

# **Mathematik in den technischen Wissenschaften**

## **eine Themen-Auswahl**

Clemens Brand

Montanuniversität Leoben

AG-Tagung AHS Mathematik  
St. Pölten, 19. März 2014

# Gliederung

## 1 Numerische Lineare Algebra

- Überbestimmte Systeme

## 2 Numerische Strömungsmechanik

- Iterative Lösungsverfahren

## 3 Diskrete Mathematik

# Gliederung

## 1 Numerische Lineare Algebra

- Überbestimmte Systeme

## 2 Numerische Strömungsmechanik

- Iterative Lösungsverfahren

## 3 Diskrete Mathematik

# Überbestimmte Systeme

Ein lineares Gleichungssystem mit mehr Gleichungen als Unbekannten heißt überbestimmt

$$\begin{array}{rcl} x & = & 1 \\ y & = & 2 \\ x + y & = & 4 \end{array} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

## Allgemein:

$$Ax = b \quad \text{mit einer } m \times n\text{-Matrix } A, \text{ wobei } m > n$$

- In der Regel hat ein solches System keine (exakte) Lösung.
- (Kompromiss-)Lösung sucht möglichst kleinen *Residuenvektor*

$$r = b - Ax$$

- Klassische kleinste-Quadrat-Lösung mit *Normalengleichungen*

$$A^T A x = A^T b$$

# Überbestimmte Systeme

Ein lineares Gleichungssystem mit mehr Gleichungen als Unbekannten heißt überbestimmt

$$\begin{array}{rcl} x & = & 1 \\ y & = & 2 \\ x + y & = & 4 \end{array} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

Allgemein:

$$Ax = b \quad \text{mit einer } m \times n\text{-Matrix } A, \text{ wobei } m > n$$

- In der Regel hat ein solches System keine (exakte) Lösung.
- (Kompromiss-)Lösung sucht möglichst kleinen Residuenvektor

$$r = b - Ax$$

- Klassische kleinste-Quadrat-Lösung mit Normalengleichungen

$$A^T A x = A^T b$$

# Überbestimmte Systeme

Ein lineares Gleichungssystem mit mehr Gleichungen als Unbekannten heißt überbestimmt

$$\begin{array}{rcl} x & = & 1 \\ y & = & 2 \\ x + y & = & 4 \end{array} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

Allgemein:

$$Ax = b \quad \text{mit einer } m \times n\text{-Matrix } A, \text{ wobei } m > n$$

- In der Regel hat ein solches System keine (exakte) Lösung.
- (Kompromiss-)Lösung sucht möglichst kleinen *Residuenvektor*

$$r = b - Ax$$

- Klassische kleinste-Quadrat-Lösung mit *Normalengleichungen*

$$A^T A x = A^T b$$

# Überbestimmte Systeme

Ein lineares Gleichungssystem mit mehr Gleichungen als Unbekannten heißt überbestimmt

$$\begin{array}{rcl} x & = & 1 \\ y & = & 2 \\ x + y & = & 4 \end{array} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

Allgemein:

$$Ax = b \quad \text{mit einer } m \times n\text{-Matrix } A, \text{ wobei } m > n$$

- In der Regel hat ein solches System keine (exakte) Lösung.
- (Kompromiss-)Lösung sucht möglichst kleinen *Residuenvektor*

$$r = b - Ax$$

- Klassische kleinste-Quadrat-Lösung mit *Normalengleichungen*

$$A^T A x = A^T b$$

# Beispiel: Wägung zweier Massen

Zwei Massen  $m_1, m_2$  werden zuerst einzeln, dann gemeinsam abgewogen.  
Die Messwerte sind:

$$m_1 = 1$$

$$m_2 = 2$$

$$m_1 + m_2 = 4$$

## Lösungsvorschläge?

- Letzte Gleichung weglassen: Fehler  $r = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- Fehler aufteilen, etwa  $m_1 = 1,5; m_2 = 2,5 \rightarrow r = \begin{bmatrix} -0,5 \\ -0,5 \\ 0 \end{bmatrix}$
- noch bessere Aufteilung...

# Beispiel: Wägung zweier Massen

Zwei Massen  $m_1, m_2$  werden zuerst einzeln, dann gemeinsam abgewogen.  
Die Messwerte sind:

$$m_1 = 1$$

$$m_2 = 2$$

$$m_1 + m_2 = 4$$

## Lösungsvorschläge?

- Letzte Gleichung weglassen: Fehler  $r = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

- Fehler aufteilen, etwa  $m_1 = 1,5; m_2 = 2,5 \rightarrow r = \begin{bmatrix} -0,5 \\ -0,5 \\ 0 \end{bmatrix}$
- noch bessere Aufteilung...

# Beispiel: Wägung zweier Massen

Zwei Massen  $m_1, m_2$  werden zuerst einzeln, dann gemeinsam abgewogen.  
Die Messwerte sind:

$$m_1 = 1$$

$$m_2 = 2$$

$$m_1 + m_2 = 4$$

## Lösungsvorschläge?

- Letzte Gleichung weglassen: Fehler  $\mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- Fehler aufteilen, etwa  $m_1 = 1,5; m_2 = 2,5 \rightarrow \mathbf{r} = \begin{bmatrix} -0,5 \\ -0,5 \\ 0 \end{bmatrix}$
- noch bessere Aufteilung...

# Beispiel: Wägung zweier Massen

Zwei Massen  $m_1, m_2$  werden zuerst einzeln, dann gemeinsam abgewogen.  
Die Messwerte sind:

$$m_1 = 1$$

$$m_2 = 2$$

$$m_1 + m_2 = 4$$

## Lösungsvorschläge?

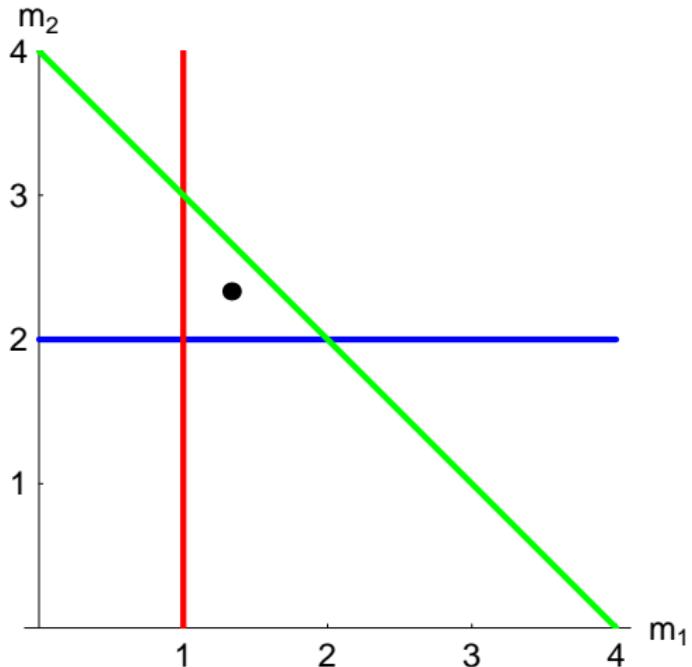
- Letzte Gleichung weglassen: Fehler  $\mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- Fehler aufteilen, etwa  $m_1 = 1,5; m_2 = 2,5 \rightarrow \mathbf{r} = \begin{bmatrix} -0,5 \\ -0,5 \\ 0 \end{bmatrix}$
- noch bessere Aufteilung...

# Geometrische Interpretation

Zeilen des Gleichungssystems sind Geradengleichungen

$$\begin{aligned}m_1 &= 1 \\m_2 &= 2 \\m_1 + m_2 &= 4\end{aligned}$$

Den drei Gleichungen entsprechen  
drei Gerade im  $\mathbb{R}^2$ . Kein  
gemeinsamer Schnittpunkt!  
„In der Mitte des Fehlerdreiecks“,  
im Punkt  $(4/3; 7/3)$  ist die  
*Summe der Fehlerquadrate*  
minimal.



$$(m_1 - 1)^2 + (m_2 - 2)^2 + (m_1 + m_2 - 4)^2 \rightarrow \min!$$

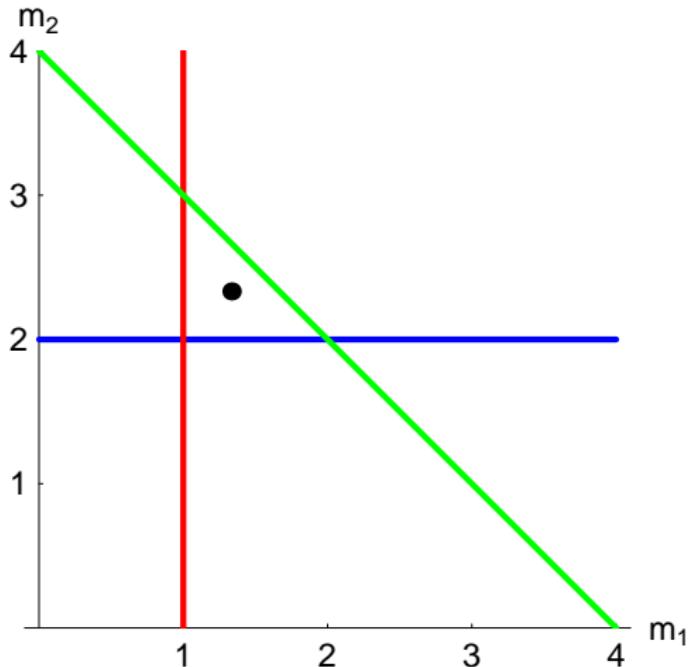
# Geometrische Interpretation

Zeilen des Gleichungssystems sind Geradengleichungen

$$\begin{aligned}m_1 &= 1 \\m_2 &= 2 \\m_1 + m_2 &= 4\end{aligned}$$

Den drei Gleichungen entsprechen  
drei Gerade im  $\mathbb{R}^2$ . Kein  
gemeinsamer Schnittpunkt!

„In der Mitte des Fehlerdreiecks“,  
im Punkt  $(4/3; 7/3)$  ist die  
*Summe der Fehlerquadrate*  
minimal.



