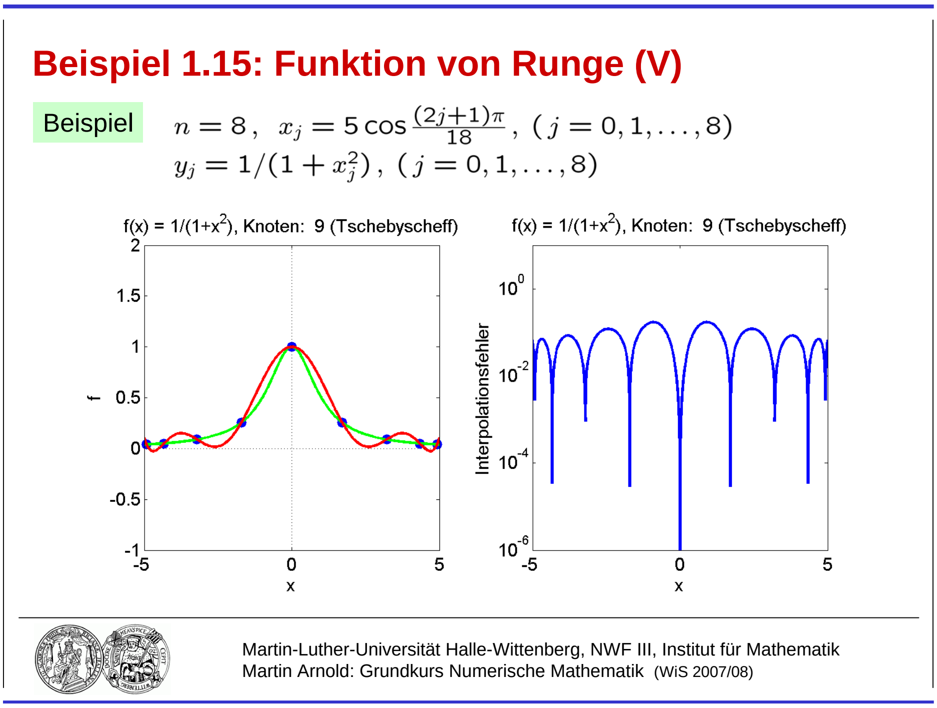


Abbildung 1.13: Interpolation der Funktion von Runge: $\Phi^{(8)}(x)$, Tschebyscheff–Stützstellen.

2 Lineare Gleichungssysteme

2.1 Gaußscher Algorithmus

Bemerkung 2.1 (Aufgabenstellung)

geg.: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $n = 10^1 \ldots 10^7$
ges.: Lösung x des linearen Gleichungssystems $Ax = b$

formal $x = A^{-1}b$, numerisch ungeeignet (Rechenaufwand zur Auswertung von A^{-1})

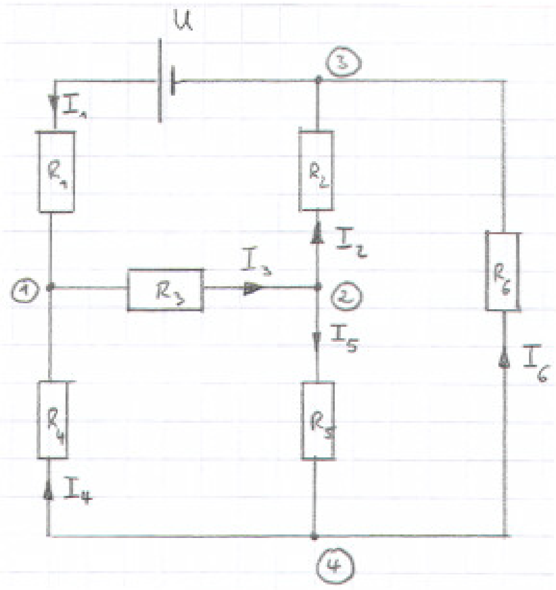
Lösbarkeit $x \in \mathbb{R}^n$ genau dann eindeutig bestimmt, wenn A regulär

$\boldsymbol{n = 2, n = 3}$ Lösung mittels Cramerscher Regel

Bemerkung 2.2 (Lineare Netzwerkmodelle)

Modellbildung komplexer Systeme (Technik, Umwelt)
Basisbausteine, verknüpft durch Ein-/Ausgangsbeziehungen und Erhaltungsgrößen

