Polytope

Theorem

Eine Menge $X \subseteq \mathbb{R}^n$ ist genau dann ein Polytop, wenn es die konvexe Hülle von einer endlichen Menge von Vektoren in $\mathbb{R}^n$ ist.

Beweis: "$\Rightarrow$:"
Sei $X = \{x \in \mathbb{R}^n \mid Ax \leq b\} \neq \emptyset$ ein Polytop.
Schreibe X als $X = \left\{x \in \mathbb{R}^n \mid \binom{x}{1} \in C\right\}$, wobei

$$C = \left\{ \begin{pmatrix} x \\ \lambda \end{pmatrix} \in \mathbb{R}^{n+1} \mid \lambda \geq 0, Ax - \lambda b \leq 0 \right\}.$$

$\Rightarrow$ C ist ein polyedrischer Kegel.
$\Rightarrow$ C wird erzeugt von endlich vielen Vektoren $\binom{x_1}{\lambda_1}, \dots, \binom{x_k}{\lambda_k}$.
X ist beschränkt $\Rightarrow$ C kann keinen Vektor $\binom{x}{\lambda}$ mit $x \neq 0$ und $\lambda \leq 0$ enthalten.
$\Rightarrow$ Können annehmen: alle λ_i sind positiv (für $i \in \{1, \dots, k\}$).
Mit Skalierung können wir annehmen: $\lambda_i = 1$ für alle $i \in \{1, \dots, k\}$.
$\Rightarrow$

$$x \in X \Leftrightarrow \exists \mu_1, \dots, \mu_k \geq 0 : \begin{pmatrix} x \\ 1 \end{pmatrix} = \mu_1 \begin{pmatrix} x_1 \\ 1 \end{pmatrix} + \cdots + \mu_k \begin{pmatrix} x_k \\ 1 \end{pmatrix}.$$

$\Rightarrow$ $X = \text{conv}(\{x_1, \dots, x_k\})$.

Polytope

Theorem

Eine Menge $X \subseteq \mathbb{R}^n$ ist genau dann ein Polytop, wenn es die konvexe Hülle von einer endlichen Menge von Vektoren in $\mathbb{R}^n$ ist.

Beweis (Fortsetzung): "$\Leftarrow$:"
Sei $X = \text{conv}(\{x_1, \ldots, x_k\})$.
Zu zeigen: X ist ein Polytop.
Sei $C = \text{cone}(\{\binom{x_1}{1}, \ldots, \binom{x_k}{1}\})$.
$\Rightarrow$

$$x \in X \Leftrightarrow \begin{pmatrix} x \\ 1 \end{pmatrix} \in C.$$

$\Rightarrow$ C polyedrisch und kann geschrieben werden als $C = \{\binom{x}{\lambda} \mid Ax + b\lambda \leq 0\}$.
$\Rightarrow X = \{x \in \mathbb{R}^n \mid Ax + b \leq 0\}$. $\Rightarrow$ X ist ein Polyeder.
X ist beschränkt, denn: Für $M = \max\{||x_i|| \mid i \in \{1, \ldots, k\}\}$ kann man $x \in X$ schreiben als $x = \sum_{i=1}^{k} \lambda_i x_i$ mit $\lambda_1, \ldots, \lambda_k \geq 0$ und $\sum_{i=1}^{k} \lambda_i = 1$, also

$$||x|| \leq \sum_{i=1}^{k} \lambda_i ||x_i|| \leq M \sum_{i=1}^{k} \lambda_i = M.$$

$\square$

Polytope

Korollar

Ein Polytop ist die konvexe Hülle seiner Ecken.

Beweis: Sei P ein Polytop mit Eckenmenge X.

P konvex, $X \subseteq P \Rightarrow \mathrm{conv}(X) \subseteq P$.

z.z.: $P \subseteq \mathrm{conv}(X)$)

Voriges Theorem: $\mathrm{conv}(X)$ ist Polyeder.

Annahme: Es gibt einen Vektor $y \in P(\mathrm{conv}(X))$

$\Rightarrow$ Es gibt einen Halbraum $H_y = \{x \in \mathbb{R}^n : c^t x \le \delta\}$

mit $\mathrm{conv}(X) \subseteq H_y$ und $y \notin H_y$

$\Rightarrow c^t y > c^t x$ für alle $x \in X$.

