Lineare Algebra (Mathe I) für Wirtschaftsinformatiker; Zusammenfassung

Artur Trzewik `sw0562@uni-essen.de`

v1.0, 26.03.1998 korrigiert 16. Februar 2000

Zusammenfassung

Warnung: für die Richtigkeit der Definitionnen kann ich nicht garantieren. Es wurde verfasst als ein Teil der Dokumatation zum Programm „Matrix“ und veröffentlich in der Hoffnung, daß es auch für andere nützlich sein könnte. Das Hauptaugenmerkt wurde auf die klare Beschreibung von Algorithmen gelegt. Es kann zugegeben schwer als eine einzige Lernvorlage dienen, weil es nichts erklärt und keine mathematische Beweise beinhalten.

Inhaltsverzeichnis

1 Beschreibung der Algorithmen; Mathe Definitionen

Das ist eine Zusammenfassung meiner Vorlesungsunterlagen aus Mathe I. Es werden alle Algorithmen grob beschreiben, ohne jegliche Beweise. Es geht vorallem darum zu beschreiben wie die Algorithmen funktionieren und welche mathematische Probleme mam mit ihnen lösen kann. Diese Beschreibung kann aber ein gutes Skript oder ein Mathebuch nicht ersetzen. Die Reihenfolge entspricht eher der Gliederung der Algorithmen als der beim wirklichen Vorlesung.

1.1 Eine Matrix

Def: Def.: eine $m \times n$ *Matrix* (m mal n): m-Zeilenzahl; n-Spaltenzahl

$$\begin{array}{ccccc} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m2} & \dots & a_{mn} \end{array}$$

Alle Matrizen werden hier ohne Klammer geschrieben. Auch Vektoren werden in der html Version ohne den gewöhnten Pfeil geschrieben. (also “a” kann auch bedeuten “a-Vektor”). Es soll es den Kontext ersichtlich sein um welche Bedeutung es sich gerade handelt.

2 Grundliegende Algorithmen

2.1 Treppenform

Def.: $c = c_{ij}$ ist in Treppenform, falls c die Nullmatrix ist, oder es eine Folge

$$1 \leq j_1 < j_2 < \ldots < j_r \leq l$$

von r Sprungstellen gibt mit

1. unterhalb der r Zeile stehen nur Nullen
2. An dem Sprungstellen stehen Elemente $c_{ij} \neq 0$
3. Links von dem Sprungstellen stehen nur Nullen

Anzahl der Sprungstellen nennt man auch Rang der Matrix

2.2 Treppennormallform

Def.: Eine Matrix ist in Treppennormalform $c = (c_{ij})$ wenn sie in Treppenform ist und wenn zusätzlich

1. Alle Sprungstellen $c_{ij} = 1$
2. In jeder Sprungstellen Spalte sind alle Elemente bis auf das Sprungstellen Element gleich Null

Treppennormalform ist eindeutig

2.3 Gauss Algorithmus

Benutzt um Lösung 3.2 des linearen Gleichungsystem 3 zu finden. k-Spaltenzahl. Bringt die Matrix auf Treppenform 2.1

1. Suche die erste Spalte Matrix c, die nicht nur Nullen erhält. Dies ist die j1-Spalte. Darin sei das Element $c_{ij1} \neq 0$ Dann vertausche die i-te Zeile mit der ersten Zeile.
2. Für i=2,. . . .,k addiere der i-ten Zeilen das $\frac{-c_{ij1}}{c_{1j1}}$ fache der 1. Zeile.
3. Wende Schritt 1 und 2 auf Matrix c2, das entsteht wenn man aus Matrix c nur die Zeile 2 bis k nimmt und j ersten Stellen abschneidet.

`Bemerkung:` c_{ij1} bedeutet: i.te Zeile j.te Spalte 1. Algorithmus Durchgang

Es gibt viele Möglichkeiten eine Matrix auf Treppenform zu bringen (Treppenform ist nicht eindeutig). Man kann, um sich die Rechnungen zu erleichtern, auch Zeilen wechseln oder Zeilen mit verschiedene Faktoren multiplizieren. Für den Rechner ist das aber keine Erleichterung. Der Algorithmus in dieser Form ändern nicht die Determinante 4.6 von Matrix.