Beispiel elektrische Schaltungen, Chip-Design



Basisbaustein: Widerstand
Ohmsches Gesetz

$$U_i = R_i \cdot I_i\,, \quad (\,i = 1, 2, \ldots, 6\,)$$

1. Kirchhoffsche Regel
In jedem Knoten: Summe der zufließenden gleich Summe der abfließenden Ströme

2. Kirchhoffsche Regel
In jeder Masche: Summe der Spannungsabfälle gleich Summe der Quellspannungen

Ergebnis Lineares Gleichungssystem

$$\begin{array}{c} (1) \\ (2) \\ (3) \\ (4) \\ [\,(1),(4),(3)\,] \\ [\,(2),(3),(4)\,] \\ [\,(1),(2),(4)\,] \\ [\,(1),(2),(3)\,] \end{array} \qquad \begin{pmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ R_1 & 0 & 0 & -R_4 & 0 & R_6 \\ 0 & R_2 & 0 & 0 & -R_5 & -R_6 \\ 0 & 0 & R_3 & R_4 & R_5 & 0 \\ R_1 & R_2 & R_3 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ U \\ 0 \\ 0 \\ U \end{pmatrix}$$

Streicht man die redundanten Gleichungen $(4) = -(1) - (2) - (3)$ und

$$[\,(1),(2),(3)\,] = [\,(1),(4),(3)\,] + [\,(2),(3),(4)\,] + [\,(1),(2),(4)\,]\,,$$

so ergibt sich $Ax = b$ mit $A \in \mathbb{R}^{6\times 6}$, $x = (I_1, I_2, \ldots, I_6)^\top \in \mathbb{R}^6$, $b = (0,0,0,U,0,0)^\top \in \mathbb{R}^6$.

Beachte

- Anteil der Nichtnullelemente in A gering $\Rightarrow$ „schwach besetzte“ Matrizen
- Position der Nichtnullelemente a_{ij} in A ergibt sich aus der sog. Topologie der Schaltung

Bemerkung 2.3 (Gestaffelte Gleichungssysteme)

Spezialfall $Rx = z$ mit regulärer oberer Dreiecksmatrix $R = (r_{ij}) \in \mathbb{R}^{n\times n}$, d. h., $r_{ii} \neq 0$, $r_{ij} = 0$, $(\, i = 1, \ldots, n;\ j = 1, \ldots, i-1\,)$.

$$\left.\begin{array}{rcl} r_{11}x_1 + r_{12}x_2 + \ \ldots\ + r_{1n}x_n & = & z_1 \\ r_{22}x_2 + \ \ldots\ + r_{2n}x_n & = & z_2 \\ \ddots \quad \vdots & = & \vdots \\ r_{nn}x_n & = & z_n \end{array}\right\} Rx = z$$

Beispiel

$$\left.\begin{array}{rcl} 10x_1 - \ \ 7x_2 + \ \ 0\cdot x_3 & = & 7 \\ 2.5x_2 + \ \ 5\cdot x_3 & = & 2.5 \\ 6.2\cdot x_3 & = & 6.2 \end{array}\right\}$$

$$\Rightarrow \quad x_3 = \frac{6.2}{6.2} = 1\,, \quad x_2 = \frac{2.5 - 5\cdot 1}{2.5} = -1\,, \quad x_1 = \frac{7 + 7\cdot(-1)}{10} = 0$$

$$x = (\,0,\ -1,\ 1\,)^\top \quad \ldots \ \text{Rückwärtssubstitution}$$

allgemein

$$x_n = \frac{z_n}{r_{nn}}\,, \qquad x_i = \frac{z_i - \sum\limits_{j=i+1}^{n} r_{ij}x_j}{r_{ii}}\,, \quad (\, i = n-1, n-2, \ldots, 1\,)$$

Algorithmus

```
for i = n : -1 : 1
  | s := z_i
  | for j = (i+1) : n
  |   | s := s - r_ij x_j
  | x_i := s/r_ii
```

Matlab–Code

```
x = zeros(size(z));
x(n) = z(n)/r(n,n);
for i=n-1:-1:1,
   x(i) = ( z(i) - r(i,i+1:n) * x(i+1:n) ) / r(i,i);
end;
```

analog $Lx = z$ mit regulärer unterer Dreiecksmatrix $L = (l_{ij}) \in \mathbb{R}^{n \times n}$,
d. h., $l_{ii} \neq 0$, $l_{ij} = 0$, $(\, i = 1, \ldots, n;\ j = i+1, \ldots, n \,)$ $\Rightarrow$ „Vorwärtssubstitution".

Bemerkung 2.4 (Gaußscher Algorithmus)

Idee Gleichungssystem $Ax = b$ in äquivalentes gestaffeltes Gleichungssystem umformen durch

- Multiplikation einer Gleichung mit einer von Null verschiedenen Zahl,
- Addition des Vielfachen einer Gleichung zu einer anderen Gleichung und / oder
- Vertauschung von Gleichungen.