$\Rightarrow$ Das Maximum von $c^t y$ wird

nicht in einer Ecke angenommen

Widerspruch, da es immer eine

Ecke gibt, in der das Maximum

angenommen wird. $\square$

Zerlegung von Polyedern

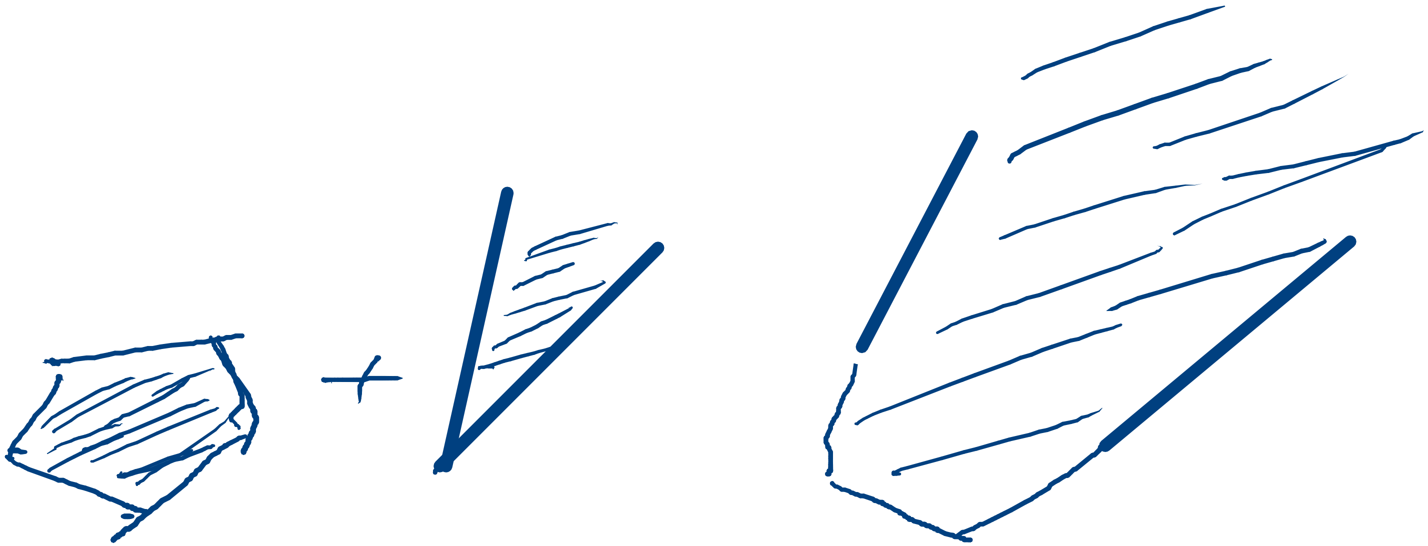
Notation: Für zwei Vektormengen $X, Y \subseteq \mathbb{R}^n$ ist die **Minkowski-Summe** definiert als:

$$X + Y := \{z \in \mathbb{R}^n \mid \exists x \in X \exists y \in Y : z = x + y\}.$$

Theorem

Es sei $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ ein Polyeder. Dann gibt es endliche Mengen $V, E \subseteq \mathbb{R}^n$, sodass

$$P = \mathrm{conv}(V) + \mathrm{cone}(E).$$



Beweis:
Der Kegel

$$C = \left\{ \begin{pmatrix} x \\ \lambda \end{pmatrix} \mid x \in \mathbb{R}^n, \lambda \in \mathbb{R}, \lambda \geq 0, Ax - \lambda b \leq 0 \right\}$$

ist polyedrisch.
$\Rightarrow$ C wird von endlich vielen Vektoren $\binom{x_1}{\lambda_1}, \ldots, \binom{x_k}{\lambda_k}$ erzeugt.
$x \in P$ gilt genau dann, wenn $\binom{x}{1} \in C$, was genau dann der Fall ist, wenn $\binom{x}{1} \in \text{cone}\left(\left\{\binom{x_1}{\lambda_1}, \ldots, \binom{x_k}{\lambda_k}\right\}\right)$.
Es gilt $\lambda_i \geq 0$, und nach Skalierung können wir $\lambda_i \in \{0, 1\}$ annehmen (für $i = 1, \ldots, k$).
Setze
$V := \{x_i \mid i \in \{1, \ldots, k\}, \lambda_i = 1\}$ und
$E := \{x_i \mid i \in \{1, \ldots, k\}, \lambda_i = 0\}$.
$\Rightarrow$ $P = \text{conv}(V) + \text{cone}(E)$. $\square$

