Vektorrechnung und -darstellung mithilfe von Maxima

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 entsprechend 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 $\begin{pmatrix} x \\ y \end{pmatrix}$ 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 in dem Kapitel zu linearen Gleichungssystemen beschriebenen beschriebenen Umgebungen `draw2d` bzw. `draw3d` genutzt. Mithilfe der folgenden Eingaben wurde Abb. 1.1 generiert:

```
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.

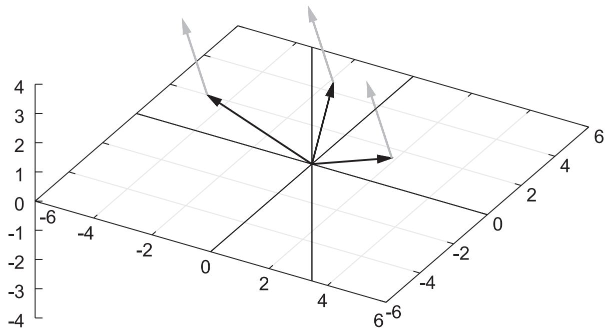

Abb. 1.1: Darstellung von Vektoren im Raum mithilfe des CAS Maxima