 v_1 + ... + \lambda_{r-1} v_{r-1} + \lambda_{r+1} v_{r+1} + ... + \lambda_n v_n$$

Also:

$$\lambda_1 v_1 + ... + \lambda_{r-1} v_{r-1} + \lambda_{r+1} v_{r+1} + ... + \lambda_n v_n + (-1) * v_r = 0,$$

d.h. B ist nicht linear unabhängig und also keine Basis.

ii)$\Rightarrow iii)$: Sei B ein Erzeugendensystem. Ist die Eindeutigkeitsbedingung nicht erfüllt, so gibt es ein $v \in V$ mit

$$v = \lambda_1 v_1, ..., \lambda_n v_n = \mu_1 v_1 + ... + \mu_n v_n$$

($\lambda_r \neq \mu_r$ für mindestens ein r).
Für ein solches r gilt:

$$v_r = \frac{\mu_1 - \lambda_1}{\lambda_r - \mu_r} v_1 + ... + \frac{\mu_{r-1} - \lambda_{r-1}}{\lambda_r - \mu_r} v_{r-1} + \frac{\mu_{r+1} - \lambda_{r+1}}{\lambda_r - \mu_r} v_{r+1} + ... + \frac{\mu_n - \lambda_n}{\lambda_r - \mu_r} v_n$$

Also ist B verkürzbar.

iii) $\Rightarrow$ **iv):** Aus iii) mit $v = 0 \in V$ folgt, direkt die lineare Unabhängigkeit. Sei $v \in V$. Dann hat v die Darstellung:

$$v = \sum_{i=1}^{n} \lambda_i v_i$$

Also

$$\sum_{i=1}^{n} \lambda_1 v_i + (-1) * v = 0,$$

d.h $(v_1, ..., v_n, v)$ ist linear abhängig.

iv) $\Rightarrow$ **i):** Aus iv) folgt: Für jedes $v \in V$ existieren $\lambda_1, ..., \lambda_n, \lambda \in K$ mit

$$\sum_{i=1}^{n} \lambda_i \lambda v = 0,$$

wobei nicht alle $\lambda_1, ..., \lambda_n, \lambda$ gleich 0 sind. Da B linear unabhängig ist, muss gelten $\lambda \neq 0$. Also:

$$v = \sum_{i=1}^{n} \frac{-\lambda_1}{\lambda} v_i,$$

d.h. B ist auch ein Erzeugendensystem.□

3.8 Folgerung

Jeder endlich erzeugte K-VR hat eine Basis.

Beweis:

Von einem endlichen Erzeugendensystem nehme man so lange Vektoren weg, bis es unverkürzbar ist. □

Konvention:

Die leere Familie von Vektoren ist linear unabhängig, also ist sie Basis von $V = \{0\}$.

Frage:

Kann ein endlich erzeugter Vektorraum Basen unterschiedlicher Länge(also mit unterschiedlich vielen Vektoren) haben?

3.9 Lemma (Austauschlemma)

Sei V ein K-VR mit Basis $B = (v_1, ..., v_n)$ und $w = \sum_{k=1}^{n} \lambda_k v_k \in V$.
Falls $\lambda_k \neq 0$, so ist $(v_1, ..., v_{k-1}, w, v_{k+1}, ..., v_n) =: B$' ebenfalls eine Basis von V.

Beweis:

Durch Umnummerierung kann man erreichen, dass $k = 1$.

- B'$= (w, v_2, ..., v_n)$ ist Erzeugendensystem:
  Sei $v \in V$. Dann: $v = \sum_{i=1}^{n} \mu_i v_i, \mu_1, ..., \mu_n \in K$. Da $\lambda_1 \neq 0$, folgt:

$$v_1 = \frac{1}{\lambda_1} w - \sum_{i=2}^{n} \frac{\lambda_i}{\lambda_1} v_i$$

  Somit:

$$v = \frac{\mu_1}{\lambda_1} w + \sum_{i=2}^{n} (\mu_i - \frac{\mu_1}{\lambda_1} \lambda_i) v_i$$

- B' ist linear unabhängig: Sei

$$0 = \mu * w + \sum_{i=2}^{n} \mu_i v_i$$

  Da $w = \sum_{i=1}^{n} \lambda_i v_i$, folgt:

$$0 = \mu \lambda_1 v_1 + \sum_{i=2}^{n} (\mu_i + \mu \lambda_i) v_i$$

  Da B linear unabhängig ist, folgt:$\mu \lambda_1 = 0$ und $(\mu_i + \mu \lambda_i) = 0$ für $i = 2, ..., n$.
  Da $\lambda_1 \neq 0$, gilt $\mu = 0$ und $\mu_i = 0$ für $i = 2, ..., n$.$\square$

3.10 Satz

Sei V ein K-VR mit Basis $B = (v_1, ..., v_r)$ und $(w_1, ..., w_n)$ eine linear unabhängige Familie in V. Dann:

i) $n \leq r$

ii) Es gibt Indizes $i_1, ..., i_n \in \{1, ..., r\}$, so dass die Familie B', die entsteht, wenn man in B $v_{i_1}, ..., v_{i_n}$ durch $w_1, ..., w_n$ ersetzt weiterhin eine Basis von V ist.

Beweis:

Induktion über $n \in \mathbb{N}_0$

- $n = 0$: Hier ist nichts zu zeigen
- $(n-1) \rightsquigarrow n$: Da auch $(w_1, ..., w_{n-1})$ linear unabhängig ist, ergibt die Induktionsannahme, dass $(n-1) \leq r$ und dass nach geeigneter Umnummerierung der Basis B $(w_1, ..., w_{n-1}, v_n, ..., v_r)$ Basis von V ist (Umnummerierung mit $i_1 = 1, ..., i_{n-1} = n-1$).
  Um $n \leq r$ zu zeigen, genügt es, $n-1 = r$ zu einem Widerspruch zu führen. In diesem Fall wäre $(w_1, ..., w_{n-1})$ eine Basis von V im Widerspruch zu Satz 3.7 iv).
  Ferne hat w_n die Darstellung

  $$w_n = \lambda_1 w_1 + ... + \lambda_{n-1} w_{n-1} + \lambda_n v_n + ... + \lambda_r v_r$$

  Da $(w_1, ..., w_n)$ linear unabhängig ist, folgt $\lambda_k \neq 0$ für ein $k = n.n+1, ..., r$. Also kann nach dem Austauschlemma v_k durch w_n ersetzt werden, ohne dass die Basiseigenschaft verloren geht.□