Der Simplex-Algorithmus

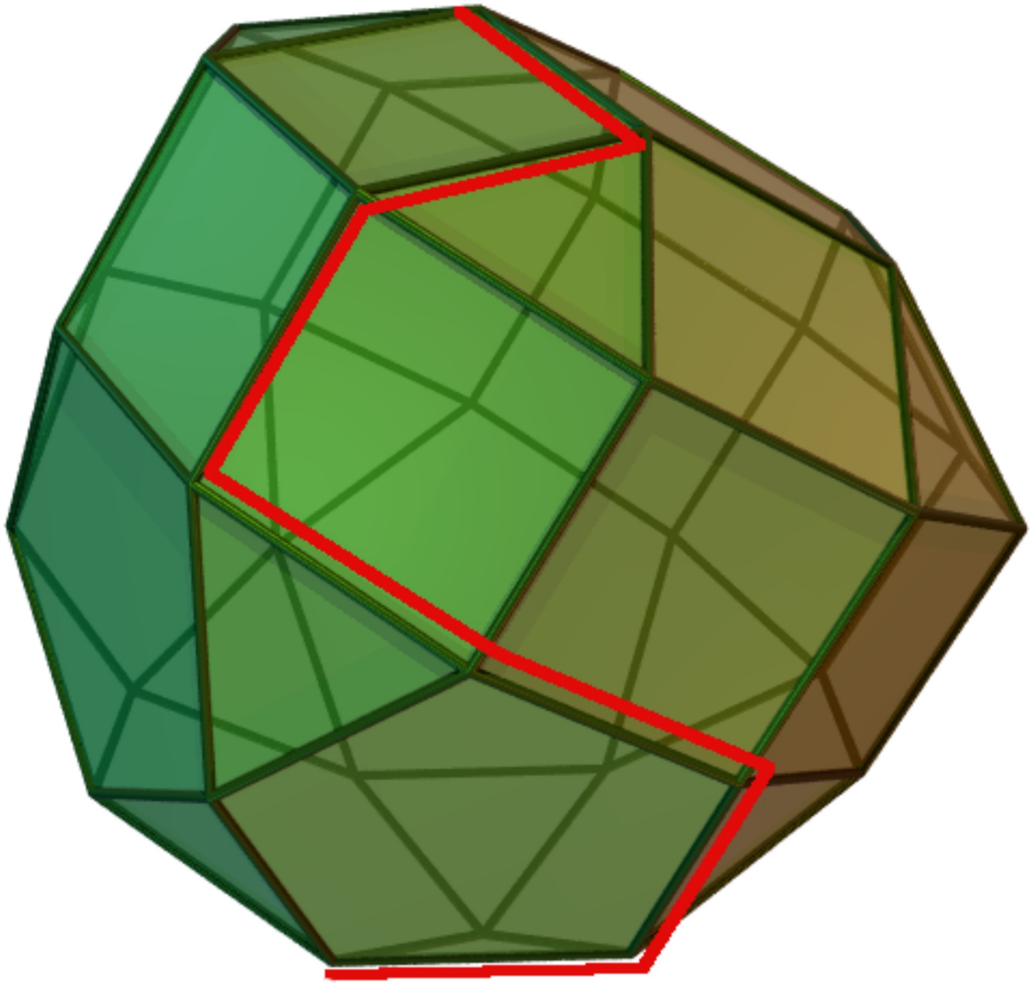
Der Simplex-Algorithmus

- Erste Algorithmus zur Lösung allgemeiner linearer Programme.
- Entwickelt von G. Dantzig [1951].
- Polynomielle Laufzeit kann nicht nachgewiesen werden, aber in der Praxis ist er oft sehr schnell.

Grundidee:

- Starte in eine Ecke des Lösungpolyeders.
- Solange es noch Nachbarecken mit besserem Zielfunktionswert gibt, lauf zu einer solchen.

Der Simplex-Algorithmus



Quelle: `https://de.wikipedia.org/`

Der Simplex-Algorithmus

Voraussetzungen:

- Wir brauchen ein spitzes Polyeder.
- Wir müssen ein Startlösung finden.