2.4 Gauss-Jordan Algorithmus

Benutzt um Lösung 3.2 des linearen Gleichungsystem 3 zu finden. Weiterentwicklung von Gauss Algorithmus 2.3. Bringt die Matrix auf Treppennormallform 2.2. k-Spaltenzahl

1. Suche die erste Spalte Matrix c, die nicht nur Nullen erhält. Dies ist die j1-Spalte. Darin sei das Element $c_{ij}1 \neq 0$ Dann vertausche die i-te Zeile mit der ersten Zeile. Dann dividiere die Zeile durch c_{ij1}
2. Für i=2,. . . .,k addiere der i-ten Zeilen das $-c_{ij1}$ fache der 1. Zeile. Mache alle Einträge unterhalb c_{ij1} zu Nullen.
3. Für alle Zeilen d=1,. . . ,i bilde $1.Zeile = 1.Zeile - c_{dj1} * (i - teZeile)$. Mache alle Einträge oberhalb c_{ij1} zu Nullen.
4. Wende Schritt 1 und 2 auf Matrix c2, das entsteht wenn man aus Matrix c nur die Zeile 2 bis k nimmt und j ersten Stellen abschneidet.

Bemerkung: Gauss und Gauss-Jordan-Algorithmus unterscheiden sich im 1. Punkt. Beim GaussJordan wird die Zeile durch Sprungstelleneintrag dieser Zeile geteilt. Dadurch erreicht man, daß Sprungstelleneintrag auf 1 skaliert wird. Das“vereinfacht„ die Rechnung im Schritt 2.

3 Lineare Gleichungssysteme

3.1 Definition

Def.: Ein lineares Gleichungssystem mit n Gleichungen hat die Form

$$\begin{array}{rcl} a_{11}*x_1+a_{12}*x_2+a_{13}*x_3+\ldots+a_{1n}*x_n & = & b_1 \\ a_{21}*x_1+a_{22}*x_2+a_{23}*x_3+\ldots+a_{2n}*x_n & = & b_2 \\ a_{31}*x_1+a_{32}*x_2+a_{33}*x_3+\ldots+a_{3n}*x_n & = & b_3 \\ : & = & : \\ a_{n1}*x_1+a_{n2}*x_2+a_{n3}*x_3+\ldots+a_{nn}*x_n & = & b_n \end{array}$$

a_{ij} sind die Koeffizienten wenn $b_i \neq 0$ dann ist das ein homogenes Gleichungssystem, andernfalls inhomogenes

2.Die Matrix der Form

$$\begin{array}{ccccc} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ : & : & : & : & \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} & bn \end{array}$$

heißt erweiterte Koeffizientenmatrix (ohne b$_i$ einfach Koeffizientenmatrix)

Man kann mit Hilfe der elementaren Umformungen (sie ändern nicht die Lösungsmenge) die Matrix in die Form bringen, in dem die Lösung des Gleichungssystem ersichtlich (leicht zu berechnen) ist. Gauss und Gauss-Jordan Algorithmus benutzen nur solche Umformungen. Zu solchen Umformungen gehören:

- Vertauschen zwei Zeilen
- Multiplikation einer Zeile mit Zahl $\neq 0$
- Addition eines Vielfachen einer Zeile zu einer anderen Zeile

3.2 Lösungstypen von linearen Gleichungsystemen (LGS)

Es gibt drei möglichen Lösungstypen von Linearen Gleichungssystemen (LGS)

1. LGS hat keine Lösung, dann Ergebnissmatrix ist leer.
2. LGS hat eine Lösung, wenn Ausgangsmatrix $m \times n$ dann Ergebnissmatrix (n-1)*1

$$\begin{array}{c} x_1 \\ x_2 \\ : \\ x_{n-1} \end{array}$$

3. LGS hat mehrere Lösungen dann ist das Ergebnissmatrix (`Bezogen auf Programm Matrix`) (n*k k< n) zusammengesetzt aus mehreren Spaltenvektoren, wobei das erste eine Lösung des LGS ist und die anderen die Basis des linearen Unterraums für das Lösung des zugehörigen homogenen Gleichungssystems. Jede mögliche Lösung des LGS kann man erreichen aus

$$\vec{x} = \vec{a} + \vec{u}_1 * b_1 + \ldots + \vec{u}_{k-1} * b_{k-1}$$

   - $\vec{x}$: Vektor für alle Lösungen
   - $\vec{a}$: Erstes Spaltenvektor des Ergebnissmatrix (Eine Lösung)
   - $\vec{u}_i$: Weiter Spaltenvektoren
   - b_i : Freie Variablen

3.3 Lösung von LGS

Algorithmus:

1. Bringe die erweiterte Koeffizientenmatrix auf Treppennormal Form mit Hilfe des Gauss-Jordan Algorithmus.
2. Falls die Letzte Spalte eine Sprungsstelle ist dann keine Lösung. Ende.
3. Es gibt nur Sprungstellen außer letzten Spalte dann ist die letzte Spalte die einzige Lösung des LGS. Ende.
4. Es gibt mehrere Lösungen.
5. Eine Lösung erhält man indem man für alle x für nicht-Sprungstellen Variablen Null einsetzt und für anderen nacheinander die Einträge der letzten Spalte
6. Finde die Lösung eines homogenen Gleichungssystems. Es gibt frei wählbare Variablen (Parameter) nämlich, die an dem Nichsprugstellen. Setzt man nacheinander eine dieser frei wählbaren Variablen gleich 1 und die anderen frei wählbaren Variablen gleich 0 und löst auf, so erhält man eine Basis des Lösungsraumes.

Bsp. (für Schritte 4-6)

$$\begin{matrix} 1 & 5 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 2 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 & 5 \end{matrix}$$

Eine Lösung

$$x_1 = 2 x_2 = 0 x_3 = 4 x_4 = 0 x_5 = 5$$

als Vektor

$$\begin{matrix} 2 \\ 0 \\ 4 \\ 0 \\ 5 \end{matrix}$$

Suche Basis des Lösung des homogenen Gleichungssystem (Schritt 6)

$$\begin{array}{cccccc} 1 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

Setzt man für $x_2 = 1$ und $x_4 = 0$ erhält man

$$\begin{array}{rcl} 1*x_1+5*1+0*x_3+0*x_3+0*x_4 & = & 0 \\ 0*x_1+0*x_2+1*x_3+2*0+0*x_4 & = & 0 \\ 0*x_1+0*x_2+0*x_3+0*x_3+0*x_4 & = & 0 \end{array}$$

also

$$x_1+5=0$$

Ein Vektor der Basis

$$\begin{array}{c} -5 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}$$

Den zweiten Vektor erhält man wenn man x_2=0 und x_4=1 einsetzt also

$$x_2+2=0$$

$$\begin{array}{c} 0 \\ -2 \\ 0 \\ 0 \\ 0 \end{array}$$

Die dazugehörige Ergebnissmatrix wäre

$$\begin{array}{ccc} 2 & -5 & 0 \\ 0 & 0 & -2 \\ 4 & 0 & 0 \\ 0 & 0 & 0 \\ 5 & 0 & 0 \end{array}$$

was bedeutet

$$\begin{array}{rcl} x_1 & = & 2-5*a \\ x_2 & = & -2*b \\ x_3 & = & 4 \\ x_4 & = & 0 \\ x_5 & = & 5 \end{array}$$

wobei a,b frei wählbar

3.4 Lineare Abbildungen und Matrizen

Sei A eine $m \times n$ Matrix, sei definiert die Abbildung ϕ

$$K^n \to K^m$$

K^n- lineares Unterraum von Dimension n

$$x \to A * x = \phi A(x)$$

$$\vec{x} = einVektor(n \times 1)$$

$$\vec{x} = \begin{matrix} x_1 \\ x_2 \\ : \\ x_n \end{matrix}$$

Die Abbildung ϕ ist linear 4.4

Def.: Die Menge $\{\phi A(x) \mid x \in K^n\}$ heißt Bild von ϕ A. Sie ist Teilmenge von K^n und ein linearer Unterraum von K^n.

Das Finden von Bild von A 3.5

Def.: Die Menge $\{x \in K^n \mid \phi A(x) = 0\}$ heißt Kern von ϕ A. Sie ist Teilmenge von K^n und ein linearer Unterraum von K^n.

Das Finden von Kern von A 3.6

Satz: Sei

$$\phi AK^n \to K^m (Aeinem \times nMatrix)$$

(A eine $m \times n$ Matrix)

Dann gilt: $dimKern\phi A + dimBild\phi A = n$

3.5 Errechnen von Bild von Matrix

Man muß die Basis des auf den Spalten Vektoren aufgespannten linearen Unterraums finden. Am einfachsten geht man vor, wenn man die Matrix auf Treppenform 3 bringt und alle Spaltenvektoren aus Ursprungsmatrix zur Lösung nimmt, die Sprungstellen haben. z.B.

$$\begin{matrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 2 \\ 2 & 3 & 4 \\ 0 & 1 & 2 \end{matrix}$$

Nach Anwendung von Gauss-Jordan Algorithmus

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Sprungstellen beim 1. und 2.Spalte kommen zur Lösung

$$\begin{matrix} 0 & 1 \\ 1 & 2 \\ 0 & 1 \\ 2 & 3 \\ 0 & 1 \end{matrix}$$

Die beiden Vektoren sind linear unabhängig und sind eine Basis des Bildes des Matrix

