Beispiele zum Computeralgebrasystem Maxima

Walter Fendt

12. November 2018
Version vom 27. November 2018

Inhaltsverzeichnis

1. Einstieg

1.1. Erste Schritte

Das Computeralgebrasystem Maxima ist sicher ein tolles Programm, wenn man es beherrscht, aber der Einstieg fällt nicht ganz leicht. Das offizielle Handbuch und die integrierte Hilfe sind nur für Freaks genießbar, und verständliche Erklärungen im WWW sind ziemlich rar. Daher versuche ich hier, anhand von Themen aus dem gymnasialen Mathematikunterricht die Verwendung zu erklären.

Eine kleine Warnung voraus: Ich bin alles andere als ein Maxima-Experte. Diese Seiten sind eher eine Dokumentation meiner „Gehversuche“ auf diesem Gebiet.

Startet man das Programm wxMaxima (bei mir derzeit Version 17.10.1), so hat man im Wesentlichen ein leeres Fenster vor sich. Nur links oben blinkt ein waagrechter Strich einsam vor sich hin. Tippt man eine kleine Rechnung ein, etwa `2+2`, sieht man zwar diese Eingabe (und sogar einen richtigen Cursor), aber auch nach Betätigung der Enter-Taste tut sich fast nichts; nur der Cursor steht jetzt eine Zeile weiter unten. Des Rätsels Lösung: Eine Maxima-Eingabe kann viele Zeilen umfassen. Mit Enter kommt man zwar in die nächste Zeile, aber die Eingabe geht weiter. Abgeschlossen wird eine Eingabe mit der Kombination **Shift-Enter**.

An dieser Stelle hat der Maxima-Neuling immerhin ein bescheidenes Erfolgserlebnis: In der nächsten Zeile erscheint das Ergebnis `4`. Außerdem sind jetzt die Zeilen markiert. Vor der Eingabezeile steht `(%i1)` (1. Input), vor der Ausgabezeile `(%o1)` (1. Output). In einer typischen Maxima-Sitzung wechseln sich Eingaben `(%i...)` und Ausgaben `(%o...)` ab. Es fällt auf, dass am Ende der Eingabezeile automatisch ein Strichpunkt eingefügt wurde. Dieser Strichpunkt drückt aus, dass der Benutzer das Ergebnis sehen will. (Eingaben, bei denen das Ergebnis nicht angezeigt werden soll, enden mit einem Dollarzeichen `$` und werden später eine Rolle spielen.)

Man kann das Programm wie einen Taschenrechner verwenden. Die vier elementaren Grundrechenarten werden durch `+`, `-`, `*` und `/` ausgedrückt, Potenzierung durch `^`. Selbstverständlich kann man Klammern verwenden (allerdings nur runde!); auch die Regel „Punkt vor Strich“ wird beachtet.

Das Programm versucht, die eingegebenen Terme korrekt zu auszurechnen, möglichst ohne Rundungsfehler. Ein kleines Problem dabei sind die **Kommazahlen**. Gibt man eine Kommazahl ein (zum Beispiel `0.5`), so verzichtet Maxima auf eine exakte Berechnung. Wer an exakten Ergebnissen interessiert ist, sollte hier `rat(0.5)` schreiben.

Bruchrechnen mit Maxima ist jetzt kein großes Problem mehr: Um etwa $\frac{3}{4} \cdot (\frac{8}{5} - \frac{5}{6})$ zu berechnen, tippt man `(3/4)*(8/5-5/6);` ein. Gemischte Zahlen wie $2\frac{7}{8}$ werden nicht unterstützt. Hier wären die Schreibweisen `2+7/8` oder `(2*8+7)/8` möglich.

Halbwegs interessant wird ein Computeralgebrasystem natürlich erst, wenn man sich mit Algebra befasst. Als einfaches Beispiel sei die Eingabe `(a-b)^3` genannt. Eigentlich würde man das Resultat $a^3 - 3a^2b + 3ab^2 - b^3$ erwarten, aber Maxima begnügt sich mit der Wiederholung der Eingabe. Wenn man das Programm zum Ausmultiplizieren und

Zusammenfassen gleichartiger Terme veranlassen will, benötigt man den Befehl `expand`. Nach Eingabe von `expand((a-b)^3);` erhält man $-b^3 + 3ab^2 - 3a^2b + a^3$. Gewöhnungsbedürftig ist hier die Reihenfolge der Summanden.

Das Auflösen von Gleichungen erfolgt mit der Maxima-Funktion `solve`. Dieser Befehl verlangt die Angabe der Gleichung und der Unbekannten, nach der aufgelöst werden soll. Beispielsweise lässt sich die quadratische Gleichung $x^2 - 7x + 12 = 0$ lösen, indem man `solve(x^2-7*x+12=0,x);` eingibt. (Man beachte, dass zwischen dem Koeffizienten `7` und der Variablen `x` ein Multiplikationszeichen nötig ist!)

Die bisher erwähnten Anwendungen machen natürlich erst einen winzigen Bruchteil der Möglichkeiten von Maxima aus. Im Hauptteil werde ich auf einige Themen eingehen, bei denen sich das Programm sinnvoll einsetzen lässt. Am intensivsten bespreche ich dabei die Analytische Geometrie, für die ich das Zusatzpaket `anageo3d` entwickelt habe. Im Anhang finden Sie Hinweise zur Installation und Verwendung dieses Pakets und eine Liste der zur Verfügung gestellten Befehle.

1.2. Kleine Tipps

- Am Beginn einer Maxima-Sitzung ist es sinnvoll, den Befehl `kill(all);` einzugeben – auch wenn er ausgesprochen blutrünstig klingt. Durch den Befehl werden frühere Definitionen gelöscht, die unter Umständen zu Fehlern führen könnten.

- In vielen Fällen wird als Nächstes ein Befehl des Typs `load(...);` folgen, um damit ein zusätzliches Maxima-Paket einzubinden.

- Insbesondere bei umfangreichen Eingabedateien sollte man immer wieder im Menü „Zellen“ den Befehl „Alle Zellen auswerten“ aufrufen.

1.3. WWW-Adressen

- `https://wxmaxima-developers.github.io/wxmaxima`
  Download von wxMaxima

- `http://www.crategus.com/books/maxima/maxima.html`
  Maxima Manual in deutscher Übersetzung (Dr. Dieter Kaiser)

- `http://www.austromath.at/daten/maxima/`
  Maxima-Onlinekurs - ACDCA (Walter Wegscheider)

- `https://www.mathematikselberlernen.de/Mathematik/Skripte/maxima_in_beispielen.pdf`
  Standardaufgaben der Sekundarstufe I und II mit Maxima lösen
  (Roland Stewen, Eike Schütze)

2. Mengenlehre

2.1. Endliche Mengen

Beispiel: Menge $A = \{a, c, b\}$

Eingabe: `set(a,c,b);`

Ausgabe: $\{a, b, c\}$

Hinweis 1: Wie man am Beispiel sieht, werden die Elemente geordnet.

Hinweis 2: Gibt man ein Element mehrfach ein, so kommt es (sinnvollerweise) in der Ausgabe nur einfach vor. Die Eingabe `set(a,b,c,a);` hätte also ebenfalls das Ergebnis $\{a, b, c\}$.

2.2. Überprüfung: Ist Objekt Element von Menge?

Diese Überprüfung wird mit der Maxima-Funktion `elementp` durchgeführt.

Beispiel: Ist x ein Element der Menge $A = \{a, b, c\}$?

Eingabe:
```
A: set(a,b,c)$
elementp(x,A);
```

Ausgabe: `false`

Interpretation: x ist kein Element von A.

Hinweis: Maxima-Befehle zur Überprüfung einer Eigenschaft enden häufig mit dem Buchstaben `p` (für „property“).

2.3. Überprüfung: Ist Menge Teilmenge von Menge?

Beispiel: Bestätige, dass $A = \{3, 5\}$ eine Teilmenge von $B = \{1, 2, 3, 4, 5, 6\}$ ist.

Eingabe:
```
A: set(3,5)$
B: set(1,2,3,4,5,6)$
subsetp(A,B);
```

Ausgabe: `true`

2.4. Schnittmenge

Der Befehl für die Schnittmenge (den Durchschnitt) zweier Mengen lautet `intersection`.

Beispiel: Gib die Schnittmenge von $A = \{a, b, c, d, e\}$ und $B = \{b, e, f, g\}$ an.

Eingabe:

```
A: set(a,b,c,d,e)$
B: set(b,e,f,g)$
intersection(A,B);
```

Ausgabe: $\{b, e\}$

Interpretation: $A \cap B = \{b, e\}$

2.5. Vereinigungsmenge

Für die Vereinigungsmenge (Vereinigung) zweier Mengen ist der Befehl `union` zuständig.

Beispiel: Gib die Vereinigungsmenge von $A = \{a, b, c, d, e\}$ und $B = \{b, e, f, g\}$ an.

Eingabe:

```
A: set(a,b,c,d,e)$
B: set(b,e,f,g)$
union(A,B);
```

Ausgabe: $\{a, b, c, d, e, f, g\}$

Interpretation: $A \cup B = \{a, b, c, d, e, f, g\}$

2.6. Differenzmenge

Beispiel: Gib die Differenzmenge von $A = \{a, b, c, d, e\}$ und $B = \{b, e, f, g\}$ an.

Eingabe:

```
A: set(a,b,c,d,e)$
B: set(b,e,f,g)$
setdifference(A,B);
```

Ausgabe: $\{a, c, d\}$

Interpretation: $A \setminus B = \{a, c, d\}$

3. Teilbarkeit

3.1. Überprüfung der Teilbarkeit

Für die Überprüfung der Teilbarkeit bietet sich die Maxima-Funktion `mod` an. `mod(a,b)` bedeutet den Rest, den man bei der Division von `a` durch `b` erhält. Eine natürliche Zahl `a` ist genau dann (ohne Rest) durch die natürliche Zahl `b` teilbar, wenn `mod(a,b)` den Wert `0` hat.

Unter Verwendung dieses Zusammenhangs kann man eine zusätzliche Maxima-Funktion `teilbar` definieren, die einen Wahrheitswert (`true` oder `false`) zurückgibt.

```
teilbar(a,b):=is(mod(a,b)=0)$
```

Erläuterung: Mithilfe der Maxima-Funktion `is` wird überprüft, ob der Divisionsrest `mod(a,b)` den Wert `0` hat. Man beachte das Dollarzeichen `$` am Ende der Eingabe. Eine Ausgabe wäre hier nicht sinnvoll; durch `$` anstelle von `;` wird sie verhindert.

Beispiel: Ist die Zahl 123456789 durch 9 teilbar?

Eingabe:	`teilbar(123456789,9);`
Ausgabe:	`true`
Interpretation:	123456789 ist (ohne Rest) durch 9 teilbar.

3.2. Teilermenge

Beispiel: Gib die Teilermenge der Zahl 60 an.

Eingabe:	`divisors(60);`
Ausgabe:	`[1,2,3,4,5,6,10,12,15,20,30,60]`
Interpretation:	$T_{60} = \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$

3.3. Überprüfung der Primzahleigenschaft

Beispiel: Ist 1234567 eine Primzahl?

Eingabe:	`primep(1234567);`
Ausgabe:	`false`
Interpretation:	123456789 ist keine Primzahl.

3.4. Primzahlentabelle

Die von Maxima zur Verfügung gestellte Funktion `next_prime` liefert zu einer gegebenen ganzen Zahl die kleinste Primzahl, die mindestens gleich dem Argument ist. Beispielsweise erhält man nach der Eingabe `next_prime(14);` das Ergebnis `17`. Mithilfe von `next_prime` wird im Folgenden eine neue Maxima-Funktion definiert, die alle Primzahlen in einem vorgegebenen Bereich aufzählt.

```
TabellePrimzahlen(min,max):=           /* Neue Maxima-Funktion */
  block([liste,p],                     /* Block mit lokalen Variablen */
    liste: [],                         /* Leere Liste */
    p: next_prime(min),                /* Startwert für p */
    while p<=max do block(             /* Solange p<=max ... */
      push(p,liste),                   /* p zur Liste hinzufügen */
      p: next_prime(p)),               /* Nächste Primzahl */
    reverse(liste)                     /* Liste (aufsteigend) */
    )$                                 /* Ende des äußeren Blocks */
```

Beispiel: Erstelle eine Liste der Primzahlen von 1 bis 20.

Eingabe: `TabellePrimzahlen(1,20);`

Ausgabe: `[2,3,5,7,11,13,17,19]`

3.5. Zerlegung in Primfaktoren

Beispiel: Zerlege die Zahl 360 in Primfaktoren.

Eingabe: `ifactors(360);`

Ausgabe: `[[2,3],[3,2],[5,1]]`

Interpretation: $360 = 2^3 \cdot 3^2 \cdot 5$

3.6. Größter gemeinsamer Teiler

Beispiel: Bestimme den größten gemeinsamen Teiler der Zahlen 96 und 54.

Eingabe: `gcd(96,54);`

Ausgabe: `6`

Interpretation: $\mathrm{ggT}(96, 54) = 6$

Hinweise: `gcd` steht für „greatest common divisor“. Der Befehl funktioniert nicht direkt, wenn mehr als zwei Zahlen gegeben sind. Beispielsweise müsste man $\mathrm{ggT}(12, 15, 18)$ durch `gcd(gcd(12,15),18)` ausdrücken.

3.7. Kleinstes gemeinsames Vielfaches

Beispiel: Bestimme das kleinste gemeinsame Vielfache der Zahlen 96 und 54.