Beispiel

$$\left.\begin{aligned} 10x_1 - 7x_2 \phantom{{}+ 6x_3} &= 7 \\ -3x_1 + 2x_2 + 6x_3 &= 4 \\ 5x_1 - x_2 + 5x_3 &= 6 \end{aligned}\right\}$$

Schritt k Addiere Vielfache der k-ten Gleichung zu den Gleichungen $k+1, \ldots, n$, so dass in der k-ten Spalte unterhalb der Hauptdiagonale die Nichtnullelemente eliminiert werden $\rightsquigarrow$ **engl.:** Gaussian elimination

$\boldsymbol{k = 1}$

$$\begin{array}{rr} & 10x_1 - 7x_2 \phantom{{}+ 6x_3} = 7 \\ \mathrm{II}' = +\frac{3}{10} * \mathrm{I} + \mathrm{II} : & - 0.1x_2 + 6x_3 = 6.1 \\ \mathrm{III}' = -\frac{5}{10} * \mathrm{I} + \mathrm{III} : & 2.5x_2 + 5x_3 = 2.5 \end{array}\Bigg\}$$

$\boldsymbol{k = 2}$

$$\begin{array}{rr} & 10x_1 - 7x_2 \phantom{{}+ 6x_3} = 7 \\ & - 0.1x_2 + 6x_3 = 6.1 \\ \mathrm{III}'' = +25 * \mathrm{II}' + \mathrm{III}' : & 155x_3 = 155 \end{array}\Bigg\}$$

Rücksubstitution $\Rightarrow$ $x = (\, 0, -1, 1 \,)^\top$.

Problem Hauptdiagonalelement $a_{kk}^{(k)} = 0$ im k-ten Eliminationsschritt

Lösung Vertauschung der k-ten Gleichung mit einer der Gleichungen $k+1, \ldots, n$ so, dass *Pivotelement* $a_{kk}^{(k)} \neq 0$ (stets möglich, falls A regulär).

Strategie Bestimme im k-ten Eliminationsschritt $p \in \{\, k, k+1, \ldots, n \,\}$ so, dass

$$|a_{pk}^{(k)}| = \max \{\, |a_{lk}^{(k)}| \;:\; l = k, k+1, \ldots, n \,\}$$

und vertausche k-te und p-te Gleichung $\Rightarrow$ (Spalten)–*Pivotisierung*
$\rightsquigarrow$ auch vorteilhaft zur Verringerung des Einflusses von Rundungsfehlern

Beispiel $k = 2$, tausche Gleichungen $\mathrm{II}' \rightleftharpoons \mathrm{III}'$

$$\left.\begin{array}{lrcl} & 10x_1 - \;\; 7x_2 & = & 7 \\ \widetilde{\mathrm{II}}' = \mathrm{III}' : & 2.5x_2 + 5x_3 & = & 2.5 \\ \widetilde{\mathrm{III}}' = \mathrm{II}' \;: & -\,0.1x_2 + 6x_3 & = & 6.1 \end{array}\right\}$$

$$\left.\begin{array}{lrcl} & 10x_1 - \;\; 7x_2 \qquad\quad & = & 7 \\ & 2.5x_2 + \;\; 5x_3 & = & 2.5 \\ \widetilde{\mathrm{III}}'' = +\frac{1}{25} * \widetilde{\mathrm{II}}' + \widetilde{\mathrm{III}}' : & 6.2x_3 & = & 6.2 \end{array}\right\}$$

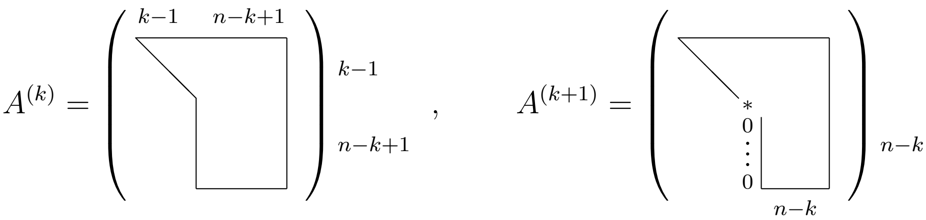
Algorithmus

```
for k = 1 : n − 1
  | p := k;  s := |a_kk|
  | for i = k + 1 : n
  |   | if |a_ik| > s then p := i ;  s := |a_ik|
  | for j = k : n
  |   | s := a_kj ;  a_kj := a_pj ;  a_pj := s
  | s := b_k ;  b_k := b_p ;  b_p := s
  | for i = k + 1 : n
  |   | l_ik := a_ik/a_kk;
  |   | b_i := b_i − l_ik · b_k
  |   | for j = k + 1 : n
  |   |   | a_ij := a_ij − l_ik · a_kj
```

Bemerkung 2.5 (LU–Zerlegung)

***k*-ter Eliminationsschritt** in Matrixschreibweise (ohne Pivotisierung)

$A^{(k)} \rightsquigarrow A^{(k+1)}$ mit



$$A^{(k+1)} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & -l_{k+1,k} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -l_{n,k} & & & 1 \end{pmatrix} \cdot A^{(k)} = (I - L^{(k)})A^{(k)}$$