Um ein spitzes Lösungspolyeder zu haben, betrachten wir LPs in Standard-Gleichungsform

$$\begin{array}{rrcl} & \max c^t x & & \\ \text{s.t.} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

Das Finden einer Startlösung werden wir später behandeln.

Der Simplex-Algorithmus

Wir wollen das folgende LP lösen (mit $A \in \mathbb{R}^{m \times n}$, $b \in R^m$):

$$
\begin{array}{lrcl}
 & \max c^t x & & \\
\text{s.t.} & Ax & = & b \\
 & x & \geq & 0
\end{array}
$$

Können annehmen:

- $\mathrm{rank}(A) = m$.
- $Ax = m$ ist lösbar.
- $m < n$.

Teilmatrizen

Notation (zur vermeidung von Doppelindizes):

- Indexmenge der Spalten einer Matrix $A \in \mathbb{R}^{m \times n}$ mit $\{1, \ldots, n\}$.
- Für $B \subseteq \{1, \ldots, n\}$ bezeichne A_B die Teilmatrix von A aus den Spalten mit Index in B.
- Ebenso bezeichnen wir für einen Vektor $x \in \mathbb{R}^n$ mit x_B den Teilvektor x aus den Einträgen mit Index in B.

Man beachte: x_B ist dann ein Vektor der Länge $|B|$, aber die Einträge sind nicht notwendigerweise von 1 bis $|B|$ indiziert, sondern die Indizes sind Element aus B.

Teilmatrizen

Beispiel:

$$A = \begin{pmatrix} 2 & 4 & 0 & -3 & 1 \\ 1 & 3 & 2 & 1 & 5 \end{pmatrix}, \quad x = \begin{pmatrix} 3 \\ -2 \\ 0 \\ 1 \\ -1 \end{pmatrix}$$

$B = \{1, 3, 4\}$:

$$A_B = \begin{pmatrix} 2 & 0 & -3 \\ 1 & 2 & 1 \end{pmatrix}, \quad x_B = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

Die Spalten von A_B und x_B sind mit 1,3,4 indiziert, also

$$A_B x_B = x_1 a_1 + x_3 a_3 + x_4 a_4 = 3 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} + 0 \cdot \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \cdot \begin{pmatrix} -3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix},$$

wobei a_1, a_2, a_3, a_4, a_5 die Spalten von A seien.

Zulässige Basislösungen

Definition

Es sei $A \in \mathbb{R}^{m \times n}$ eine Matrix mit Rang m und $b \in \mathbb{R}^m$ ein Vektor. Sei $B \subseteq \{1, \ldots, n\}$ mit $|B| = m$, sodass A_B regulär ist. Setze $N := \{1, \ldots, n\} \setminus B$.

(a) Wir nennen B eine **Basis** von A. Der Vektor x mit $x_B = A_B^{-1} b$ und $x_N = 0$ heißt **Basislösung von $Ax = b$ zur Basis B**.

(b) Wenn x eine Basislösung von $Ax = b$ zur Basis B ist, dann heißen die Variablen x_j mit $j \in B$ **Basisvariablen**, und die Variablen x_j mit $j \in N$ heißen **Nicht-Basisvariablen**.

(c) Eine Basislösung x heißt **zulässig** (feasible), wenn $x \geq 0$. eine Basis heißt **zulässig**, wenn die zugehörige Basislösung zulässig ist.

(d) Eine zulässige Basislösung x für eine Basis B heißt **nicht-degeneriert** if $A_B^{-1} b > 0$. Andernfalls heißt sie **degeneriert**.

$$\underbrace{\left[\begin{array}{c|c} A_B & A_N \end{array}\right]}_{A} \begin{pmatrix} x_1 \\ \vdots \\ x_m \\ 0 \\ \vdots \\ 0 \end{pmatrix} = b$$

Zulässige Basislösungen

Beispiel:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

- Für $B = \{1, 2\}$: $A_B = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$ mit Basislösung $(1, 0, 0, 0)$, die zulässig und degeneriert ist.
- Gleiche Basislösungen für $B = \{1, 3\}$ und $B = \{1, 4\}$.
- Für $B = \{2, 3\}$: $A_B = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ mit nicht zulässiger Basislösung $(0, 2, -1, 0)$.
- Für $B = \{2, 4\}$: $A_B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ mit zulässiger nicht-degenerierter Basislösung $(0, 1, 0, 1)$.