Eingabe:	`lcm(96,54);`
Ausgabe:	`864`
Interpretation:	$\text{kgV}(96, 54) = 864$

Hinweise: `lcm` steht für „least common multiple“. Der Befehl funktioniert – im Gegensatz zu `gcd` – auch für mehr als zwei gegebene Zahlen.

4. Algebra

4.1. Zusammenfassen gleichartiger Terme

Beispiel: Vereinfache den Term $2a - 5b + 3a + 8b - 7a$.

Eingabe: `2*a-5*b+3*a+8*b-7*a;`

Ausgabe: $3b - 2a$

Hinweis: Bei der Eingabe darf das Multiplikationszeichen `*` zwischen Koeffizient und Variable nicht weggelassen werden.

4.2. Auflösen von Klammern

Beispiel: Schreibe den Term $a - (b + (c - d))$ ohne Klammern.

Eingabe: `a-(b+(c-d));`

Ausgabe: $d - c - b + a$

4.3. Multiplikation einfacher Terme

Beispiel: Vereinfache den Term $5x^2y \cdot (-3xy^3z)$.

Eingabe: `5*x^2*y*(-3*x*y^3*z);`

Ausgabe: $-15x^3y^4z$

Hinweis: Potenzen werden durch das Symbol `^` ausgedrückt.

4.4. Ausmultiplizieren

Zum Ausmultiplizieren benötigt man den Maxima-Befehl `expand`. Dieser Befehl wandelt ein Produkt von Summen in eine Summe von Produkten um und fasst anschließend gleichartige Terme zusammen.

Beispiel: Multipliziere den Term $(a - 5b)(2a + 5b)$ aus.

Eingabe: `expand((a-5*b)*(2*a+5*b));`

Ausgabe: $-25b^2 - 5ab + 2a^2$

4.5. Faktorisieren

Faktorisieren einer Summe oder Differenz bedeutet die Umwandlung in ein Produkt. Dafür ist der Maxima-Befehl `factor` zuständig.

Beispiel: Zerlege den Term $-4a^2b + 10ab^2$ in Faktoren.

Eingabe: `factor(-4*a^2*b+10*a*b^2);`

Ausgabe: $2ab(5b - 2a)$

4.6. Vereinfachung von Bruchtermen

Bruchterme können mit dem Befehl `ratsimp` vereinfacht werden.

Beispiel: Vereinfache: $\frac{x+5}{2x+3} - \frac{x+4}{2x-3} + \frac{21}{4x^2-9}$

Eingabe: `ratsimp((x+5)/(2*x+3)-(x+4)/(2*x-3)+21/(4*x^2-9));`

Ausgabe: $-\frac{2}{2x-3}$

Bemerkung: Etwas schöner wäre hier das Ergebnis $\frac{2}{3-2x}$.

4.7. Auflösen von Bruchgleichungen

Beispiel: Löse die Gleichung $\frac{3x-5}{x^2-1} - \frac{2x+9}{x^2-x} = \frac{1}{x+1}$ nach x auf.

Eingabe:
```
gl: (3*x-5)/(x^2-1)-(2*x+9)/(x^2-x)=1/(x+1);
solve(gl,x);
```

Ausgabe: $[x = -\frac{3}{5}]$

Hinweis 1: Es ist sinnvoll, insbesondere bei komplizierten Gleichungen, der Gleichung einen Namen zu geben (hier `gl`) und dann die Gleichung und den `solve`-Befehl auf zwei Zeilen zu verteilen. Gleichwertig, aber unübersichtlicher, wäre im Beispiel:

```
solve((3*x-5)/(x^2-1)-(2*x+9)/(x^2-x)=1/(x+1),x);
```

Hinweis 2: Der `solve`-Befehl erwartet die Angabe der Gleichung und (nach einem Komma) die Angabe der Unbekannten, nach der aufgelöst werden soll.

Hinweis 3: Da eine Gleichung auch mehrere Lösungen haben kann, gibt der `solve`-Befehl eine Liste von Lösungen aus, hier eine Liste mit einer Lösung. Listen erkennt man an den eckigen Klammern.

4.8. Lineare Gleichungssysteme mit mehreren Unbekannten

Beispiel: Löse das folgende Gleichungssystem nach den Unbekannten x, y und z auf:

$$\begin{array}{llcrcrcl}
(1) & 3x & - & 5y & - & 6z & = & -3 \\
(2) & 2x & & & + & z & = & 6 \\
(3) & 3x & - & 5y & + & z & = & 4
\end{array}$$

Eingabe:

```
gl1: 3*x-5*y-6*z = -3$
gl2: 2*x+z = 6$
gl3: 3*x-5*y+z = 4$
linsolve([gl1,gl2,gl3],[x,y,z]);
```

Ausgabe: $[x = \frac{5}{2}, y = \frac{9}{10}, z = 1]$

5. Analysis

5.1. Grenzwerte

Beispiel: Durch $f(x) = \dfrac{x-1-\ln x}{(x-1)^2}$ ist eine Funktion f mit der Definitionsmenge $D = \mathbb{R}^+ \setminus \{1\}$ gegeben. Berechne die Grenzwerte $\lim\limits_{x\to 0} f(x)$, $\lim\limits_{x\to 1} f(x)$ und $\lim\limits_{x\to\infty} f(x)$.

Eingabe:
```
f(x) := (x-1-log(x))/(x-1)^2$
limit(f(x),x,0);
limit(f(x),x,1);
limit(f(x),x,inf);
```

Ausgabe: infinity

$\frac{1}{2}$

0

Interpretation: $\lim\limits_{x\to 0} f(x) = \infty$; $\lim\limits_{x\to 1} f(x) = \frac{1}{2}$; $\lim\limits_{x\to\infty} f(x) = 0$

Hinweis: Ein wenig Vorsicht ist angebracht. Gibt man `limit(f(x),x,-inf);` ein, so erhält man als Ergebnis 0, obwohl der Grenzwert $\lim\limits_{x\to -\infty} f(x)$ aufgrund der Definitionsmenge gar nicht definiert ist.

5.2. Differentialrechnung

5.2.1. Ableitungsfunktion

Beispiel: Bestimme die Ableitungsfunktion f' von $f:\ x \mapsto \dfrac{x}{x^2-1}$.

Eingabe:
```
diff(x/(x^2-1),x);
```

Ausgabe: $\dfrac{1}{(x-1)^2} - \dfrac{2x}{(x-3)^3}$

Das Ergebnis ist richtig, aber nicht besonders „schön“, und sollte daher nachbearbeitet werden.

Eingabe:
```
ratsimp(%)
```

Ausgabe: $-\dfrac{x+1}{x^3-3x^2+3x-1}$

Eingabe:
```
factor(%)
```

Ausgabe: $-\dfrac{x+1}{(x-1)^3}$

5.3. Integralrechnung

5.3.1. Stammfunktion

Beispiel: Ermittle für $f:\ x \mapsto \dfrac{x}{e^x}$ den Funktionsterm einer Stammfunktion F.

Eingabe: `integrate(x/%e^x,x);`

Ausgabe: $(-x-1)\%e^{-x}$

Interpretation: $F(x) = (-x-1)\,e^{-x} + C$

5.3.2. Bestimmtes Integral

Beispiel: Berechne das bestimmte Integral $\int\limits_1^2 x^2\,dx$.

Eingabe: `integrate(x^2,x,1,2);`

Ausgabe: $\frac{7}{3}$

Interpretation: $\int\limits_1^2 x^2\,dx = \frac{7}{3}$

6. Vektorrechnung (Dimension 3)

6.1. Darstellung von Vektoren

Vektoren werden durch Listen ausgedrückt, also durch eckige Klammern, in denen – getrennt durch Kommata – die Koordinaten stehen.

Beispiel: $\vec{v} = \begin{pmatrix} 3 \\ -4 \\ 1 \end{pmatrix}$

Eingabe: `[3,-4,1]$`

Hinweis: Bei Kommazahlen ist Vorsicht geboten. Eingaben wie `1.2` führen zu fehlerhaften Ergebnissen, stattdessen sollte hier `rat(1.2)` oder `6/5` eingegeben werden.

6.2. Schreibweise als Spaltenvektor

Die übliche Schreibweise von Vektoren lässt sich mit dem Befehl `matrix` realisieren.

```
Vektor(v) := matrix([v[1]],[v[2]],[v[3]])$
```

Erläuterung: Der `matrix`-Befehl erwartet als Eingabeparameter Listen, die den Zeilen der gegebenen Matrix entsprechen. Bei einem Spaltenvektor enthält jede Zeile nur ein Element. Daher werden die Koordinaten `v[1]`, `v[2]` und `v[3]` jeweils in eine Liste (eckige Klammer) gesteckt.

6.3. Addition und Subtraktion von Vektoren

Addition und Subtraktion von Vektoren werden durch `+` bzw. `-` ausgedrückt.

Beispiel: Berechne die Summe der Vektoren $\begin{pmatrix} 3 \\ 0 \\ -4 \end{pmatrix}$ und $\begin{pmatrix} -3 \\ 2 \\ 8 \end{pmatrix}$.

Eingabe: `[3,0,-4]+[-3,2,8];`

Ausgabe: `[0,2,4]`

Interpretation: $\begin{pmatrix} 3 \\ 0 \\ -4 \end{pmatrix} + \begin{pmatrix} -3 \\ 2 \\ 8 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 4 \end{pmatrix}$

6.4. S-Multiplikation

Für die S-Multiplikation, also für die Multiplikation eines Skalars (einer Zahl) mit einem Vektor, verwendet man den Stern `*`.

Beispiel: Multipliziere die Zahl 3 mit dem Vektor $\begin{pmatrix} 2 \\ -7 \\ 1 \end{pmatrix}$.

Eingabe: `3*[2,-7,1];`

Ausgabe: `[6,-21,3]`

Interpretation: $3 \begin{pmatrix} 2 \\ -7 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ -21 \\ 3 \end{pmatrix}$

6.5. Skalarprodukt zweier Vektoren

Das Skalarprodukt wird durch einen Punkt ausgedrückt.

Beispiel: Berechne das Skalarprodukt der Vektoren $\begin{pmatrix} 2 \\ -7 \\ 4 \end{pmatrix}$ und $\begin{pmatrix} -5 \\ 3 \\ 1 \end{pmatrix}$.

Eingabe: `[2,-7,4].[-5,3,1];`

Ausgabe: `-27`

Interpretation: $\begin{pmatrix} 2 \\ -7 \\ 4 \end{pmatrix} \circ \begin{pmatrix} -5 \\ 3 \\ 1 \end{pmatrix} = -27$

6.6. Betrag eines Vektors

Hier wird eine einfache Maxima-Funktion für den Betrag eines Vektors (anschaulich die Länge des Vektorpfeils) definiert. Sie beruht auf der Beziehung $|\overrightarrow{v}| = \sqrt{\overrightarrow{v} \circ \overrightarrow{v}}$.

```
Betrag(v):=sqrt(v.v)$
```

Beispiel: Berechne den Betrag des Vektors $\overrightarrow{v} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$.

Eingabe:
```
v: [1,-2,3]$
Betrag(v);
```

Ausgabe: $\sqrt{14}$

Interpretation: $|\overrightarrow{v}| = \left| \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} \right| = \sqrt{14}$

6.7. Kreuzprodukt zweier Vektoren

Hierfür ist es sinnvoll, eine neue Maxima-Funktion `Kreuzprodukt` zu definieren.

```
Kreuzprodukt(v1,v2) :=
   [v1[2]*v2[3]-v1[3]*v2[2],             /* 1. Koordinate */
    v1[3]*v2[1]-v1[1]*v2[3],             /* 2. Koordinate */
    v1[1]*v2[2]-v1[2]*v2[1]]$            /* 3. Koordinate */
```

Beispiel: Berechne das Kreuzprodukt der Vektoren $\begin{pmatrix} -5 \\ 2 \\ 6 \end{pmatrix}$ und $\begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix}$.

Eingabe: `Kreuzprodukt([-5,2,6],[2,-4,1]);`

Ausgabe: `[26,17,16]`

Interpretation: $\begin{pmatrix} -5 \\ 2 \\ 6 \end{pmatrix} \times \begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix} = \begin{pmatrix} 26 \\ 17 \\ 16 \end{pmatrix}$

6.8. Lineare Abhängigkeit von zwei Vektoren

Zur Überprüfung, ob zwei gegebene Vektoren linear abhängig sind, definiert man sinnvollerweise eine neue Maxima-Funktion, die einen Wahrheitswert (`true` oder `false`) ausgibt. In dieser Maxima-Funktion wird die Tatsache verwendet, dass zwei Vektoren im $\mathbb{R}^3$ genau dann linear abhängig sind, wenn ihr Kreuzprodukt der Nullvektor ist.

```
linearAbhaengig2(v1,v2) := is(Kreuzprodukt(v1,v2)=[0,0,0])$
```

Hinweis: Da die Methode `Kreuzprodukt` verwendet wird, muss selbstverständlich auch ihre Definition (siehe 6.7) eingebunden werden.

Beispiel: Überprüfe die lineare Abhängigkeit der Vektoren $\begin{pmatrix} -4 \\ 7 \\ 8 \end{pmatrix}$ und $\begin{pmatrix} 2 \\ -3{,}5 \\ -4 \end{pmatrix}$.

Eingabe:
```
v1: [-4,7,8]$
v2: [2,rat(-3.5),-4]$
linearAbhaengig2(v1,v2);
```

Ausgabe: `true`

Interpretation: Die Vektoren $\begin{pmatrix} -4 \\ 7 \\ 8 \end{pmatrix}$ und $\begin{pmatrix} 2 \\ -3{,}5 \\ -4 \end{pmatrix}$ sind linear abhängig.