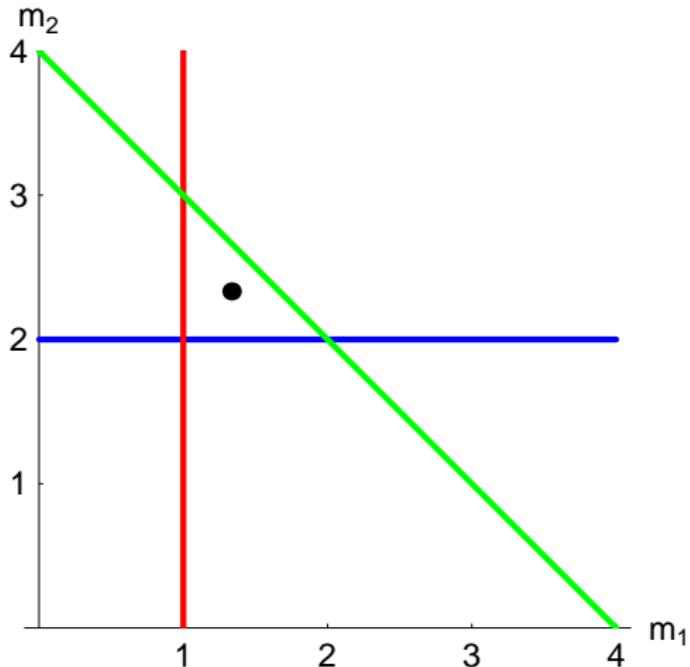
$$(m_1 - 1)^2 + (m_2 - 2)^2 + (m_1 + m_2 - 4)^2 \rightarrow \min!$$

# Geometrische Interpretation

Zeilen des Gleichungssystems sind Geradengleichungen

$$\begin{aligned}m_1 &= 1 \\m_2 &= 2 \\m_1 + m_2 &= 4\end{aligned}$$

Den drei Gleichungen entsprechen  
drei Gerade im  $\mathbb{R}^2$ . Kein  
gemeinsamer Schnittpunkt!  
„In der Mitte des Fehlerdreiecks“,  
im Punkt  $(4/3; 7/3)$  ist die  
*Summe der Fehlerquadrate*  
minimal.



$$(m_1 - 1)^2 + (m_2 - 2)^2 + (m_1 + m_2 - 4)^2 \rightarrow \min!$$

# Methode der kleinsten Fehlerquadrate

Suche  $m_1, m_2$  so dass

$$(m_1 - 1)^2 + (m_2 - 2)^2 + (m_1 + m_2 - 4)^2 \rightarrow \min!$$

Differenziere nach  $m_1$  und  $m_2$ , setze Ableitungen gleich Null  $\rightarrow$

$$\begin{array}{rcl} 2m_1 &+& m_2 = 5 \\ m_1 &+& 2m_2 = 6 \end{array}$$

Führt man diese Rechnung allgemein für ein überbestimmtes System  $Ax = b$  durch, erhält man die *Normalengleichungen*  $A^T A x = A^T b$ .

hier:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

# Zwei mögliche geometrische Interpretationen

## Im Raum der Lösungsvektoren

- Gleichungen entsprechen Geraden (Ebenen, Hyperebenen)
- Optimal-Lösung im Schnittbereich (Fehlerdreieck, -Polyeder, ...)
- Nachteil: „Mitte des Fehlerdreiecks“ nicht exakt definiert.

## Im Raum der rechten-Seite-Vektoren

- Matrix mal Vektor ergibt *Linearkombination der Spaltenvektoren*
- Bei „zu wenig“ Spaltenvektoren lässt sich nicht jede rechte Seite als Linearkombination erreichen.
- Der geringste Abstand zwischen Linearkombination und rechter Seite ist Normalabstand.

Es kommt eh gleich ein Beispiel...

# Zwei mögliche geometrische Interpretationen

## Im Raum der Lösungsvektoren

- Gleichungen entsprechen Geraden (Ebenen, Hyperebenen)
- Optimal-Lösung im Schnittbereich (Fehlerdreieck, -Polyeder, ...)
- Nachteil: „Mitte des Fehlerdreiecks“ nicht exakt definiert.

## Im Raum der rechten-Seite-Vektoren

- Matrix mal Vektor ergibt *Linearkombination der Spaltenvektoren*
- Bei „zu wenig“ Spaltenvektoren lässt sich nicht jede rechte Seite als Linearkombination erreichen.
- Der geringste Abstand zwischen Linearkombination und rechter Seite ist Normalabstand.

Es kommt eh gleich ein Beispiel...

# Zwei mögliche geometrische Interpretationen

## Im Raum der Lösungsvektoren

- Gleichungen entsprechen Geraden (Ebenen, Hyperebenen)
- Optimal-Lösung im Schnittbereich (Fehlerdreieck, -Polyeder, ...)
- Nachteil: „Mitte des Fehlerdreiecks“ nicht exakt definiert.

## Im Raum der rechten-Seite-Vektoren

- Matrix mal Vektor ergibt *Linearkombination der Spaltenvektoren*
- Bei „zu wenig“ Spaltenvektoren lässt sich nicht jede rechte Seite als Linearkombination erreichen.
- Der geringste Abstand zwischen Linearkombination und rechter Seite ist Normalabstand.

Es kommt eh gleich ein Beispiel...

# Zwei mögliche geometrische Interpretationen

## Im Raum der Lösungsvektoren

- Gleichungen entsprechen Geraden (Ebenen, Hyperebenen)
- Optimal-Lösung im Schnittbereich (Fehlerdreieck, -Polyeder, ...)
- Nachteil: „Mitte des Fehlerdreiecks“ nicht exakt definiert.

## Im Raum der rechten-Seite-Vektoren

- Matrix mal Vektor ergibt *Linearkombination der Spaltenvektoren*
- Bei „zu wenig“ Spaltenvektoren lässt sich nicht jede rechte Seite als Linearkombination erreichen.
- Der geringste Abstand zwischen Linearkombination und rechter Seite ist Normalabstand.

Es kommt eh gleich ein Beispiel...

# Interpretation von $Ax = b$

Zwei Möglichkeiten

## Im Raum der x-Vektoren

Jede *Zeile* der Matrix ist ein Normalvektor in einer Geradengleichung für  $x \in \mathbb{R}^2$ , Ebenengleichung für  $x \in \mathbb{R}^3, \dots$

Lösung  $x$  ist *Schnittpunkt* aller Geraden (Ebenen, Hyperebenen. . .)

## Im Raum der b-Vektoren

Jede *Spalte* der Matrix  $A$  ist ein Vektor. Das Produkt  $Ax$  ist eine Linearkombination der Spaltenvektoren.

Lösung  $x$  gibt genau die Koeffizienten an, mit denen sich  $b$  als *Linearkombination der Spaltenvektoren* von  $A$  erreichen lässt.

# Interpretation von $Ax = b$

Zwei Möglichkeiten

## Im Raum der x-Vektoren

Jede Zeile der Matrix ist ein Normalvektor in einer Geradengleichung für  $x \in \mathbb{R}^2$ , Ebenengleichung für  $x \in \mathbb{R}^3, \dots$

Lösung  $x$  ist *Schnittpunkt* aller Geraden (Ebenen, Hyperebenen...)

## Im Raum der b-Vektoren

Jede Spalte der Matrix  $A$  ist ein Vektor. Das Produkt  $Ax$  ist eine Linearkombination der Spaltenvektoren.

Lösung  $x$  gibt genau die Koeffizienten an, mit denen sich  $b$  als *Linearkombination der Spaltenvektoren* von  $A$  erreichen lässt.

# Interpretation von $Ax = b$

Zwei Möglichkeiten

## Im Raum der x-Vektoren

Jede *Zeile* der Matrix ist ein Normalvektor in einer Geradengleichung für  $x \in \mathbb{R}^2$ , Ebenengleichung für  $x \in \mathbb{R}^3, \dots$

Lösung  $x$  ist *Schnittpunkt* aller Geraden (Ebenen, Hyperebenen. . .)

## Im Raum der b-Vektoren

Jede *Spalte* der Matrix  $A$  ist ein Vektor. Das Produkt  $Ax$  ist eine Linearkombination der Spaltenvektoren.

Lösung  $x$  gibt genau die Koeffizienten an, mit denen sich  $b$  als *Linearkombination der Spaltenvektoren* von  $A$  erreichen lässt.

# Beispiel: Farbmischung mit RGB-Vektoren

Welches Grau gibt Rot und Blau?

Gegeben: zwei Farbtöne

$$\text{RGB} \begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix}$$



$$\text{RGB} \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix}$$



Gesucht: neutrale Grauschattierung RGB 150 150 150

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} x + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} y = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}$$

Optimale Lösung erreicht

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} 0,588 + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} 0,674 = \begin{bmatrix} 151 \\ 130 \\ 164 \end{bmatrix}$$

# Beispiel: Farbmischung mit RGB-Vektoren

Welches Grau gibt Rot und Blau?

Gegeben: zwei Farbtöne

$$\text{RGB} \begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix}$$



$$\text{RGB} \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix}$$



Gesucht: neutrale Grauschattierung RGB 150 150 150

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} x + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} y = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}$$

Optimale Lösung erreicht

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} 0,588 + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} 0,674 = \begin{bmatrix} 151 \\ 130 \\ 164 \end{bmatrix}$$

# Beispiel: Farbmischung mit RGB-Vektoren

Welches Grau gibt Rot und Blau?

Gegeben: zwei Farbtöne

$$\text{RGB} \begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix}$$



$$\text{RGB} \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix}$$



Gesucht: neutrale Grauschattierung RGB 150 150 150

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} x + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} y = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}$$

Optimale Lösung erreicht

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} 0,588 + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} 0,674 = \begin{bmatrix} 151 \\ 130 \\ 164 \end{bmatrix}$$

# Beispiel: Farbmischung mit RGB-Vektoren

Welches Grau gibt Rot und Blau?

Gegeben: zwei Farbtöne

$$\text{RGB} \begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix}$$



$$\text{RGB} \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix}$$



Gesucht: neutrale Grauschattierung RGB 150 150 150

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} x + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} y = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}$$



Optimale Lösung erreicht

$$\begin{bmatrix} 200 \\ 50 \\ 50 \end{bmatrix} 0,588 + \begin{bmatrix} 50 \\ 150 \\ 200 \end{bmatrix} 0,674 = \begin{bmatrix} 151 \\ 130 \\ 164 \end{bmatrix}$$



# Gleichungssystem interpretiert als Linearkombination der Spaltenvektoren

nochmals am Massen-Wägung-Beispiel illustriert

$$\begin{array}{rcl} x & = & 1 \\ y & = & 2 \\ x + y & = & 4 \end{array} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} y = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

Die *spaltenweise* Interpretation eines Gleichungssystems ist weniger geläufig als die *zeilenweise*. Sie ist aber ebenso hilfreich und wichtig!

