14 Das Gaußsche Eliminationsverfahren

Wir beschreiben nun den Gaußschen Algorithmus zur Lösung eines quadratischen Gleichungssystems

$$\sum_{k=1}^{n} a_{ik}x_k = y_i, \quad i = 1, \ldots, n,$$

bzw. ausführlich geschrieben

$$\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & y_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & y_2 \\
 & & & & \cdots & & & & \\
a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & y_n.
\end{array}$$

In der Mathematik versteht man allgemein unter einem *Algorithmus* ein schematisches Verfahren zur Lösung eines mathematischen Problems. Der Gaußsche Algorithmus basiert auf Überlegungen, die Gauß 1801 in seinen „Disquisitiones arithmeticae“ publiziert hat. Seine Grundidee besteht darin, das vorgelegte Gleichungssystem durch fortgesetzte Elimination in (höchstens) $n-1$ Schritten zu überführen in ein äquivalentes Gleichungssystem in *Dreiecksgestalt*

$$\begin{array}{ccccccccc}
b_{11}x_1 & + & b_{12}x_2 & + & \cdots & + & b_{1n}x_n & = & z_1 \\
 & & b_{22}x_2 & + & \cdots & + & b_{2n}x_n & = & z_2 \\
 & & & & \cdots & & & & \\
 & & & & b_{n-1,n-1}x_{n-1} & + & b_{n-1,n}x_n & = & z_{n-1} \\
 & & & & & & b_{nn}x_n & = & z_n
\end{array}$$

mit $b_{ii} \neq 0$ für $i = 1, \ldots, n$. Aus diesem gestaffelten System lassen sich die Unbekannten dann durch rückwärts Einsetzen, beginnend mit x_n, der Reihe nach berechnen. Zwei Gleichungssysteme werden dabei *äquivalent* genannt, wenn ihre Lösungen übereinstimmen.

Wir betrachten zunächst eine invertierbare Matrix A. Zur Elimination von x_1 subtrahieren wir für $i = 2, \ldots, n$ von der i–ten Zeile das a_{i1}/a_{11}–fache der ersten Zeile. Hierzu muß $a_{11} \neq 0$ sein. Da die Spalten der invertierbaren Matrix A linear unabhängig sind, lässt sich dies durch Vertauschen von Zeilen des Gleichungssystems stets erreichen.

Es entsteht

$$
\begin{array}{ccccccccc}
b_{11}x_1 & + & b_{12}x_2 & + & \cdots & + & b_{1n}x_n & = & z_1 \\[2ex]
 & & a_{22}^{(2)}x_2 & + & \cdots & + & a_{2n}^{(2)}x_n & = & y_2^{(2)} \\[2ex]
 & & & & \cdots & & & & \\[2ex]
 & & a_{n2}^{(2)}x_2 & + & \cdots & + & a_{nn}^{(2)}x_n & = & y_n^{(2)}
\end{array}
$$

mit den neuen Koeffizienten

$$
\begin{array}{rcll}
b_{1k} & := & a_{1k}^{(1)}, & \qquad k = 1, \ldots, n, \\[2ex]
a_{ik}^{(2)} & := & a_{ik}^{(1)} - \dfrac{a_{i1}^{(1)} a_{1k}^{(1)}}{a_{11}^{(1)}}, & \qquad i, k = 2, \ldots, n,
\end{array}
$$

und den neuen rechten Seiten

$$
z_1 := y_1^{(1)}, \quad y_i^{(2)} := y_i^{(1)} - \frac{a_{i1}^{(1)} y_1^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \ldots, n.
$$

Dabei haben wir mit den Koeffizienten und rechten Seiten des Ausgangssystems $a_{ik}^{(1)} := a_{ik}$, $y_i^{(1)} := y_i$ gesetzt. Hierdurch ist das ursprüngliche System aus n Gleichungen für n Unbekannte $x_1, \ldots, x_n$ äquivalent überführt in ein System aus $n-1$ Gleichungen für $n-1$ Unbekannte $x_2, \ldots, x_n$.