Bemerkung 2.4: Gaußscher Algorithmus

$$
\begin{array}{l}
\texttt{for } k = 1 : n-1 \\
\quad p := k; \quad s := |a_{kk}| \\
\quad \texttt{for } i = k+1 : n \\
\quad\quad \texttt{if } |a_{ik}| > s \texttt{ then } p := i\,; \quad s := |a_{ik}| \\
\quad \texttt{for } j = k : n \\
\quad\quad s := a_{kj}\,; \quad a_{kj} := a_{pj}\,; \quad a_{pj} := s \\
\quad s := b_k\,; \quad b_k := b_p\,; \quad b_p := s \\
\quad \texttt{for } i = k+1 : n \\
\quad\quad l_{ik} := a_{ik}/a_{kk}; \\
\quad\quad b_i := b_i - l_{ik} \cdot b_k \\
\quad\quad \texttt{for } j = k+1 : n \\
\quad\quad\quad a_{ij} := a_{ij} - l_{ik} \cdot a_{kj}
\end{array}
$$

Martin-Luther-Universität Halle-Wittenberg, NWF III, Institut für Mathematik
Martin Arnold: Grundkurs Numerische Mathematik (WiS 2007/08)

Abbildung 2.1: Gaußscher Algorithmus.

mit

$$
L^{(k)} = \begin{pmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & l_{k+1,k} & 0 & & \\ & & & \vdots & & \ddots & \\ & & & l_{n,k} & & & 0 \end{pmatrix}, \quad A^{(1)} := A\,, \quad A^{(n)} =: U \quad \text{(obere Dreiecksmatrix)},
$$

$$
(I - L^{(n-1)}) \cdots (I - L^{(1)}) A = U\,.
$$

Auflösen nach A

$$
A = LU \quad \text{mit} \quad L := (I - L^{(1)})^{-1} \cdots (I - L^{(n-1)})^{-1}\,.
$$

Wegen $L^{(i)} L^{(j)} = 0$, ($i \le j$), ist

$$
(I - L^{(i)})(I + L^{(i)}) = I - L^{(i)} + L^{(i)} = I \quad \Rightarrow \quad (I - L^{(i)})^{-1} = (I + L^{(i)})\,.
$$

Außerdem erhält man $(I + L^{(i)})(I + L^{(j)}) = I + L^{(i)} + L^{(j)}$, ($i \le j$), also insgesamt

$$
L = (I + L^{(1)}) \cdots (I + L^{(n-1)}) = I + L^{(1)} + L^{(2)} + \ldots + L^{(n-1)} \quad \ldots \text{ untere Dreiecksmatrix.}
$$

Ergebnis Gauß–Algorithmus berechnet *LU–Zerlegung* von A: $A = L \cdot U$ mit der oberen Dreiecksmatrix U aus dem gestaffelten linearen Gleichungssystem und der unteren Dreiecksmatrix L, die die Eliminationskoeffizienten l_{ik} enthält und deren Hauptdiagonalelemente $= 1$ sind.

praktisch Abspeicherung der Nicht–Diagonalelemente von L unterhalb der Hauptdiagonalen von A.

Pivotisierung Spaltenpivotisierung $\Rightarrow$ $LU = PA$ mit *Permutationsmatrix* P

Beispiel wie oben

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{P} \underbrace{\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ -3/10 & -1/25 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix}}_{U}$$

Lösung linearer Gleichungssysteme

Schritt 1 Berechne mittels Gauß–Algorithmus mit Spaltenpivotisierung die LU–Zerlegung $PA = LU$.

Für jede rechte Seite: **Schritt 2** Vorwärtssubstitution

$$Ly = Pb \rightsquigarrow y$$

Schritt 3 Rückwärtssubstitution

$$Ux = y \rightsquigarrow x$$

Software
- Matlab–Kommando `lu`
- `LAPACK` (`FORTRAN`, `C`), über `http://www.netlib.org`
  `DGETRF`, `DGETRS`

Rechenaufwand Gemessen in flops ... Floating point operations
(1 Gleitpunktaddition, 1 Gleitpunktmultiplikation)

Schritt 1 $\frac{n^3}{3} + \mathcal{O}(n^2)$ Rechenoperationen, vgl. Bemerkung 1.7

Schritt 2+3 jeweils $\frac{n^2}{2} + \mathcal{O}(n)$ Rechenoperationen $\Rightarrow$ $n^2 + \mathcal{O}(n)$ Rechenoperationen pro linearem Gleichungssystem

Bemerkung 2.6 (Symmetrische Koeffizientenmatrizen, Cholesky–Zerlegung)
Wichtiger Spezialfall: $Ax = b$ mit $A = A^\top$, insbesondere auch symmetrische, positiv definite Koeffizientenmatrizen A, d. h.

$$x^\top Ax > 0\,, \quad (\,x \in \mathbb{R}^n \setminus \{0\}\,)\,.$$

$\boldsymbol{A = A^\top}$ Gauß–Algorithmus ohne Pivotisierung $\rightsquigarrow$ $A = L \cdot U$