Hinweis: Beim zweiten Vektor darf man nicht `-3.5` als zweite Koordinate eingeben.

6.9. Lineare Abhängigkeit von drei Vektoren

Drei Vektoren im $\mathbb{R}^3$ sind genau dann linear abhängig, wenn ihre Determinante gleich 0 ist. Auf diesem Zusammenhang beruht die folgende Maxima-Funktion:

```
linearAbhaengig3(v1,v2,v3) := block(
  [d],
  d: determinant(matrix(v1,v2,v3)),
  is(d=0)
  )$
```

Beispiel:

Überprüfe, ob $\overrightarrow{v1} = \begin{pmatrix} 3 \\ 6 \\ -2 \end{pmatrix}$, $\overrightarrow{v2} = \begin{pmatrix} 5 \\ -2 \\ -4 \end{pmatrix}$ und $\overrightarrow{v3} = \begin{pmatrix} 11 \\ 10 \\ -8 \end{pmatrix}$ linear abhängig sind.

Eingabe:

```
v1: [3,6,-2]$
v2: [5,-2,-4]$
v3: [11,10,-8]$
linearAbhaengig3(v1,v2,v3);
```

Ausgabe: `true`

Interpretation: Die Vektoren $\overrightarrow{v1}$, $\overrightarrow{v2}$ und $\overrightarrow{v3}$ sind linear abhängig.

6.10. Winkel zwischen zwei Vektoren

Mit der Beziehung $\overrightarrow{a} \circ \overrightarrow{b} = |\overrightarrow{a}||\overrightarrow{b}|\cos\varphi$ lässt sich (näherungsweise) der Winkel φ berechnen, den die zugehörigen Vektorpfeile einschließen. Die folgende Maxima-Funktion realisiert eine solche Berechnung.

```
Winkel(v1,v2) := acos(float(v1.v2/(Betrag(v1)*Betrag(v2))))$
```

Erläuterung: Das Skalarprodukt `v1.v2` wird durch das Produkt der Beträge von `v1` und `v2` dividiert. Die Maxima-Funktion `float` bewirkt, dass näherungsweise statt symbolisch gerechnet wird. Zum Schluss wird mit der Umkehrfunktion der Kosinusfunktion (`acos`) der Winkel ermittelt, und zwar im Bogenmaß. Das Resultat liegt im Intervall $[0; \pi]$.

6.11. Optimierung eines Richtungsvektors

Die Vektorgleichungen von Geraden und Ebenen enthalten Richtungsvektoren, bei denen es nur auf die Richtung, aber nicht auf den Betrag ankommt. Der folgende Befehl `Optimierung` liefert zu einem gegebenen Richtungsvektor ein geeignetes Vielfaches (natürlich nicht das 0-fache), bei dem die Koordinaten nach Möglichkeit ganzzahlig sind und keine gemeinsamen Faktoren enthalten. Außerdem sollen nicht unnötig viele Minuszeichen vorkommen.

```
Optimierung(v) := block(
  [f,w],
  f: lcm(denom(v[1]),denom(v[2]),denom(v[3])),
  w: f*v,
  f: gcd(gcd(w[1],w[2]),w[3]),
  w: (1/f)*w,
  if w[1]<0 and w[2]<0 and w[3]<0 then return(-w),
  if w[1]<0 and w[2]<0 then return(-w),
  if w[1]<0 and w[3]<0 then return(-w),
  if w[2]<0 and w[3]<0 then return(-w),
  if w[1]<0 and w[2]=0 and w[3]=0 then return(-w),
  if w[1]=0 and w[2]<0 and w[3]=0 then return(-w),
  if w[1]=0 and w[2]=0 and w[3]<0 then return(-w),
  w
  )$
```

7. Analytische Geometrie (Dimension 3)

Aufbauend auf dem letzten Abschnitt (Vektorrechnung), sollen hier Maxima-Funktionen entwickelt werden, mit denen man Standardaufgaben aus der Analytischen Geometrie lösen kann. Auf Absicherung gegen fehlerhafte Eingaben habe ich dabei weitgehend verzichtet, damit das Wesentliche der Programmierung besser zu sehen ist.

Die Koordinaten werden hier mit x, y, z bezeichnet; dadurch verringert sich in manchen Fällen ein wenig die Schreibarbeit. Wenn Sie großen Wert auf die ebenfalls weit verbreiteten Bezeichnungen x_1, x_2, x_3 legen, müssten Sie einige der Maxima-Quelltexte geringfügig abändern, vor allem im Zusammenhang mit der Normalenform der Ebenengleichung.

Punkte und ihre Ortsvektoren werden grundsätzlich mit demselben Großbuchstaben benannt; der Ortsvektor von Punkt A erhält also die Bezeichnung $\overrightarrow{A}$.

7.1. Punkte

7.1.1. Darstellung von Punkten

Punkte werden genauso dargestellt wie ihre Ortsvektoren, also durch Listen (eckige Klammern), in denen die Koordinaten stehen.

Beispiel: Durch $P(4|-1|2)$ wird ein Punkt P festgelegt.

Eingabe: `P: [4,-1,2]$`

Hinweis 1: Kommazahlen (in Maxima-Schreibweise mit Punkt) führen im Allgemeinen zu falschen Ergebnissen. Daher sollte man etwa für `1.25` entweder `rat(1.25)` oder `5/4` eingeben.

Hinweis 2: Die einzelnen Koordinaten eines Punktes `P` werden durch `P[1]`, `P[2]` und `P[3]` ausgedrückt.

7.1.2. Bildpunkt bei Punktspiegelung

Wird eine Punktspiegelung mit Zentrum Z auf einen Punkt P angewandt, so erhält man den Ortsvektor des Bildpunkts P' mithilfe der einfachen Beziehung $\overrightarrow{P'} = 2\overrightarrow{Z} - \overrightarrow{P}$. Damit lässt sich eine einfache Maxima-Funktion formulieren:

```
SpiegelpunktPP(z,p) := 2*z-p$
```

Beispiel: Welche Koordinaten hat der Bildpunkt von $A(1|2|3)$ bezüglich der Punktspiegelung mit dem Zentrum $Z(-5|2|4)$?

Eingabe:
```
A: [1,2,3]$
Z: [-5,2,4]$
```

Ausgabe: `[-11,2,5]`

Interpretation: Der Bildpunkt von A ist $A'(-11|2|5)$.

7.2. Geraden

7.2.1. Darstellung von Geraden (Punkt-Richtungs-Form)

Eine Gerade wird im Folgenden immer durch eine Liste, bestehend aus einem Stützvektor (Liste von drei Zahlen) und einem Richtungsvektor (Liste von drei Zahlen), dargestellt. Eine solche Liste entspricht einer Parametergleichung (Vektorgleichung) in Punkt-Richtungs-Form.

Beispiel: Die Gerade g ist festgelegt durch die Gleichung $\overrightarrow{X} = \begin{pmatrix} -4 \\ 3 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 7 \\ -2 \\ 2 \end{pmatrix}$.

Eingabe: `g: [[-4,3,0],[7,-2,2]]$`

Hinweis 1: Wie bei Punkten führen Kommazahlen im Allgemeinen zu falschen Ergebnissen. Daher sollte man beispielsweise statt `0.5` entweder `rat(0.5)` oder `1/2` eingeben.

Hinweis 2: Der Name des Parameters spielt keine Rolle.

Hinweis 3: Durch `g[1]` erhält man den Stützvektor, durch `g[2]` den Richtungsvektor der Geradengleichung.

7.2.2. Optimerung des Richtungsvektors

Der Maxima-Befehl `OptimierungG` sorgt eventuell beim Richtungsvektor für „schönere“ Zahlen.

```
OptimierungG(g) := [g[1],Optimierung(g[2])]$
```

Beispiel: Geradengleichung $\overrightarrow{X} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} -2 \\ -1{,}6 \\ \frac{4}{3} \end{pmatrix}$

Eingabe:
```
g: [[0,0,1],[-2,rat(-1.6),4/3]]$
OptimierungG(g);
```

Ausgabe: $[[0, 0, 1], [15, 12, -10]]$

7.2.3. Ausgabe einer Parametergleichung

Der folgende Maxima-Befehl dient dazu, eine Parametergleichung möglichst „naturgetreu“ darzustellen.

```
AusgabeG(g,par) := block(
  simp: false,
  print("X = ",Vektor(g[1])+par*Vektor(g[2])),
  simp: true,
  ""
  )$
```

Beispiel: Geradengleichung $\overrightarrow{X} = \begin{pmatrix} 0 \\ 2 \\ -3 \end{pmatrix} + \alpha \begin{pmatrix} 3 \\ -5 \\ 4 \end{pmatrix}$

Eingabe:

```
g: [[0,2,-3],[3,-5,4]]$
AusgabeG(g,alpha);
```

Ausgabe: $X = \begin{bmatrix} 0 \\ 2 \\ -3 \end{bmatrix} + \alpha \begin{bmatrix} 3 \\ -5 \\ 4 \end{bmatrix}$

Hinweis: Der Parametername lambda wird nicht als griechischer Buchstabe dargestellt, sondern ausgeschrieben, für den Parameternamen gamma wird der Großbuchstabe Γ ausgegeben.

7.2.4. Zwei-Punkte-Form der Geradengleichung

Ist eine Gerade durch zwei Punkte A und B gegeben, so eignet sich der Differenzvektor $\vec{B} - \vec{A}$ als Richtungsvektor einer Geradengleichung.
Die Gleichung lautet dann $\vec{X} = \vec{A} + \lambda(\vec{B} - \vec{A})$.

Die einfache Maxima-Funktion `GeradePP` dient dem Aufstellen einer solchen Gleichung:

```
GeradePP(p1,p2) := [p1,p2-p1]$
```

Beispiel: Gib eine Gleichung der Geraden g durch $A(-4|7|5)$ und $B(2|-3|2)$ an.

Eingabe:

```
A: [-4,7,5]$
B: [2,-3,2]$
g: GeradePP(A,B);
```

Ausgabe: $[[-4, 7, 5], [6, -10, -3]]$

Interpretation: $\overrightarrow{X} = \begin{pmatrix} -4 \\ 7 \\ 5 \end{pmatrix} + \lambda \begin{pmatrix} 6 \\ -10 \\ -3 \end{pmatrix}$

7.2.5. Überprüfung: Liegt Punkt auf Gerade?

Ein Punkt P (mit Ortsvektor $\overrightarrow{P}$) liegt genau dann auf der Geraden mit der Gleichung $\overrightarrow{X} = \overrightarrow{A} + \lambda\overrightarrow{u}$, wenn die Vektoren $\overrightarrow{AP} = \overrightarrow{P} - \overrightarrow{A}$ und $\overrightarrow{u}$ linear abhängig sind. Damit lässt sich eine einfache Maxima-Funktion für solche Überprüfungen formulieren:

```
enthaltenPG(p,g) := linearAbhaengig2(p-g[1],g[2])$
```

Erläuterung: `p-g[1]` entspricht dem Vektor $\overrightarrow{P} - \overrightarrow{A}$, `g[2]` dem Richtungsvektor der Geraden.

Beispiel:

Liegt der Punkt $P(7|0|10)$ auf der Geraden g mit der Gleichung $\overrightarrow{X} = \begin{pmatrix} -1 \\ 4 \\ 2 \end{pmatrix} + \alpha \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix}$?

Eingabe:	`P: [7,0,10]$`
	`g: [[-1,4,2],[2,-1,2]]$`
	`enthaltenPG(P,g);`

Ausgabe:	true
Interpretation:	Der Punkt P liegt auf der Geraden g.

7.2.6. Überprüfung: Sind zwei Geraden parallel?

Um die Parallelität zweier Geraden zu überprüfen, genügt es, ihre Richtungsvektoren zu betrachten. Die Geraden sind genau dann zueinander parallel, wenn die Richtungsvektoren linear abhängig sind. Die neue Maxima-Funktion `parallelGG` gibt die Antwort (true oder false).

```
parallelGG(g1,g2) := linearAbhaengig2(g1[2],g2[2])$
```

Beispiel: Sind die Geraden mit den Gleichungen $g:\ \overrightarrow{X} = \begin{pmatrix} 3 \\ 5 \\ -6 \end{pmatrix} + \alpha \begin{pmatrix} 1 \\ 1 \\ 5 \end{pmatrix}$ und $h:\ \overrightarrow{X} = \begin{pmatrix} -2 \\ 8 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 3 \\ 3 \\ 15 \end{pmatrix}$ zueinander parallel?

Eingabe:	`g: [[3,5,-6],[1,1,5]]$`
	`h: [[-2,8,0],[3,3,15]]$`
	`parallelGG(g,h);`

Ausgabe	true
Interpretation:	g und h sind zueinander parallel.

7.2.7. Überprüfung: Sind zwei Geraden gleich?

Diese Überprüfung ist geringfügig komplizierter als die letzte. Daher benötigt man für die Überprüfungsfunktion `gleichGG` einen Block mit zwei Befehlen. Wenn die Geraden nicht parallel sind, können sie erst recht nicht gleich sein. Sind die Geraden parallel, so genügt es zu überprüfen, ob der Aufpunkt der einen Geraden auf der anderen Geraden liegt.

```
gleichGG(g1,g2) := block(
  if not parallelGG(g1,g2) then return(false),
  enthaltenPG(g2[1],g1)
  )$
```

Beispiel: Zeige, dass die durch $g:\ \overrightarrow{X} = \begin{pmatrix} 0 \\ 6 \\ -1 \end{pmatrix} + \alpha \begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}$ und $h:\ \overrightarrow{X} = \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} -2 \\ 3 \\ -1 \end{pmatrix}$ gegebenen Geraden gleich sind.

