A. Filler, Humboldt-Universität zu Berlin

Nutzung des CAS Maxima für Berechnungen und Visualisierungen in der Linearen Algebra und Analytischen Geometrie

1 Lösen linearer Gleichungssysteme

Um das LGS

$$\begin{array}{rrcr} \text{(I)} & 2x+3y+\ \ z & = & 3 \\ \text{(II)} & x-3y+2z & = & -1 \\ \text{(III)} & 3x+5y-\ \ z & = & 4 \end{array}$$

zu lösen, gibt man in Maxima ein:

```
(%i1) solve([2*x+3*y+z=3, x-3*y+2*z=-1, 3*x+5*y-z=4 ], [x,y,z]);
```

und übergibt diese Anweisung an Maxima zur Berechnung durch gleichzeitiges Drücken der `Shift-` und `Enter`-Taste.Maxima gibt sofort die Lösungsmenge in folgender Form aus:

(%o1) $[[x=\frac{1}{3}, y=\frac{2}{3}, z=\frac{1}{3}]]$

Alternativ zur Eingabe des Befehls kann auch der Dialog „Gleichung lösen“ im Menü „Gleichungen“ aufgerufen werden, wo dann die Gleichungen und die Variablen, nach denen sie gelöst werden sollen, eingegeben werden.

Für *nicht lösbare LGS* erhält man für die Lösungsmenge die Ausgabe `[]`, zum Beispiel:

```
(%i2) solve([3*x-4*y+8*z=16, 6*x-y+5*z=-4, -x-y+z=-8/3], [x,y,z]);
```

(%o2) $[\,]$

Für ein *LGS mit einparametriger Lösungsmenge*, das aus drei Gleichungen mit drei Lösungsvariablen besteht, erhält man:

```
(%i3) solve([3*x-4*y+8*z=2, 6*x-y+5*z=4, -x-y+z=-2/3], [x,y,z]);
solve: dependent equations eliminated: (3)
```

(%o3) $[[x=-\frac{12\,\%r1-14}{21}, y=\frac{11\,\%r1}{7}, z=\%r1]]$

Maxima hat also z als Parameter gewählt und ihm den Namen `%r1` gegeben. Interessant ist die Information `solve: dependent equations eliminated: (3)`. Die Software hat also erkannt, dass eine Gleichung von den anderen abhängt und diese „eliminiert“.

Leicht lässt sich auch die Wahl einer anderen Variablen als Parameter erreichen, wenn bekannt ist, dass ein LGS unendlich viele Lösungen besitzt:

```
(%i4) solve([3*x-4*y+8*z=2, 6*x-y+5*z=4, -x-y+z=-2/3], [x,y]);
solve: dependent equations eliminated: (3)
```

(%o4) $[[x=-\frac{12\,z-14}{21}, y=\frac{11\,z}{7}]]$

Maxima nimmt in diesem Falle z als konstant an. Die Ausgabe lässt sich aber auch in dem Sinne interpretieren, dass z als Parameter gewählt wird, denn z kann beliebige Werte annehmen.

Oft ist es sinnvoll, ein *LGS mit Symbolen für die Koeffizienten* in Maxima einzugeben, diesen aber dann *konkrete Werte zuzuweisen*, wenn man Lösungen für bestimmte Koeffizienten erhalten möchte. Wir betrachten dazu das LGS

$$\begin{array}{c} ax+by=c \\ dx+ey=f \end{array}$$

und lösen es mit Maxima (mit den Lösungsvariablen x, y):

```
(%i5) solve([a*x+b*y=c, d*x+e*y=f], [x,y]);
```

(%o5) $[[x=-\frac{c\,e-b\,f}{b\,d-a\,e}, y=\frac{c\,d-a\,f}{b\,d-a\,e}]]$

Weist man aber den Koeffizienten $a \ldots f$ Werte zu, so erhält man dasselbe Ergebnis wie bei der direkten Eingabe des LGS ohne Symbole für die Koeffizienten:

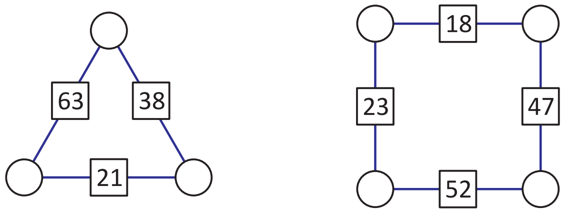
```
(%i6) a:2; b:-1; c:-8; d:4; e:5; f:-4;
      solve([a*x+b*y=c, d*x+e*y=f], [x,y]);
```

(%o6) $[[x = -\frac{22}{7}, y = \frac{12}{7}]]$

Achtung: Löscht man die Eingaben für `a`, `b`, `c`, `d`, `e`, `f` und löst das LGS erneut, so erhält man nicht mehr die allgemeine Lösung, sondern die spezielle Lösung für die zuletzt eingegebenen Werte. Wurden nämlich Symbole mit Werten belegt, so „merkt“ sich Maxima diese Werte und lässt z. B. `a`, `b`, `c`, `d`, `e` und `f` nicht mehr als freie Variablen zu. Um alle Werte zu löschen und die entsprechenden Symbole zu „befreien“, gibt man `kill(all);` ein oder wählt den Menübefehl „Maxima → Speicher löschen“.

Aufgabe: Arithmogons

Arithmogons sind ein aus der Grundschule bekanntes Übungsformat. In der Grundschulversion stehen in den Eck-Kreisen eines Dreiecks Quadrats in konkrete Zahlen. Die Summe von zwei Eckzahlen muss in das dazwischen liegende quadratische Kästchen geschrieben werden.

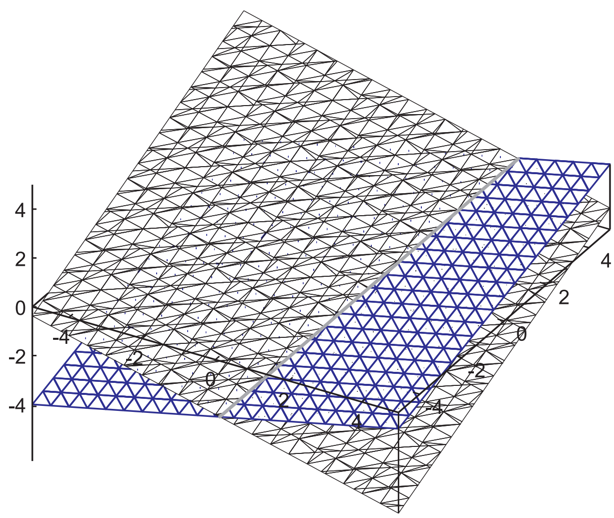


Aufgabe: Hier wird nun eine komplexere Aufgabe gestellt: Es sind die Zahlen in den Quadraten vorgegeben. In die Eck-Kreise müssen Zahlen so eingefügt werden, dass die Zahlen in den Quadraten jeweils die Summe der zugehörigen Eckzahlen sind.

2 Graphische Darstellungen der Lösungsmengen von LGS

In Maxima stehen Funktionen zur Darstellung „implizit“ gegebener Objekte zur Verfügung. Mit „implizit“ ist gemeint, dass Graphen (Geraden oder Ebenen) nicht direkt (explizit) durch Funktionsgleichungen, sondern durch Gleichungen gegeben sind und Maxima selbst Lösungen bestimmen muss, um die betreffenden Geraden oder Ebenen darzustellen. Durch die folgende Eingabe werden die durch zwei Gleichungen beschriebenen Ebenen dargestellt:

```
Gl1 : (3/2)*x - 2*y + 4*z = 1;
Gl2 : x + y - 3*z = 2;
load(draw)$
draw3d(user_preamble = ["set size ratio 1"], surface_hide = true,
       color = black, implicit(Gl1, x,-5,5, y,-5,5, z,-5,5),
       color = blue , implicit(Gl2, x,-5,5, y,-5,5, z,-5,5)  );
```