Sei $D := \operatorname*{diag}_{1\le i\le n} u_{ii}$ $\Rightarrow$ $D^{-1}U$ ist obere Dreiecksmatrix mit Hauptdiagonalelementen $= 1$.

$$A^\top = (L \cdot D \cdot D^{-1}U)^\top = (D^{-1}U)^\top DL^\top$$

Aus der Eindeutigkeit der LU–Zerlegung (!) folgt $D^{-1}U = L^\top$, also $A = LDL^\top$.
Berechnung mit $\frac{n^3}{6} + \mathcal{O}(n^2)$ Rechenoperationen möglich.
Pivotisierung: gleichzeitiger Zeilen- und Spaltentausch, um Symmetrie zu erhalten.

***A* symmetrisch, positiv definit** Man zeigt, dass der Gauß–Algorithmus ohne Pivotisierung stets durchführbar ist. Wegen $0 < y^\top Ay$ für $y := (L^\top)^{-1}x$ ist

$$0 < y^\top Ay = ((L^\top)^{-1}x)^\top (LDL^\top)((L^\top)^{-1}x) = (L^\top (L^\top)^{-1}x)^\top D(L^\top (L^\top)^{-1}x) = x^\top Dx\,,$$

also ist auch D positiv definit: $D = \underset{1\le i\le n}{\operatorname{diag}} d_i$ mit $d_i > 0$.

Cholesky–Zerlegung von *A*

$$A = \hat{L}\hat{L}^\top \quad \text{mit} \quad \hat{L} := L \cdot D^{1/2}\,, \quad D^{1/2} = \operatorname{diag}_i \sqrt{d_i}$$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} \hat{l}_{11} & & & \\ \hat{l}_{21} & \hat{l}_{22} & & \\ \vdots & \vdots & \ddots & \\ \hat{l}_{n1} & \hat{l}_{n2} & \cdots & \hat{l}_{nn} \end{pmatrix} \begin{pmatrix} \hat{l}_{11} & \hat{l}_{21} & \cdots & \hat{l}_{n1} \\ & \hat{l}_{22} & \cdots & \hat{l}_{n2} \\ & & \ddots & \vdots \\ & & & \hat{l}_{nn} \end{pmatrix}$$

Algorithmus

$\texttt{for}\ k = 1 : n$

$\quad \hat{l}_{kk} := \Big(a_{kk} - \sum_{j=1}^{k-1} \hat{l}_{kj}^2\Big)^{1/2}$

$\quad \texttt{for}\ i = (k+1) : n$

$\quad\quad \hat{l}_{ik} := \Big(a_{ik} - \sum_{j=1}^{k-1} \hat{l}_{ij}\hat{l}_{kj}\Big) \,/\, \hat{l}_{kk}$

Vorwärts- / Rückwärtssubstitution
wie im allgemeinen Fall, beachte jedoch, dass nur $\hat{L}$, nicht $\hat{L}^\top$ abgespeichert wird.

Rechenaufwand $\frac{n^3}{6} + \mathcal{O}(n^2)$ Rechenoperationen, n Quadratwurzeln

2.2 Lineare Ausgleichsrechnung

Bemerkung 2.7 (Methode der kleinsten Quadrate)

geg.: $A \in \mathbb{R}^{m\times n}$, $m > n$, $\operatorname{rank}(A) = n$, $b \in \mathbb{R}^m$
ges.: „Lösung“ $x \in \mathbb{R}^n$ von $Ax = b$

Methode der kleinsten Quadrate (Gauß)
Da i. Allg. keine *klassische Lösung* $x \in \mathbb{R}^n$ mit $Ax - b = 0$ existiert, sucht man als *verallgemeinerte Lösung* ein $x \in \mathbb{R}^n$ so, dass

$$\|Ax - b\|_2 \to \min\,. \qquad (*)$$

Hierbei bezeichnet $\|v\|_2 := \sqrt{v^\top v} = \Big(\sum_{i=1}^m v_i^2\Big)^{1/2}$ die *euklidische Vektornorm* des Vektors $v = (v_i)_{i=1}^m \in \mathbb{R}^m$, vgl. Abschnitt 3.2.