Eingabe:	`g: [[0,6,-1],[2,-3,1]]$`
	`h: [[2,3,0],[-2,3,-1]]$`
	`gleichGG(g,h);`

Ausgabe:	true
Interpretation:	Die Geraden g und h sind gleich.

7.2.8. Überprüfung: Schneiden sich zwei Geraden?

Um zu testen, ob sich zwei Geraden mit den Gleichungen $\overrightarrow{X} = \overrightarrow{A} + \alpha\overrightarrow{u}$ und $\overrightarrow{X} = \overrightarrow{B} + \beta\overrightarrow{v}$ in einem Punkt schneiden, wird man zuerst prüfen, ob parallele Geraden vorliegen; in diesem Fall ist klar, dass sich die beiden Geraden nicht schneiden. Zwei nicht parallele Geraden schneiden sich genau dann in einem Punkt, wenn die Richtungsvektoren $\overrightarrow{u}$ und $\overrightarrow{v}$ sowie der Verbindungsvektor $\overrightarrow{B} - \overrightarrow{A}$ komplanar, also linear abhängig sind.

```
schneidendGG(g1,g2) := block(
  if parallelGG(g1,g2) then return(false),
  linearAbhaengig3(g1[2],g2[2],g2[1]-g1[1])
  )$
```

Beispiel: Begründe rechnerisch, dass die Gerade $g:\ \overrightarrow{X} = \begin{pmatrix} 11 \\ -8 \\ 14 \end{pmatrix} + \lambda \begin{pmatrix} 3 \\ -4 \\ 7 \end{pmatrix}$ die x-Achse schneidet.

Eingabe:	`g: [[11,-8,14],[3,-4,7]]$`
	`a: [[0,0,0],[1,0,0]]$`
	`schneidendGG(g,a);`

Ausgabe:	true
Interpretation:	Die Gerade g schneidet die x-Achse.

7.2.9. Überprüfung: Sind zwei Geraden windschief?

Diese Überprüfung funktioniert ganz ähnlich wie die letzte: Zwei Geraden mit den Gleichungen $\overrightarrow{X} = \overrightarrow{A} + \alpha\overrightarrow{u}$ und $\overrightarrow{X} = \overrightarrow{B} + \beta\overrightarrow{v}$ sind genau dann windschief, wenn die Richtungsvektoren $\overrightarrow{u}$ und $\overrightarrow{v}$ sowie der Verbindungsvektor $\overrightarrow{B} - \overrightarrow{A}$ *nicht* komplanar, also *nicht* linear abhängig sind.

```
windschiefGG(g1,g2) := not linearAbhaengig3(g1[2],g2[2],g2[1]-g1[1])$
```

Beispiel: Bestätige, dass die Geraden mit den Gleichungen $g : \overrightarrow{X} = \lambda \begin{pmatrix}1\\1\\1\end{pmatrix}$ und $h : \overrightarrow{X} = \begin{pmatrix}0\\0\\1\end{pmatrix} + \mu \begin{pmatrix}1\\0\\-1\end{pmatrix}$ windschief sind.

Eingabe:	`g: [[0,0,0],[1,1,1]]$`
	`h: [[0,0,1],[1,0,-1]]$`
	`windschiefGG(g,h);`

Ausgabe:	true
Interpretation:	Die beiden Geraden sind tatsächlich windschief.

7.2.10. Überprüfung: Schneiden sich zwei Geraden senkrecht?

Zum Test, ob zwei Geraden sich senkrecht schneiden, eignet sich das Skalarprodukt der beiden Richtungsvektoren.

```
senkrechtGG(g1,g2) := block(
  if not schneidendGG(g1,g2) then return(false),
  is(g1[2].g2[2]=0)
  )$
```

Beispiel: Zeige, dass die durch $\overrightarrow{X} = \begin{pmatrix}9\\-21\\17\end{pmatrix} + \alpha \begin{pmatrix}2\\-5\\4\end{pmatrix}$ und $\overrightarrow{X} = \begin{pmatrix}-4\\-11\\-9\end{pmatrix} + \beta \begin{pmatrix}1\\2\\2\end{pmatrix}$ gegebenen Geraden sich senkrecht schneiden.

Eingabe: `g: [[9,-21,17],[2,-5,4]]$`
`h: [[-4,-11,-9],[1,2,2]]$`
`senkrechtGG(g,h);`

Ausgabe: true

Interpretation: Die beiden Geraden schneiden sich senkrecht.

7.2.11. Lagebeziehung zweier Geraden

Die folgende Maxima-Funktion fasst die zuletzt besprochenen Überprüfungen zusammen. Man beachte, dass senkrechtes Schneiden als eigener Fall betrachtet wird.

```
LageGG(g1,g2) := block(
  if gleichGG(g1,g2) then return("Geraden gleich"),
  if parallelGG(g1,g2) then
    return("Geraden parallel, aber nicht gleich"),
  if senkrechtGG(g1,g2) then return("Geraden schneiden sich senkrecht"),
  if schneidendGG(g1,g2) then return("Geraden schneiden sich"),
  "Geraden windschief"
  )$
```

Beispiel: Wie liegen die Geraden, die durch die Gleichungen $g:\ \overrightarrow{X} = \begin{pmatrix}-3\\0\\4\end{pmatrix} + \sigma\begin{pmatrix}2\\0\\-7\end{pmatrix}$ und $h:\ \overrightarrow{X} = \begin{pmatrix}2\\-2\\0\end{pmatrix} + \tau\begin{pmatrix}1\\-5\\-6\end{pmatrix}$ gegeben sind, zueinander?

Eingabe: `g: [[-3,0,4],[2,0,-7]]$`
`h: [[2,-2,0],[1,-5,-6]]$`
`LageGG(g,h);`

Ausgabe: Geraden windschief

7.2.12. Schnittpunkt zweier Geraden

Die Berechnung des Schnittpunkts zweier Geraden, sofern existent und eindeutig, erfolgt üblicherweise durch Gleichsetzen der zugehörigen Parametergleichungen. Die so entstehende Vektorgleichung (hier `vg`) mit zwei Unbekannten (hier `a` und `b`) lässt sich in ein Gleichungssystem mit drei Gleichungen zerlegen. Die hier definierte Maxima-Funktion `Gleichungssystem` nimmt diese Zerlegung vor.

Das Gleichungssystem wird nun durch den Maxima-Befehl **`linsolve`** nach den Unbekannten aufgelöst. Die Lösung (`lsg`) ist eine Liste nach dem Schema `[a=..., b=...]`. Der Wert des Parameters `a` ist die rechte Seite des ersten Listenelements und lässt sich daher als **`rhs`**`(lsg[1])` ausdrücken. Einsetzen dieses Parameterwerts in die Gleichung der ersten Geraden ergibt schließlich den Ortsvektor des gesuchten Schnittpunkts.

```
Gleichungssystem(vg) := block(
  [ls,rs],
  ls: lhs(vg),
  rs: rhs(vg),
  [ls[1]=rs[1],ls[2]=rs[2],ls[3]=rs[3]]
  )$

SchnittpunktGG(g1,g2) := block(
  [vg,a,b,lsg],
  if gleichGG(g1,g2) then return("Geraden gleich!"),
  if parallelGG(g1,g2) then return("Geraden parallel"),
  if windschiefGG(g1,g2) then return("Geraden windschief"),
  vg: g1[1]+a*g1[2]=g2[1]+b*g2[2],
  lsg: linsolve(Gleichungssystem(vg),[a,b]),
  g1[1]+rhs(lsg[1])*g1[2]
  )$
```

Beispiel: In welchem Punkt schneiden sich die Geraden mit den Gleichungen
$g:\ \overrightarrow{X} = \begin{pmatrix} 12 \\ 7 \\ -14 \end{pmatrix} + \alpha \begin{pmatrix} 3 \\ 2 \\ -4 \end{pmatrix}$ und $h:\ \overrightarrow{X} = \begin{pmatrix} 9 \\ -4 \\ -1 \end{pmatrix} + \beta \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$?

Eingabe:	`g: [[12,7,-14],[3,2,-4]]$`
	`h: [[9,-4,-1],[-3,1,1]]$`
	`SchnittpunktGG(g,h);`

Ausgabe:	$[0, -1, 2]$
Interpretation:	g und h schneiden sich im Punkt $(0\vert -1\vert 2)$.

7.2.13. Parallele zu Gerade durch Punkt

Diese Aufgabenstellung ist sehr einfach. Der gegebene Punkt dient als Aufpunkt, der Richtungsvektor der gegebenen Geraden wird übernommen.

```
ParalleleGP(g,p) := [p,g[2]]$
```

Beispiel: Die Gerade g ist festgelegt durch $\overrightarrow{X} = \begin{pmatrix} -6 \\ 2 \\ 7 \end{pmatrix} + \lambda \begin{pmatrix} 5 \\ -8 \\ 2 \end{pmatrix}$. Gib eine Gleichung an für die Parallele zu g durch den Punkt $(2|2|2)$.

Eingabe: `g: [[-6,2,7],[5,-8,2]]$`
`P: [2,2,2]$`
`ParalleleGP(g,P);`

Ausgabe: $[[2,2,2],[5,-8,2]]$

Interpretation: $\overrightarrow{X} = \begin{pmatrix}2\\2\\2\end{pmatrix} + \mu \begin{pmatrix}5\\-8\\2\end{pmatrix}$

7.2.14. Fußpunkt des Lotes zu Gerade durch Punkt

Der Fußpunkt des Lotes von einem Punkt auf eine Gerade, anders ausgedrückt die senkrechte Projektion eines Punkts auf eine Gerade, wird in der Analytischen Geometrie des Öfteren benötigt. Daher wird hier eine entsprechende Maxima-Funktion eingeführt.

```
FusspunktGP(g,p) := block(
  [par],
  par: (g[2].(p-g[1]))/(g[2].g[2]),
  g[1]+par*g[2]
  )$
```

Erläuterung: Gegeben seien eine Gerade g mit Gleichung $\overrightarrow{X} = \overrightarrow{A} + \lambda\overrightarrow{u}$ und ein Punkt P mit Ortsvektor $\overrightarrow{P}$. Den Ortsvektor $\overrightarrow{F}$ des Fußpunkts kann man ermitteln, indem man eine Gleichung aufstellt für die Ebene, die zu g senkrecht ist und durch P geht. $\overrightarrow{u} \circ (\overrightarrow{X} - \overrightarrow{P}) = 0$ ist eine solche Gleichung. Für $\overrightarrow{X}$ setzt man jetzt den Rechenausdruck aus der Geradengleichung ein:

$$\overrightarrow{u} \circ (\overrightarrow{A} + \lambda\overrightarrow{u} - \overrightarrow{P}) = 0$$

So erhält man den Parameterwert $\lambda = \dfrac{\overrightarrow{u} \circ (\overrightarrow{P} - \overrightarrow{A})}{\overrightarrow{u} \circ \overrightarrow{u}}$ für den gesuchten Fußpunkt. Diesen Wert muss man nur noch in die Geradengleichung einsetzen.

Beispiel: Vom Punkt $P(6|-2|1)$ wird ein Lot auf die Gerade g mit der Gleichung $\overrightarrow{X} = \begin{pmatrix}0\\5\\-2\end{pmatrix} + \alpha \begin{pmatrix}-4\\0\\3\end{pmatrix}$ gefällt. Welche Koordinaten hat der Fußpunkt dieses Lotes?

Eingabe: `g: [[0,5,-2],[-4,0,3]]$`
`P: [6,-2,1]$`
`FusspunktGP(g,P);`

Ausgabe: $[\frac{12}{5}, 5, -\frac{19}{5}]$

Interpretation: Der Fußpunkt des Lotes ist $F(2{,}4|5|-3{,}8)$.

7.2.15. Lot zu Gerade durch Punkt

Mithilfe der zuletzt definierten Maxima-Funktion `FusspunktGP` (siehe 7.2.14) lässt sich leicht ein Befehl für das Lot zu einer gegebenen Geraden durch einen gegebenen Punkt formulieren.

```
LotGP(g,p) := block(
  if enthaltenPG(p,g) then
    return("Lot nicht eindeutig definiert!"),
  OptimierungG(GeradePP(p,FusspunktGP(g,p)))
  )$
```

Erläuterung: Liegt der gegebene Punkt auf der gegebenen Geraden, so ist das Lot (im Sinne von Lotgerade) nicht eindeutig definiert, und es erfolgt eine Fehlermeldung. Im Normalfall dagegen wird zunächst der Fußpunkt des Lotes ermittelt. Anschließend wird zum gegebenen Punkt und zum Fußpunkt die Verbindungsgerade gebildet. Weil die Koordinaten des Fußpunkts meist nicht ganzzahlig sind, sorgt am Ende die Maxima-Funktion `OptimierungG` für einen Richtungsvektor mit „schönen“ Zahlen.

7.2.16. Spiegelpunkt bei Achsenspiegelung

Die Maxima-Funktion `FusspunktGP` (siehe 7.2.14) ermöglicht auch die Bestimmung von Bildpunkten bezüglich einer Achsenspiegelung.

```
SpiegelpunktGP(a,p) := 2*FusspunktGP(a,p)-p$
```

Beispiel: Der Punkt $P(-2|3|-4)$ wird an der Geraden mit der Gleichung $\overrightarrow{X} = \begin{pmatrix}0\\0\\1\end{pmatrix} + \lambda \begin{pmatrix}1\\5\\0\end{pmatrix}$ gespiegelt. Welche Koordinaten hat der Spiegelpunkt P'?

Eingabe:

```
g: [[0,0,1],[1,5,0]]$
P: [-2,3,-4]$
SpiegelpunktGP(g,P);
```

Ausgabe: $[3, 2, 6]$

Interpretation: Der Spiegelpunkt ist $P'(3|2|6)$.