Die gleiche Operation wird nun auf das verbleibende Gleichungssytem für $x_2, \ldots, x_n$ angewandt. Addition eines Vielfachen einer Zeile zu einer anderen Zeile läßt den Zeilenrang und damit auch den Spaltenrang unverändert. Daher sind die Spalten des verbleibenden $(n-1) \times (n-1)$–Gleichungssystems wiederum linear unabhängig. Gegebenenfalls nach erneuter Zeilenvertauschung haben wir also $a_{22}^{(2)} \neq 0$ und können wie oben nun mit der zweiten Gleichung die zweite Unbekannte x_2 eliminieren. So fortfahrend gelangen wir zum

Gauß Algorithmus. Das lineare Gleichungssystem

$$\begin{array}{ccccccccc} a_{11}^{(1)}x_1 & + & a_{12}^{(1)}x_2 & + & \cdots & + & a_{1n}^{(1)}x_n & = & y_1^{(1)} \\ a_{21}^{(1)}x_1 & + & a_{22}^{(1)}x_2 & + & \cdots & + & a_{2n}^{(1)}x_n & = & y_2^{(1)} \\ & & & & \cdots & & & & \\ a_{n1}^{(1)}x_1 & + & a_{n2}^{(1)}x_2 & + & \cdots & + & a_{nn}^{(1)}x_n & = & y_n^{(1)} \end{array}$$

mit invertierbarer Matrix $A = (a_{ik}^{(1)})$ wird durch $n-1$ Eliminationsschritte gemäß der Vorschrift

$$\begin{aligned} a_{ik}^{(m+1)} &:= a_{ik}^{(m)} - \frac{a_{im}^{(m)}a_{mk}^{(m)}}{a_{mm}^{(m)}}, \qquad i,k = m+1,\ldots,n, \\ y_i^{(m+1)} &:= y_i^{(m)} - \frac{a_{im}^{(m)}y_m^{(m)}}{a_{mm}^{(m)}}, \qquad i = m+1,\ldots,n, \end{aligned}$$

mit $m = 1,\ldots,n-1$ äquivalent überführt in Dreiecksgestalt

$$\begin{array}{ccccccccc} b_{11}x_1 & + & b_{12}x_2 & + & \cdots & + & b_{1n}x_n & = & z_1 \\ & & b_{22}x_2 & + & \cdots & + & b_{2n}x_n & = & z_2 \\ & & & & \cdots & & & & \\ & & & & b_{n-1,n-1}x_{n-1} & + & b_{n-1,n}x_n & = & z_{n-1} \\ & & & & & & b_{nn}x_n & = & z_n \end{array}$$

mit

$$b_{ik} := a_{ik}^{(i)}, \quad k = i,\ldots,n, \; i = 1,\ldots,n,$$

und

$$z_i := y_i^{(i)}, \quad i = 1,\ldots,n.$$

Die für die Durchführbarkeit notwendige Bedingung $a_{mm}^{(m)} \neq 0$ läßt sich stets durch Zeilenvertauschungen erfüllen.

Die Rechenvorschriften des Gaußschen Algorithmus lassen sich im folgenden Schema zusammenfassen.

$$\boxed{\begin{array}{cc} \mathrm{a} & \mathrm{b} \\ \mathrm{c} & \mathrm{d} \end{array}}$$

Hier veranschaulicht das Rechteck die noch der Elimination zu unterwerfende Restmatrix einschließlich der rechten Seite. a ist das Eliminationselement. Die Elemente b in der Eliminationszeile bleiben unverändert, die Elemente c in der Eliminationsspalte werden durch Null ersetzt (mit Ausnahme des Eliminationselements a), und die übrigen Elemente d werden ausgetauscht nach der Regel

$$d \to d - \frac{bc}{a} \,.$$

In Pseudocode läßt sich der Gauß Algorithmus wie folgt darstellen:

1. Vorwärts Elimination:

for $m = 1, \ldots, n-1$ do

  for $j = m+1, \ldots, n$ do

    for $k = m+1, \ldots, n$ do $a_{jk} = a_{jk} - \dfrac{a_{jm}a_{mk}}{a_{mm}}$

    $y_j = y_j - \dfrac{a_{jm}y_m}{a_{mm}}$

2. Rückwärts Einsetzen:

for $m = n, n-1, \ldots, 1$ do $x_m = y_m$

  for $k = m+1, \ldots, n$ do $x_m = x_m - a_{mk}x_k$

  $x_m := \dfrac{x_m}{a_{mm}}$

Wir geben zur Illustration das folgende kleine

Beispiel 14.1 Betrachte das 3×3–Gleichungssystem

$$\begin{array}{rcrcrcr} 2x_1 & + & 4x_2 & + & x_3 & = & 4 \\ 2x_1 & + & 6x_2 & - & x_3 & = & 10 \\ x_1 & + & 5x_2 & + & 2x_3 & = & 2. \end{array}$$

Hier liefern zwei Eliminationsschritte

2	4	1	4
2	6	-1	10
1	5	2	2

2	4	1	4
	2	-2	6
	3	1.5	0

2	4	1	4
	2	-2	6
		4.5	-9

Durch rückwärts Einsetzen ergibt sich der Reihe nach $x_3 = -2$, $x_2 = 1$ und $x_1 = 1$. $\square$

Bei einer nicht invertierbaren Matrix mit Rang r bricht das Eliminationsverfahren nach r Schritten ab. Die Matrix des Restgleichungssystem für die Unbekannnten $x_{r+1}, \ldots, x_n$ ist die Nullmatrix, denn andrenfalls wäre der Rang von A von r verschieden. Das vorgelegte Gleichungssystem ist also lösbar genau dann, wenn die aus der Elimination hervorgehenden rechten Seiten

$$z_{r+1} = \cdots = z_n = 0$$

erfüllen. Die Lösungen errechnen sich aus dem gestaffelten System, indem die $x_{r+1}, \ldots, x_n$ beliebig gewählt werden, und dann $x_r, \ldots, x_1$ durch rückwärts Einsetzen ermittelt werden. Auf diese Weise erhalten wir die $(n-r)$–dimensionale lineare Lösungsmannigfaltigkeit.

Um den Einfluß von Rundungsfehlern möglichst gering zu halten, sorgt man bei der praktischen Durchführung des Gaußschen Algorithmus nicht nur für $a_{mm}^{(m)} \neq 0$, sondern man vertauscht Zeilen und Spalten so, daß das Eliminationselement $a_{mm}^{(m)}$ das betragsgrößte Element der noch der Elimination zu unterwerfenden Restmatrix ist (*vollständige Pivotisierung*), bzw. das betragsgrößte Element der Eliminationszeile oder -spalte (*teilweise Pivotisierung*). Dies führt natürlich zu einer Erhöhung des Rechenaufwands. Bei der Programmierung des Gauß Algorithmus mit Pivotsuche werden diese Vertauschungen allerdings nicht explizit vorgenommen, d.h. man beläßt das Pivotelement auf seinem Platz und notiert die Reihenfolge der Eliminationszeilen und -spalten.

In Pseudocode läßt sich der Gauß Algorithmus mit vollständiger Pivotsuche wie folgt darstellen:

1. Vorwärts Elimination:
for $m = 1, \ldots, n-1$ do $z(m) = \text{true},\ s(m) = \text{true}$
for $m = 1, \ldots, n-1$ do
  $p = 0$
  for $j = 1, \ldots, n$ do if z(j) then
    for $k = 1, \ldots, n$ do if $s(k)$ then if $\text{abs}(a_{jk}) > p$
      then $p = \text{abs}(a_{jk}),\ pz = j,\ ps = k$
  $z(pz) = \text{false},\ s(ps) = \text{false},\ ez(m) = pz,\ es(m) = ps$

  for $j = 1, \ldots, n$ do if $z(j)$ then
    for $k = 1, \ldots, n$ do if $s(k)$ then $a_{jk} = a_{jk} - \dfrac{a_{j,ps} a_{pz,k}}{a_{pz,ps}}$
  $y_j = y_j - \dfrac{a_{j,ps} y_{pz}}{a_{pz,ps}}$