Es öffnet sich ein separates Grafikfenster, in welchem sich die Darstellung mit der Maus interaktiv drehen und dadurch aus verschiedenen Richtungen betrachten lässt. Die Darstellung erfolgt in dem Bereich, der durch die x-, y- und z-Werte -5 und 5 begrenzt wird. Der Darstellungsbereich lässt sich verändern, indem die Werte in `x,-5,5`, `y,-5,5`, `z,-5,5` (für beide Gleichungen) angepasst werden. Die Option `surface_hide = true` bewirkt, dass verdeckte Bereiche von Ebenen nicht dargestellt werden. Die Ebenen werden zunächst recht grob aufgelöst durch Dreiecke dargestellt. Für eine feinere Auflösung kann vor der mit `color = blue` beginnenden Zeile eine Zeile

```
x_voxel = 20, y_voxel=20, z_voxel=20,
```

eingefügt werden. Höhere Werte bewirken hierbei eine feiner aufgelöste Darstellung, verlängern aber die für die Berechnung der Grafiken benötigte Zeit.

3 Vektorrechnung und -darstellung mit Maxima

Für einfache Aufgaben der Vektorrechnung und die Darstellung von Vektoren im Zweidimensionalen lässt sich u. a. gut die dynamische Mathematiksoftware *GeoGebra* verwenden. Da in der S II jedoch vor allem Vektorrechnung und Analytische Geometrie im Dreidimensionalen betrieben wird, ist es notwendig, Software zu verwenden, die sowohl die in der Schule auftretenden Berechnungen mit Vektoren ausführen als auch dreidimensionale Darstellungen generieren kann. Hierfür ist Maxima gut geeignet.

Eingabe von Vektoren und einfache Berechnungen in Maxima

Vektoren werden in dem CAS Maxima als n-Tupel in eckigen Klammern eingegeben, z. B. `[x,y]` und `[x,y,z]`. Diese lassen sich addieren und mit reellen Zahlen multiplizieren. So gibt Maxima z. B. auf die Eingaben

```
u:[3,2,12]$  v:[-8,12.5,13]$  lambda:4/3$
u+v;  lambda*u;  u+lambda*v;
```

die Ergebnisse $[-5, 14.5, 25]$, $[4, \frac{8}{3}, 16]$ und $[-\frac{23}{3}, 18.666, \frac{88}{3}]$ aus.

Es ist zu beachten, dass Maxima bei der Eingabe von Dezimalzahlen annimmt, dass es sich um Näherungswerte handelt und auch Näherungswerte ausgibt (wie 18.666). Hingegen erfolgen bei der Eingabe von ganzen Zahlen oder Brüchen exakte Berechnungen, wie man an den Ausgaben $\frac{8}{3}$, $-\frac{23}{3}$ und $\frac{88}{3}$ sieht.

Linearkombinationen

Um Vektoren als Linearkombinationen anderer Vektoren darzustellen, sind lineare Gleichungssysteme zu lösen. Um die Komponenten von Vektoren nicht einzeln eingeben zu müssen, kann in Maxima auf Komponenten in der Form `[x,y]` bzw. `[x,y,z]` eingegebener Vektoren zurückgegriffen werden, z. B. mittels `v[2]` auf die zweite (y-) Komponente eines zuvor angegebenen Vektors $\vec{v}$. Dies wird in dem folgenden Beispiel genutzt, um die Koeffizienten bei der Darstellung des Vektors $\vec{x}$ als Linearkombination der Vektoren $\vec{u}$, $\vec{v}$ und $\vec{w}$ zu ermitteln.

```
x:[1,3,2]$   u:[-1,3,-4]$   v:[3,-5,-2]$   w:[7,-9,-4]$
solve([ lambda * u[1] + mu * v[1] + nu * w[1]= x[1],
        lambda * u[2] + mu * v[2] + nu * w[2]= x[2],
        lambda * u[3] + mu * v[3] + nu * w[3]= x[3] ],
      [lambda, mu, nu] );
```

Als Ausgabe erhält man: `[[lambda=-3/10, mu=-21/5, nu=19/10]]`.