# Orthogonalitätsbedingung im Bildraum

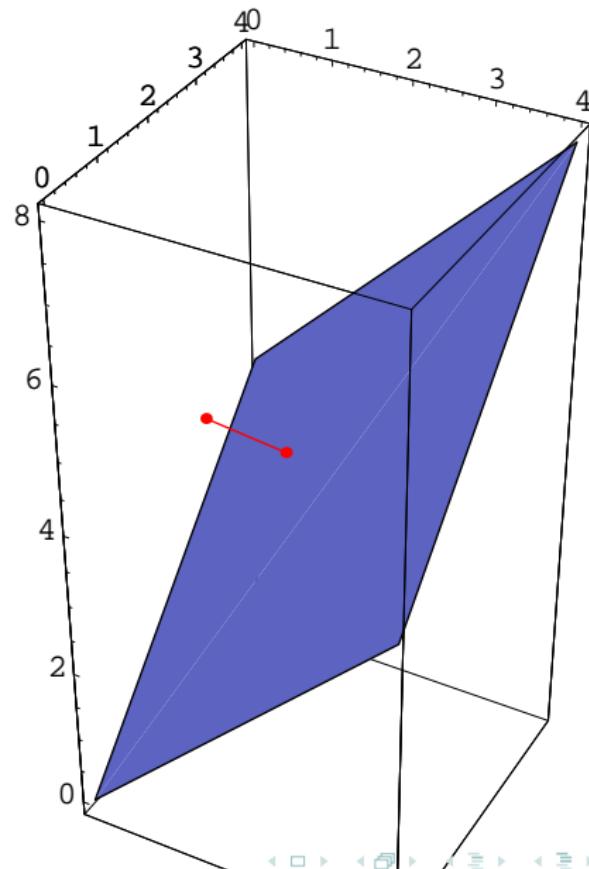
Der Ausdruck  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$

entspricht der  
Parameterdarstellung einer Ebene  
im  $\mathbb{R}^3$ .

Der Punkt  $(1; 2; 4)$  liegt nicht auf  
dieser Ebene. Der Residuenvektor

$$\mathbf{r} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$$

hat minimale Länge, wenn er auf  
die Ebene normal steht.

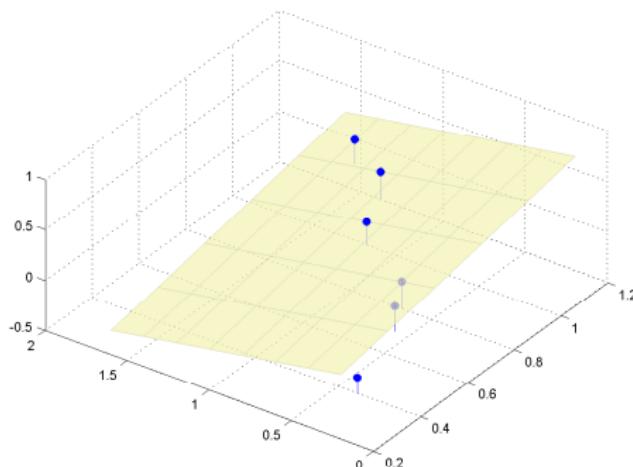


# Lineares Modell in zwei Variablen

## Anpassen einer Ausgleichs-Ebene: Beispiel aus der Matlab-Hilfe

Angenommen, eine Größe  $y$  hängt von zwei Parametern  $x_1$  und  $x_2$  ab. Folgende Messwerte liegen vor:

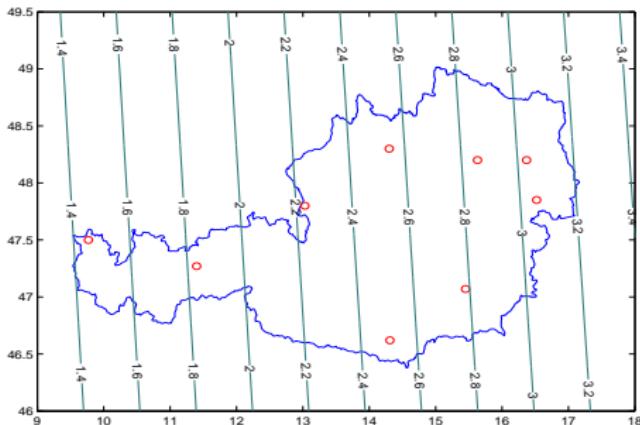
$x_1$ :	0.2	0.5	0.6	0.8	1.0	1.1
$x_2$ :	0.1	0.3	0.4	0.9	1.1	1.4
$y$ :	0.17	0.26	0.28	0.23	0.27	0.24



Wir nehmen ein lineares Modell  
 $y = a_0 + a_1x_1 + a_2x_2$  an und setzen die gegebenen Datentripel ein  
→ führt auf ein System von 6 linearen Gleichungen in den 3 unbekannten Koeffizienten  $a_0, a_1, a_2$ .  
→ Kleinstes-Quadrat-Lösung liefert Ebene mit „bestmöglicher“ Anpassung an Daten

# Lineares Modell in zwei Variablen

Beispiel: Magnetische Deklinationswerte 2008.5 in Österreich



Wien	$3^{\circ}$	00'
Eisenstadt	$3^{\circ}$	02'
St. Pölten	$2^{\circ}$	54'
Graz	$2^{\circ}$	47'
Linz	$2^{\circ}$	33'
Klagenfurt	$2^{\circ}$	30'
Salzburg	$2^{\circ}$	15'
Innsbruck	$1^{\circ}$	53'
Bregenz	$1^{\circ}$	24'

Daten: ZAMG

## Kleinste-Quadrat-Anpassung

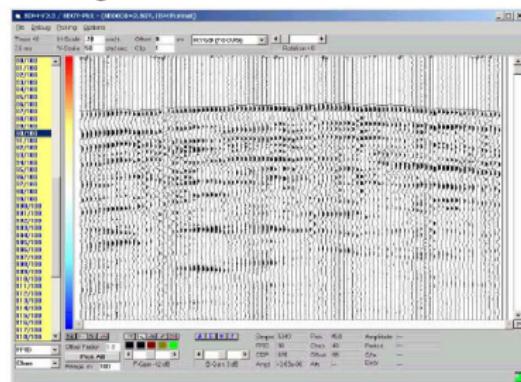
liefert Modell:

$$\delta = -2.0987 + 0.2365\lambda + 0.0261\phi$$

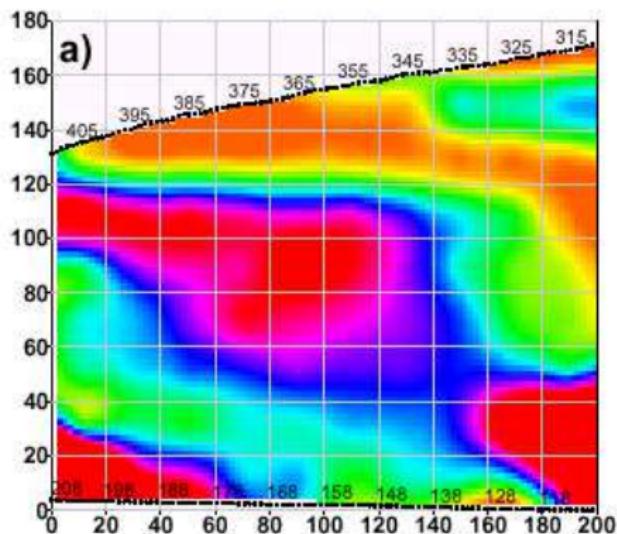
# Seismik bestimmt Gesteinsparameter

Aus Laufzeiten seismischer Wellen zwischen verschiedenen Sendern und Empfängern lässt sich durch *tomographische Inversion* ein Modell des Gesteinskörpers rekonstruieren. Erfordert *Lösung großer überbestimmter Gleichungssysteme*.

Erfassung der Ersteinsatzlaufzeiten



Inversionsmodell

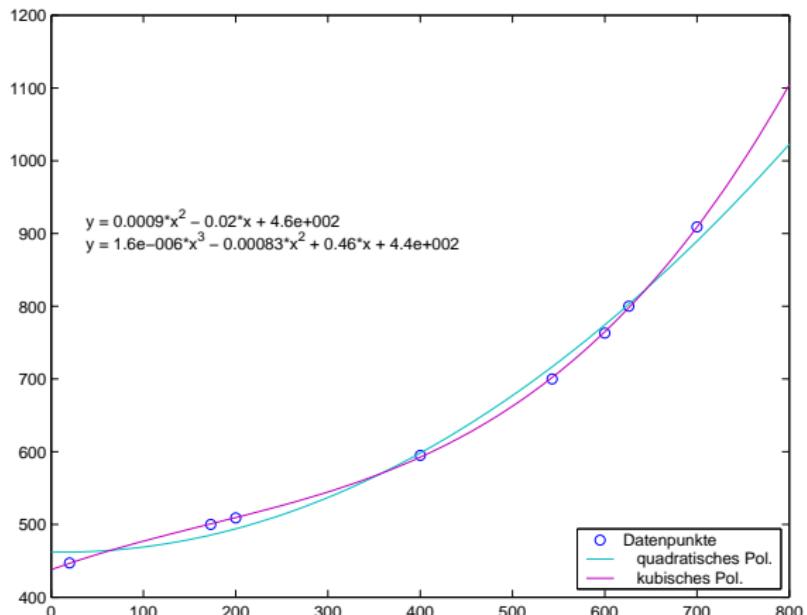


Quelle: Messungen über dem Semmering-Tunnel, Dr. Rudolf Fruhwirt, Angew. Geophysik-Joanneum Research, Leoben, 2000.

# Anpassen eines Polynoms an Datenpunkte

Spezifische Wärmekapazität  
von kohlenstoffarmem Stahl in  
 $J/kg\text{ K}$  für  $20\text{ C} \leq T \leq 700\text{ C}$

$T$	$c_p$
20	447
173	500
200	509
400	595
543	700
600	763
626	800
700	909



Die Abbildung illustriert *polynomiale Regression* (quadratisch und kubisch) an die gegebenen Datenpunkte.