7.3. Ebenen

7.3.1. Darstellung von Ebenen (Punkt-Richtungs-Form)

Eine Ebene wird hier entsprechend dargestellt wie eine Gerade, nämlich als Liste, die einen Stützvektor (Liste von drei Zahlen) und zwei Richtungsvektoren (Listen von jeweils drei Zahlen) enthält. Diese Darstellung entspricht einer Parametergleichung der Ebene in Punkt-Richt-Richtungs-Form:

$$\overrightarrow{X} = \overrightarrow{A} + \lambda\overrightarrow{u} + \mu\overrightarrow{v}$$

Beispiel:

Die Ebene E ist festgelegt durch die Gleichung $\overrightarrow{X} = \begin{pmatrix} 1 \\ -4 \\ 5 \end{pmatrix} + \lambda \begin{pmatrix} 3 \\ 7 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} -4 \\ 8 \\ 5 \end{pmatrix}$.

Eingabe: `E: [[1,-4,5],[3,7,0],[-4,8,5]]$`

Hinweis: Durch `E[1]` erhält man den Stützvektor, durch `E[2]` und `E[3]` die Richtungsvektoren der Ebenengleichung.

7.3.2. Optimierung der Richtungsvektoren

Wie bei Geradengleichungen ist es oft sinnvoll, für Richtungsvektoren mit einfacheren Zahlen zu sorgen.

```
OptimierungE(e) := [e[1],Optimierung(e[2]),Optimierung(e[3])]$
```

Beispiel: Ebenengleichung $\overrightarrow{X} = \begin{pmatrix} 2 \\ -7 \\ 6 \end{pmatrix} + \alpha \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} + \beta \begin{pmatrix} -\frac{1}{2} \\ \frac{5}{6} \\ \frac{3}{4} \end{pmatrix}$

Eingabe:
```
[[2,-7,6],[10,20,30],[-1/2,5/6,3/4]]$
OptimierungE(E);
```

Ausgabe: $[[2, -7, 6], [1, 2, 3], [-6, 10, 9]]$

7.3.3. Ausgabe einer Parametergleichung

Die Ausgabe des folgenden Maxima-Befehls entspricht in etwa der Form, die ein menschlicher Rechner von einer Parametergleichung erwartet.

```
AusgabeE(e,par1,par2) := block(
  simp: false,
  print("X = ",Vektor(e[1])+par1*Vektor(e[2])+par2*Vektor(e[3])),
  simp: true,
  ""
  )$
```

Erläuterung: Mithilfe der Variablen `simp` wird die automatische Vereinfachung vor der Ausgabe aus- und später wieder eingeschaltet. Durch die leere Zeichenkette `""` wird die Ausgabe `true` unterdrückt.

Beispiel: Ebenengleichung $\overrightarrow{X} = \begin{pmatrix} -4 \\ 0 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} 1 \\ -1 \\ 5 \end{pmatrix} + \beta \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix}$

Eingabe:

```
E: [[-4,0,1],[1,-1,5],[2,3,0]]$
AusgabeE(E,alpha,beta);
```

Ausgabe: $X = \begin{bmatrix} -4 \\ 0 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ -1 \\ 5 \end{bmatrix} + \beta \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix}$

7.3.4. Drei-Punkte-Form der Ebenengleichung

Eine Ebene ist eindeutig definiert durch drei Punkte A, B und C, wenn diese nicht auf einer Geraden liegen. Als Stützvektor kann $\overrightarrow{A}$ dienen, als Richtungsvektoren sind $\overrightarrow{B} - \overrightarrow{A}$ und $\overrightarrow{C} - \overrightarrow{A}$ geeignet. Damit erhält man die Gleichung $\overrightarrow{X} = \overrightarrow{A} + \lambda \left(\overrightarrow{B} - \overrightarrow{A}\right) + \mu \left(\overrightarrow{C} - \overrightarrow{A}\right)$. Diese Überlegung wird durch folgende Maxima-Funktion umgesetzt:

```
EbenePPP(p1,p2,p3) := [p1,p2-p1,p3-p1]$
```

Beispiel: Stelle eine Gleichung auf für die Ebene E, die durch die Punkte $A(2|-6|7)$, $B(-3|2|0)$ und $C(5|8|2)$ geht.

Eingabe:

```
A: [2,-6,7]$
B: [-3,2,0]$
C:[5,8,2]$
EbenePPP(A,B,C);
```

Ausgabe: `[[2,-6,7],[-5,8,-7],[3,14,-5]]`

Interpretation: $\overrightarrow{X} = \begin{pmatrix} 2 \\ -6 \\ 7 \end{pmatrix} + \lambda \begin{pmatrix} -5 \\ 8 \\ -7 \end{pmatrix} + \mu \begin{pmatrix} 3 \\ 14 \\ -5 \end{pmatrix}$

7.3.5. Ebene mit gegebener Normalenform

Die folgende Maxima-Funktion `EbeneNF` dient der Umwandlung einer Koordinatengleichung (Normalenform) in eine Parametergleichung (Vektorform).

```
EbeneNF(nf) := block(
  [ls,n1,n2,n3,n0,e],
  ls: expand(lhs(nf)-rhs(nf)),
  n1: coeff(ls,x),
  n2: coeff(ls,y),
  n3: coeff(ls,z),
  n0: ls-n1*x-n2*y-n3*z,
  if n1#0 then e: [[-n0/n1,0,0]]
  else if n2#0 then e: [[0,-n0/n2,0]]
  else e: [[0,0,-n0/n3]],
  if n2#0 or n3#0 then e: append(e,[[0,n3,-n2]]),
  if n1#0 or n3#0 then e: append(e,[[-n3,0,n1]]),
  if length(e)<3 then e: append(e,[[n2,-n1,0]]),
  OptimierungE(e)
  )$
```

Erläuterung: Zuerst wird in der Gleichung durch Subtraktion der rechten Seite alles auf eine Seite gebracht, sodass die Gleichung dem Schema $n_1x+n_2y+n_3z+n_0 = 0$ entspricht. Aus dieser Gleichung lassen sich die Koeffizienten n_1, n_2, n_3 und der konstante Summand n_0 ermitteln.

Als Aufpunkt der gesuchten Parametergleichung eignet sich prinzipiell jeder Punkt der Ebene. Hier wird als Aufpunkt ein Spurpunkt verwendet, das heißt ein Schnittpunkt der Ebene mit einer der drei Koordinatenachsen. Dabei ist zu beachten, dass unter Umständen nur ein oder zwei solche Spurpunkte existieren. Existierende Spurpunkte erhält man durch die Ausdrücke $(-\frac{n_0}{n_1}|0|0)$, $(0|-\frac{n_0}{n_2}|0)$ beziehungsweise $(0|0|-\frac{n_0}{n_3})$. Der Ortsvektor des gewählten Spurpunkts (Stützvektor) wird als erstes Element in eine neue Liste `e` übernommen.

Nun fehlen noch die Richtungsvektoren. Bei den Vektoren

$$\begin{pmatrix} 0 \\ n_3 \\ -n_2 \end{pmatrix}, \begin{pmatrix} -n_3 \\ 0 \\ n_1 \end{pmatrix} \text{ und } \begin{pmatrix} n_2 \\ -n_1 \\ 0 \end{pmatrix}$$

erkennt man leicht mithilfe des Skalarprodukts, dass sie senkrecht zum Normalenvektor oder gleich dem Nullvektor sind. Zwei solche Vektoren werden am Ende der Liste `e` hinzugefügt. Der Aufruf von `OptimierungE` bewirkt unter Umständen bei den Richtungsvektoren einfachere Zahlen.

Beispiel: Eine Ebene E ist durch die Gleichung $2x + y - 2z + 15 = 0$ gegeben. Gib eine Parametergleichung dieser Ebene an.

Eingabe:
```
nf: 2*x+y-2*7+15=0$
EbeneNF(nf);
```

Ausgabe: $[[-\frac{15}{2}, 0, 0], [0, 2, 1], [1, 0, 1]]$

Interpretation: $\overrightarrow{X} = \begin{pmatrix} -\frac{15}{2} \\ 0 \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$

7.3.6. Normalenvektor einer Ebene

Oft wird für Berechnungen ein Normalenvektor einer Ebene benötigt, den man zum Beispiel mit dem Kreuzprodukt der beiden Richtungsvektoren bestimmen kann. Dies führt zur folgenden (noch nicht optimalen) Maxima-Funktion:

```
NormalenvektorE(e) := Kreuzprodukt(e[2],e[3])$
```

Beispiel: Bestimme einen Normalenvektor zur Ebene E mit der Gleichung

$\overrightarrow{X} = \begin{pmatrix} 5 \\ 0 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix} + \mu \begin{pmatrix} -1 \\ 6 \\ 2 \end{pmatrix}$.

Eingabe:
```
E: [[5,0,-1],[3,2,4],[-1,6,2]]$
n: NormalenvektorE(E);
```

Ausgabe: $[-20, -10, 20]$

Interpretation: $\overrightarrow{n} = \begin{pmatrix} -20 \\ -10 \\ 20 \end{pmatrix}$ ist ein Normalenvektor der Ebene E.

Auch Vielfache dieses Vektors (außer dem 0-fachen) sind mögliche Normalenvektoren. Um einen „schöneren“ Normalenvektor zu erhalten, wird wieder die Maxima-Funktion `Optimierung` (siehe 6.11) verwendet.

Der verbesserte Maxima-Befehl für die Bestimmung eines Normalenvektors lautet nun:

```
NormalenvektorE(e) := Optimierung(Kreuzprodukt(e[2],e[3]))$
```

Neubearbeitung des letzten Beispiels ergibt:

Eingabe:
```
E: [[5,0,-1],[3,2,4],[-1,6,2]]$
n: NormalenvektorE(E);
```

Ausgabe: $[2, 1, -2]$

Interpretation: $\overrightarrow{n} = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix}$ ist ein Normalenvektor der Ebene E.

7.3.7. Normalenform einer Ebene

Ebenengleichungen in Normalenform sind oft einfacher zu handhaben als Vektorgleichungen. Die folgende Maxima-Funktion ermöglicht das einfache Aufstellen einer solchen Normalenform.

```
NormalenformE(e) := block(
  [n],
  n: NormalenvektorE(e),
  expand(n.([x,y,z]-e[1]))=0
  )$
```

Erläuterung: Die Normalenform lässt sich vektoriell formulieren: $\vec{n} \circ (\vec{X} - \vec{A}) = 0$. Dabei ist $\vec{n}$ ein Normalenvektor und $\vec{A}$ der Stützvektor der Parametergleichung. Die Maxima-Formulierung der Gleichung lautet `n.([x,y,z]-e[1])=0`. Der Befehl `expand` bewirkt das Ausmultiplizieren und Zusammenfassen von Summanden.

Beispiel: Bestimme eine Normalenform für $E: \vec{X} = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} + \alpha \begin{pmatrix} 4 \\ 1 \\ -1 \end{pmatrix} + \beta \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix}$.

Eingabe:
```
E: [[2,4,3],[4,1,-1],[-3,0,1]]$
NormalenformE(E);
```

Ausgabe: $x - y + 3z - 7 = 0$

7.3.8. Überprüfung: Liegt Punkt in Ebene?

Ein Punkt P mit Ortsvektor $\vec{P}$ liegt genau dann in der Ebene mit der Gleichung $\vec{X} = \vec{A} + \lambda \vec{u} + \mu \vec{v}$, wenn die drei Vektoren $\vec{P} - \vec{A}$, $\vec{u}$ und $\vec{v}$ linear abhängig sind.

```
enthaltenPE(p,e) := linearAbhaengig3(p-e[1],e[2],e[3])$!
```

Beispiel: Bestätige durch Rechnung, dass der Punkt $P(-13|38|24)$ in der Ebene mit der Gleichung $\vec{X} = \begin{pmatrix} 0 \\ 4 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} -1 \\ 2 \\ 5 \end{pmatrix} + \mu \begin{pmatrix} 1 \\ -3 \\ 0 \end{pmatrix}$ liegt.

Eingabe:
```
P: [-13,38,24]$
E: [[0,4,-1],[-1,2,5],[1,-3,0]]$
enthaltenPE(P,E);
```

Ausgabe: true

Interpretation: P liegt in E.

7.3.9. Überprüfung: Ist Gerade parallel zu Ebene?

Eine Gerade mit Gleichung $\overrightarrow{X} = \overrightarrow{A} + \lambda\overrightarrow{u}$ ist genau dann parallel zu einer Ebene mit Gleichung $\overrightarrow{X} = \overrightarrow{B} + \alpha\overrightarrow{v} + \beta\overrightarrow{w}$, wenn die drei Richtungsvektoren $\overrightarrow{u}$, $\overrightarrow{v}$ und $\overrightarrow{w}$ linear abhängig sind.

```
parallelGE(g,e) := linearAbhaengig3(g[2],e[2],e[3])$!
```

Beispiel: Begründe rechnerisch, dass die durch $\overrightarrow{X} = \begin{pmatrix} 5 \\ -4 \\ 2 \end{pmatrix} + \lambda \begin{pmatrix} 3 \\ -13 \\ -2 \end{pmatrix}$ festgelegte Gerade g zur Ebene E mit der Gleichung $\overrightarrow{X} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} 2 \\ -3 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}$ parallel ist.

Eingabe:

```
g: [[5,-4,2],[3,-13,-2]]$
E: [[1,1,1],[2,-3,0],[5,1,2]]$
parallelGE(g,E);
```

Ausgabe: true

Interpretation: g ist parallel zu E.

