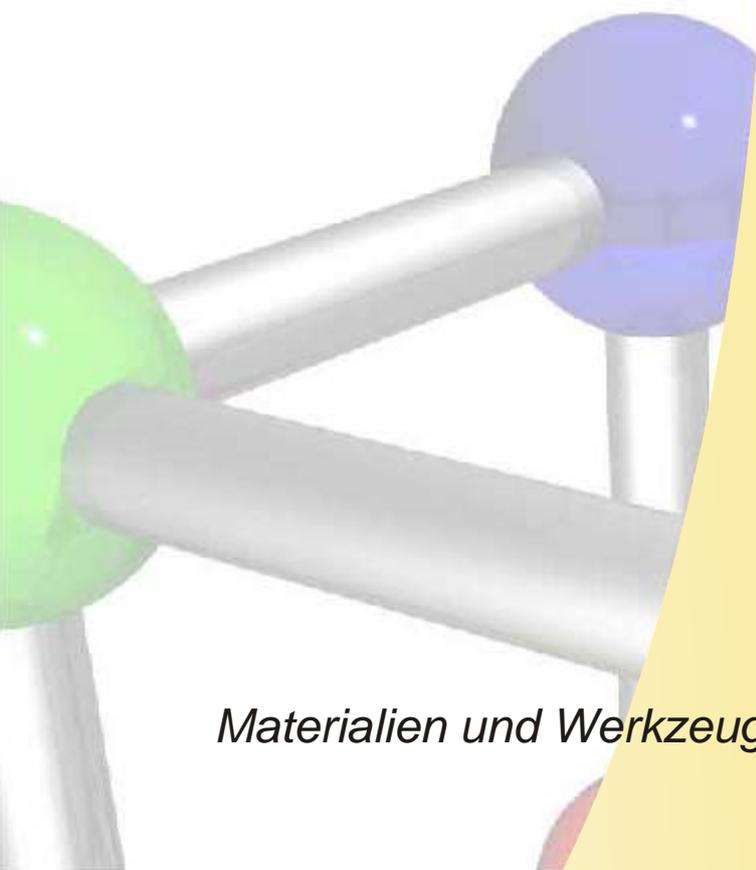


*Mathematik 1 x anders*  
*Band 10*

*Analytische Geometrie*  
*mit **Mu**PAD*

*Kai Gehrs*

*Materialien und Werkzeuge für computerunterstütztes Lernen*





Kai Gehrs

# **Analytische Geometrie mit MuPAD**

Mathematik 1 × anders: Materialien und Werkzeuge für  
computerunterstütztes Lernen

SciFace Software

Dipl.-Math. Kai Gehrs  
Universität Paderborn  
Fakultät EIM–M (AutoMATH Institut)  
MuPAD Schule-Team  
Pohlweg 98  
33100 Paderborn

`schule@mupad.de`

**Analytische Geometrie mit MuPAD**, Mathematik 1×anders: Materialien und Werkzeuge für computerunterstütztes Lernen, Band 10. Paderborn: SciFace Software GmbH & Co. KG, September 2004.

Erste Auflage: September 2004

SciFace Software GmbH & Co. KG Paderborn

Alle Rechte vorbehalten. Diese Arbeit ist urheberrechtlich geschützt und darf nicht in andere Sprachen übersetzt oder publiziert werden ohne ausdrückliche Erlaubnis von SciFace Software GmbH & Co. KG.