Skalarprodukt

Für das (im folgenden Kapitel behandelte) Skalarprodukt zweier Vektoren wird ein Punkt `.` (dot) verwendet.

Eingabe:
```
u:[3,4,-1]$  v:[-12,5,7]$
u.v;
```
Ausgabe: `-23`

Vektorprodukt

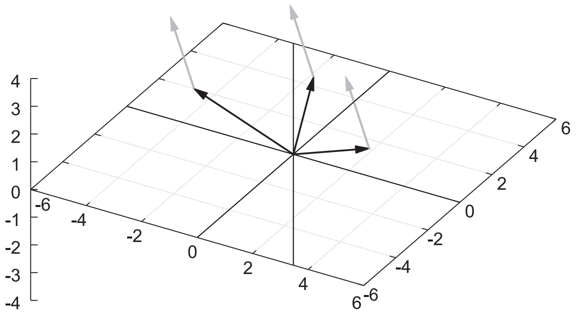
Das Vektorprodukt zweier Vektoren $\vec{u}$ und $\vec{v}$ wird mittels `express(u~v)` berechnet. Dazu muss vorher mittels `load("vect")` das Paket „`vect`" geladen werden.

Eingabe:
```
load("vect")$
u:[1,2,2]$  v:[-4,2,2]$
express(u~v);
```
Ausgabe: `[0,-10,10]`

Graphische Darstellungen von Vektoren als Pfeile

Mithilfe von `vector([x0,y0],[x,y])` bzw. `vector([x0,y0,z0],[x,y,z])` lässt sich ein Vektor $\binom{x}{y}$ der Ebene bzw. $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ des Raumes als Pfeil mit dem Anfangspunkt $P_0(x_0; y_0)$ bzw. $P_0(x_0; y_0; z_0)$ darstellen. Dazu werden die bereits beschriebenen Umgebungen `draw2d` bzw. `draw3d` genutzt.

```
load(draw)$
  o:[0,0,0]$  u:[1,3,-1]$  v:[-3,-1,2]$  w:[-1,3,1]$
  draw3d( user_preamble = ["set size ratio 1"], grid=true,
        xyplane=0, xaxis=true, yaxis=true, zaxis=true,
        xrange = [-6,6], yrange = [-6,6], zrange = [-4,4],
        color = black, vector(o,u),   vector(o,v),   vector(o,w),
        color = red,   vector(u,u+v), vector(v,u+v), vector(w,u+v)
    )
```



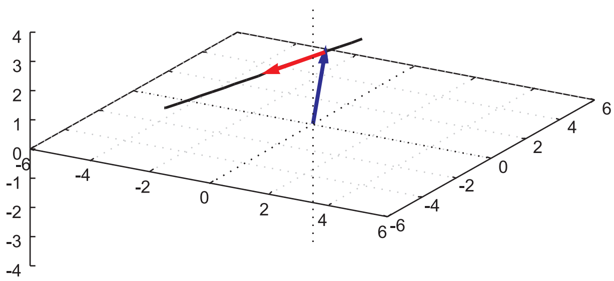
Die Werte für `xrange`, `yrange`, `zrange` sind anzupassen, wenn (in Abhängigkeit von den Komponenten der zu visualisierenden Vektoren) andere Ausschnitte des Raumes dargestellt werden sollen. Die Anschaulichkeit räumlicher Darstellungen ist insbesondere dadurch gegeben, dass diese mit der Maus „gedreht“ und aus verschiedenen Richtungen betrachtet werden können.