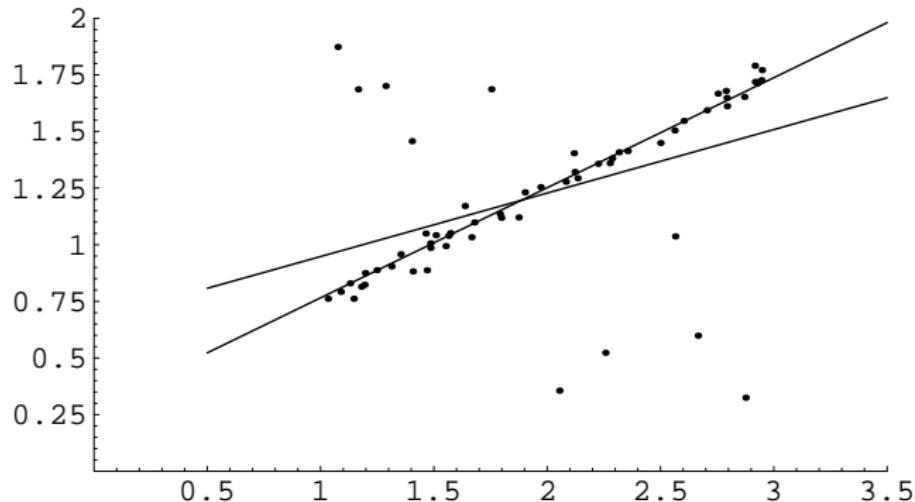
# Was dabei schiefgehen kann

Remember Murphy's Law: "If anything can go wrong, it will"

- Normalengleichungen für größere  $n$  schlecht konditioniert
- Abhilfe: Daten skalieren. Anderere Lösungswege ( $QR$ -Zerlegung, Singulärwertzerlegung), andere Ansatzfunktionen (Orthogonalpolynome)
- Methode der kleinsten Quadrate wird durch Ausreißer stark irritiert
- Abhilfe: Robuste Methoden, Minimierung der Summe der absoluten Fehler (Minimierung in der 1-Norm statt in der 2-Norm)

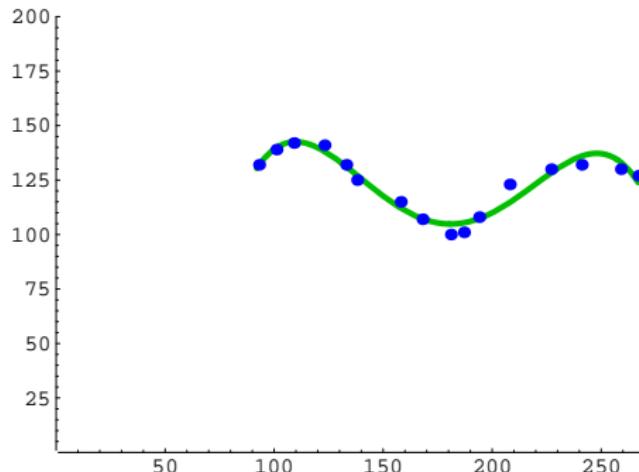
# Lineare Regression

Einfacher Spezialfall der polynomialen Regression



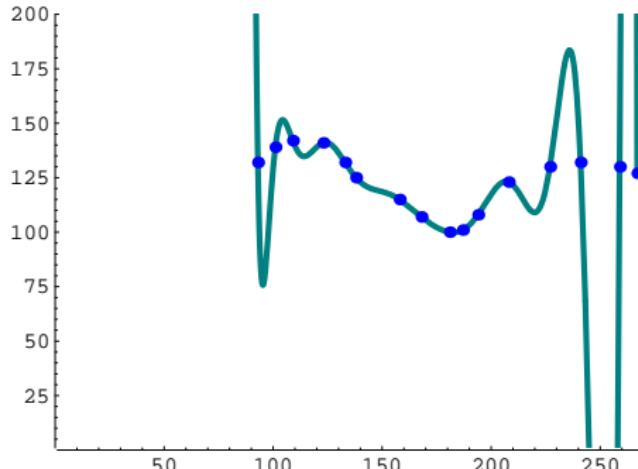
Anpassen einer Geraden an Datenpunkte. Die Ausgleichsgerade nach der Methode der *kleinsten Quadrate* lässt sich von den wenigen Ausreisern stark ablenken. Minimieren des *absoluten Fehlers* legt eine wesentlich plausiblere Gerade durch die Daten.

# Approximation durch polynomiale Regression



16 Datenpunkte sind gegeben. Ein Approximationspolynom vierten Grades modelliert den Verlauf der Daten ganz passabel. Es hängt vom Modell ab, ob es Sinn macht, mehr Parameter (höheren Grad) zu verwenden. Ein Polynom 15. Grades (16 freie Parameter) könnte die Daten exakt modellieren, aber...

# Datenanpassung mit zu hohem Polynomgrad



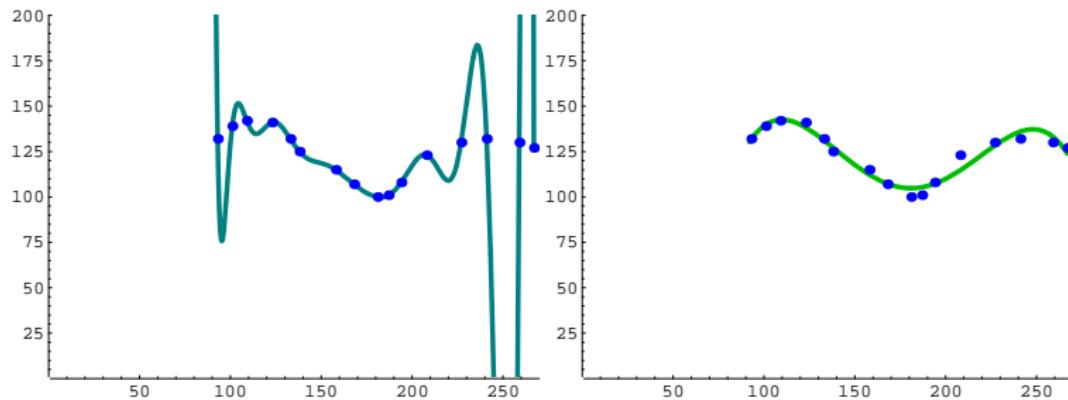
Der Fehler an den Datenpunkten verschwindet zwar, das Polynom oszilliert aber heftig.

Typisch für Polynome hohen Grades. Nur *sehr* glatte Funktionen lassen sich gut durch Polynome hohen Grades gut annähern, und auch das nur in kleinen Bereichen (Beispiel: Potenzreihen)

# Woher die Daten kommen

Ob eine Approximation ausreichend gut ist, hängt unter anderem auch davon ab, was die Daten beschreiben sollen...

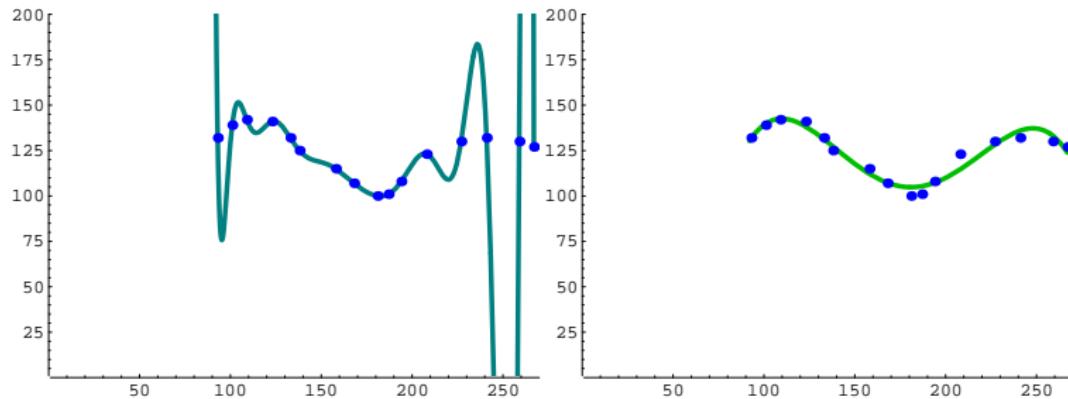
In diesem Fall sind es Punkte in einer kurvenreichen Computergraphik:



# Woher die Daten kommen

Ob eine Approximation ausreichend gut ist, hängt unter anderem auch davon ab, was die Daten beschreiben sollen...

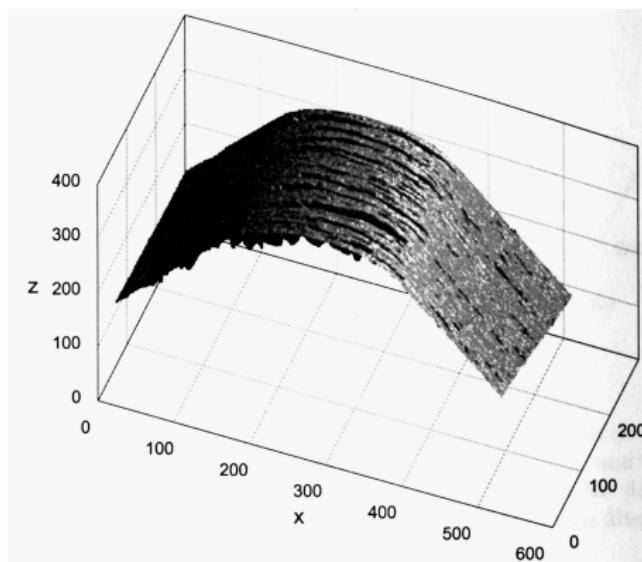
In diesem Fall sind es Punkte in einer kurvenreichen Computergraphik:



# Erkennen von Oberflächen-Defekten

Ein Anwendungsbeispiel (MUL Dissertation Ingo Reindl, 2006)

Polynomiale Regression approximiert Oberflächen-Querschnitte



Pixeldaten aus Oberflächenerfassung  
der abgerundeten Kante eines  
rohgewalzten Stahlblocks

Polynom → Referenzquerschnitt  
Abweichung → Oberflächendefekte

# Überbestimmte nichtlineare Systeme

Beispiel: Standortbestimmung durch Trilateration

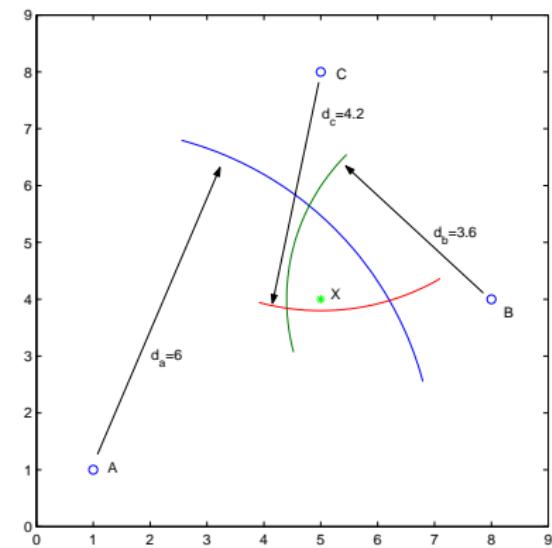
Die Abstände von drei festen Punkten  $A, B, C$  zu einem unbekannten Punkt  $X$  sind (etwas ungenau) bekannt. Gesucht ist eine möglichst gute Positionsbestimmung.

$$\sqrt{(x_1 - 1)^2 + (x_2 - 1)^2} = 6$$

$$\sqrt{(x_1 - 8)^2 + (x_2 - 4)^2} = 3.6$$

$$\sqrt{(x_1 - 5)^2 + (x_2 - 8)^2} = 4.2$$

Den drei Gleichungen entsprechen drei Kreise im  $\mathbb{R}^2$ . Sie haben keinen gemeinsamen Schnittpunkt.