© 2004 SciFace Software GmbH & Co. KG, Paderborn

Umschlaggestaltung: SciFace Software GmbH & Co. KG, Paderborn

Stand: 12. Oktober 2004

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>2</b>
<b>Über dieses Buch</b>	<b>3</b>
<b>1 Motivation</b>	<b>5</b>
1.1 Analytische Geometrie: Was ist das?	5
1.2 Motivation: Von der Analysis zur Analytischen Geometrie	6
<b>2 Vektoren und Vektorräume: Rechnen mit Pfeilen</b>	<b>12</b>
2.1 Technische-Vorbemerkungen zum Umgang mit MuPAD	12
2.2 Vektoren in der Ebene: Der $\mathbb{R}^2$	12
2.3 Vektoren im Raum: Der $\mathbb{R}^3$	23
2.4 Länge von Vektoren	30
2.4.1 Abstand zweier Punkte	35
2.5 Standardskalarprodukt und Winkel zwischen Vektoren	38
2.6 Kreuzprodukt von Vektoren	42
2.7 Erste elementare Anwendungen in der Geometrie	47
<b>3 Geraden und Ebenen</b>	<b>51</b>
3.1 Beschreibung von Geraden	51
3.1.1 Parameterform von Geraden im $\mathbb{R}^2$ und $\mathbb{R}^3$	51
3.1.2 Koordinatenform von Geraden im $\mathbb{R}^2$	54
3.2 Lagebeziehung von Geraden und Punkten im $\mathbb{R}^2$ und $\mathbb{R}^3$	58
3.3 Lagebeziehung zweier Geraden	61
3.3.1 Parallele und identische Geraden	63
3.3.2 Windschiefe und sich schneidende Geraden	65
3.3.3 Zusammenfassung: Lagebeziehung zweier Geraden im Raum	69
3.4 Beschreibung von Ebenen	70
3.4.1 Parameterform von Ebenen	71
3.4.2 Koordinatenform von Ebenen	76
3.5 Lagebeziehung von Ebenen und Punkten	79
3.6 Lagebeziehung von Gerade und Ebene	82
3.6.1 Ebene in Parameterform	83
3.6.2 Ebene in Koordinatenform	86
3.7 Lagebeziehung zweier Ebenen	89
3.7.1 Beide Ebenen in Parameterform	89

3.7.2	Eine Ebene in Koordinatenform, eine Ebene in Parameterform . . . . .	94
3.7.3	Beide Ebenen in Koordinatenform . . . . .	96
3.8	Koordinatenachsen und Koordinatenebenen . . . . .	98
3.9	Schnittwinkelberechnung . . . . .	100
3.9.1	Schnittwinkel zweier Geraden . . . . .	100
3.9.2	Schnittwinkel von Gerade und Ebene . . . . .	102
3.9.3	Schnittwinkel zweier Ebenen . . . . .	104
3.10	Abstandsprobleme . . . . .	106
3.10.1	Abstand eines Punktes von einer Geraden im $\mathbb{R}^3$ . . . . .	106
3.10.2	Abstand zweier paralleler Geraden im $\mathbb{R}^3$ . . . . .	112
3.10.3	Abstand windschiefer Geraden im $\mathbb{R}^3$ . . . . .	114
3.10.4	Abstand eines Punktes von einer Ebene im $\mathbb{R}^3$ . . . . .	118
3.10.5	Abstand einer Gerade von einer Ebene im $\mathbb{R}^3$ . . . . .	122
3.10.6	Noch einmal Abstand zweier windschiefer Geraden . . . . .	125
<b>4</b>	<b>Kreise und Kugeln</b> . . . . .	<b>130</b>
4.1	Beschreibung von Kreisen im $\mathbb{R}^2$ . . . . .	130
4.2	Lagebeziehung von Punkten und Kreisen im $\mathbb{R}^2$ . . . . .	137
4.3	Lagebeziehung von Geraden und Kreisen im $\mathbb{R}^2$ . . . . .	139
4.3.1	Geraden in Koordinatenform . . . . .	140
4.3.2	Geraden in Parameterform . . . . .	142
4.4	Beschreibung von Kugeln . . . . .	147
4.5	Lagebeziehung von Punkten und Kugeln . . . . .	151
4.6	Lagebeziehung von Geraden und Kugeln . . . . .	153
4.7	Lagebeziehung von Ebenen und Kugeln . . . . .	159
4.7.1	Tangentialebenen . . . . .	167
<b>5</b>	<b>Aufgaben aus Abiturprüfungen</b> . . . . .	<b>170</b>
5.1	Grundkursniveau . . . . .	170
5.2	Leistungskursniveau . . . . .	182
<b>6</b>	<b>Bemerkungen zur Kompatibilität mit verschiedenen MuPAD-Versionen</b> . . . . .	<b>193</b>
<b>7</b>	<b>Interaktive Arbeitsblätter zu diesem Buch</b> . . . . .	<b>196</b>
7.1	Interaktive Arbeitsblätter zu thematischen Abschnitten . . . . .	196
7.2	Lösungen zu den Übungsaufgaben . . . . .	197
	<b>Literaturverzeichnis</b> . . . . .	<b>200</b>