Übertragung auf Systems der Form $\tilde{A}x \leq b, x \geq 0$.

- Es sei $\tilde{A} \in \mathbb{R}^{m \times \tilde{n}}$.
- Wir nennen einen Vektor $x^* \in \mathbb{R}^{\tilde{n}}$ mit $\tilde{A}x^* \leq b$ und $x^* \geq 0$ eine (zulässige/degenerierte) Basislösung, falls x^*, s^* mit $s^* := b - \tilde{A}x^*$ eine solche von $\tilde{A}x + I_m s = b, x \geq 0, s \geq 0$ (mit $n := \tilde{n} + m$ Variablen) ist.
- Also: In einer zulässigen Basislösung von $\tilde{A}x \leq b, x \geq 0$ muss die Zahl an Nebenbedingungen, die mit Gleichheit erfüllt sind, (inklusive Nicht-Negativitäts-Nebenbedingungen) mindestens $n - m = \tilde{n}$ sein.
- In einer *nicht-degenerierten* zulässigen Basislösung muss die Zahl der mit Gleichheit erfüllten Nebenbedingungen *genau* $\tilde{n}$ sein.

Beispiel:

Betrachte:

$$\begin{array}{rcrcl} x_1 & + & x_2 & \leq & 1 \\ 2x_1 & + & x_2 & \leq & 2 \\ x_1 & , & x_2 & \geq & 0 \end{array}$$

Mit Slack-Variablen s_1 und s_2 führt diese zu:

$$\begin{array}{rcrcrcrcl} x_1 & + & x_2 & + & s_1 & & & = & 1 \\ 2x_1 & + & x_2 & & & + & s_2 & = & 2 \\ x_1 & , & x_2 & , & s_1 & , & s_2 & \geq & 0 \end{array}$$

Die ist äquivalent zu dem schon betrachteten System

$$\begin{array}{rcrcrcrcl} x_1 & + & x_2 & + & x_3 & & & = & 1 \\ 2x_1 & + & x_2 & & & + & x_4 & = & 2 \\ x_1 & , & x_2 & , & x_3 & , & x_4 & \geq & 0 \end{array}$$

Es erlaubt eine geometrische Interpretation, wenn man sich auf die ersten beiden Variablen beschränkt.