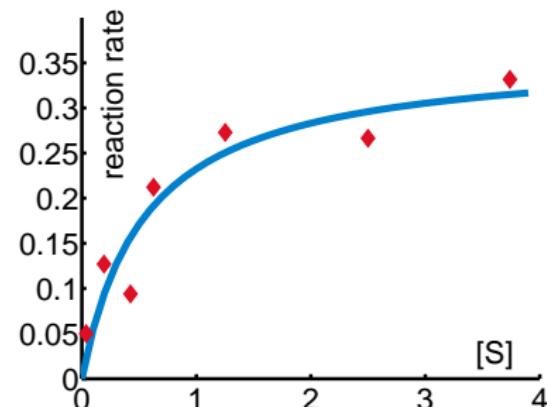


# Überbestimmte nichtlineare Systeme

Beispiel zu *Gauss-Newton-Algorithm* in der engl. Wikipedia

Biologie-Experimente liefern Datenpunkte zu Substanz-Konzentration und Reaktionsrate in einer durch Enzyme vermittelten Reaktion.  
Gesucht ist eine Modell-Kurve der Form

$$y = a \frac{x}{b + x}$$



Einsetzen der Datenpunkte für  $x$  und  $y$  ergibt sieben nichtlineare Gleichungen in den beiden Unbekannten  $a, b$ .

# Überbestimmte nichtlineare Systeme

Lösung durch Linearisierung und Iteration

$$\mathbf{f}(\mathbf{x}) = \mathbf{b}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{f}(\mathbf{x}), \mathbf{b} \in \mathbb{R}^m, \quad m > n$$

Ausgehend von einem Startvektor  $\mathbf{x}^{(0)}$  bestimmt man eine Korrektur  $\Delta\mathbf{x}$  aus der Lösung des überbestimmten *linearen Systems mit der Jacobimatrix  $D_f$*

$$D_f \cdot \Delta\mathbf{x} = \mathbf{b} - \mathbf{f}(\mathbf{x})$$

und gewinnt eine verbesserte Lösung  $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}$ .

Für die Konvergenz der Iteration kann *Unterrelaxation* (Dämpfung) notwendig sein:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \omega \Delta\mathbf{x} \text{ mit Unterrelaxationsfaktor } 0 < \omega \leq 1.$$

# Überbestimmte nichtlineare Systeme

Lösung durch Linearisierung und Iteration

$$\mathbf{f}(\mathbf{x}) = \mathbf{b}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{f}(\mathbf{x}), \mathbf{b} \in \mathbb{R}^m, \quad m > n$$

Ausgehend von einem Startvektor  $\mathbf{x}^{(0)}$  bestimmt man eine Korrektur  $\Delta\mathbf{x}$  aus der Lösung des überbestimmten *linearen Systems mit der Jacobimatrix  $D_f$*

$$D_f \cdot \Delta\mathbf{x} = \mathbf{b} - \mathbf{f}(\mathbf{x})$$

und gewinnt eine verbesserte Lösung  $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}$ .

Für die Konvergenz der Iteration kann *Unterrelaxation* (Dämpfung) notwendig sein:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \omega \Delta\mathbf{x} \text{ mit Unterrelaxationsfaktor } 0 < \omega \leq 1.$$

# Gliederung

## 1 Numerische Lineare Algebra

- Überbestimmte Systeme

## 2 Numerische Strömungsmechanik

- Iterative Lösungsverfahren

## 3 Diskrete Mathematik

# Numerische Strömungsmechanik

## Schritt 1: Physik liefert Modellgleichungen

### Massenerhaltung

Alles, was in ein Volumen hineinfließt, muss auf der anderen Seite wieder raus oder erhöht die Dichte.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0.$$

### Impulserhaltung

Eine vergleichsweise einfache Form (Euler-Gleichung)

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} + \frac{1}{\rho} \nabla p = 0$$

### Plus

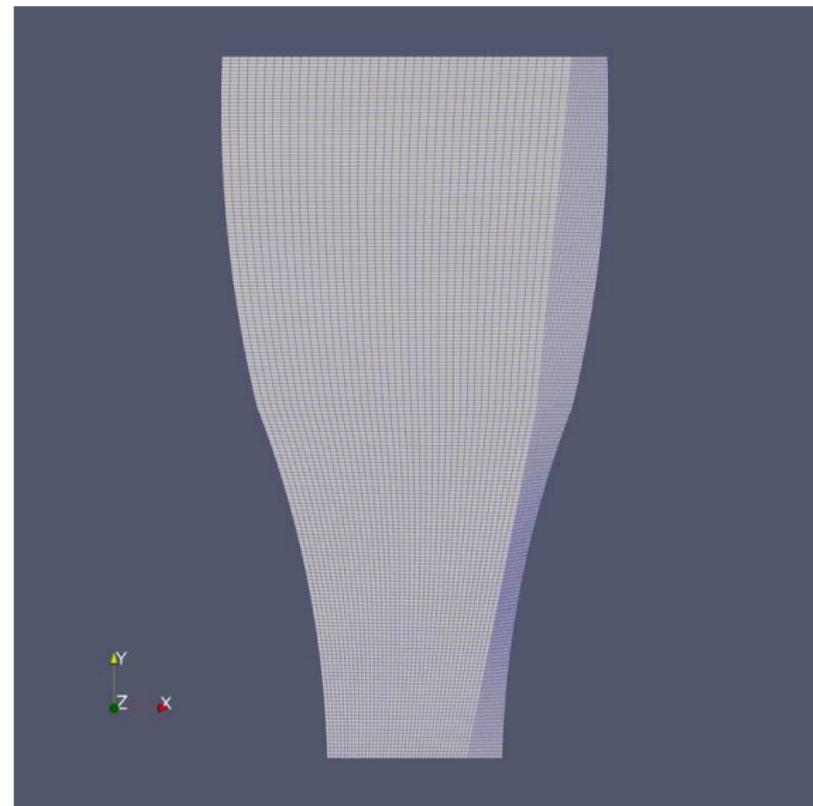
Energieerhaltung, Wärmeleitung, chemische Reaktionen, Phasenumwandlungen...

# Numerische Strömungsmechanik

## Schritt 2: Rechengitter zerlegt Geometrie in Zellen, Elemente, Punkte

Ein 2-D Gebiet, mit einem feinen Rechengitter überzogen.

Präprozessor modelliert Geometrie, erzeugt Gitter, setzt Rand- und Anfangsbedingungen



# Numerische Strömungsmechanik

Schritt 3: Finite-Volumen Methode fordert Erhaltungssätze für jede Gitterzelle

## Massen- und Impulsbilanz

Was während eines kleinen Zeitintervalls über die Zellgrenzen zu Nachbarzellen fließt oder von Nachbarzellen hereinfließt, ändert Werte in der Zelle

## Zeitschritt-Schleife

Ausgehend von Anfangsbedingungen wird Lösung von einem Zeitschritt zum nächsten berechnet.

## Andere Verfahren

Finite-Elemente-Methode

Finite-Differenzen-Methode

# Numerische Strömungsmechanik

## Schritt 4: Nachbearbeitung, Interpretation, Verifikation der Resultate

**Studenten-Projekt:** Zwei-Phasen-Strömung, Einström-Vorgang. Die Auflösung der Phasengrenze ist rechnerisch anspruchsvoll.

Simulation: OpenFOAM. Visualisierung: ParaView.

# Stationäre Wärmeleitung

Ein einfacheres Modell illustriert die Prinzipien

Quadratisches Gebiet mit  $3 \times 3$  Gitterzellen im Inneren.

Temperatur 100 am linken Rand, 0 an den anderen Rändern.

	0	0	0	
100	$T_7$	$T_8$	$T_9$	0
100	$T_4$	$T_5$	$T_6$	0
100	$T_1$	$T_2$	$T_3$	0
	0	0	0	

Gesucht sind die  
Gleichgewichts-Temperaturwerte  
 $T_1, T_2, \dots, T_9$  an den neun inneren  
Zell-Mittelpunkten

Wärmefluss über die Grenzfläche  $A$  zwischen zwei Zellen im Abstand  $\Delta x$ ,  
am Beispiel Zelle 5 → 6

$$Q_{56} = (T_5 - T_6) \frac{\lambda A}{\Delta x}$$

# Stationäre Wärmeleitung

## Bilanzgleichung für Zelle 5

	0	0	0	
100	$T_7$	$T_8$	$T_9$	0
100	$T_4$	$T_5$	$T_6$	0
100	$T_1$	$T_2$	$T_3$	0
	0	0	0	

Summe aller Wärmeflüsse im Gleichgewicht ist 0

$$Q_{56} + Q_{58} + Q_{54} + Q_{52} = 0$$

Der Faktor  $\frac{\lambda A}{\Delta x}$  kürzt sich heraus, es bleibt:

$$(T_5 - T_6) + (T_5 - T_8) + (T_5 - T_4) + (T_5 - T_2) = 0$$

zusammengefasst und sortiert:

$$-T_2 - T_4 + 4T_5 - T_6 - T_8 = 0$$

# Stationäre Wärmeleitung

## Alle neun Bilanzgleichungen

In den anderen Bilanzgleichungen treten Rand-Werte auf; die landen auf der rechten Seite.

	0	0	0	
100	$T_7$	$T_8$	$T_9$	0
100	$T_4$	$T_5$	$T_6$	0
100	$T_1$	$T_2$	$T_3$	0
	0	0	0	

$$\begin{aligned}4T_1 - T_2 - T_4 &= 100 \\-T_1 + 4T_2 - T_3 - T_5 &= 0 \\-T_2 + 4T_3 - T_6 &= 0 \\-T_1 + 4T_4 - T_5 - T_7 &= 100 \\-T_2 - T_4 + 4T_5 - T_6 - T_8 &= 0 \\-T_3 - T_5 + 4T_6 - T_9 &= 0 \\-T_4 + 4T_7 - T_6 &= 100 \\-T_5 - T_7 + 4T_8 - T_6 &= 0 \\-T_6 - T_8 + 4T_9 &= 0\end{aligned}$$

# Stationäre Wärmeleitung

## Matrix-Struktur des Gleichungssystems

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & & & 4 & -1 & 0 & -1 & & \\ -1 & & & -1 & 4 & -1 & & -1 & \\ -1 & & & -1 & 0 & -1 & 4 & & -1 \\ & & & -1 & & & 4 & -1 & 0 \\ & & & -1 & & & -1 & 4 & -1 \\ & & & & & -1 & 0 & -1 & 4 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \\ 100 \\ 0 \\ 0 \\ 100 \\ 0 \\ 0 \end{bmatrix}$$

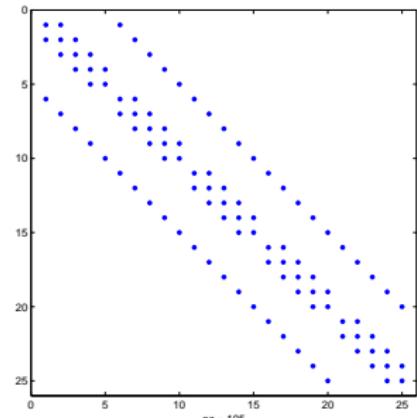
(Matrix-Elemente 0 sind großteils gar nicht mehr aufgeschrieben).