7.3.10. Überprüfung: Liegt Gerade in Ebene?

Die Überprüfung, ob eine Gerade g in einer Ebene E liegt, erfordert zwei Teilschritte: Zuerst testet man, ob g parallel zu E ist. Trifft dies nicht zu, dann kann die Gerade nicht in der Ebene liegen. Weiß man, dass g zu E parallel ist, so muss man nur noch überprüfen, ob der Aufpunkt von g in E liegt.

```
enthaltenGE(g,e) := block(
  if not parallelGE(g,e) then return(false),
  enthaltenPE(g[1],e)
  )$
```

Beispiel: Überprüfe für das letzte Beispiel, also $g: \overrightarrow{X} = \begin{pmatrix} 5 \\ -4 \\ 2 \end{pmatrix} + \lambda \begin{pmatrix} 3 \\ -13 \\ -2 \end{pmatrix}$ und $E: \overrightarrow{X} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} 2 \\ -3 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}$, ob die Gerade g in der Ebene E liegt.

Eingabe:

```
g: [[5,-4,2],[3,-13,-2]]$
E: [[1,1,1],[2,-3,0],[5,1,2]]$
enthaltenGE(g,E);
```

Ausgabe: false

Interpretation: g liegt nicht in E.

7.3.11. Überprüfung: Schneidet Gerade Ebene?

Eine Gerade und eine Ebene im Raum schneiden sich genau dann, wenn sie nicht parallel sind. Ein Maxima-Befehl für diese Überprüfung ist damit leicht zu formulieren:

```
schneidendGE(g,e) := not parallelGE(g,e)$
```

7.3.12. Überprüfung: Schneidet Gerade Ebene senkrecht?

Ob eine Gerade eine Ebene senkrecht schneidet, lässt sich leicht mithilfe eines Normalenvektors der Ebene prüfen.

```
senkrechtGE(g,e) := linearAbhaengig2(g[2],NormalenvektorE(e))$
```

Beispiel: $g: \overrightarrow{X} = \begin{pmatrix} 4 \\ -1 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 3 \\ 8 \end{pmatrix}; \quad E: \overrightarrow{X} = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix} + \sigma \begin{pmatrix} 1 \\ 5 \\ -2 \end{pmatrix} + \tau \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix}$

Schneidet die Gerade g die Ebene E senkrecht?

Eingabe:

```
g: [[4,-1,-1],[1,3,8]]$
E: [[2,0,-1],[1,5,-2],[3,-1,0]]$
senkrechtGE(g,E);
```

Ausgabe: true

Interpretation: g schneidet E tatsächlich senkrecht.

7.3.13. Lagebeziehung von Gerade und Ebene

Mit dem folgende Maxima-Befehl erfährt man, wie eine Gerade in Bezug auf eine Ebene liegt.

```
LageGE(g,e) := block(
  if enthaltenGE(g,e) then return("Gerade liegt in Ebene"),
  if parallelGE(g,e) then
    return("Gerade parallel zur Ebene, aber außerhalb"),
  if senkrechtGE(g,e) then return("Gerade schneidet Ebene senkrecht"),
  "Gerade schneidet Ebene"
  )$
```

Beispiel: Wie liegt die Gerade mit der Gleichung $g: \overrightarrow{X} = \begin{pmatrix} 2 \\ -7 \\ 4 \end{pmatrix} + \lambda \begin{pmatrix} 2 \\ -3 \\ 0 \end{pmatrix}$ zur Ebene mit der Gleichung $E: \overrightarrow{X} = \begin{pmatrix} 1 \\ 3 \\ -2 \end{pmatrix} + \sigma \begin{pmatrix} 0 \\ -2 \\ 5 \end{pmatrix} + \tau \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$?

Eingabe:	`g: [[2,-7,4],[2,-3,0]]$`
	`E: [[1,3,-2],[0,-2,5],[2,2,3]]$`
	`LageGE(g,E);`

Ausgabe:	Gerade schneidet Ebene

7.3.14. Schnittpunkt von Gerade und Ebene

Die folgende Maxima-Funktion `SchnittpunktGE` berechnet den Schnittpunkt einer Geraden und einer Ebene, sofern dieser existiert und eindeutig ist; andernfalls wird eine Fehlermeldung ausgegeben.

```
SchnittpunktGE(g,e) := block(
  [vg,a,b,c,lsg],
  if enthaltenGE(g,e) then return("Gerade in Ebene!"),
  if parallelGE(g,e) then return("Gerade parallel zur Ebene!"),
  vg: g[1]+a*g[2]=e[1]+b*e[2]+c*e[3],
  lsg: linsolve(Gleichungssystem(vg),[a,b,c]),
  g[1]+rhs(lsg[1])*g[2]
  )$
```

Erläuterung: Durch Gleichsetzen der Parametergleichungen von Gerade und Ebene erhält man eine Vektorgleichung (`vg`) der Form

$$\overrightarrow{A} + a\overrightarrow{u} = \overrightarrow{B} + b\overrightarrow{v} + c\overrightarrow{w}.$$

Diese Vektorgleichung wird mit dem Befehl `Gleichungssystem` (siehe 7.2.12) in ein System von drei Gleichungen umgewandelt und mit `linsolve` nach den Unbekannten `a`, `b` und `c` aufgelöst. Zum Schluss wird der Wert des Parameters `a` (in Maxima-Schreibweise `rhs(lsg[1])`) in die Geradengleichung $\overrightarrow{X} = \overrightarrow{A} + a\overrightarrow{u}$ eingesetzt.

Beispiel: Wo schneidet die durch $g: \ \overrightarrow{X} = \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix} + \lambda \begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}$ gegebene Gerade g die x-y-Ebene?

Eingabe:	`[[2,-1,2],[2,-3,1]]$`
	`E: [[0,0,0],[1,0,0],[0,1,0]]$`
	`SchnittpunktGE(g,E);`

Ausgabe:	$[-2, 5, 0]$
Interpretation:	g und E schneiden sich im Punkt $(-2\|5\|0)$.

7.3.15. Überprüfung: Sind zwei Ebenen parallel?

Die Parallelität zweier Ebenen überprüft man am einfachsten mithilfe der Normalenvektoren. Zwei Ebenen sind genau dann parallel zueinander, wenn ihre Normalenvektoren linear abhängig sind.

```
parallelEE(e1,e2) := block(
  [n1,n2],
  n1: NormalenvektorE(e1),
  n2: NormalenvektorE(e2),
  linearAbhaengig2(n1,n2)
  )$
```

Beispiel: Durch $\overrightarrow{X} = \begin{pmatrix} 1 \\ -4 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} + \mu \begin{pmatrix} 1 \\ -8 \\ 0 \end{pmatrix}$ und $\overrightarrow{X} = \begin{pmatrix} 2 \\ -6 \\ 2 \end{pmatrix} + \alpha \begin{pmatrix} 2 \\ -5 \\ 3 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix}$ sind zwei Ebenen festgelegt. Handelt es sich um parallele Ebenen?

Eingabe

```
E1: [[1,-4,3],[0,2,1],[1,-8,0]]$
E2: [[2,-6,2],[2,-5,3],[1,0,4]]$
parallelEE(E1,E2);
```

Ausgabe: true

Interpretation: E_1 und E_2 sind zueinander parallel.

7.3.16. Überprüfung: Sind zwei Ebenen gleich?

Will man herausfinden, ob zwei Ebenen gleich sind, so ist es sinnvoll, zuerst die Parallelität zu untersuchen. Sind die Ebenen nicht parallel, so können sie auch nicht gleich sein. Bei parallelen Ebenen genügt es zu überprüfen, ob der Aufpunkt der einen Ebene in der anderen Ebene liegt. Der Maxima-Befehl `gleichEE` ist daher folgendermaßen definiert:

```
gleichEE(e1,e2) := block(
  if not parallelEE(e1,e2) then return(false),
  enthaltenPE(e1[1],e2)
  )$
```

Beispiel: Zeige, dass die Ebenen mit den Gleichungen $\overrightarrow{X} = \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 1 \\ 5 \end{pmatrix} + \mu \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$ und $3x - 5y + z - 9 = 0$ übereinstimmen.

Eingabe:

```
E1: [[3,1,5],[0,1,5],[1,0,-3]]$
E2: EbeneNF(3*x-5*y+z-9=0)$
gleichEE(E1,E2);
```

Ausgabe: true

Interpretation: Die Ebenen stimmen überein.

7.3.17. Überprüfung: Schneiden sich zwei Ebenen?

Zwei Ebenen schneiden sich genau dann in einer Geraden, wenn sie nicht parallel sind.

```
schneidendEE(e1,e2) := not parallelEE(e1,e2)$
```

7.3.18. Überprüfung: Schneiden sich zwei Ebenen senkrecht?

Diesen Spezialfall überprüft man am besten mithilfe der Normalenvektoren.

```
senkrechtEE(e1,e2) := is(NormalenvektorE(e1).NormalenvektorE(e2)=0)$
```

Beispiel: Schneiden sich die Ebenen mit den Gleichungen $E_1: \ x + 5y - 2z + 4 = 0$ und $E_2: \ \overrightarrow{X} = \begin{pmatrix} 2 \\ 5 \\ -4 \end{pmatrix} + \alpha \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} + \beta \begin{pmatrix} 5 \\ 0 \\ 1 \end{pmatrix}$? Wenn ja, schneiden sie sich senkrecht?

Eingabe:

```
E1: EbeneNF(x+5*y-2*z+4=0)$
E2: [[2,5,-4],[1,-2,3],[5,0,1]]$
schneidendEE(E1,E2);
senkrechtEE(E1,E2);
```

Ausgabe: true
false

Interpretation: Die Ebenen E_1 und E_2 schneiden sich, aber nicht senkrecht.

7.3.19. Lagebeziehung zweier Ebenen

Die folgende Maxima-Funktion `LageEE` findet heraus, wie zwei Ebenen zueinander liegen.

```
LageEE(e1,e2) := block(
  if gleichEE(e1,e2) then return("Ebenen gleich!"),
  if parallelEE(e1,e2) then return("Ebenen parallel, aber nicht gleich"),
  if senkrechtEE(e1,e2) then return("Ebenen schneiden sich senkrecht"),
  "Ebenen schneiden sich"
  )$
```

Beispiel: Wie liegen die Ebenen mit den Gleichungen $3x-5z-2 = 0$ und $3x-5z+4 = 0$ zueinander?

Eingabe:

```
E1: EbeneNF(3*x-5*z-2=0)$
E2: EbeneNF(3*x-5*z+4=0)$
LageEE(E1,E2);
```

Ausgabe: Ebenen parallel, aber nicht gleich

7.3.20. Schnittgerade zweier Ebenen

Die Bestimmung der Schnittgeraden von Ebenen gehört eindeutig zu den komplizierteren Standardaufgaben der Analytischen Geometrie. Die hier definierte Maxima-Funktion `SchnittgeradeEE` verwendet folgendes Verfahren: Zuerst wird die zweite Ebene (`e2`) in Normalenform dargestellt; der zugehörige Normalenvektor ist mit `n2` bezeichnet. Die Parametergleichung der ersten Ebene, hier mit den Parametern `a` und `b`, wird in die Normalenform der zweiten Ebene eingesetzt, was zu einer Gleichung `gl` mit den Unbekannten `a` und `b` führt. `c0` ist der konstante Summand in dieser Gleichung, `c1` und `c2` sind die Koeffizienten von `a` und `b`. Die Gleichung wird nun mit dem Maxima-Befehl `solve` nach dem Parameter `b` aufgelöst; `lsg` ist (bei nicht parallelen Ebenen) eine Liste mit einem Element, das die Lösung `b=...` enthält. Der Ausdruck `rhs(lsg[1])` entspricht dem von `a` abhängigen Rechenausdruck für den Parameter `b`. Dieser Rechenausdruck wird nun in die Parametergleichung der ersten Ebene eingesetzt; anschließend wird ausmultipliziert und zusammengefasst (Maxima-Befehl `expand`). Damit hat man einen vektoriellen, von `a` abhängigen Ausdruck. Im letzten Schritt wird dieser Ausdruck zerlegt in einen Summanden, der `a` nicht enthält (`coeff(g,a,0)`, Stützvektor) und einen Summanden nach dem Schema „Parameter mal Richtungsvektor“. Der Richtungsvektor entspricht `coeff(g,a)` und kann mit der Maxima-Funktion `Optimierung` (siehe 6.11) „verschönert“ werden.

```
SchnittgeradeEE(e1,e2) := block(
  [n2,c0,c1,c2,gl,a,b,lsg,g],
  if gleichEE(e1,e2) then return("Ebenen gleich!"),
  if parallelEE(e1,e2) then return("Ebenen parallel!"),
  n2: NormalenvektorE(e2),
  c0: n2.(e1[1]-e2[1]),
  c1: n2.e1[2],
  c2: n2.e1[3],
  gl: c0+c1*a+c2*b=0,
  lsg: solve(gl,b),
  g: expand(e1[1]+a*e1[2]+rhs(lsg[1])*e1[3]),
  [coeff(g,a,0),Optimierung(coeff(g,a))]
  )$
```

Beispiel: Durch $E_1: \ x + 5y - 2z + 4 = 0$ und $E_2: \ \overrightarrow{X} = \begin{pmatrix} 2 \\ 5 \\ -4 \end{pmatrix} + \alpha \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} + \beta \begin{pmatrix} 5 \\ 0 \\ 1 \end{pmatrix}$ sind zwei Ebenen E_1 und E_2 festgelegt. Bestimme die Schnittgerade der beiden Ebenen.