4 Geraden und Ebenen

Darstellung von Geraden im dreidimensionalen Raum

Einzugeben sind ein Stützvektor $\vec{p}_0$ und ein Richtungsvektor $\vec{a}$ einer Geraden. Gezeichnet werden durch die folgenden Anweisungen die dadurch beschriebene Gerade sowie Pfeile, die $\vec{p}_0$ und $\vec{a}$ repräsentieren.

```
(%i5)  load(draw)$
       p0:[1,-1,3]$
       a:[-2.5,1,-1.5]$
       draw3d( user_preamble = ["set size ratio 1"], grid=true,
               xyplane=0, xaxis=true, yaxis=true, zaxis=true,
               xrange = [-6,6],yrange = [-6,6],zrange = [-4,4],
               color = black, line_width=2, xu_grid = 100,
                   parametric(p0[1]+t*a[1], p0[2]+t*a[2], p0[3]+t*a[3], t,-0.66,5),
               color = blue, line_width=3, head_length=0.2, vector([0,0,0],p0),
               color = red, vector(p0,a)
           )$
```



Lagebeziehungen von Geraden, Schnittpunktbestimmung

Es sind Stützvektoren $\vec{p}_0$, $\vec{q}_0$ und Richtungsvektoren $\vec{a}$, $\vec{b}$ zweier Geraden einzugeben. Danach werden (falls vorhanden) Parameter s und t für einen Schnittpunkt P mit $\vec{p} = \vec{p}_0 + t\vec{a}$ und $\vec{q} = \vec{q}_0 + s\vec{b}$ berechnet.

```
(%i9)  p0:[4,2,1]$       a:[-2,-1,1/2]$
       q0:[-2,4,1]$      b:[-1,-3,1]$
       solve([ p0[1]+t*a[1]=q0[1]+s*b[1],
               p0[2]+t*a[2]=q0[2]+s*b[2],
               p0[3]+t*a[3]=q0[3]+s*b[3] ],
             [t,s]);
```

solve : *dependentequationseliminated* : (1)

(%o13) $[[t = 4, s = 2]]$

Falls sich als Lösung des obigen LGS die leere Menge [] ergibt, so existiert kein Schnittpunkt. Falls Lösungen für s und t ausgegeben werden, so werden diese in die Parameterdarstellungen eingesetzt, es müsste sich jeweils derselbe Vektor ergeben (Ortsvektor des Schnittpunktes).

```
(%i14) t: 4$
       s: 2$
       p0 + t*a;
       q0 + s*b;
```