## Vorwort

Die Entwicklung von MuPAD begann 1990 an der Universität Paderborn mit einem Forschungsprojekt zur Lösung spezieller Problemstellungen im Bereich der Dynamischen Systeme. MuPAD wurde jedoch sehr bald zu einem universellen Werkzeug zum symbolischen und exakten sowie numerischen Rechnen. Darüber hinaus können zwei- und dreidimensionale mathematische Sachverhalte in hoher Darstellungsqualität visualisiert und interaktiv manipuliert werden. Die Entwicklung von MuPAD wurde bereits 1993 von der Forschungsgemeinschaft mit dem Deutsch-Österreichischen Hochschul-Software Preis und 1994 mit dem European Academic Software Award honoriert.

Im Februar 1997 wurde als Teilausgliederung aus der Universität Paderborn das Unternehmen SciFace Software GmbH & Co. KG gegründet, um MuPAD in enger Kooperation mit der MuPAD-Forschungsgruppe weiter zu entwickeln und den zunehmenden Anforderungen der Benutzer u.a. hinsichtlich vielfältiger und mehrsprachiger Dokumentationen zum System und modernen Benutzerschnittstellen gerecht zu werden.

Forschungen aus der Universität werden von SciFace Software aufgegriffen, zu marktreifen Entwicklungen ausgebaut und in das System MuPAD integriert. Aufgrund dieser erfolgreichen und engen Zusammenarbeit von SciFace Software mit der Universität Paderborn wurde das Unternehmen 1998 mit dem Förderpreis des Technologie Forum Paderborn e.V. für hervorragende Leistungen auf dem Gebiet der Zusammenarbeit zwischen Wirtschaft und Wissenschaft ausgezeichnet. MuPAD wird u.a. zur Forschung und Lehre an Universitäten eingesetzt und findet mittlerweile auch verstärkt Einzug in den Mathematikunterricht der gymnasialen Oberstufe. MuPAD trägt dabei ergänzend und unterstützend zur Lehre von Mathematik bei.

Das starke Interesse und die vielfältigen Tätigkeiten von SciFace Software in dem Bereich der Lehre von Mathematik zeigt sich auch an der engen Zusammenarbeit mit bedeutenden deutschen Verlagen und Herstellern von Unterrichtsoftware mit dem Ziel, über gemeinsame Anstrengungen heraus Lösungen zu schaffen, den Schülern und Studenten u.a. einen interaktiven, explorativen Zugang zu mathematischen Sachverhalten und ein Web-unterstütztes Lernen zu ermöglichen.

Schreiben Sie uns, wenn Sie Fragen oder Anregungen zum Thema "MuPAD in der Lehre" haben. Nehmen Sie Teil an der Entwicklung einer modernen Mathematiksoftware und senden Sie uns Ihre Vorschläge, Kritiken und Fragen an [schule@mupad.de](mailto:schule@mupad.de).

Paderborn, September 2004  
Dr. Andreas Sorgatz,  
SciFace Software

## Über dieses Buch

Dieses Buch richtet sich sowohl an Lehrerinnen und Lehrer als auch ausdrücklich an Schülerinnen und Schüler sowie Studentinnen und Studenten. Es ist als Begleitbuch zu den üblichen Standardlehrbüchern gedacht, die im Unterricht der Sekundarstufe II zur Vermittlung der Inhalte der Linearen Algebra und Analytischen Geometrie verwendet werden.

Die mathematischen Inhalte der Linearen Algebra und Analytischen Geometrie werden hier weniger genau und detailliert erklärt, als man es aus konventionellen Lehrbüchern gewohnt ist. Das vorliegende Buch ist mit Absicht nicht so angelegt worden, dass auf konventionelle Lehrmaterialien verzichtet werden kann. Anregungen bei der Wahl und Formulierung insbesondere der Übungsaufgaben haben die Bücher [2], [9], [10] und [11] geliefert. Die gewählte Reihenfolge, in der die entsprechenden mathematischen Themengebiete sowie ihre geometrischen Anwendungen vorgestellt werden, ist angelehnt an [2] und [9]. Die Abituraufgaben in Kapitel 5 sind [2] entnommen. “Analytische Geometrie mit MuPAD” soll den üblichen Schulunterricht vor allem durch zwei Aspekte anreichern: Beweise und Herleitungen von Formeln aus dem Bereich der Vektorrechnung, die durchaus dem Verständnis der mathematischen Theorie zuträglich sind, aber aus Gründen des Zeitmangels oder des bloßen Rechenaufwands im Schulunterricht (zum Teil) entfallen (müssen), werden mit dem Computeralgebra-System MuPAD durchgeführt.