Eigenschaft: Für Vektoren $y = (y_i) \in \mathbb{R}^n,\ z = (z_i) \in \mathbb{R}^{m-n}$ gilt

$$\|\begin{pmatrix} y \\ z \end{pmatrix}\|_2^2 = \sum_{i=1}^n y_i^2 + \sum_{i=1}^{m-n} z_i^2 = \|y\|_2^2 + \|z\|_2^2 \,.$$

Das *Kleinste–Quadrate–Problem* $(*)$ ist äquivalent zu

$$\|Ax - b\|_2^2 = \sum_{i=1}^m \Big(\sum_{j=1}^n a_{ij}x_j - b_i\Big)^2 \to \min \,.$$

Notwendige Bedingung für Minimum

$$0 \overset{!}{=} \frac{\partial}{\partial x_k}\|Ax - b\|_2^2 = 2\sum_{i=1}^m \Big(\sum_{j=1}^n a_{ij}x_j - b_i\Big) a_{ik} \,, \quad (\, k = 1, \dots, n \,) \,,$$

$$\sum_{i=1}^m a_{ik}\Big(\sum_{j=1}^n a_{ij}x_j\Big) = \sum_{i=1}^m a_{ik} b_i \,, \quad (\, k = 1, \dots, n \,) \,,$$

Gaußsche Normalgleichungen $\boxed{A^\top A x = A^\top b}$

Wegen $A^\top A = (A^\top A)^\top \in \mathbb{R}^{n\times n}$ und $\operatorname{rank}(A) = n$ ist $A^\top A$ symmetrisch und positiv definit:

$$\xi \in \mathbb{R}^n \setminus \{0\} \quad\Rightarrow\quad \xi^\top(A^\top A)\xi = (\xi^\top A^\top)(A\xi) = (A\xi)^\top(A\xi) = \|A\xi\|_2^2 > 0 \,,$$

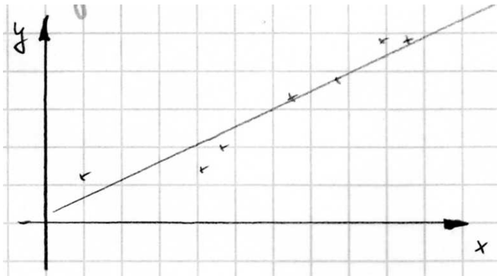
denn $A\xi \neq 0$ wegen $\xi \neq 0$ und $\operatorname{rank}(A) = n$.

Lösung der Normalgleichungen mittels Cholesky–Zerlegung

Problem Bei Verwendung der Gaußschen Normalgleichungen reagiert die numerische Lösung oft sehr empfindlich auf Rundungsfehler.

Alternative Orthogonalisierungsverfahren, vgl. Bemerkung 2.10.

Beispiel 2.8 (Lineare Regression)



geg.: Messdaten (x_i, y_i), $(\, i = 1, \dots, m\,)$, mit Messfehlern behaftet
ges.: Gerade $y = a + bx$, die die Messwerte „möglichst gut“ approximiert:
$y_i \approx a + bx_i$, $(\, i = 1, \dots, m\,)$

Ansatz $\displaystyle\sum_{i=1}^{m} (a + bx_i - y_i)^2 \to \min$

Matrixschreibweise

$$A \begin{pmatrix} a \\ b \end{pmatrix} = y \quad \text{mit} \quad y = (y_1, \dots, y_m)^\top \in \mathbb{R}^m, \quad A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \in \mathbb{R}^{m \times 2}, \quad n = 2$$

Normalgleichungen $A^\top A \cdot \begin{pmatrix} a \\ b \end{pmatrix} = A^\top y$

$$\begin{pmatrix} m & \displaystyle\sum_{j=1}^{m} x_j \\ \displaystyle\sum_{j=1}^{m} x_j & \displaystyle\sum_{j=1}^{m} x_j^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{j=1}^{m} y_j \\ \displaystyle\sum_{j=1}^{m} x_j y_j \end{pmatrix} \qquad \Rightarrow \quad a, b \in \mathbb{R}$$

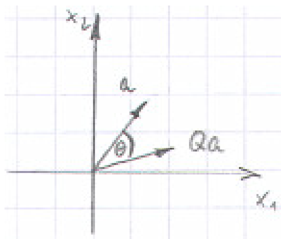
Bemerkung 2.9 (Orthogonale Transformationen)

Idee Verwende orthogonale Transformationen, um lineare Gleichungssysteme und lineare Ausgleichsprobleme in äquivalente Probleme einfacherer Gestalt umzuformen.

***n* = 2** Drehungen, Spiegelungen

hier Drehmatrizen

$$Q = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$



Literatur zu Spiegelungsmatrizen:
Stoer, Deuflhard/Hohmann

allgemein Givens–Drehungen, Householder–Spiegelungen

hier Givens–Drehungen

$$G_{kl} = \begin{array}{cc} \left(\begin{array}{ccccccccccc} 1 &&&&&&&&&& \\ & \ddots &&&&&&&&& \\ && 1 &&&&&&&& \\ &&& c &&&& s &&& \\ &&&& 1 &&&&&& \\ &&&&& \ddots &&&&& \\ &&&&&& 1 &&&& \\ &&& -s &&&& c &&& \\ &&&&&&&& 1 && \\ &&&&&&&&& \ddots & \\ &&&&&&&&&& 1 \end{array}\right) & \begin{array}{c} \\ \\ \\ l \\ \\ \\ \\ k \\ \\ \\ \\ \end{array} \\ \begin{array}{ccccccccccc} & & & l & & & & k & & & \end{array} & \end{array} \in \mathbb{R}^{m \times m}$$