### Schablone

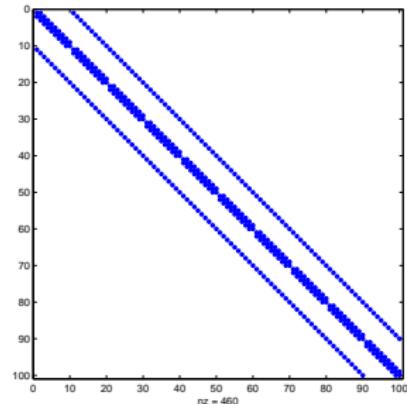
Jede Gleichung verknüpft höchstens 5 Unbekannte nach dem Muster  
„4-mal der Wert im Punkt minus Werte im Süden, Westen, Norden, Osten“

# Matrix-Struktur größerer Gitter

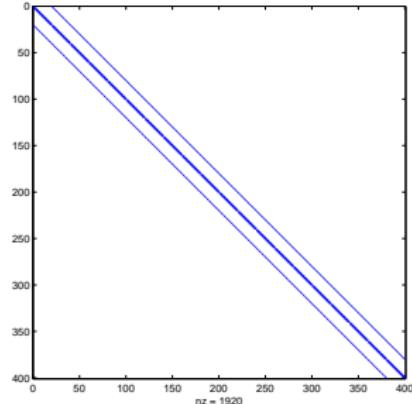
$5 \times 5$



$10 \times 10$



$20 \times 20$



## Schwach besetzte Matrix

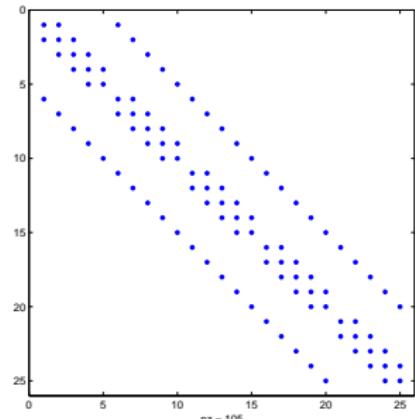
Die meisten Positionen sind mit Nullen besetzt

Für solche Matrizen gibt es spezielle Lösungsverfahren – dadurch lassen sich auch Systeme mit Millionen von Unbekannten lösen.

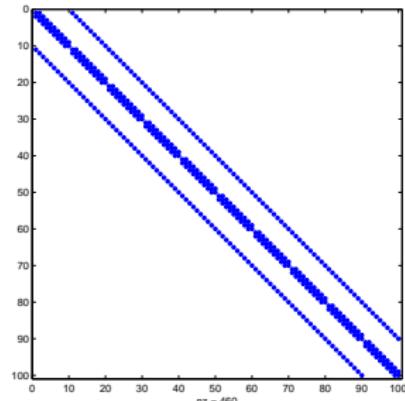
Bei voll besetzten Matrizen dieser Größe wäre Gauß-Elimination völlig unmöglich!

# Matrix-Struktur größerer Gitter

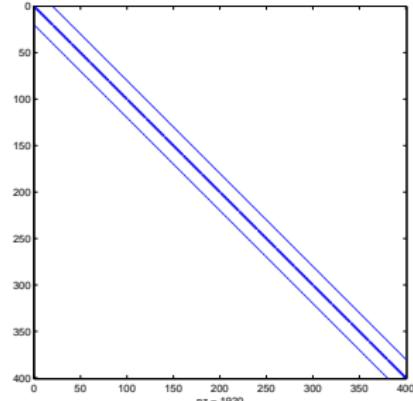
$5 \times 5$



$10 \times 10$



$20 \times 20$



## Schwach besetzte Matrix

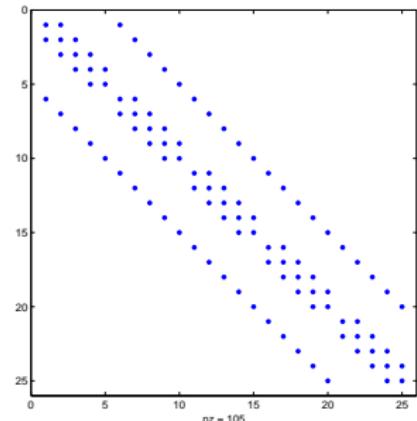
Die meisten Positionen sind mit Nullen besetzt

Für solche Matrizen gibt es spezielle Lösungsverfahren – dadurch lassen sich auch Systeme mit Millionen von Unbekannten lösen.

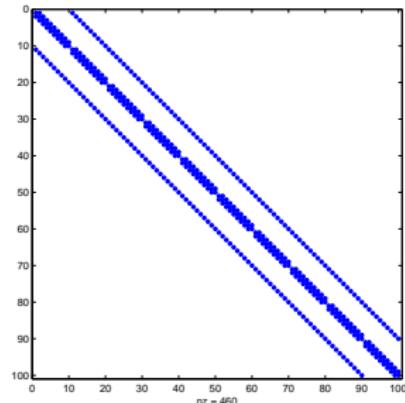
Bei voll besetzten Matrizen dieser Größe wäre Gauß-Elimination völlig unmöglich!

# Matrix-Struktur größerer Gitter

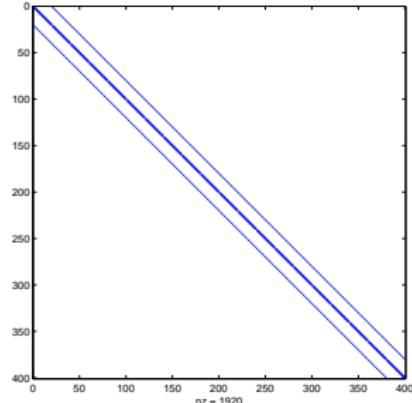
$5 \times 5$



$10 \times 10$



$20 \times 20$



## Schwach besetzte Matrix

Die meisten Positionen sind mit Nullen besetzt

Für solche Matrizen gibt es spezielle Lösungsverfahren – dadurch lassen sich auch Systeme mit Millionen von Unbekannten lösen.

Bei voll besetzten Matrizen dieser Größe wäre Gauß-Elimination völlig unmöglich!

# Iterative Lösungsverfahren

Wir lösen die Gleichungen nach den Diagonal-Termen auf

$$4T_1 - T_2 - T_4 = 100$$

$$-T_1 + 4T_2 - T_3 - T_5 = 0$$

$$-T_2 + 4T_3 - T_6 = 0$$

$$-T_1 + 4T_4 - T_5 - T_7 = 100$$

$$-T_2 - T_4 + 4T_5 - T_6 - T_8 = 0$$

$$-T_3 - T_5 + 4T_6 - T_9 = 0$$

$$-T_4 + 4T_7 - T_6 = 100$$

$$-T_5 - T_7 + 4T_8 - T_6 = 0$$

$$-T_6 - T_8 + 4T_9 = 0$$

$$4T_1 = 100 + T_2 + T_4$$

$$4T_2 = 0 + T_1 + T_3 + T_5$$

$$4T_3 = 0 + T_2 + T_6$$

$$4T_4 = 100 + T_1 + T_5 + T_7$$

$$4T_5 = 0 + T_2 + T_4 + T_6 + T_8$$

$$4T_6 = 0 + T_3 + T_5 + T_9$$

$$4T_7 = 100 + T_4 T_6$$

$$4T_8 = 0 + T_5 + T_7 + T_6$$

$$4T_9 = 0 + T_6 + T_8$$

# Iterative Lösungsverfahren

Wir lösen die Gleichungen nach den Diagonal-Termen auf

$$4T_1 - T_2 - T_4 = 100$$

$$T_1 = (100 + T_2 + T_4)/4$$

$$-T_1 + 4T_2 - T_3 - T_5 = 0$$

$$T_2 = (T_1 + T_3 + T_5)/4$$

$$-T_2 + 4T_3 - T_6 = 0$$

$$T_3 = (T_2 + T_6)/4$$

$$-T_1 + 4T_4 - T_5 - T_7 = 100$$

$$T_4 = (100 + T_1 + T_5 + T_7)/4$$

$$-T_2 - T_4 + 4T_5 - T_6 - T_8 = 0$$

$$T_5 = (T_2 + T_4 + T_6 + T_8)/4$$

$$-T_3 - T_5 + 4T_6 - T_9 = 0$$

$$T_6 = (T_3 + T_5 + T_9)/4$$

$$-T_4 + 4T_7 - T_6 = 100$$

$$T_7 = (100 + T_4 + T_6)/4$$

$$-T_5 - T_7 + 4T_8 - T_6 = 0$$

$$T_8 = (T_5 + T_7 + T_6)/4$$

$$-T_6 - T_8 + 4T_9 = 0$$

$$T_9 = (T_6 + T_8)/4$$

# Iterative Lösungsverfahren

Wir lösen die Gleichungen nach den Diagonal-Termen auf

$$4T_1 - T_2 - T_4 = 100$$

$$-T_1 + 4T_2 - T_3 - T_5 = 0$$

$$-T_2 + 4T_3 - T_6 = 0$$

$$-T_1 + 4T_4 - T_5 - T_7 = 100$$

$$-T_2 - T_4 + 4T_5 - T_6 - T_8 = 0$$

$$-T_3 - T_5 + 4T_6 - T_9 = 0$$

$$-T_4 + 4T_7 - T_6 = 100$$

$$-T_5 - T_7 + 4T_8 - T_6 = 0$$

$$-T_6 - T_8 + 4T_9 = 0$$

$$T_1 = (100 + T_2 + T_4)/4$$

$$T_2 = (T_1 + T_3 + T_5)/4$$

$$T_3 = (T_2 + T_6)/4$$

$$T_4 = (100 + T_1 + T_5 + T_7)/4$$

$$T_5 = (T_2 + T_4 + T_6 + T_8)/4$$

$$T_6 = (T_3 + T_5 + T_9)/4$$

$$T_7 = (100 + T_4 + T_6)/4$$

$$T_8 = (T_5 + T_7 + T_6)/4$$

$$T_9 = (T_6 + T_8)/4$$

# Iterative Lösungsverfahren

Wir lösen die Gleichungen nach den Diagonal-Termen auf

$$4T_1 - T_2 - T_4 = 100$$

$$-T_1 + 4T_2 - T_3 - T_5 = 0$$

$$-T_2 + 4T_3 - T_6 = 0$$

$$-T_1 + 4T_4 - T_5 - T_7 = 100$$

$$-T_2 - T_4 + 4T_5 - T_6 - T_8 = 0$$

$$-T_3 - T_5 + 4T_6 - T_9 = 0$$

$$-T_4 + 4T_7 - T_6 = 100$$

$$-T_5 - T_7 + 4T_8 - T_6 = 0$$

$$-T_6 - T_8 + 4T_9 = 0$$

$$T_1 = (100 + T_2 + T_4)/4$$

$$T_2 = (T_1 + T_3 + T_5)/4$$

$$T_3 = (T_2 + T_6)/4$$

$$T_4 = (100 + T_1 + T_5 + T_7)/4$$

$$T_5 = (T_2 + T_4 + T_6 + T_8)/4$$

$$T_6 = (T_3 + T_5 + T_9)/4$$

$$T_7 = (100 + T_4 + T_6)/4$$

$$T_8 = (T_5 + T_7 + T_6)/4$$

$$T_9 = (T_6 + T_8)/4$$

Einsetzen von Startwerten rechts liefert verbesserte Werte links → iteriere!