3.6 Errechnen von Kern von Matrix

Kern ϕA ist die Lösung der Abbildung $\phi A(x) = 0$

Der Nullvektor gehört immer zur Lösung also z.B. der Kern von

$$\begin{matrix} 1 & 2 & 3 \\ 2 & 1 & 0 \\ 2 & 4 & 6 \end{matrix}$$

ist die Lösungsmenge von LGS 3

$$\begin{aligned} x_1 + 2 * x_2 + 3 * x_3 &= 0 \\ 2 * x_1 + 1 * x_2 &= 0 \\ 2 * x_1 + 4 * x_2 + 6 * x_3 &= 0 \end{aligned}$$

Hier in Programm wird zum Matrix eine Nullspalte addiert und durch GaussJordan 2.4 Algorithmus die Lösungsmenge 3.2 ausgerechnet. Weil Kern ein lineares Unterraum ist, wird die eine spezielle Lösung (immer Nullvektor) abgeschnitten. Die übrigen Vektoren (wenn vorhanden) bilden die Basis des gesuchten Kerns. Hier:

$$\begin{matrix} 1 & 2 & 3 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 4 & 6 & 0 \end{matrix}$$

nach Gauss-Jordan

$$\begin{matrix} 1 & 0 & \frac{1}{3} & 0 \\ 0 & 1 & 1\frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

eine Lösung ist Nullvektor; Kern ist

$$\begin{matrix} -\frac{1}{3} \\ -1\frac{1}{3} \\ 0 \end{matrix}$$

der Kern ist nicht eindeutig (Es gibt viele unterschiedliche Kerne von einer Matrix)

3.7 Lösung von (LGS) und lineare Abbildungen

Ein LGS kann man als lineare Abbildung betrachten

$$\begin{array}{rcl} a_{11} * x_1 + a_{12} * x_2 + a_{13} * x_3 + \ldots + a_{1n} * x_n & = & b_1 \\ a_{21} * x_1 + a_{22} * x_2 + a_{23} * x_3 + \ldots + a_{2n} * x_n & = & b_2 \\ a_{31} * x_1 + a_{32} * x_2 + a_{33} * x_3 + \ldots + a_{3n} * x_n & = & b_3 \\ : & = & : \\ a_{n1} * x_1 + a_{n2} * x_2 + a_{n3} * x_3 + \ldots + a_{nn} * x_n & = & b_n \end{array}$$

als $A * \vec{x} = \vec{b}$ wo A die Koeffizienten Matrix 3

betrachte A: $\phi A(x) = A * x : K^n \to K^m$ dann $\phi A(x) = b$

Dann gilt:

- LGS ist genau dann lösbar, wenn b in Bild 3.5 von $\phi A(x)$
- Lösungsmenge=spezielle Lösung+Kern 3.6 $\phi A(x)$

4 Weitere Operationen auf Matrizen

4.1 Matrix-Multiplikation

Sei A eine $m \times n$ Matrix

sei B eine n*t Matrix

$$A = \begin{matrix} a_{11} & \ldots & a_{1n} \\ : & : & : \\ a_{m1} & \ldots & a_{mn} \end{matrix} \quad B = \begin{matrix} b_{11} & \ldots & b_{1t} \\ : & : & : \\ b_{n1} & \ldots & a_{nt} \end{matrix}$$

Dann ist C=A*B eine $m \times t$ Matrix mit

$$C = \begin{matrix} c_{11} & \ldots & c_{1t} \\ : & : & : \\ c_{m1} & \ldots & c_{mt} \end{matrix}$$

c_{ij}- i_te Zeile von A; j_te Spalte von B

$$c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + a_{i3} * b_{3j} + \ldots + a_{in} b_{nj}$$

4.2 Matrix-Addition

Sei A eine $m \times n$ Matrix sei B eine $m \times n$ Matrix

$$A = \begin{matrix} a_{11} & \dots & a_{1n} \\ : & : & : \\ a_{m1} & \dots & a_{mn} \end{matrix} \quad B = \begin{matrix} b_{11} & \dots & b_{1n} \\ : & : & : \\ b_{m1} & \dots & a_{mn} \end{matrix}$$