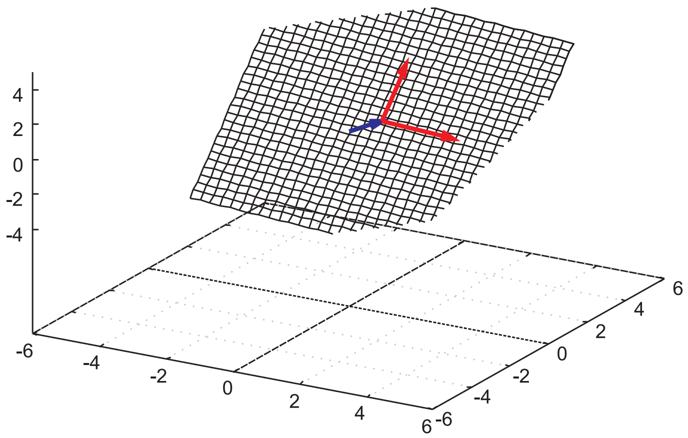
(%o16) $[-4, -2, 3]$

(%o17) $[-4, -2, 3]$

Darstellung von Ebenen, die durch Parameterdarstellungen gegeben sind

Einzugeben sind ein $\vec{p}_0$ und zwei Richtungsvektoren $\vec{a}$, $\vec{b}$ einer Ebene. Gezeichnet werden durch die folgenden Anweisungen die dadurch beschriebene Gerade sowie Pfeile, die $\vec{p}_0$, $\vec{a}$ und $\vec{b}$ repräsentieren. Eventuell sind die das Parameterintervalle (`s,-2,2, t,-2,2`) zu verändern. In Abhängigkeit von darzustellenden Bereich des Raumes müssen die Darstellungsintervalle `xrange = [-7.5,7.5]`, `yrange = [-7.5,7.5]`, `zrange = [-5,5]` verändert werden. Dabei ist darauf zu achten, dass eine etwa gleiche Achsenskalierung entsteht. Dies gelingt meist mit `xrange = yrange =` ca. $1{,}5 \cdot$ `zrange`.

```
(%i6)  load(draw)$
       p0:[1,0,1]$
       a:[-1,3,1]$
       b:[3,-1.5,1]$
       draw3d( user_preamble = ["set size ratio 1"], grid=true,
               xaxis=true, yaxis=true,
               xrange = [-6,6], yrange = [-6,6], zrange = [-5,5],
               color = black,
               parametric_surface (p0[1]+s*a[1]+t*b[1],
                                   p0[2]+s*a[2]+t*b[2],
                                   p0[3]+s*a[3]+t*b[3],
                                   s,-2,2, t,-2,2),
               color = blue, line_width=3, head_length=0.2, head_angle=20,
               vector([0,0,0],p0),
               color = red, vector(p0,a), vector(p0,b)
           )$
```



Lagebeziehungen und Schnittgeraden zweier Ebenen

Es werden zunächst Stützvektoren $\vec{p}_0$, $\vec{q}_0$ und Richtungsvektoren $\vec{a}$, $\vec{b}$, $\vec{c}$ und $\vec{d}$ zweier Ebenen eingegeben. Danach werden (falls vorhanden) Parameter s, t und u, v für einen Schnittpunkt P mit $\vec{p} = \vec{p}_0 + s\vec{a} + t\vec{b}$ und $\vec{q} = \vec{q}_0 + u\vec{c} + v\vec{d}$ berechnet.

```
(%i12) kill(all)$
       p0:[6,5,3]$  /*EBENE1:*/
       a:[1,7,10]$
       b:[-3,0,3]$
       q0:[1,9,7]$  /*EBENE2:*/
       c:[2,11,8]$
       d:[2,-4,-1]$
       solve([ p0[1]+s*a[1]+t*b[1]=q0[1]+u*c[1]+v*d[1],
               p0[2]+s*a[2]+t*b[2]=q0[2]+u*c[2]+v*d[2],
               p0[3]+s*a[3]+t*b[3]=q0[3]+u*c[3]+v*d[3] ],
             [s,t,u,v]);
```

(%o7) $[[s = \%r1 - 1, t = 2 - \%r1, u = \%r1 - 1, v = \%r1]]$

Falls sich als Lösung des obigen LGS die leere Menge [] ergibt, so existiert keine Schnittgerade. Falls Lösungen für s, t und u, v ausgegeben werden, so werden diese in die Parameterdarstellungen eingesetzt, es müsste sich jeweils dieselbe Gerade ergeben.

```
(%i8)  s:%r-1$
       t:2-%r$
       u:%r-1$
       v:%r$
       ratsimp(p0 + s*a + t*b);
       ratsimp(q0 + u*c + v*d);
```

(%o12) $[4\,\%r - 1, 7\,\%r - 2, 7\,\%r - 1]$

(%o13) $[4\,\%r - 1, 7\,\%r - 2, 7\,\%r - 1]$

5 Matrizenrechnung mit Maxima

Eingabe von Matrizen

Matrizen werden durch `matrix([Zeile 1], [Zeile 2], ..., [Zeile m]);` eingegeben.

```
(%i1)  A:matrix([1,3,5,2,-4],
                [0,2,3,1,-3],
                [1,-3,-4,-1,5],
                [4,6,11,5,2] );
```

(%o1) $\begin{pmatrix} 1 & 3 & 5 & 2 & -4 \\ 0 & 2 & 3 & 1 & -3 \\ 1 & -3 & -4 & -1 & 5 \\ 4 & 6 & 11 & 5 & 2 \end{pmatrix}$

Rangbestimmung

Der Rang einer Matrix **A** wird (nach deren Eingabe) mittels `rank(A)` berechnet.

```
(%i2)  rank(A);
```

(%o2) 3

Matrizenmultiplikation

Für das Produkt zweier Matrizen (bzw. einer Matrix und eines Spaltenvektors) wird (wie für das Skalarprodukt von Vektoren) ein Punkt `.` gesetzt.