# Iterative Lösungsverfahren

Einfaches Prinzip: ersetze Wert der Gitterzelle durch Mittelwert der Nachbarn

- Einfach zu programmieren, wenn  $T$ -Werte (inklusive Randwerte) in 2D-Feld gespeichert sind:

```
for i=2:n-1
    for j=2:n-1
        T(i,j) = 0.25*(T(i-1,j)+T(i+1,j)...
                        +T(i,j-1)+T(i,j+1));
    end
end
```

- In dieser Form ein *Gauß-Seidel-Verfahren*.
- Es fehlen noch Konvergenztests, Abbruchbedingungen,...
- Allerdings nach heutigen Standards *viel* zu langsam

# Iterative Lösungsverfahren

Einfaches Prinzip: ersetze Wert der Gitterzelle durch Mittelwert der Nachbarn

- Einfach zu programmieren, wenn  $T$ -Werte (inklusive Randwerte) in 2D-Feld gespeichert sind:

```
for i=2:n-1
    for j=2:n-1
        T(i,j) = 0.25*(T(i-1,j)+T(i+1,j)...
                        +T(i,j-1)+T(i,j+1));
    end
end
```

- In dieser Form ein *Gauß-Seidel-Verfahren*.
- Es fehlen noch Konvergenztests, Abbruchbedingungen,...
- Allerdings nach heutigen Standards *viel* zu langsam

# Iterative Lösungsverfahren

Einfaches Prinzip: ersetze Wert der Gitterzelle durch Mittelwert der Nachbarn

- Einfach zu programmieren, wenn  $T$ -Werte (inklusive Randwerte) in 2D-Feld gespeichert sind:

```
for i=2:n-1
    for j=2:n-1
        T(i,j) = 0.25*(T(i-1,j)+T(i+1,j)...
                        +T(i,j-1)+T(i,j+1));
    end
end
```

- In dieser Form ein *Gauß-Seidel-Verfahren*.
- Es fehlen noch Konvergenztests, Abbruchbedingungen,...
- Allerdings nach heutigen Standards *viel* zu langsam

# Iterative Lösungsverfahren

Einfaches Prinzip: ersetze Wert der Gitterzelle durch Mittelwert der Nachbarn

- Einfach zu programmieren, wenn  $T$ -Werte (inklusive Randwerte) in 2D-Feld gespeichert sind:

```
for i=2:n-1
    for j=2:n-1
        T(i,j) = 0.25*(T(i-1,j)+T(i+1,j)...
                        +T(i,j-1)+T(i,j+1));
    end
end
```

- In dieser Form ein *Gauß-Seidel-Verfahren*.
- Es fehlen noch Konvergenztests, Abbruchbedingungen,...
- Allerdings nach heutigen Standards *viel* zu langsam

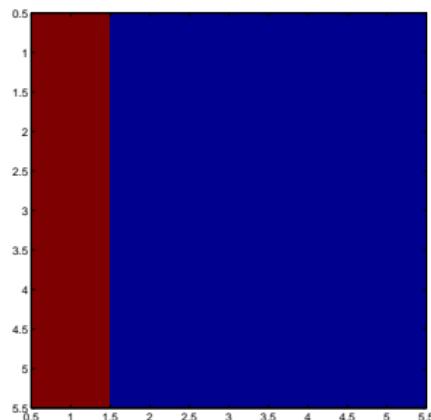
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	0	0	0	0
100	0	0	0	0
100	0	0	0	0
	0	0	0	

Anfangsbedingung



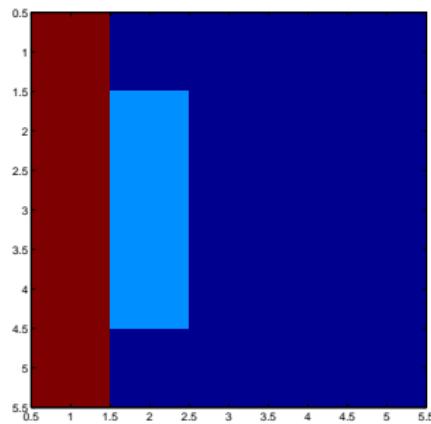
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	25	0	0	0
100	25	0	0	0
100	25	0	0	0
	0	0	0	

Schritt 1



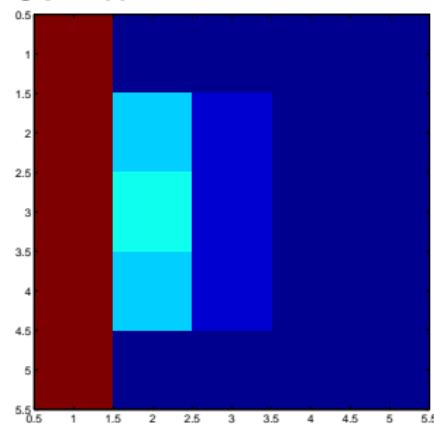
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	31	6	0	0
100	38	6	0	0
100	31	6	0	0
	0	0	0	

Schritt 2



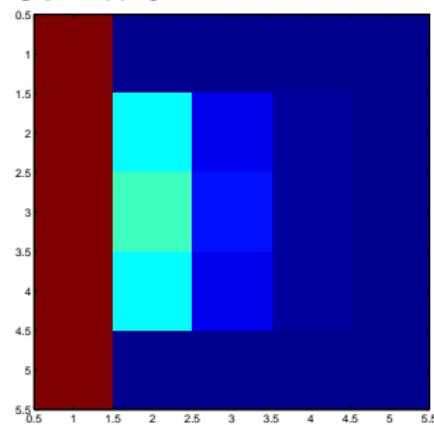
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	36	9	2	0
100	42	13	2	0
100	36	9	2	0
	0	0	0	

Schritt 3



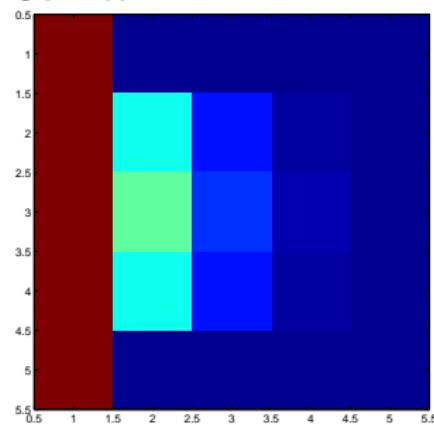
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	38	13	3	0
100	46	16	4	0
100	38	13	3	0
	0	0	0	

Schritt 4



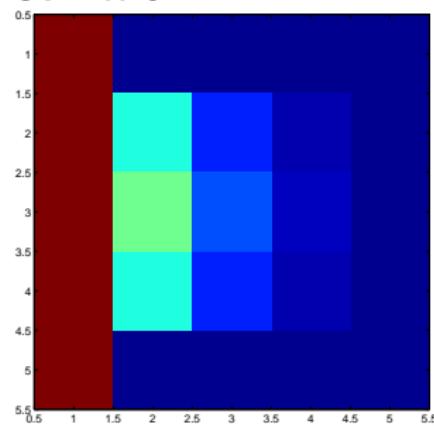
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	40	14	4	0
100	48	19	5	0
100	40	14	4	0
	0	0	0	

Schritt 5



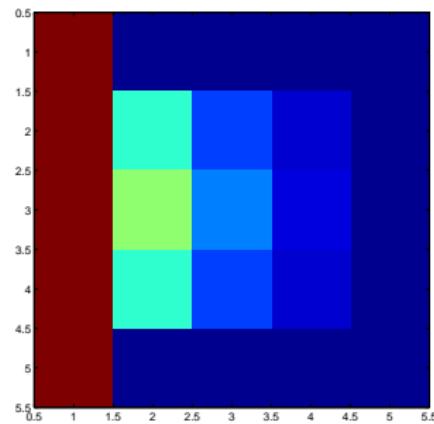
# Ablauf der Iteration

bei  $3 \times 3$  inneren Zellen

In dieser Form ein *Jacobi-Verfahren*: alle Gitter-Werte werden gleichzeitig aktualisiert. Noch langsamer als das Gauß-Seidel-Verfahren, aber für Parallel-Rechner geeignet.

	0	0	0	
100	42	18	7	0
100	52	24	9	0
100	42	18	7	0
	0	0	0	

Schritt 10



# Stationäre Temperaturverteilung

100 × 100-Gitter, insgesamt 10.000 Iterationsschritte

# Gliederung

## 1 Numerische Lineare Algebra

- Überbestimmte Systeme

## 2 Numerische Strömungsmechanik

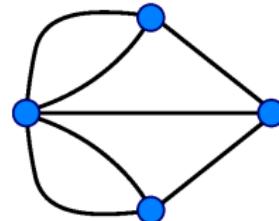
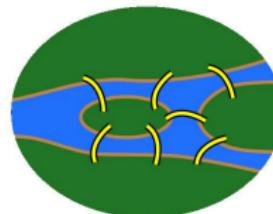
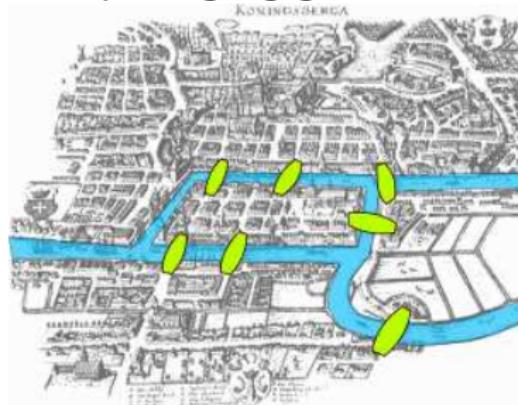
- Iterative Lösungsverfahren

## 3 Diskrete Mathematik

# Leonhard Euler und die Brücken von Königsberg

Kennen Sie sicher schon...

Ein Spaziergang genau einmal über jede der sieben Brücken?



## Eulerkreis

Genau dann, wenn von jedem Knoten eine gerade Anzahl von Kanten ausgehen.