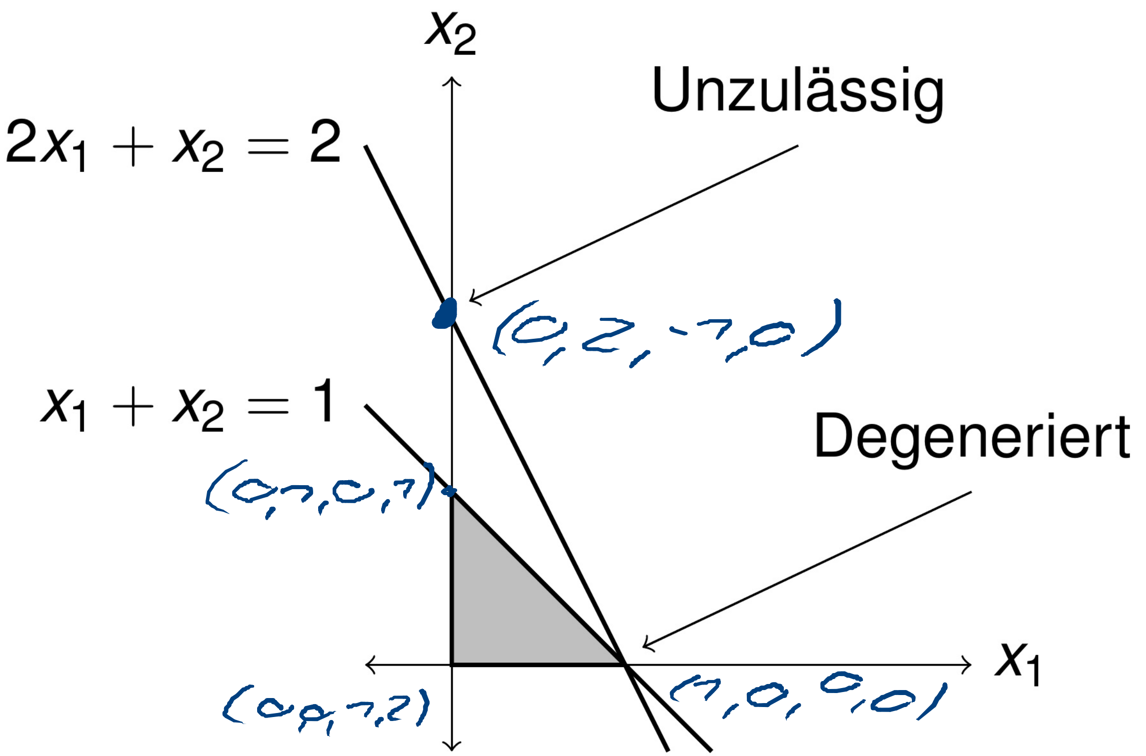


Figure: Basislösungen des vorigen Beispiels, projiziert auf den $\mathbb{R}^2$.

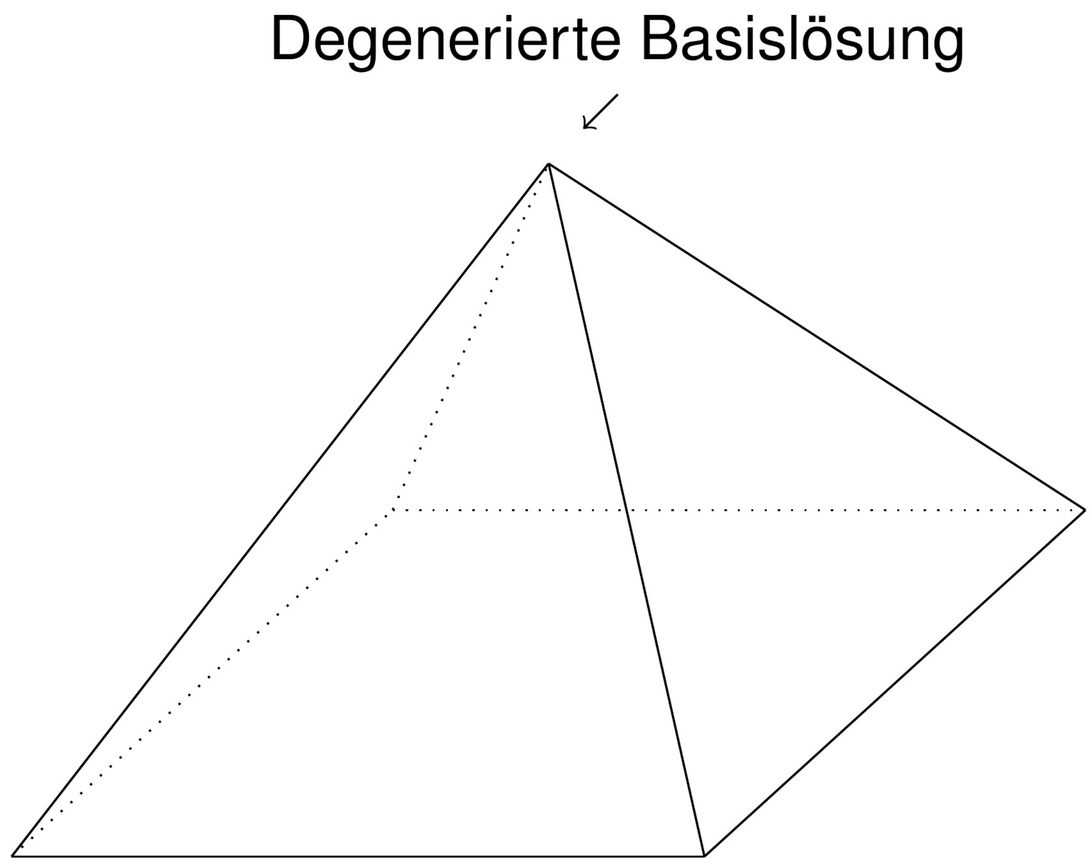


Figure: Eine degenerierte Lösung im $\mathbb{R}^3$.