Es ist dabei durchaus klar, dass ein Beweis, den man mit einem Computeralgebra-System durchführt, einen anderen Lerneffekt hat, als wenn man ihn mit Papier und Bleistift auf traditionelle Weise führt. Da aber viele Beweise wie etwa das Nachrechnen, dass das Kreuzprodukt zweier allgemeiner Vektoren im  $\mathbb{R}^3$  immer senkrecht auf denselben steht, Schülerinnen und Schüler einen recht erheblichen Rechenaufwand abfordern (und damit verhältnismäßig viel wertvolle Unterrichtszeit verloren geht), wird in der unterrichtlichen Praxis verständlicherweise häufig ganz darauf verzichtet. MuPAD wird in diesem Kontext von uns als “Rechenknecht” verwandt frei nach dem Motto: “Besser man führt den Beweis mit einem Computeralgebra-System als gar nicht!”.

Der zweite und noch wichtigere Aspekt ist die Tatsache, dass Computeralgebra-Systeme im allgemeinen und MuPAD im besonderen hervorragende Möglichkeiten zur Visualisierung mathematischer Sachverhalte bieten.

Die Erfahrung zeigt, dass insbesondere in der Analytischen Geometrie das intuitive

Verständnis sowie das räumliche Vorstellungsvermögen von Schülerinnen und Schülern signifikant erhöht wird, wenn der Unterricht durch passende Visualisierungen angereichert wird. Da die Visualisierung von dreidimensionalen Objekten an der Tafel oder auf einem Blatt Papier häufig recht schwierig und zeitaufwändig ist, bietet sich hier der Einsatz von MuPAD in besonderer Weise an.

Viele mathematische Verfahrensweisen im Kontext der Analytischen Geometrie können mit Hilfe von MuPAD in einer einzelnen Grafik oder in einer Animation visualisiert werden. Frei nach dem Motto: "Ein Bild sagt mehr als tausend Worte!" wurde im Rahmen dieses Buches versucht, einige Grafiken und Animationen zu erzeugen, die im Unterricht direkt eingesetzt werden können. Beispiele für solche Visualisierungen sind im einfachsten Fall etwa Darstellungen von Paaren von Geraden und/oder Ebenen im Raum zur Veranschaulichung diverser Lagebeziehungen.

Solche Grafiken lassen sich in MuPAD stets mit einem Doppelklick "aktivieren" und können so vom Betrachter interaktiv erforscht werden. Das Grafik-Programm VCam (in MuPAD integriert) gestattet dabei u.a. das Drehen von und Zoomen in dreidimensionalen Grafiken, so dass Visualisierungen stets nicht nur in statischer Form ausgedruckt, sondern vielmehr am Computer interaktiv erforscht werden können. Für die PDF-Version dieses Buches gilt: Um diese Interaktivität zu unterstützen, finden sich bei jeder mit MuPAD erzeugten Grafik die zwei Symbole  und . Mit einem Mausklick auf das Symbol  kann die in der PDF-Version dieses Buches eingebettete Grafik zusätzlich mit einem unter MS Windows verfügbaren Grafik-Programm in Form einer PNG-Datei geöffnet werden. Mit einem Mausklick auf das Symbol  lässt sich die zugehörige Grafik in dem Grafik-Programm VCam öffnen und dort – wie oben angedeutet – interaktiv erforschen. Hier lassen sich insbesondere auch Animationen abspielen. Bei Animationen, die in diesem Buch immer durch Filmstreifen gekennzeichnet sind, findet sich darüber hinaus zusätzlich das Symbol . Ein Mausklick darauf öffnet die Animation in Form einer AVI-Datei, die dann sich dann automatisch z.B. mit dem MS Media Player (standardmäßig unter den aktuelleren Versionen von MS Windows verfügbar) abspielen. Zu den Abschnitten dieses Buches wurden darüber hinaus MuPAD-Arbeitsblätter erstellt, die in der interaktiven Fassung per Mausklick verfügbar sind und dann direkt in MuPAD bearbeitet werden können. Eine Übersicht dieser Arbeitsblätter findet sich am Ende des Buchs auf Seite **197**.