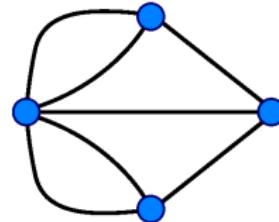
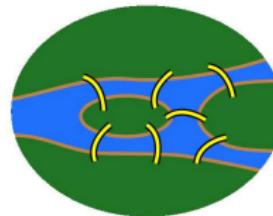
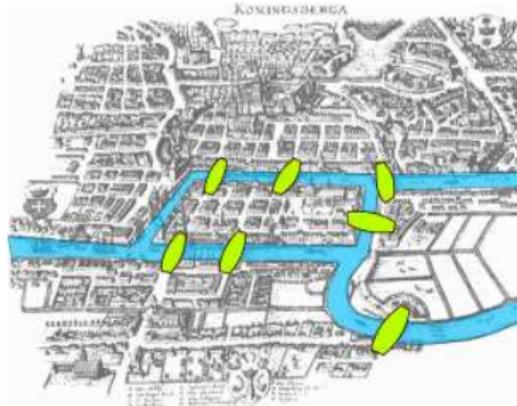
## Eulerweg

Wenn von allen Knoten bis auf zwei (nämlich Start- bzw. Endpunkt) eine gerade Anzahl von Kanten ausgehen.

# Leonhard Euler und die Brücken von Königsberg

Kennen Sie sicher schon...

Ein Spaziergang genau einmal über jede der sieben Brücken?



## Eulerkreis

Genau dann, wenn von jedem Knoten eine gerade Anzahl von Kanten ausgehen.

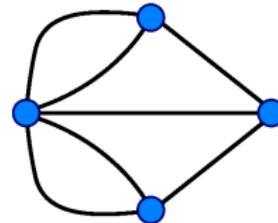
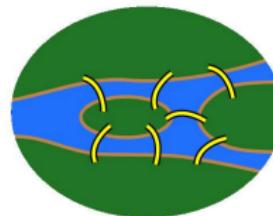
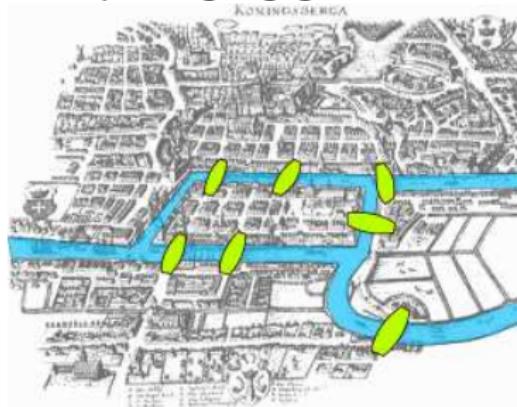
## Eulerweg

Wenn von allen Knoten bis auf zwei (nämlich Start- bzw. Endpunkt) eine gerade Anzahl von Kanten ausgehen.

# Leonhard Euler und die Brücken von Königsberg

Kennen Sie sicher schon...

Ein Spaziergang genau einmal über jede der sieben Brücken?



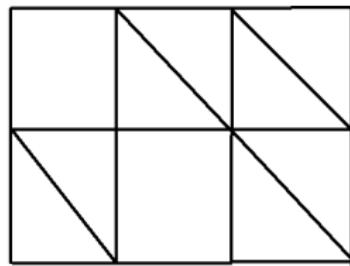
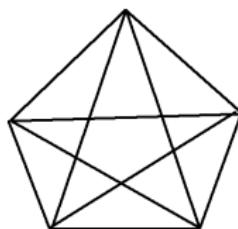
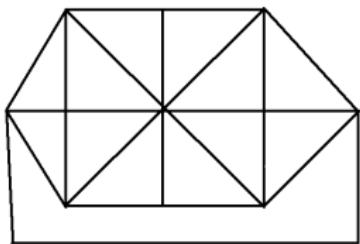
## Eulerkreis

Genau dann, wenn von jedem Knoten eine gerade Anzahl von Kanten ausgehen.

## Eulerweg

Wenn von allen Knoten bis auf zwei (nämlich Start- bzw. Endpunkt) eine gerade Anzahl von Kanten ausgehen.

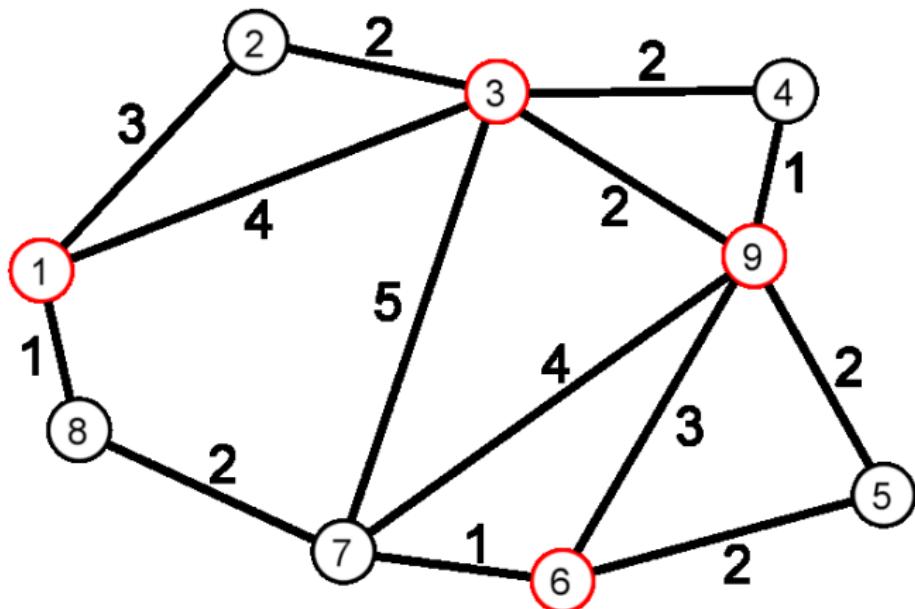
# Lassen sich die Figuren in einem Zug zeichnen?



# Schneeräumung und chinesische Briefträger

Jede Straße einmal, möglichst wenig Straßen mehrfach, Start=Ziel.

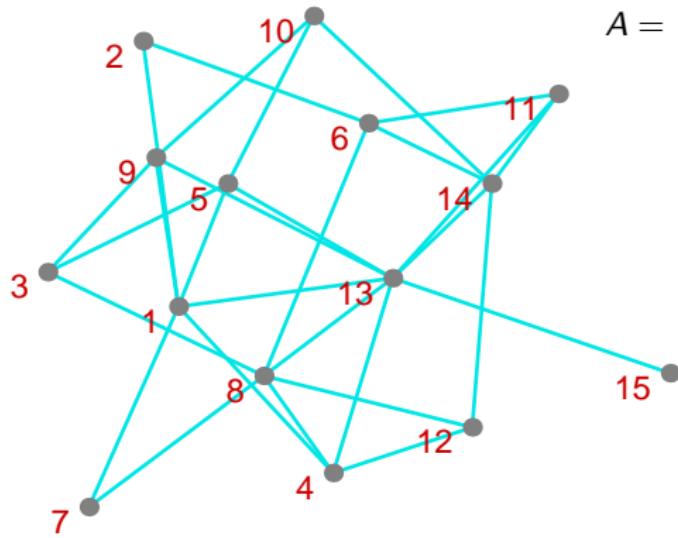
In diesem Graphen-Modell eines Stadtplans geben die Zahlen neben den Kanten jeweils die Länge der Straße an. Möglichst kurze Briefträger-Route?



# Wer ist wie gut vernetzt

## Ordne Knoten nach Wichtigkeit

Ein Netzwerk (Verkehrsverbindungen, verlinkte Seiten im Internet, soziales Netz... , mathematisch: ein Graph) lässt sich durch seine *Adjazenzmatrix* beschreiben

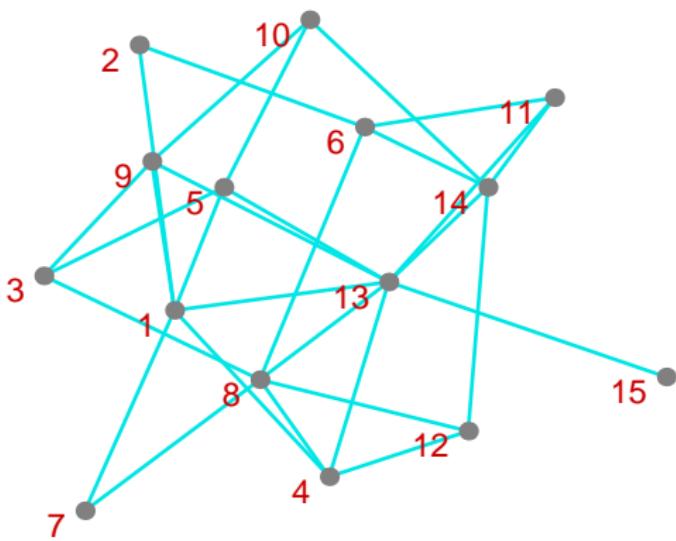


$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ \vdots & \ddots \end{bmatrix}$$

# Vektoriteration

## Eine Ranking-Methode

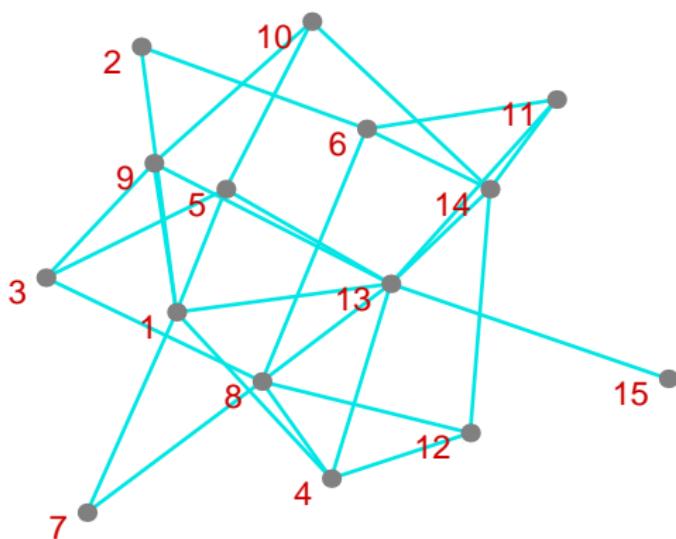
Beginne mit  $x^{(0)}$  = lauter-Einsen-Vektor. Bilde  $x^{(1)} = Ax^{(0)}$ , usw. Skaliere Ergebnis.



	74
	27
	41
	64
	56
	45
	32
$x =$	72
	56
	38
	46
	44
	100
	60
	22

# Ordne Knoten nach Wichtigkeit

Wer will das wissen?



Die Idee, Links zwischen Internet-Seiten auf diese Art mathematisch zu untersuchen, war 1997 Bestandteil einer akademischen Forschungsarbeit an der Stanford University.

Der Student, der damals dran gearbeitet hat (ein gewisser Larry Page), hat bis heute seine Dissertation nicht abgeschlossen. Aber Sie können ja googeln, ob er es inzwischen doch zu etwas gebracht hat.