Simplex Algorithm: Example I

$$
\begin{array}{lrcrcrcrcrcl}
\max & x_1 & + & x_2 & & & & & & & & \\
s.t. & -x_1 & + & x_2 & + & x_3 & & & & & = & 1 \\
 & x_1 & & & & & + & x_4 & & & = & 3 \\
 & & & x_2 & & & & & + & x_5 & = & 2 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0
\end{array}
$$

Simplex Algorithm: Example I

$$
\begin{array}{rrcrcrcrcrcl}
\max & x_1 & + & x_2 & & & & & & & & \\
s.t. & -x_1 & + & x_2 & + & x_3 & & & & & = & 1 \\
 & x_1 & & & & & + & x_4 & & & = & 3 \\
 & & & x_2 & & & & & + & x_5 & = & 2 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0
\end{array}
$$

Initial basis: $\{3, 4, 5\}$. $\Rightarrow$ $A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Simplex Algorithm: Example I

$$
\begin{array}{lrcrcrcrcrcl}
\max & x_1 & + & x_2 & & & & & & & & \\
s.t. & -x_1 & + & x_2 & + & x_3 & & & & & = & 1 \\
 & x_1 & & & & & + & x_4 & & & = & 3 \\
 & & & x_2 & & & & & + & x_5 & = & 2 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0
\end{array}
$$

Initial basis: $\{3, 4, 5\}$. $\quad \Rightarrow \quad A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Simplex tableau:

$$
\begin{array}{rclcrcr}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}
$$

Simplex Algorithm: Example I

$$
\begin{array}{rrcrcrcrcrcl}
\max & x_1 & + & x_2 & & & & & & & & \\
s.t. & -x_1 & + & x_2 & + & x_3 & & & & & = & 1 \\
 & x_1 & & & & & + & x_4 & & & = & 3 \\
 & & & x_2 & & & & & + & x_5 & = & 2 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0
\end{array}
$$

Initial basis: $\{3, 4, 5\}$. $\Rightarrow$ $A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Simplex tableau:

$$
\begin{array}{rcrcrcr}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}
$$

Recent solution: $(0, 0, 1, 3, 2)$

Simplex Algorithm: Example I

$$
\begin{array}{rcrcrcr}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}
$$

Increase exactly one of the non-basic variables with positive coefficient in the objective function.

Simplex Algorithm: Example I

$$
\begin{array}{rclclcl}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}
$$

Increase exactly one of the non-basic variables with positive coefficient in the objective function.

We choose x_2. How much can we increase it?

Simplex Algorithm: Example I

$$\begin{array}{rcrcrcr} x_3 & = & 1 & + & x_1 & - & x_2 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 2 & & & - & x_2 \\ \hline z & = & & & x_1 & + & x_2 \end{array}$$

Increase exactly one of the non-basic variables with positive coefficient in the objective function.

We choose x_2. How much can we increase it?

Constraints:

$x_3 = 1 + x_1 - x_2$: x_2 cannot get larger than 1.

$x_4 = 3 - x_1$: no constraint on x_2.

$x_5 = 2 - x_2$: x_2 cannot get larger than 2.

Simplex Algorithm: Example I

$$
\begin{array}{rcrcrcr}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}
$$

Increase exactly one of the non-basic variables with positive coefficient in the objective function.

We choose x_2. How much can we increase it?

Constraints:

$x_3 = 1 + x_1 - x_2$: x_2 cannot get larger than 1.

$x_4 = 3 - x_1$: no constraint on x_2.

$x_5 = 2 - x_2$: x_2 cannot get larger than 2.

Strictest constraint: $x_3 = 1 + x_1 - x_2$

$\Rightarrow$ Replace 3 by 2 in B.

Simplex Algorithm: Example I

First tableau:

$$\begin{array}{rclllll}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}$$

Replace 3 by 2 in the basis B: $B = \{2, 4, 5\}$:

$x_2 = 1 + x_1 - x_3$.

Second tableau:

$$\begin{array}{rclllll}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}$$

Simplex Algorithm: Example I

First tableau:

$$\begin{array}{rcrcrcr} x_3 & = & 1 & + & x_1 & - & x_2 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 2 & & & - & x_2 \\ \hline z & = & & & x_1 & + & x_2 \end{array}$$

Replace 3 by 2 in the basis B: $B = \{2, 4, 5\}$:

$x_2 = 1 + x_1 - x_3$.