Beispiele:

```
A:matrix([9,3,8,2],[5,1,8,1]);
B:matrix([4,2],[5,7],[3,3]);
B.A;
```

```
C:matrix([3,2,8],[1,0,9],[3,2,1]);
x:matrix([3],[2],[-7]);
C.x;
```

Bildung transponierter Matrizen

Nach vorheriger Eingabe einer Matrix $\mathbf{A}$ wird ihre Transponierte durch

```
transpose(A);
```

gebildet.

Berechnung inverser Matrizen

Nach Eingabe einer regulären Matrix $\mathbf{A}$ wird ihre Inverse durch

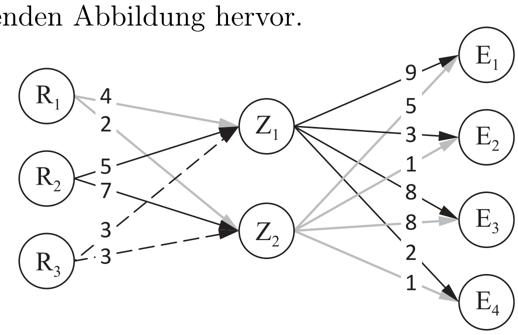
```
invert(A);
```

berechnet. Bei einer nicht regulären Matrix erscheint die Meldung

```
Division by 0 -- an error
```

5.1 Beispiel zur Matrizenmultiplikation: Materialverflechtung

Aus drei verschiedenen Rohstoffen R_1, R_2 und R_3 werden in einem Produktionsablauf zwei Zwischenprodukte Z_1, Z_2 hergestellt, welche dann zu vier Endprodukten E_1, E_2, E_3 und E_4 weiterverarbeitet werden. Wie viele Mengeneinheiten der verschiedenen Rohstoffe zur Herstellung der jeweiligen Zwischenprodukte und wie viele Zwischenprodukte zur Herstellung der verschiedenen Endprodukte benötigt werden, geht aus der folgenden Abbildung hervor.



In den folgenden Tabellen sind die der Abbildung entnommenen Mengenangaben wiedergegeben. Dabei geben die Spalten den Bedarf an Rohstoffen bzw. Zwischenprodukten für die jeweiligen Zwischen- bzw. Endprodukte an.

	Z_1	Z_2
R_1	4	2
R_2	5	7
R_3	3	3

	E_1	E_2	E_3	E_4
Z_1	9	3	8	2
Z_2	5	1	8	1

Es soll berechnet werden, wie viele Mengeneinheiten (ME) der verschiedenen Rohstoffe für die Produktion von 60 (ME) des Endproduktes E_1, 150 (ME) E_2, 40 (ME) E_3 sowie 200 (ME) E_4 erforderlich sind. Dazu stellen wir zunächst die gewünschten Mengen e_1, e_2, e_3, e_4 der Endprodukte E_1, E_2, E_3, E_4 als Spaltenvektor dar:

$$\text{Outputvektor:} \quad \vec{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \begin{pmatrix} 60 \\ 150 \\ 40 \\ 200 \end{pmatrix}.$$

Zunächst ermitteln wir den Bedarf an Zwischenprodukten, indem wir $\vec{e}$ mit einer *Verflechtungsmatrix* verknüpfen, welche die rechte Bedarfstabelle enthält:

$$\mathbf{A}_{\mathrm{Z}\to\mathrm{E}} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{pmatrix} = \begin{pmatrix} 9 & 3 & 8 & 2 \\ 5 & 1 & 8 & 1 \end{pmatrix}.$$