Dann ist C=A+B eine $m \times n$ Matrix mit

$$C = \begin{matrix} c_{11} & \dots & c_{1n} \\ : & : & : \\ c_{m1} & \dots & c_{mn} \end{matrix}$$

c_{ij}- i_te Zeile von A,B; j_te Spalte von A,B

$$c_{ij} = a_{ij} + b_{ij}$$

4.3 Matrixsubstraktion

Sei A eine $m \times n$ Matrix sei B eine $m \times n$ Matrix

$$A = \begin{matrix} a_{11} & \dots & a_{1n} \\ : & : & : \\ a_{m1} & \dots & a_{mn} \end{matrix} \quad B = \begin{matrix} b_{11} & \dots & b_{1n} \\ : & : & : \\ b_{m1} & \dots & a_{mn} \end{matrix}$$

Dann ist C=A-B eine $m \times n$ Matrix mit

$$C = \begin{matrix} c_{11} & \dots & c_{1n} \\ : & : & : \\ c_{m1} & \dots & c_{mn} \end{matrix}$$

c_{ij}- i_te Zeile von A,B; j_te Spalte von A,B

$$c_{ij} = a_{ij} - b_{ij}$$

4.4 linearer Unterraum

Def.: linearer Unterraum

Eine Teilmenge U von K^n heißt linearer Unterraum wenn gilt,

- für alle $a \in K$ (a Element von K) und $u \in U$ u Element von U gilt a*u ist ein Element in U
- für alle $u_1 \in U \wedge u_2 \in U \Longrightarrow u_1 + u_2 \in U$;für alle u_1 Element von U und u_2 Element von U $u_1 + u_2$ ist wieder in U

4.5 Inverse der Matrix

Es gibt nur eine Inverse von $n \times n$ Matrizen A^{-1} ist Eine Inverse von A wenn gilt $A^{-1} * A = I_n$ wobei In ist eine Einheitsmatrix wie folgt

$$I_n = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

für n=4

I_n ist ein neutrales Element bezüglich Multiplikation 4.1 Matrix A heißt dann inventierbar Man findet eine Inverse, indem man zur gegebenen Matrix eine Einheitsmatrix dazu fügt und anschließend mit Gauss-Jordan 2.4 behandelt z.B zur Matrix

$$\begin{matrix} 5 & 10 & 20 \\ 1 & 1 & 1 \\ 12 & 12 & 20 \end{matrix}$$

eine Einheitsmatrix zufügt

$$\begin{matrix} 5 & 10 & 20 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 12 & 12 & 20 & 0 & 0 & 1 \end{matrix}$$

nach Gauss-Jordan

$$\begin{matrix} 1 & 0 & 0 & -\frac{1}{5} & -1 & \frac{1}{4} \\ 0 & 1 & 0 & \frac{1}{5} & \frac{7}{2} & -\frac{3}{8} \\ 0 & 0 & 1 & 0 & -\frac{3}{2} & \frac{1}{8} \end{matrix}$$

Die Inverse ist dann

$$\begin{matrix} -\frac{1}{5} & -1 & \frac{1}{4} \\ \frac{1}{5} & \frac{7}{2} & -\frac{3}{8} \\ 0 & -\frac{3}{2} & \frac{1}{8} \end{matrix}$$

Eine Matrix ist nicht inventierbar wenn nach Gauss-Jordan am Anfang der Matrix keine Einheitsmatrix ersichtlich wird.

4.6 Determinante der Matrix

Die Determinante ist nur auf $n \times n$ Matrizen definiert. Im Programm kann auch eine Determinante aus $n \times (n+1)$ Matrix ausgerechnet werden. Die letzte Spalte wird einfach ignoriert. Es soll noch Leute geben, die mit Hilfe der Determinanten LGS 3 lösen. Ohne hier drauf genauer anzugehen muß man sagen Determinanten sind OUT.

Es gibt grundsätzlich zwei Methoden um eine Determinante auszurechnen

- Rekursiv (In dem Programm nicht implementiert), sie ist wegen exponential wachsenden Aufwand nicht bevorzuziehen.
- Durch Hilfe von Gauss-Algorithmus 2.3. Man benutzt dabei zwei Eigenschaften:
  - Addiert man das x-Fache einer Zeile zu einer anderen Zeile, dann ändert sich die Determinante nicht.
  - Vertauscht man zwei Zeilen, dann ändert sich die Determinante um Faktor -1.

Ist die Matrix in Treppenform 3 bekommt man die Determinante, indem man alle Diagonalen Einträge (mit Index i,i) miteinander multipliziert und gegeben falls um Faktor aus Punkt b) korrigiert.