2. Rückwärts Einsetzen:
for $m = 1, \ldots, n-1$ do $x_m = 0$
for $m = n, n-1, \ldots, 1$ do $p := y_{ez(m)}$
  for $k = 1, \ldots, n$ do $p = p - a_{ez(m),k} x_k$
  $x_{es(m)} = \dfrac{p}{a_{ez(m),es(m)}}$

Daß die teilweise Pivotsuche nicht immer vor dem Einfluß von Rundungsfehlern schützt, belegt folgendes Beispiel.

Beispiel 14.2 Wir betrachten das Gleichungssystem

$$\begin{array}{rcrcr} x_1 & + & 200x_2 & = & 100 \\ x_1 & + & x_2 & = & 1 \end{array}$$

mit der exakten Lösung $x_1 = 100/199 = 0.503\ldots,\ x_2 = 99/199 = 0.497\ldots$. Wir rechnen nun in 2-stelliger Gleitkommaarithmetik. Auswahl von a_{11} als Pivotelement (bei Spaltenpivotsuche) führt auf

$$\begin{array}{rcrcr} x_1 & + & 200x_2 & = & 100 \\ & & -200x_2 & = & -99 \end{array}$$

($199 = 200$ in 2-stelliger Rechnung!) mit dem Resultat $x_2 = 0.50$ ($0.495 = 0.50$ in 2-stelliger Rechnung!) aus der zweiten Zeile, und dann $x_1 = 0$ aus der ersten Zeile. Dagegen führt die Wahl von a_{12} als Pivotelement (vollständige Pivotsuche) auf

$$\begin{array}{rcrcl} x_1 & + & 200x_2 & = & 100 \\ \\ x_1 & & & = & 0.5 \end{array}$$

($0.995 = 1.00$ in 2-stelliger Rechnung!) und dann auf die in 2-stelliger Rechnung korrekte Lösung $x_1 = x_2 = 0.5$. □

Zur Abschätzung des Rechenaufwands bestimmen wir die Anzahl der im Gauß Algorithmus erforderlichen Multiplikationen. Die Anzahl der Multiplikationen für die Berechnung aller Lösungen aus der Dreicksform nennen wir α_n. Für sie gilt die Rekursion

$$\alpha_n = \alpha_{n-1} + n,$$

denn es sind noch n Multiplikationen erforderlich, um x_1 mit der ersten Zeile aus den schon berechneten $x_2, \ldots, x_n$ zu ermitteln. Wegen $\alpha_1 = 1$ folgt also

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

Bezeichne β_n die Anzahl der Multiplikationen für die Elimination in Dreicksform gleichzeitig für r rechte Seiten. Für sie gilt die Rekursion

$$\beta_n = \beta_{n-1} + (n+r)(n-1),$$

denn die Elimination von x_1 aus der ersten Zeile (bei r verschiedenen rechten Seiten) bedarf pro Zeile $n+r$ Multiplikationen in insgesamt $n-1$ Zeilen. Aus $\beta_1 = 0$ folgt also

$$\beta_n = \sum_{k=1}^{n}(k+r)(k-1) = \frac{n^3}{3} - \frac{n}{3} + \frac{n(n-1)r}{2} \ .$$

Daraus ergibt sich für die Gesamtzahl durch Addition $r\alpha_n + \beta_n$ der

Satz 14.3 *Der Gauß Algorithmus benötigt zur Lösung eines $n \times n$–Gleichungssystems simultan für r verschiedene rechte Seiten insgesamt*

$$\frac{n^3}{3} + rn^2 - \frac{n}{3}$$

Multiplikationen.