Die Arbeitsblätter stehen außerdem auf dem Web-Portal des MuPAD-Schule Teams unter der Adresse <http://www.mupad.de/schule+studium/material/> zum kostenlosen Download bereit. Die Erklärungen innerhalb der Arbeitsblätter sind jedoch sehr viel knapper gehalten als es in diesem Buch der Fall ist.

Kai Gehrs  
Paderborn, September 2004

# 1 Motivation

## 1.1 Analytische Geometrie: Was ist das?

Als Erfinder der Analytischen Geometrie wird gemeinhin der Philosoph und Mathematiker René Descartes (geboren 1596 in La Haye und gestorben 1650 in Stockholm) genannt (“Cogito ergo sum – Ich denke, also bin ich”).

In der Analytischen Geometrie werden wir uns hier – im Gegensatz zur Analysis, wo man in der Schule fast ausschließlich Funktionen in der Ebene (d.h. im Zweidimensionalen) untersucht – vornehmlich mit mathematischen Objekten im dreidimensionalen Raum beschäftigen. Die Objekte des Interesses sind dabei weniger kompliziert als etwa die Funktionen, die gemeinhin im Analysisunterricht untersucht werden (gebrochenrationale Funktionen,  $e$ -Funktionen etc.). Es handelt sich in erster Linie um Punkte, Geraden und Ebenen im Raum.

Wir wollen uns dann mit Fragestellungen beschäftigen wie etwa:

- Welche möglichen Lagebeziehungen gibt es zwischen zwei Geraden im Raum?
- Wie sieht der Schnitt zweier Ebenen aus?
- Wie weit sind zwei parallele Geraden im Raum voneinander entfernt?

und vieles mehr. Um diese Fragen, die offensichtlich rein geometrischer Natur sind, beantworten zu können, bedarf es eines geeigneten mathematischen Modells, um die Objekte des Interesses (Punkte im Raum, Geraden im Raum, Ebenen im Raum) mathematisch korrekt beschreiben und mit ihnen “rechnen” zu können. Denn ganz im Gegenteil zur Euklidischen Geometrie, die man in der Sekundarstufe I an Schulen betreibt und in der man Problem- oder Fragestellungen vornehmlich dadurch löst, dass man eine Konstruktion mit Bleistift, Lineal und Zirkel anfertigt, wollen wir geometrische Behauptungen “nachrechnen” können.

An diesem Punkt kommt die Vektorrechnung ins Spiel, die uns mit ihren Konzepten das nötige mathematische Modell liefert, um geometrische Probleme und Fragestellung mit Hilfe mathematisch korrekter Berechnungen lösen bzw. beantworten zu können. Wichtig ist, dass wir immer im Hinterkopf behalten, dass für uns die Vektorrechnung, die einen Teil des umfassenden Gebiets der Linearen Algebra ausmacht, lediglich ein Hilfsmittel ist, um die geometrischen Fragestellungen aus der Analytischen Geometrie zu beantworten (in sich ist die Lineare Algebra ein geschlossenes Gebiet, das eigene

Aufmerksamkeit verdient und mit dem man sich viel viel länger und intensiver auseinandersetzen kann, als wir es hier tun wollen). Vergisst man dieses, so mag ein Teil der Vektorrechnung, die wir uns im zweiten Kapitel aneignen wollen, als etwas "trockener Stoff" anmuten. Um den immer vorhandenen Querbezug zur räumlichen Geometrie und damit die mathematische Intuition zu fördern, werden wir immer parallel zur mathematischen Theorie Visualisierungsmöglichkeiten mit MuPAD andeuten.

## 1.2 Motivation: Von der Analysis zur Analytischen Geometrie

In der Analysis werden lineare Funktionen in der Regel durch ihre Steigung  $m$  und ihren Ordinatenabschnitt  $b$  beschrieben, d.h. eine Gerade ist von der Form

$$y = m \cdot x + b.$$

Beispiele für solche Geraden sehen wir in Abbildung 1.1.