Second tableau:

$$\begin{array}{rcrcrcr} x_2 & = & 1 & + & x_1 & - & x_3 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 1 & - & x_1 & + & x_3 \\ \hline z & = & 1 & + & 2x_1 & - & x_3 \end{array}$$

Recent solution: $(0, 1, 0, 3, 1)$

Simplex Algorithm: Example I

Second tableau:

$$
\begin{array}{rcrcrcr}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}
$$

Simplex Algorithm: Example I

Second tableau:

$$
\begin{array}{rcrcrcr}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}
$$

Only one candidate: x_1

Simplex Algorithm: Example I

Second tableau:

$$
\begin{array}{rcrcrcr}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}
$$

Only one candidate: x_1

$x_5 = 1 - x_1 + x_3$ is critical. Replace 5 by 1 in B: $B = \{1, 2, 4\}$.

$x_1 = 1 + x_3 - x_5$.

Third tableau:

$$
\begin{array}{rcrcrcr}
x_1 & = & 1 & + & x_3 & - & x_5 \\
x_2 & = & 2 & & & - & x_5 \\
x_4 & = & 2 & - & x_3 & + & x_5 \\
\hline
z & = & 3 & + & x_3 & - & 2x_5
\end{array}
$$

Simplex Algorithm: Example I

Second tableau:

$$
\begin{array}{rcrcrcr}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}
$$

Only one candidate: x_1
$x_5 = 1 - x_1 + x_3$ is critical. Replace 5 by 1 in B: $B = \{1, 2, 4\}$.
$x_1 = 1 + x_3 - x_5$.
Third tableau:

$$
\begin{array}{rcrcrcr}
x_1 & = & 1 & + & x_3 & - & x_5 \\
x_2 & = & 2 & & & - & x_5 \\
x_4 & = & 2 & - & x_3 & + & x_5 \\
\hline
z & = & 3 & + & x_3 & - & 2x_5
\end{array}
$$

Recent solution: $x = (1, 2, 0, 2, 0)$.

Simplex Algorithm: Example I

Third tableau:

$$
\begin{array}{rcrcrcr}
x_1 & = & 1 & + & x_3 & - & x_5 \\
x_2 & = & 2 & & & - & x_5 \\
x_4 & = & 2 & - & x_3 & + & x_5 \\
\hline
z & = & 3 & + & x_3 & - & 2x_5
\end{array}
$$

Simplex Algorithm: Example I

Third tableau:

$$
\begin{array}{rclclcl}
x_1 & = & 1 & + & x_3 & - & x_5 \\
x_2 & = & 2 & & & - & x_5 \\
x_4 & = & 2 & - & x_3 & + & x_5 \\
\hline
z & = & 3 & + & x_3 & - & 2x_5
\end{array}
$$

Only one candidate: x_3

$x_4 = 2 - x_3 + x_5$ is critical. Replace 4 by 3 in B: $B = \{1, 2, 3\}$.

$x_3 = 2 - x_4 + x_5$

Fourth tableau:

$$
\begin{array}{rclclcl}
x_1 & = & 3 & - & x_4 & & \\
x_2 & = & 2 & & & - & x_5 \\
x_3 & = & 2 & - & x_4 & + & x_5 \\
\hline
z & = & 5 & - & x_4 & - & x_5
\end{array}
$$

Simplex Algorithm: Example I

Third tableau:

$$\begin{array}{rcrcrcr} x_1 & = & 1 & + & x_3 & - & x_5 \\ x_2 & = & 2 & & & - & x_5 \\ x_4 & = & 2 & - & x_3 & + & x_5 \\ \hline z & = & 3 & + & x_3 & - & 2x_5 \end{array}$$

Only one candidate: x_3
$x_4 = 2 - x_3 + x_5$ is critical. Replace 4 by 3 in B: $B = \{1, 2, 3\}$.
$x_3 = 2 - x_4 + x_5$
Fourth tableau:

$$\begin{array}{rcrcrcr} x_1 & = & 3 & - & x_4 & & \\ x_2 & = & 2 & & & - & x_5 \\ x_3 & = & 2 & - & x_4 & + & x_5 \\ \hline z & = & 5 & - & x_4 & - & x_5 \end{array}$$

Recent solution: $x = (3, 2, 2, 0, 0)$.

Simplex Algorithm: Example I

Fourth tableau:

$$
\begin{array}{rcrcrcr}
x_1 & = & 3 & - & x_4 & & \\
x_2 & = & 2 & & & - & x_5 \\
x_3 & = & 2 & - & x_4 & + & x_5 \\
\hline
z & = & 5 & - & x_4 & - & x_5
\end{array}
$$

Recent solution: $x = (3, 2, 2, 0, 0)$.

This is an optimum solution!