5 Optimierungsverfahren von linearen Ungleichungssystemen (Simplex)

Problem: Suche die beste (optimale) Lösung eines Ungleicheungssystem bezüglich einer zu maximierenden Funktion $G(\vec{x})$.

1. Bringe alle Ungleichungen in die Form

$$\begin{array}{rcl} a_{11}*x_1+a_{12}*x_2+a_{13}*x_3+\ldots+a_{1n}*x_n & \geq & b_1 \\ a_{21}*x_1+a_{22}*x_2+a_{23}*x_3+\ldots+a_{2n}*x_n & \geq & b_2 \\ a_{31}*x_1+a_{32}*x_2+a_{33}*x_3+\ldots+a_{3n}*x_n & \geq & b_3 \\ : & : & : \\ a_{m1}*x_1+a_{m2}*x_2+a_{m3}*x_3+\ldots+a_{mn}*x_n & \geq & b_m \end{array}$$

   evtl. durch Multiplikation mit -1 und Einsetzen von Gleichungen in die Ungleichungen

2. sei gegeben eine zu maximierende Funktion G

$$G(x_1+\ldots+x_n)=G(\vec{x})=g_0+g_1*x_1+g_2*x_2+\ldots+g_n*x_n=g_0+\vec{g}(\vec{x})$$

   (wenn G zu Minimieren ist, dann durch -1 Multiplizieren)

3. Die Ausgangsmatrix zum Simplex Algorithmus hat das Aussehen

$$\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} & b_2 \\ : & : & : & : & : & b_3 \\ a_{m1} & a_{m2} & a_{m2} & \ldots & a_{mn} & b_m \\ g_1 & g_2 & g_3 & \ldots & g_n & G \end{array}$$

   Letzte Spalte heißt Gewinnspalte Allgemein:

$$\begin{array}{c|c} A & b \\ g & G \end{array}$$

   Bedeutet $A(\vec{x}) \geq \vec{b}$ und $G = g_0$

4. Finde eine Lösung des Ungleichungsystems 5.3 sei $\vec{u}$ diese Lösung $A * \vec{u} \leq \vec{b}$, dann $k = A\vec{x} - \vec{b}$ Vektor $\vec{k}$ hat Einträge nur größer oder gleich Null Ausgangsmatrix zum Eckenfindungsalgorithmus (m+1)*(n+1)

$$\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & k1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & k2 \\ : & : & : & : & : & k3 \\ a_{m1} & a_{m2} & a_{m2} & \dots & a_{mn} & km \\ g_1 & g_2 & g_3 & \dots & g_n & G \end{array}$$

$$G = G(\vec{u}) = g_0 + g_1 * u_1 + g_2 * u_2 + \dots + g_n * u_n$$

5. Benutze Ekenfindung-Algorithmus 5.2
6. Wenn der Eckenfindung Algorithmus beim Maximumkriterium (1. Schritt) abbricht dann gilt. Wir haben l verschiedene Spalten verarbeitet und daher l verschiedene Zeilen erhalten, die jetzt Standartbasis Vektoren sind $(0, \dots, 1, \dots, 0)$ mit n+1 Koordinaten und mit “1” an einer der ersten n-Stellen. Es seien $i_1, \dots, i_l$ diese Zeilenindizies. Wir setzen zusätzlich voraus, daß die Gewinnzeile g(l) von B(l) nur negative Einträge erhält

   $$g_i(l) \geq 0$$

   Wenn nicht dann fahre fort mit Schritt 8 Dann gibt es eine optimale Lösung(en) auf $Ax \geq b$ diese erhält man folgendermaßen. Man nehme die Zeile $i_1 \dots i_l$ von A und b der Ausgangsmatirx und betrachtet

   $$(\overline{A}) * \vec{x} = (\overline{b})$$

   Dies ist lösbar und jede Lösung 3.2 y davon erfühlt $A * \vec{y} \geq b$ und $G(\vec{y})$ ist ein Maximum von $A * y \geq b$.
7. Wenn der Eckenfindung-Algorithmus beim Quotientenkriterium endet (2. Schritt) (zwar Spalte keine Zeile), dann ist die Gewinnfunktion G auf der Lösung Menge $A * x \geq b$ nach oben unbeschränkt (kein Maximum)
8. Benutze Eckenaustausch-Algorithmus. Beim ihm gelten die Selbe Regel wie beim Eckenfindung-Algorithmus 5.2. Nur die Spalten werden nicht markiert und der Algorithmus endet wenn alle Einträge in der Gewinnzeile g_i negativ sind. Es kann passieren, daß dieser Algorithmus nicht endet (nicht terminiert). Ich konnte leider keine Abbruchbedingung in der Literatur finden, obwohl solche existiert. Im Programm endet der Algorithmus nach n Schritten.