Eingabe:
```
E1: EbeneNF(x+5*y-2*z+4=0)$
E2: [[2,5,-4],[1,-2,3],[5,0,1]]$
SchnittgeradeEE(E1,E2);
```

Ausgabe: $[[2, 0, 3], [13, -1, 4]]$

Interpretation: Gleichung der Schnittgeraden: $\overrightarrow{X} = \begin{pmatrix} 2 \\ 0 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 13 \\ -1 \\ 4 \end{pmatrix}$

7.3.21. Parallelebene zu Ebene durch Punkt

Für eine solche Parallelebene ist schnell eine Gleichung hingeschrieben. Man nimmt den gegebenen Punkt als Aufpunkt und übernimmt die beiden Richtungsvektoren der ursprünglichen Ebene.

```
ParallelebeneEP(e,p) := [p,e[2],e[3]]$
```

Beispiel: Gesucht ist die Parallelebene zur x-y-Ebene durch $P(0|0|1)$.

Eingabe:
```
E: [[0,0,0],[1,0,0],[0,1,0]]$
P: [0,0,1]$
ParallelebeneEP(E,P);
```

Ausgabe: $[[0, 0, 1], [1, 0, 0], [0, 1, 0]]$

Interpretation: $\overrightarrow{X} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

7.3.22. Fußpunkt des Lotes zu Ebene durch Punkt

Der Fußpunkt des Lotes zu einer gegebenen Ebene durch einen gegebenen Punkt, anders ausgedrückt die senkrechte Projektion eines Punkts auf eine Ebene, benötigt man relativ häufig.

```
FusspunktEP(e,p) := block(
  [n,par],
  n: NormalenvektorE(e),
  par: (n.(e[1]-p))/(n.n),
  p[1]+par*n
  )$
```

Erläuterung: Ist $\overrightarrow{n}$ ein Normalenvektor der Ebene E und $\overrightarrow{A}$ der Ortsvektor eines beliebigen Punkts von E, so ist $\overrightarrow{n} \circ (\overrightarrow{X} - \overrightarrow{A}) = 0$ eine Gleichung der Ebene. Das Lot von Punkt P auf die Ebene E lässt sich durch $\overrightarrow{X} = \overrightarrow{P} + \lambda \overrightarrow{n}$ beschreiben. Durch Einsetzen der Geradengleichung in die Ebenengleichung ergibt sich $\overrightarrow{n} \circ (\overrightarrow{P} + \lambda \overrightarrow{n} - \overrightarrow{A}) = 0$ und weiter $\lambda = \dfrac{\overrightarrow{n} \circ (\overrightarrow{A} - \overrightarrow{P})}{\overrightarrow{n} \circ \overrightarrow{n}}$, in Maxima-Schreibweise `(n.(e[1]-p)/(n.n)`. Setzt man diesen Parameterwert in die Geradengleichung ein, so erhält man den Ortsvektor des gesuchten Fußpunkts.

Beispiel: Vom Punkt $P(3|-4|7)$ wird ein Lot auf die durch $2x - y + 2z - 6 = 0$ definierte Ebene gefällt. Bestimme den Fußpunkt dieses Lotes.

Eingabe:
```
E: EbeneNF(2*x-y+2*z-6=0)$
P: [3,-4,7]$
FusspunktEP(E,P);
```
Ausgabe: $[-1, 5, -1]$

Interpretation: $F(-1|5|-1)$ ist der gesuchte Fußpunkt.

7.3.23. Lot zu Ebene durch Punkt

Eine Gleichung des Lotes zu einer gegebenen Ebene durch einen gegebenen Punkt lässt sich problemlos aufstellen:

```
LotEP(e,p) := [p,NormalenvektorE(e)]$
```

Beispiel: Zum vorigen Beispiel (Ebene $E: \ 2x - y + 2z - 6 = 0$, Punkt $P(3|-4|7)$) ist eine Gleichung des Lotes anzugeben.

Eingabe:
```
E: EbeneNF(2*x-y+2*z-6=0)$
P: [3,-4,7]$
LotEP(E,P);
```
Ausgabe: $[[3, -4, 7], [2, -1, 2]]$

Interpretation: $\overrightarrow{X} = \begin{pmatrix} 3 \\ -4 \\ 7 \end{pmatrix} + \alpha \begin{pmatrix} 2 \\ -1 \\ 2 \end{pmatrix}$

7.3.24. Spiegelpunkt bei Ebenenspiegelung

Mithilfe der Maxima-Funktion `FusspunktEP` lässt sich ohne Weiteres der Spiegelpunkt eines Punktes bezüglich einer Ebene ermitteln.

```
SpiegelpunktEP(e,p) := 2*FusspunktEP(e,p)-p$
```

Beispiel: Bestimme für das letzte Beispiel (Ebene $E: \ 2x - y + 2z - 6 = 0$, Punkt $P(3|-4|7)$) den Spiegelpunkt P'.

Eingabe: `E: EbeneNF(2*x-y+2*z-6=0)$`
`P: [3,-4,7]$`
`SpiegelpunktEP(E,P);`

Ausgabe: $[-5, 14, -9]$

Interpretation: $P'(-5|14|-9)$ ist der Spiegelpunkt von P bezüglich E.

7.3.25. Lotebene zu Gerade durch Punkt

Sind eine Gerade g und ein Punkt P gegeben, so lässt sich leicht eine Normalenform für die Lotebene zu g durch P finden, da der Richtungsvektor von g als Normalenvektor der Lotebene geeignet ist.

```
LotebeneGP(g,p) := EbeneNF(g[2].([x,y,z]-p)=0)$
```

Beispiel: Zur Geraden g mit der Gleichung $\overrightarrow{X} = \begin{pmatrix} 5 \\ -2 \\ 4 \end{pmatrix} + \lambda \begin{pmatrix} 8 \\ 0 \\ -3 \end{pmatrix}$ soll die Lotebene bestimmt werden, die durch den Punkt $P(2|-1|4)$ geht.

Eingabe: `g: [[5,-2,4],[8,0,-3]]$`
`P: [2,-1,4]$`
`E: LotebeneGP(g,P);`
`NormalenformE(E);`

Ausgabe: $[[\frac{1}{2}, 0, 0], [0, 1, 0], [3, 0, 8]]$
$8x - 3z - 4 = 0$

Interpretation: $\overrightarrow{X} = \begin{pmatrix} 0{,}5 \\ 0 \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 3 \\ 0 \\ 8 \end{pmatrix}; \quad 8x - 3z - 4 = 0$

7.4. Abstandsberechnungen

7.4.1. Abstand zweier Punkte

Der Abstand zweier Punkte ergibt sich als Betrag des Verbindungsvektors, das heißt als Betrag der Differenz der beiden Ortsvektoren.

```
AbstandPP(p1,p2) := Betrag(p2-p1)$
```

Beispiel: Berechne den Abstand der Punkte $A(5|-2|-1)$ und $B(3|1|0)$.

Eingabe: `A: [5,-2,-1]$`
`B: [3,1,0]$`
`AbstandPP(A,B);`

Ausgabe: $\sqrt{14}$

7.4.2. Abstand zwischen Punkt und Gerade

Die folgende Maxima-Funktion `AbstandPG` liefert den Abstand zwischen einem Punkt P (Ortsvektor $\overrightarrow{P}$) und einer Geraden g (Gleichung $\overrightarrow{X} = \overrightarrow{A} + \lambda\overrightarrow{u}$).

```
AbstandPG(p,g) := AbstandPG(p,g) := Betrag(p-FusspunktGP(g,p))$
```

Erläuterung: Mit dem schon definierten Maxima-Befehl `FusspunktGP` (siehe 7.2.14) wird der Fußpunkt des Lotes von P auf g bestimmt. Der gesuchte Abstand ist der Betrag des Verbindungsvektors dieses Fußpunkts mit P.

Beispiel: Wie groß ist der Abstand des Punktes $P(-5|2|-1)$ von der Geraden g, die durch $\overrightarrow{X} = \begin{pmatrix}-2\\0\\7\end{pmatrix} + \lambda\begin{pmatrix}2\\1\\-2\end{pmatrix}$ gegeben ist?

Eingabe:
```
P: [-5,2,-1]$
g: [[-2,0,7],[2,1,-2]]$
AbstandPG(P,g);
```

Ausgabe: $\sqrt{61}$

7.4.3. Abstand zwischen Punkt und Ebene

```
AbstandPE(p,e) := block(
  [n],
  n: NormalenvektorE(e),
  abs(n.(p-e[1])/Betrag(n))
  )$
```

Erläuterung: Den Abstand eines Punktes P von einer Ebene E erhält man, indem man in der hesseschen Normalenform der Ebenengleichung ($\frac{1}{|\overrightarrow{n}|}\overrightarrow{n}\circ(\overrightarrow{X}-\overrightarrow{A}) = 0$) den Ortsvektor $\overrightarrow{P}$ für $\overrightarrow{X}$ einsetzt. Der Rechenausdruck für den Abstand (ohne Vorzeichen) lautet also $\left|\frac{\overrightarrow{n}\circ(\overrightarrow{P}-\overrightarrow{A})}{|\overrightarrow{n}|}\right|$, in Maxima-Schreibweise `abs(n.(p-e[1])/Betrag(n))`.

Beispiel: Berechne den Abstand des Punktes $P(0|-9|1)$ von der Ebene E mit der Gleichung $2x - 2y - z - 12 = 0$.

Eingabe:
```
P: [0,-9,1]$
E: EbeneNF(2*x-2*y-z-12=0)$
AbstandPE(P,E);
```

Ausgabe: $\frac{5}{3}$

7.4.4. Abstand zweier Geraden

```
AbstandGG(g1,g2) := block([e],
  if parallelGG(g1,g2) then return(AbstandPG(g1[1],g2)),
  e: [g1[1],g1[2],g2[2]],
  AbstandPE(g2[1],e)
  )$
```

Erläuterung: Sind die gegebenen Geraden parallel, so berechnet man mit der Maxima-Funktion `AbstandPG` den Abstand des Aufpunkts der einen Geraden (`g1[1]`) von der anderen Geraden (`g2`). Sind die Geraden nicht parallel, so wird – unter Verwendung des Aufpunkts von `g1` und der beiden Richtungsvektoren – eine Hilfsebene `e` definiert. Der gesuchte Abstand lässt sich dann mithilfe der zuvor definierten Maxima-Funktion `AbstandPE` ermitteln.

Falls sich `g1` und `g2` schneiden, handelt es sich bei der Hilfsebene `e` um die Ebene, in der beide Geraden liegen. Unter dieser Voraussetzung liefert die Berechnung den Abstand 0. Falls die gegebenen Geraden windschief zueinander sind, enthält die Hilfsebene die erste Gerade und ist parallel zur zweiten, sodass ein Abstand ungleich 0 herauskommt.

Beispiel: Wie groß ist der Abstand der parallelen Geraden mit den Gleichungen $g: \overrightarrow{X} = \begin{pmatrix} 2 \\ -7 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 5 \\ 1 \\ -2 \end{pmatrix}$ und $h: \overrightarrow{X} = \begin{pmatrix} -2 \\ 4 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} 5 \\ 1 \\ -2 \end{pmatrix}$?

Eingabe:

```
g: [[2,-7,3],[5,1,-2]]$
h: [[-2,4,1],[5,1,-2]]$
AbstandGG(g,h);
```

Ausgabe: $\frac{29}{\sqrt{6}}$

Beispiel: Durch $g: \overrightarrow{X} = \begin{pmatrix} 1 \\ 4 \\ -3 \end{pmatrix} + \alpha \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$ und $h: \overrightarrow{X} = \begin{pmatrix} -5 \\ -2 \\ 2 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \\ 4 \end{pmatrix}$ sind zwei windschiefe Geraden g und h gegeben. Berechne den Abstand der beiden Geraden.

Eingabe:

```
g: [[1,4,-3],[1,1,-3]]$
h: [[-5,-2,2],[0,1,4]]$
AbstandGG(g,h);
```

Ausgabe: $\frac{13}{\sqrt{66}}$

7.4.5. Abstand zwischen Gerade und Ebene

```
AbstandGE(g,e) := block(
  if schneidendGE(g,e) then return(0),
  AbstandPE(g[1],e)
  )$
```

Erläuterung: Falls die Gerade `g` die Ebene `e` schneidet, ist der Abstand gleich 0. Andernfalls, wenn die Gerade zur Ebene parallel ist, verwendet man die schon definierte Maxima-Funktion `AbstandPE` und bestimmt damit den Abstand des Aufpunkts von `g` von der Ebene `e`.

Beispiel: Durch $g:\ \overrightarrow{X} = \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} -4 \\ 6 \\ 3 \end{pmatrix}$ und $E:\ 3x - 4y + 12z - 6 = 0$ sind eine Gerade g und eine Ebene E definiert. Ermittle den Abstand.

Eingabe:

```
g: [[5,-1,1],[-4,6,3]]$
E: EbeneNF(3*x-4*y+12*z-6=0)$
AbstandGE(g,E);
```

Ausgabe: $\frac{25}{13}$

7.4.6. Abstand zweier Ebenen

```
AbstandEE(e1,e2) := block(
  if schneidendEE(e1,e2) then return(0),
  AbstandPE(e1[1],e2)
  )$
```

Erläuterung: Bei zwei sich schneidenden Ebenen ist der gesuchte Abstand gleich 0. Sind `e1` und `e2` dagegen parallel, so kann man wieder den Befehl `AbstandPE` verwenden und damit den Abstand berechnen, den irgendein Punkt von `e1` (zum Beispiel der Aufpunkt) von der Ebene `e2` hat.