Als Ergebnis der Multiplikation $\mathbf{A}_{\mathrm{Z}\to\mathrm{E}} \circ \vec{e}$ erhält man einen Vektor $\vec{z} = \binom{z_1}{z_2}$ sein, der den Bedarf an Zwischenprodukten angibt. Der so ermittelte „Zwischenvektor" wird nun auf dieselbe Weise mit der Verflechtungsmatrix

$$\mathbf{B}_{\mathrm{R}\to\mathrm{Z}} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 5 & 7 \\ 3 & 3 \end{pmatrix}$$

verknüpft, die den Rohstoffbedarf für die Herstellung von Zwischenprodukten angibt. Als Ergebnis erhält man einen „Inputvektor" $\vec{r}$, dessen Komponenten die benötigten Rohstoffmengen sind:

$$\vec{r} = \mathbf{B}_{\mathrm{R}\to\mathrm{Z}} \circ \vec{z}$$

Aufgabe: Ermitteln Sie für das gegebene Beispiel den Inputvektor.

5.2 Beispiel zur Matrizenmultiplikation: Populationsmatrizen

Aus den Eiern eines Käfers schlüpfen nach einem Monat Larven. Nach einem weiteren Monat werden diese zu Käfern, die nach einem Monat jeweils 8 Eier legen und dann sofort sterben. Nur aus einem Viertel der Eier werden Larven, die anderen Eier werden gefressen oder verenden. Von den Larven wird die Hälfte zu Käfern, die andere Hälfte stirbt.

Zeitpunkt t	Zeitpunkt $t+1$ Monat
Käfer	8 Eier
Ei	$\frac{1}{4}$ Larve
Larve	$\frac{1}{2}$ Käfer

Eine andere Darstellungsweise des Vorgangs (die mit der Darstellung von Materialverflechtungen in Tabellen vergleichbar ist) ist die folgende:

Ei wird zu	Larve wird zu	Käfer wird zu	
		8	Eier
$\frac{1}{4}$			Larven
	$\frac{1}{2}$		Käfer

Diese Tabelle schreiben wir nun als *Populationsmatrix*: $\mathbf{P} = \begin{pmatrix} 0 & 0 & 8 \\ \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix}$.

Es seien zu einem Zeitpunkt gleiche Anzahlen von 1000 Eiern, 1000 Larven und 1000 Käfern vorhanden. Diese Angabe fassen wir zu einem *Populationsvektor* $\vec{p}_0 = \begin{array}{c} \begin{pmatrix} 1000 \\ 1000 \\ 1000 \end{pmatrix} \end{array} \begin{array}{l} \text{(Eier)} \\ \text{(Larven)} \\ \text{(Käfer)} \end{array}$ zusammen.

Aufgaben:

- Ermitteln Sie den Populationsvektor nach einem Monat, nach zwei Monaten und nach drei Monaten.
- Erklären Sie das Ergebnis nach drei Monaten, indem Sie $\mathbf{P} \circ \mathbf{P} \circ \mathbf{P}$ berechnen.

LÖSUNG Arithmogons

Schreibt man in die Eckkreise die Lösungsvariablen a, b und c im Dreiecksfall, a, b, c und d im Quadratfall, so ergeben sich die beiden LGS

$$\begin{aligned} a+b &= 63 \\ b+c &= 38 \\ c+a &= 21 \end{aligned} \qquad \begin{aligned} a+b &= 18 \\ b+c &= 47 \\ c+d &= 52 \\ d+a &= 23 \end{aligned}$$

Interessanterweise hat das erste System die eindeutige Lösung $\begin{pmatrix} 23 \\ 40 \\ -2 \end{pmatrix}$, während das zweite System die einparametrige Lösung $\begin{pmatrix} 23-r \\ r-5 \\ 52-r \\ r \end{pmatrix} = \begin{pmatrix} 23 \\ -5 \\ 52 \\ 0 \end{pmatrix} + r \cdot \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$ mit $r \in \mathbb{R}$ hat.

LÖSUNG Materialverflechtung

Für die Herstellung der gewünschten Mengen an Endprodukten werden also 8780 (ME) des Rohstoffs R_1, 15340 (ME) R_2 und 8040 (ME) R_3 benötigt.