5.1 Pivotieren

Spaltenpivotierung an der Stelle $_{ij}$

$$\begin{array}{ccccc} b_{11} & b_{12} & b_{13} & \dots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2n} \\ : & : & : & : & : \\ b_{m1} & b_{m2} & b_{m2} & \dots & b_{mn} \end{array}$$

sei das Element $b_{ij} \neq 0$

1. Multipliziere die i-te Spalte mit $\frac{1}{b_{ij}}$. Dann steht an der Stelle 1
2. Für alle Spalten verschieden von j-ten Führe aus

$$k - teSpalte = k - teSpalte - b_{ik} * j - teSpalte,$$

   dann stehen an allen (i,k)-ten Stellen (mit $k \neq j$) nur Nullen

5.2 Eckenfindung-Algorithmus

Ein Teil von Simplex Algorithmus

1. Schritt (Maximumkriterium) Wähle einen Spaltenindex j mit $1 \leq j \leq n$ (letzte Spalte mit b's ausgeschlossen), derart daß die j-te Spalte von $B \neq \vec{0}$ (Nullvektor) ist und, daß $\mid g_j \mid$ den größten Wert hat. Markiere diese Spalte. Wenn ein solches j nicht existiert dann ende des Algorithmus. (Wenn mehrere j mit dieser Eigenschaft dann kleinste j)
2. Schritt (Quotientenkriterium) Ist die j-te Spalte von 1.Schritt

$$\begin{matrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \\ g_j \end{matrix}$$

   Dann betrachte die "relevante Zeilenmenge„ R

$$R = \{i \mid a_{ij} \neq 0 \wedge a_{ij} g_j \leq 0\}$$

   Wenn R die leere Menge $R = \emptyset$ ist, dann Ende des Algorithmus Wenn R nicht leer ist, dann wähle i Element Reell ($i \in \Re$) aus , so daß

$$\frac{k_i}{\mid a_{ij} \mid} \leq \frac{k_l}{\mid a_{ij} \mid}$$

   für alle l Element R

   (Wenn mehrere i dann nimmt das kleinste)
3. Schritt. Führe für B Spaltenpivotierung 5.1 an der Stelle (i,j) Man erhält neue Matrix B2 wobei Zeilenvektor von B2 so aussieht 0… 1… 0 i-te Zeile 1 in der j-ten Spalte
4. Schritt: gehe zum 1. Schritt . Laß allerdings die zuvor markierten Spalten bei der Auswahl von j-Außer Betracht.

(Der Eckenfindung-Algorithmus endet spätesten nach n Runden)

5.3 Das Suchen nach einer speziellen Lösung von $Ax \geq b$

1. wenn alle $b_i \leq 0(i = 1, .., m)$ dann ist $\vec{u} = 0$ (Vektor) eine Lösung von $A * x \geq b$ sonst,
2. Betrachte die Matrix

$$\hat{A} = A \begin{matrix} -1 \\ -1 \\ : \\ -1 \end{matrix}$$

   $\hat{A}$ ist eine $m \times (n+1)$ Matrix. Man fügt zu Matrix A einfach eine Spalte mit nur "-1„ zu

3. füge zu x eine neue Koordinate

$$\hat{x} = \begin{matrix} x_1 \\ x_2 \\ : \\ x_n \\ x_{n+1} \end{matrix}$$

   Betrachte $\hat{A} * \hat{x} \geq b$

4. Sei $w = max(b_i | 1 \leq i \leq m)$ dann ist

$$\hat{u} = \begin{matrix} 0 \\ 0 \\ : \\ -w \end{matrix}$$

   eine Lösung von $\hat{A} * \hat{x} \geq b$

5. Ist $\hat{v} = \begin{matrix} v_1 \\ v_2 \\ : \\ v_{n+1} \end{matrix}$ eine Lösung von $\hat{A} * \hat{x} \geq b$ (Suche nach dieser Lösung mit Eckenfindung evtl. Eckenaustauschalgorithmus 5) mit $v_{n+1} \geq 0$ dann ist $v = \begin{matrix} v_1 \\ v_2 \\ : \\ v_n \end{matrix}$ eine Lösung von $A * x \geq b$
6. Wenn für alle Lösungen $\hat{v}$ von $\hat{A} * \hat{x} \geq b$ immer gilt $v_{n+1} < 0$ dann hat $A * x \geq b$ keine Lösung