mit $c = \cos\theta\,,\ \ s = \sin\theta\,,\ \ c^2 + s^2 = 1\,.$

Bestimme θ so, dass $\big(G_{kl}A\big)_{kl} = 0$:

$$a_{kl} \ \mapsto\ -s a_{ll} + c a_{kl} \ \overset{!}{=}\ 0 \quad \text{und} \quad c^2 + s^2 = 1\,.$$

$$|a_{kl}| > |a_{ll}| \ : \qquad \tau := \frac{a_{ll}}{a_{kl}}\,, \quad s := \frac{1}{\sqrt{1+\tau^2}}\,, \quad c := s \cdot \tau$$

$$|a_{kl}| \le |a_{ll}| \ : \qquad \tau := \frac{a_{kl}}{a_{ll}}\,, \quad c := \frac{1}{\sqrt{1+\tau^2}}\,, \quad s := c \cdot \tau$$

Bemerkung 2.10 (QR–Zerlegung)

geg.: $A \in \mathbb{R}^{m \times n}\,,\ \ m \ge n\,,\ \ \operatorname{rank}(A) = n$

Schrittweise Elimination der Nichtnullelemente a_{ij}, $(\,j < i\,)$ mittels Givens–Drehungen:

$$A \to G_{21}A \to G_{31}G_{21}A \to \ \ldots\ \to G_{m1}\cdots G_{31}G_{21}A \to G_{32}G_{m1}\cdots G_{31}G_{21}A \to \ \ldots\ \to R$$

mit

$$G_{mn}G_{m-1,n}\ \cdots\ G_{n+1,n}G_{m,n-1}\ \cdots\ G_{32}G_{m1}\ \cdots\ G_{31}G_{21}A = R = \begin{array}{cc} \begin{pmatrix} \widetilde{R} \\ \cdots \\ 0 \end{pmatrix} & \begin{array}{l} {\scriptstyle n} \\ \\ {\scriptstyle m-n} \end{array} \\ {\scriptstyle n} & \end{array}$$

QR–Zerlegung $A = QR$ mit der orthogonalen Matrix

$$Q := G_{21}^\top G_{31}^\top\ \cdots\ G_{m,n-1}^\top G_{n+1,n}^\top\ \cdots\ G_{m,n}^\top\,.$$

Wegen $n = \operatorname{rank}(A) = \operatorname{rank}(R) = \operatorname{rank}(\widetilde{R})$ ist $\widetilde{R} \in \mathbb{R}^{n \times n}$ regulär.

Lösung regulärer linearer Gleichungssysteme

1. QR–Zerlegung mittels Givens–Drehungen, $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ Rechenoperationen
2. $Ax = b \;\Leftrightarrow\; QRx = b \;\Leftrightarrow\; Rx = z\,,\;\; Qz = b$

   2a) $z = Q^\top b = G_{n,n-1}\,(G_{n,n-2}\,(G_{n-1,n-2}\,(\,\cdots\, G_{31}(G_{21}b)\,\cdots\,)))$
   Berechnung durch sukzessive Auswertung von Matrix–Vektor–Produkten $G_{kl}w$, keine explizite Berechnung von Q

   2b) Löse $Rx = z$ mittels Rücksubstitution.

Besonders geeignet für Gleichungssysteme $Ax = b$, deren Lösung sehr empfindlich gegenüber Rundungsfehlern ist, und zur numerischen Rangbestimmung von A.

Lösung von Kleinste–Quadrate–Problemen
Wegen $\|w\|_2^2 = w^\top w$ ist $\|w\|_2$ invariant gegenüber orthogonalen Transformationen:

$$\|Qw\|_2^2 = (Qw)^\top (Qw) = w^\top (Q^\top Q) w = w^\top w = \|w\|_2^2$$

$$\Rightarrow \quad \|Ax - b\|_2^2 = \|QRx - b\|_2^2 = \|Q(Rx - Q^\top b)\|_2^2 = \|Rx - Q^\top b\|_2^2 = \|\widetilde{R}x - z_1\|_2^2 + \|z_2\|_2^2$$

mit $\quad Q^\top b = \begin{pmatrix} z_1 \\ \cdots \\ z_2 \end{pmatrix} \begin{matrix} n \\ \\ m-n \end{matrix} \in \mathbb{R}^m\,.$

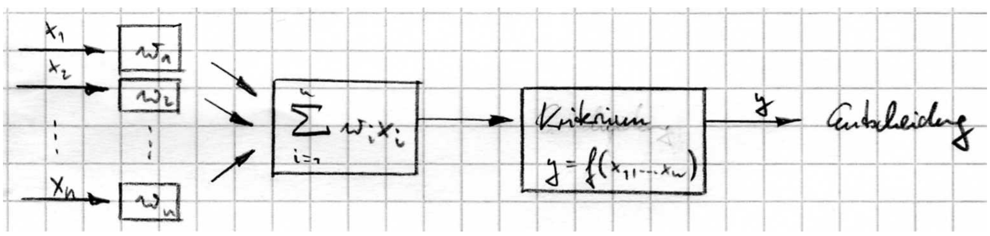
Lösung: Berechne $Q^\top b$ und löse $\widetilde{R}x = z_1$ mittels Rücksubstitution.