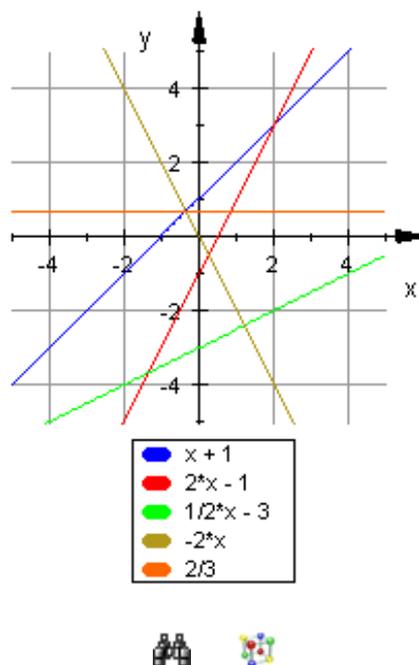


Abbildung 1.1: Einige Geraden in einem 2D-Koordinatensystem

Wodurch ist eine solche Gerade bestimmt? Die Antwort ist schlicht und ergreifend: "Durch zwei verschiedene Punkte auf der Geraden." Haben wir zwei Punkte gegeben, so können wir gedanklich ein Lineal anlegen und beide Punkte zu einer Strecke verbinden, die wir dann an den Endpunkten zu der eigentlichen Geraden verlängern.

Betrachten wir das konkrete Beispiel der Geraden  $y = x + 1$ . Sicherlich liegen die Punkte  $A(-1|0)$  und  $B(1|2)$  auf der Geraden (um dies zu prüfen, setzen wir die  $x$ -Koordinate

für  $x$  und die  $y$ -Koordinate für  $y$  in die Geradengleichung ein und sehen, dass auf beiden Seiten der Gleichung dasselbe steht). Die Gerade ergibt sich geometrisch, indem wir  $A$  und  $B$  verbinden und anschließend die Strecke an den Enden zu der Geraden verlängern (siehe Abbildung 1.2).

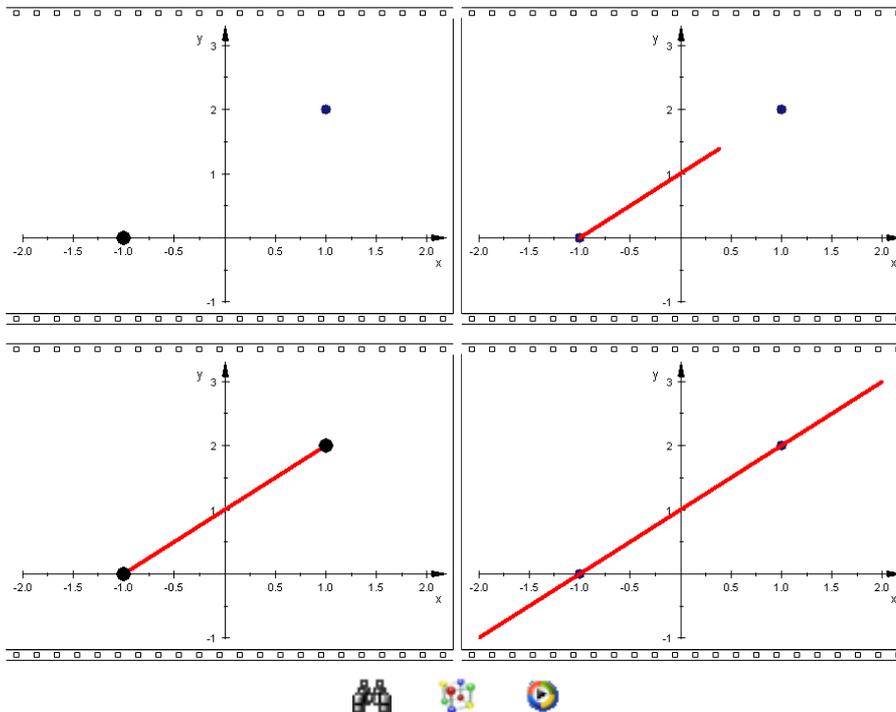


Abbildung 1.2: Konstruktion einer Geraden

Abbildung 1.2 zeigt vier Einzelbilder einer mit MuPAD erzeugten Animation. Solche Animationen werden im gesamten Buch durch die angedeuteten Filmstreifen gekennzeichnet (diese tauchen im Übrigen auch in der Dokumentation der MuPAD-Grafik-Bibliothek auf, wenn Animationen beschrieben werden).