Beispiel: Berechne den Abstand der parallelen Ebenen mit den Gleichungen $-2x + y + 2z - 9 = 0$ und $-2x + y + 2z + 5 = 0$.

Eingabe:

```
E1: EbeneNF(-2*x+y+2*z-9=0)$
E2: EbeneNF(-2*x+y+2*z+5=0)$
AbstandEE(E1,E2);
```

Ausgabe: $\frac{14}{3}$

7.5. Winkelberechnungen

Die schon definierte Maxima-Funktion `Winkel` (Winkel zwischen zwei Vektoren, siehe 6.10) wird im Folgenden mehrfach zur Schnittwinkelberechnung verwendet. Es wird jeweils eine Variante für Winkel im Bogenmaß und für Winkel im Gradmaß definiert.

7.5.1. Schnittwinkel zweier Geraden

Der Schnittwinkel zweier Geraden lässt sich mithilfe der Richtungsvektoren bestimmen.

```
WinkelGG(g1,g2) := block(
  [w],
  if gleichGG(g1,g2) then return(0),
  if not schneidendGG(g1,g2) then
    return("Geraden schneiden sich nicht!"),
  w: Winkel(g1[2],g2[2]),
  min(w,float(%pi)-w)
  )$
```

Erläuterung: Bei identischen Geraden ist der Rückgabewert 0, bei Geraden ohne gemeinsamen Punkt erfolgt eine Fehlermeldung. Falls die Geraden sich schneiden, wird die Maxima-Funktion `Winkel` auf die beiden Richtungsvektoren losgelassen. Falls das Ergebnis w (Winkel im Bogenmaß) größer als $\frac{\pi}{2}$ ist, wenn also die beiden Richtungsvektoren „auseinandergehen"), wird es durch $\pi - w$ ersetzt, da man üblicherweise den kleinsten Winkel zwischen den gegebenen Geraden angibt.

```
WinkelGG_Grad(g1,g2) := block(
  [w],
  w: WinkelGG(g1,g2),
  if not floatnump(w) then return(w),
  float(w*180/%pi)
  )$
```

Beispiel: Unter welchem Winkel (Gradmaß) schneiden sich die Geraden mit den Gleichungen $g_1: \overrightarrow{X} = \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} + \lambda \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}$ und $g_2: \overrightarrow{X} = \begin{pmatrix} 9 \\ -1 \\ -7 \end{pmatrix} + \mu \begin{pmatrix} -2 \\ 0 \\ 3 \end{pmatrix}$?

Eingabe:

```
g1: [[1,3,4],[-1,2,1]]$
g2: [[9,-1,-7],[-2,0,3]]$
WinkelGG_Grad(g1,g2);
```

Ausgabe: 55.51861062801841

7.5.2. Schnittwinkel zwischen Gerade und Ebene

Zur Ermittlung eines solchen Schnittwinkels verwendet man den Richtungsvektor der Geradengleichung und einen Normalenvektor der Ebene. Die Berechnung entspricht weitgehend der vorigen, erfordert aber am Ende einen zusätzlichen Schritt, nämlich die Ergänzung des berechneten Winkels auf $\frac{\pi}{2}$ (entsprechend 90°).

```
WinkelGE(g,e) := block(
  [n,w],
  if enthaltenGE(g,e) then return(0),
  if parallelGE(g,e) then return("Gerade schneidet Ebene nicht!"),
  n: NormalenvektorE(e),
  w: Winkel(g[2],n),
  w: min(w,float(%pi)-w),
  float(%pi/2-w)
  )$

WinkelGE_Grad(g,e) := block(
  [w],
  w: WinkelGE(g,e),
  if not floatnump(w) then return(w),
  float(w*180/%pi)
  )$
```

Beispiel: Berechne den Schnittwinkel (Gradmaß) zwischen der durch $\overrightarrow{X} = \begin{pmatrix}1\\2\\4\end{pmatrix} + \lambda \begin{pmatrix}2\\0\\-1\end{pmatrix}$ festgelegten Geraden g und der Ebene E mit der Parametergleichung $\overrightarrow{X} = \begin{pmatrix}2\\5\\-3\end{pmatrix} + \sigma \begin{pmatrix}1\\1\\0\end{pmatrix} + \tau \begin{pmatrix}4\\1\\-6\end{pmatrix}$.

Eingabe:

```
g: [[1,2,4],[2,0,-1]]$
E: [[2,5,-3],[1,1,0],[4,1,-6]]$
WinkelGE_Grad(g,E);
```

Ausgabe: 26.56505117707798

7.5.3. Schnittwinkel zweier Ebenen

Den Schnittwinkel zweier Ebenen erhält man mithilfe von zugehörigen Normalenvektoren.

```
WinkelEE(e1,e2) := block(
  [n1,n2,w],
  if gleichEE(e1,e2) then return(0),
  if parallelEE(e1,e2) then return("Ebenen schneiden sich nicht!"),
  n1: NormalenvektorE(e1),
  n2: NormalenvektorE(e2),
  w: Winkel(n1,n2),
  min(w,float(%pi)-w)
  )$

WinkelEE_Grad(e1,e2) := block(
  [w],
  w: WinkelEE(e1,e2),
  if not floatnump(w) then return(w),
  float(w*180/%pi)
  )$
```

Beispiel: Unter welchem Winkel (Gradmaß) schneiden sich die Ebenen mit den Parametergleichungen $\overrightarrow{X} = \begin{pmatrix} 2 \\ 5 \\ 4 \end{pmatrix} + \alpha \begin{pmatrix} -1 \\ 2 \\ 4 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}$ und $\overrightarrow{X} = \begin{pmatrix} -1 \\ 4 \\ 0 \end{pmatrix} + \gamma \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} + \delta \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix}$?

Eingabe:

```
E1: [[2,5,4],[-1,2,4],[1,0,3]]$
E2: [[-1,4,0],[1,1,2],[-3,0,1]]$
WinkelEE_Grad(E1,E2);
```

Ausgabe: 47.45260500767188

7.6. Sonstiges

7.6.1. Flächeninhalt eines Dreiecks

Der Flächeninhalt eines Dreiecks lässt sich leicht mit dem Kreuzprodukt berechnen.

```
FlaecheDreieckPPP(p1,p2,p3) := block(
  [kp],
  kp: Kreuzprodukt(p2-p1,p3-p1),
  Betrag(kp)/2
  )$
```

Beispiel: Welchen Flächeninhalt hat das gleichseitige Dreieck mit den Ecken $A(1|0|0)$, $B(0|1|0)$ und $C(0|0|1)$?

Eingabe:

```
A: [1,0,0]$
B: [0,1,0]$
C: [0,0,1]$
FlaecheDreieckPPP(A,B,C);
```

Ausgabe: $\frac{\sqrt{3}}{2}$

7.6.2. Volumen eines Tetraeders

Kaum komplizierter ist die Bestimmung des Tetraedervolumens mithilfe des sogenannten Spatprodukts.

```
VolumenTetraederPPPP(p1,p2,p3,p4) := block(
  [kp,sp],
  kp: Kreuzprodukt(p2-p1,p3-p1),
  sp: kp.(p4-p1),
  abs(sp)/6
  )$
```

Beispiel: Berechne das Volumen des Tetraeders mit den Ecken $A(1|-5|0)$, $B(-3|2|0)$, $C(-5|-8|0)$ und $D(1|1|4)$.

Eingabe:

```
A: [1,-5,0]$
B: [-3,2,0]$
C: [-5,-8,0]$
D: [1,1,4]$
VolumenTetraederPPPP(A,B,C,D);
```

Ausgabe: 36

A. Maxima-Paket anageo3d (Vektorrechnung und Analytische Geometrie)

A.1. Hinweise zu Installation und Verwendung

- Laden Sie die Datei `http://www.walter-fendt.de/download/anageo3d.wxmx` herunter und speichern Sie diese Datei in einem beliebigen Verzeichnis.
- Starten Sie das Programm wxMaxima und öffnen Sie die zuvor gespeicherte Datei.
- Führen Sie im Menü „Zellen“ den Befehl „Alle Zellen neu auswerten“ aus. Dadurch wird das Zusatzpaket `anageo3d` im Verzeichnis wxMaxima installiert.
- Wenn Sie das Zusatzpaket verwenden wollen, sollte Ihre Maxima-Sitzung mit den Befehlen `kill(all);` und `load(anageo3d);` beginnen. Danach können Sie die Befehle verwenden, die in der folgenden Liste aufgeführt sind.

A.2. Liste der Befehle

`AbstandEE(e1,e2);`	Abstand zweier Ebenen
`AbstandGE(g,e);`	Abstand zwischen Gerade und Ebene
`AbstandGG(g1,g2);`	Abstand zweier Geraden
`AbstandPE(p,e);`	Abstand zwischen Punkt und Ebene
`AbstandPG(p,g);`	Abstand zwischen Punkt und Gerade
`AbstandPP(p1,p2);`	Abstand zweier Punkte
`AusgabeE(e,par1,par2);`	Ausgabe einer Parametergleichung einer Ebene
`AusgabeG(g,par);`	Ausgabe einer Parametergleichung einer Geraden
`Betrag(v);`	Betrag eines Vektors
`EbeneNF(nf);`	Ebene mit gegebener Normalenform
`EbenePPP(p1,p2,p3);`	Ebene durch drei Punkte
`enthaltenGE(g,e);`	Liegt Gerade in Ebene?
`enthaltenPE(p,e);`	Liegt Punkt in Ebene?
`enthaltenPG(p,g);`	Liegt Punkt auf Gerade?
`FlaecheDreieckPPP(p1,p2,p3);`	Flächeninhalt eines Dreiecks
`FusspunktEP(e,p);`	Fußpunkt des Lotes zu Ebene durch Punkt
`FusspunktGP(g,p);`	Fußpunkt des Lotes zu Gerade durch Punkt
`GeradePP(p1,p2);`	Gerade durch zwei Punkte
`gleichEE(e1,e2);`	Sind Ebenen gleich?
`gleichGG(g1,g2);`	Sind Geraden gleich?
`Gleichungssystem(vg);`	Umwandlung einer Vektorgleichung in ein Gleichungssystem
`Kreuzprodukt(v1,v2);`	Kreuzprodukt zweier Vektoren

`LageEE(e1,e2);`	Lagebeziehung zweier Ebenen
`LageGE(g,e);`	Lagebeziehung von Gerade und Ebene
`LageGG(g1,g2);`	Lagebeziehung zweier Geraden
`linearAbhaengig2(v1,v2);`	Sind zwei Vektoren linear abhängig?
`linearAbhaengig3(v1,v2,v3);`	Sind drei Vektoren linear abhängig?
`linearUnabhaengig2(v1,v2);`	Sind zwei Vektoren linear unabhängig?
`linearUnabhaengig3(v1,v2,v3);`	Sind drei Vektoren linear unabhängig?
`LotEP(e,p);`	Lot zu Ebene durch Punkt
`LotebeneGP(g,p);`	Lotebene zu Gerade durch Punkt
`LotGP(g,p);`	Lot zu Gerade durch Punkt
`NormalenformE(e);`	Normalenform einer Ebene
`NormalenvektorE(e);`	Normalenvektor einer Ebene
`Optimierung(v);`	Optimierung eines Richtungsvektors
`OptimierungE(e);`	Optimierung einer Parametergleichung einer Ebene
`OptimierungG(g);`	Optimierung einer Parametergleichung einer Geraden
`ParallelebeneEP(e,p);`	Parallelebene zu Ebene durch Punkt
`ParalleleGP(g,p);`	Parallele zu Gerade durch Punkt
`parallelEE(e1,e2);`	Sind Ebenen parallel?
`parallelGE(g,e);`	Ist Gerade parallel zu Ebene?
`parallelGG(g1,g2);`	Sind Geraden parallel?
`schneidendEE(e1,e2);`	Schneiden sich Ebenen?
`schneidendGE(g,e);`	Schneidet Gerade Ebene?
`schneidendGG(g1,g2);`	Schneiden sich Geraden?
`SchnittgeradeEE(e1,e2);`	Schnittgerade zweier Ebenen
`SchnittpunktGE(g,e);`	Schnittpunkt von Gerade und Ebene
`SchnittpunktGG(g1,g2);`	Schnittpunkt zweier Geraden
`senkrechtEE(e1,e2);`	Schneiden sich Ebenen senkrecht?
`senkrechtGE(g,e);`	Schneidet Gerade Ebene senkrecht?
`senkrechtGG(g1,g2);`	Schneiden sich Geraden senkrecht?
`SpiegelpunktEP(e,p);`	Spiegelpunkt bei Ebenenspiegelung
`SpiegelpunktGP(a,p);`	Spiegelpunkt bei Achsenspiegelung
`SpiegelpunktPP(z,p);`	Spiegelpunkt bei Punktspiegelung
`Vektor(v);`	Spaltenvektor
`VolumenTetraederPPPP(p1,p2,p3,p4);`	Volumen eines Tetraeders
`windschiefGG(g1,g2);`	Sind Geraden windschief?

`Winkel(v1,v2);`	Winkel zwischen zwei Vektoren (Bogenmaß)
`WinkelEE(e1,e2);`	Schnittwinkel zweier Ebenen (Bogenmaß)
`WinkelEE_Grad(e1,e2);`	Schnittwinkel zweier Ebenen (Gradmaß)
`WinkelGE(g,e);`	Schnittwinkel von Gerade und Ebene (Bogenmaß)
`WinkelGE_Grad(g,e);`	Schnittwinkel von Gerade und Ebene (Gradmaß)
`WinkelGG(g1,g2);`	Schnittwinkel zweier Geraden (Bogenmaß)
`WinkelGG_Grad(g1,g2);`	Schnittwinkel zweier Geraden (Gradmaß)