Der Rechenaufwand ist also von der Größenordung $n^3/3$. Verdopplung der Zahl der Unbekannten erhöht die Rechenzeit für die Elimination um den Faktor 8. Bei 3μ sec $= 3 \cdot 10^{-6}$ sec pro Multiplikation dauert die Auflösung eines Gleichungssystems mit $n = 10^3$ Unbekannten etwa 15 min und mit $n = 10^4$ Unbekannten 15000 min $\approx$ 10 Tage. Für große Gleichungssysteme sind daher später zu betrachtende Iterationsverfahren den Eliminationsverfahren überlegen. Spalten- oder Zeilenpivotisierung erfordert zusätzlich maximal $O(n^2)$ Operationen, die vollständige Pivotisierung erfordert zusätzlich $O(n^3)$ Operationen, und wird daher selten angewandt.

Der Gaußsche Algorithmus ermöglicht auch die Berechnung der Inversen von Matrizen. Die Inverse ergibt sich durch simultanes Auflösen von n Gleichungssystemen mit den Spalten der Einheitsmatrix als rechten Seiten. Die Lösungen liefern die Spalten der inversen Matrix.

Wir veranschaulichen im folgenden, daß das Gaußsche Eliminationsverfahren auf eine LR–Zerlegung führt.

Definition 14.4 *Eine Zerlegung einer Matrix A in ein Produkt*

$$A = LR$$

aus einer unteren (linken) Dreiecksmatrix L und einer oberen (rechten) Dreiecksmatrix R heißt eine LR–Zerlegung*.*

Es ist leicht zu sehen, daß das Produkt zweier unterer (oberer) Dreiecksmatrizen wieder eine untere (obere) Dreiecksmatrix ist. Vorwärts- bzw. Rückwärtseinsetzen zeigt, das untere bzw. obere Dreiecksmatrizen mit nichtverschwindenden Diagonalelementen invertierbar sind, und daß die Inversen wiederum untere bzw. obere Dreiecksmatrizen sind.

Satz 14.5 *Die Durchführung des Gaußschen Eliminationsverfahrens (ohne Zeilen- und Spaltenvertauschungen) für eine invertierbare Matrix A liefert eine LR–Zerlegung.*

Beweis. Im ersten Eliminationsschritt wird von der i–ten Zeile das a_{i1}/a_{11}–fache der ersten Zeile subtrahiert, d.h. $A^{(1)} = A$ wird von links mit der invertierbaren unteren Dreiecksmatrix

$$L^{(1)} = \begin{pmatrix} 1 & & & & \\ -\frac{a_{21}^{(1)}}{a_{11}^{(1)}} & 1 & & & \\ -\frac{a_{31}^{(1)}}{a_{11}^{(1)}} & & 1 & & \\ & & \cdots & & \\ -\frac{a_{n1}^{(1)}}{a_{11}^{(1)}} & & & & 1 \end{pmatrix}$$

multipliziert. Dann hat $A^{(2)} = L^{(1)}A^{(1)}$ die Gestalt

$$A^{(2)} = \begin{pmatrix} a_{11} & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ & & \cdots & & \\ 0 & * & * & * & * \end{pmatrix}.$$

Die gleiche Elimination wird auf die verbleibende $(n-1) \times (n-1)$–Matrix angewandt. Die zugehörende $(n-1)\times(n-1)$–Eliminationsmatrix wird durch Diagonalelement 1 zu einer invertierbaren $n \times n$–Dreiecksmatrix $L^{(2)}$ ergänzt. Fortgesetztes Eliminieren ergibt also nach $n-1$ Schritten

$$L^{(n-1)} \cdots L^{(1)} A = R$$

mit invertierbaren unteren Dreiecksmatrizen $L^{(1)}, \ldots, L^{(n-1)}$ und einer oberen Dreicksmatrix R. Daraus folgt

$$A = LR$$

mit der Inversen L des Produkts $L^{(n-1)} \cdots L^{(1)}$. □

Wir vermerken, daß nicht zu jeder invertierbaren Matrix eine LR–Zerlegung existiert. So hat zum Beispiel

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

keine LR–Zerlegung. Da jedoch der Gaußsche Algorithmus bei invertierbaren Matrizen mit Zeilenvertauschung stets durchführbar ist, läßt sich zu jeder invertierbaren Matrix A eine Permutationsmatrix P angeben, so daß PA eine LR–Zerlegung besitzt.