Leider hat die angegebene “Konstruktionsbeschreibung” im Prinzip recht wenig mit unserer mathematischen Beschreibung der Geraden in Form der Gleichung  $y = x + 1$  zu tun. Wir wollen daher eine alternative und intuitivere Beschreibung der Geraden finden, der wir die “geometrische Konstruktion” besser ansehen.

Stellen wir uns vor, die Gerade wäre eine Straße, die durch einen Zielort  $Z$  (ein allgemeiner Punkt der Geraden) verläuft. Diesen Punkt wollen wir von einem Startpunkt aus mit einem Fahrrad oder einem Auto erreichen. Unser Startpunkt soll der Ursprung des Koordinatensystems  $U(0|0)$  sein. Wir wissen, dass es eine Straße vom Startpunkt aus hin zu dem “Ort”  $A$  gibt, durch den die Straße führt, die auch durch den Zielort  $Z$  verläuft. Angenommen, wir haben nur die “Orte”  $A(-1|0)$  und  $B(1|2)$  als Orientierungspunkte auf der Geraden, mit denen wir im folgenden arbeiten können. Wie gelangen wir

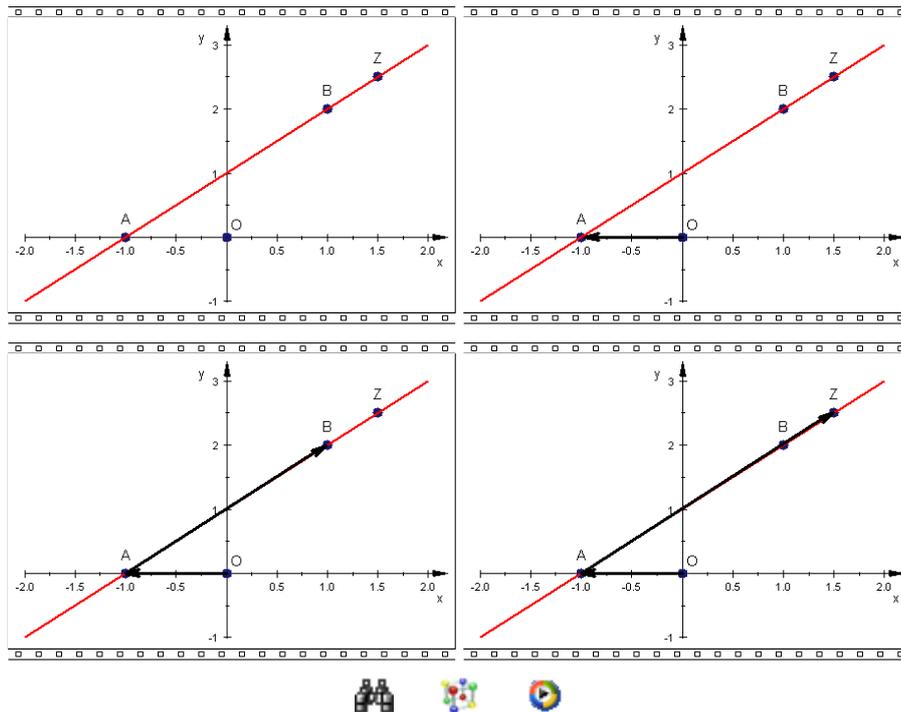


Abbildung 1.3: Wegbeschreibung im Zweidimensionalen

also von unserem Startpunkt zu dem dem Zielort  $Z$ ? Die Situation ist in Abbildung 1.3 verdeutlicht. Wie im praktischen Leben, geben wir eine *Wegbeschreibung* an, die zum Beispiel so aussehen könnte:

1. Fahre z.B. vom Startpunkt  $(0|0)$  aus mit dem Fahrzeug in Richtung  $A(0|-1)$ .
2. Fahre dann von  $A(0|-1)$  die Straße entlang in Richtung  $B(1|2)$  bis Du  $X$  erreicht hast.

Wenn auch schematisch stark vereinfacht (die Straßen sind in unserem Beispiel natürlich alle pfeilgerade), so zeigt dieses Beispiel die Grundidee einer alternativen Beschreibung von Geraden in der Ebene auf. Wir werden im nächsten Kapitel lernen, wie man die in Abbildung 1.3 angedeuteten Pfeile mathematisch exakt beschreiben und später mit ihrer Hilfe auch eine Alternative zu der bekannten Geradengleichung  $y = x + 1$  finden kann.