Beispiel 2.11 (Neuronale Netze)

Idee Entscheidungen fällen auf Grundlage einer Vielzahl von Einzelinformationen in Anlehnung an Entscheidungsprozesse im menschlichen Hirn.

Beispiel Technik Wäsche in Waschmaschine „stark verschmutzt“
Messung der Wassertemperatur und Wassertrübung an verschiedenen Punkten
Grundlage der Entscheidung: Erfahrung
Konsequenz einer Fehlentscheidung: Lerneffekt

Neuronales Netz (Ein–Schicht–Modell, linear)



Verhalten des Netzes bestimmt durch *Gewichte* $w_1, \ldots, w_n$

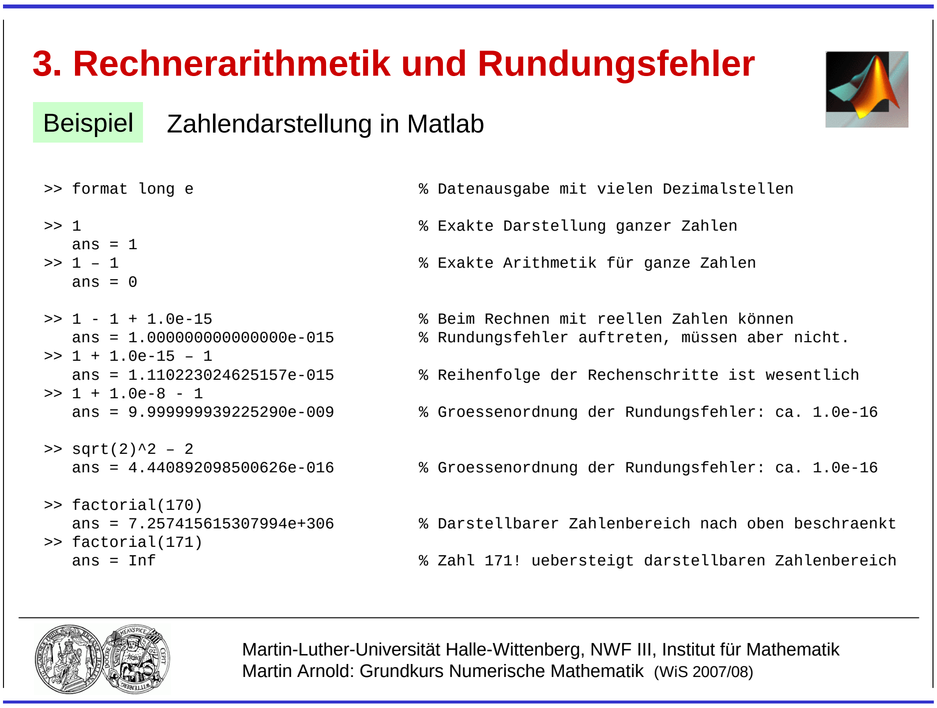


Abbildung 3.1: Rechnen in Gleitpunktarithmetik: Beispiel Matlab.

Training des Netzes Wähle $w_1, \ldots, w_n$ so, dass eine große Zahl von Tests mit vorgegebenen Eingangsdaten $(x_1^{(j)}, \ldots, x_n^{(j)})^\top$ und bekannten Resultaten $y^{(j)}$ möglichst gut wiedergegeben wird:

$$\sum_{i=1}^{n} x_i^{(j)} w_i \approx y^{(j)}\,, \quad (\, j = 1, \ldots, m\,)$$

$\rightsquigarrow$ überbestimmtes lineares Gleichungssystem, Bestimmung von $(w_1, \ldots, w_n)$ als Kleinste–Quadrate–Lösung

praktisch Lösung der Normalgleichungen oder Lösung mittels QR–Zerlegung.

3 Rechnerarithmetik und Rundungsfehler

3.1 Gleitpunktarithmetik

Bemerkung 3.1 (Gleitpunktzahlen)

a) Ganzzahlige Datentypen (INTEGER) mit exakter Arithmetik

$$-\text{MaxInt} - 1\,,\ \ldots\ ,\ -1,\ 0,\ 1,\ \ldots\ ,\text{MaxInt}\,,$$

z. B. für Indizes in Laufanweisungen.

b) *Normalisierte Gleitpunktdarstellung* (engl.: floating point numbers) zur Darstellung reeller Zahlen

$$F := \{\, y \,:\, y = \pm m * \beta^{e-t}\,\} \subset \mathbb{R}$$