Anmerkungen: Man kann den Algorithmus verkürzen. Es wird nur eine Lösung von $\hat{A} * \hat{x} \geq b$ nicht unbedingt die optimale Lösung gesucht. Man kann den o.g. Algorithmus abbrechen sobald der Gewinn positiv ist. Es gibt dann eine Lösung des Gleichungssystem

$$A * s = k^{(l)} - k$$
$$k = A * u - b$$

(k ist der Anfangsvektor) für die gilt

$$G(s+u) \geq 0$$

Jede Lösung von $A*x = k(l) + b + G(l) * Einheitsvektor$ erfühlt $A*v \geq b$

6 Andere Verfahren

6.1 Nährungslösung

Sei A eine $m \times n$ Matrix und b Element Rm dann gilt für das Gleichungssystem

$$(A^t * A) * x = A^t * b$$

A^t ist eine Trasponente von A

ist immer lösbar und die Lösungen sind die besten Lösungen von $A*x = b$

"beste Lösung„ Das heißt wenn u diese Lösung ist dann für alle v

$$| A*u - b | < | Av - b |$$

also ist $| A*u - b |$ minimal. $| a |$ ist die Länge von a und ist definiert als

$$| a | = \sqrt{a_1 + a_2 + \ldots + a_n}$$

6.2 Trasponente

Die Transponente von A $m \times n$ Matrix ist eine A^t $n \times m$ Matrix, wo die Spalten von A^t die Zeilen von A sind.

6.3 Determinante Rekursiv

Die Determinante läßt sich auch rekursiv berechnen. Diese Methode wird auch als Standartmethode betrachtet. Im Tkmatrix wird es aber nur verwendet um charakteristisches Polynom zu berechnen. Seit Version 0.5 wird es mit Hilfe von einen Backtracking Algorithmus vermieden, daß Determinanten mehrfach von gleichen Matrizen berechnet werden. Das wird aber mit hohen Speicheraufwand bezahlt.

Zuerst Hilfsdefinition:

Def.: Streichungs Matrix Sei A eine [m$\times$ n] Matrix über den Körper K. Unter der ij-ten Streichungsmatrix A_{ij} versteht man die Matrix, die man erhält wenn man aus A die i-te Zeile und j-te Spalte streicht.

Def.: Determinante Determinante von A [n$\times$ n] wird folgendermaßen definiert (induktic, Entwicklung nach 1. Spalte):

$$det(A) = a_{11}det(A_{11}) - a_{21}det(A_{21}) + .. - a_{n1}det(A_{n1})$$

Vorsicht wechselnde Zeichen + - wenn A=(a) ([1$\times$ 1] Matrix dann det(A)=a

6.4 Charakteristisches Polynom

Charkteristische Polynom wird gebildet um Eigenwerte (und nacher Eignevektoren) der Matrix zu finden.

Satz: Nullstellen des charakteristischen Polynom sind Eigenwerte dazugehörigen Matrix

Formal wird charakteristisches Polynom gebildet durch

$$det(A - xI)$$

wo I Einheitsmatrix ist. Was zu folge hat, daß man rekursiv Determinante (siehe 6.3 von der Matrix ausrechnen muß

$$\begin{array}{ccccc} a_{11}-x & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22}-x & a_{23} & \dots & a_{2n} \\ : & : & : & : & : \\ a_{m1} & a_{m2} & a_{m2} & \dots & a_{mn}-x \end{array}$$

Dabei muß man selbstverständlich, die so erhaltene Polynomen hantieren

6.5 Matrixspiel

Matrixspiele lassen sich mit Hilfe von Simplexverfahren lösen. D.h zu jedem Matrixspiel, bei dem kein Sattelpunkt existiert, läßt sich mit Hilfe von Simplexverfahren eine optimale gemischte Strategie finden. Auch jedes Optiemirungsproblem hat eine Entsprechung als Matrixspiel (Duealitätsatz). Im Program wurde das Verfahren aus dem Buch (Georg Schrage, Rüdeger Baumann, „Strategiespiele; Computerorientierte Einfürung in Algorithmen der Spieltheorie“, R. Oldenbourg Verlag, München Wien 1984) benutzt.

Dazugehörige Simpexverfahren ist unterschiedlich zu dem übrigen im Program benutzten Verfahren. Zu den übrigen Erläuterungen verweise ich auf das obige Buch.

Das Programm berechnet den Wert des Spieles und die optimale Strategien für beide Spieler. Ergebnismatrix wird die optimale Strategie des ersten Spielers.