Auch wenn wir uns hier zunächst keine Gedanken über die mathematische Beschreibung machen, so stellt sich doch die Frage: Worin besteht der Mehrwert dieser Art von *Wegbeschreibungen* zur Beschreibung von Geraden?

Die Antwort ist: Man kann die Beschreibung von Geraden durch Gleichungen wie etwa  $y = x + 1$  nicht vom Zweidimensionalen ins Dreidimensionale übertragen. Für Geraden

im Raum gibt es keine solche ‐Darstellungsform‐. Abbildung 1.4 zeigt, was wir f ur ein geometrisches Objekt erhalten, wenn wir die Geradengleichung  $y = x + 1$  einfach so als Gleichung eines mathematischen Objekts im Dreidimensionalen auffassen. Abbildung 1.4 zeigt aber keine Gerade, sondern vielmehr eine ‐Ebene‐. Warum dies der Fall ist, k onnen wir erst sp ater im Detail verstehen.

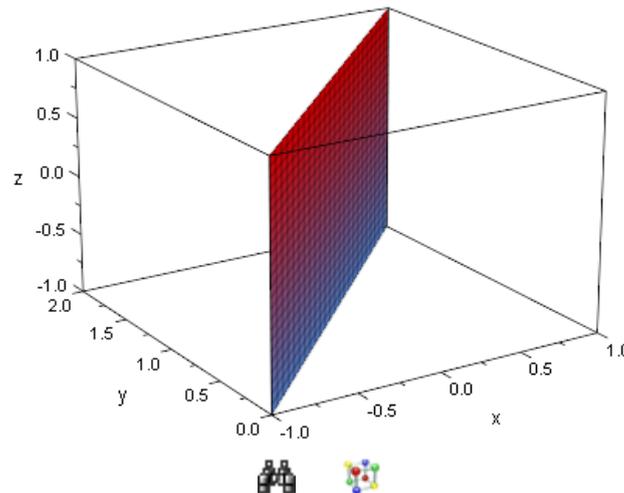


Abbildung 1.4: 2D-Geradengleichung  $y = x + 1$  im Raum

Intuitiv kann man dies wie folgt einsehen: Die Gleichung  $y = x + 1$  gibt nur implizit Bedingungen vor, an die sich die Koordinaten  $x$  und  $y$  ‐zu halten‐ haben. Die dritte Koordinate im Raum (wir nennen sie  $z$ -Koordinate) unterliegt keinerlei Einschr ankungen. Die  $z$ -Koordinate entspricht im Prinzip der ‐H ohe‐, in der ein Punkt im Raum angebracht ist. Wenn wir nun aber die H ohe durch die mathematische Gleichung  $y = x + 1$  gar nicht festlegen, so entsteht aus einer Geradengleichung im Zweidimensionalen eine Ebenengleichung im Dreidimensionalen (siehe auch Abbildung 1.5).

Zur uck zur Beschreibung von Geraden, die wir uns im Raum wie gerade ‐Fluglinien‐ vorstellen k onnen. Wir haben gesehen, dass die aus der Analysis bekannte Geradengleichung f ur den dreidimensionalen Fall nicht geeignet ist. Dagegen k onnen wir, wenn wir Punkte  $A$ ,  $B$  und  $Z$  im Raum gegeben haben und eine ‐Fluglinie‐, die durch  $A$  und  $B$  verl auft, bei der Wegbeschreibung von einem Startpunkt hin entlang der ‐Fluglinie‐ durch  $A$  und  $B$  genau so vorgehen, wie wir es oben f ur den zweidimensionalen Fall getan haben. Die Situation ist in Abbildung 1.6 verdeutlicht. Die *Wegbeschreibung* ist wieder ganz einfach:

1. Fliege z.B. vom Startpunkt aus mit dem Flugzeug in Richtung  $A$ .
2. Fliege dann von  $A$  die Fluglinie in Richtung  $B$  bis Du  $X$  erreicht hast.

Wir haben bereits jetzt ein Prinzip kennen gelernt, wie man Geraden sowohl im Zweidimensionalen als auch im Dreidimensionalen beschreiben kann. Auf anschaulicher Ebene ist dieses Prinzip klar. Wir m ussen jetzt daran arbeiten, das n otige mathematische

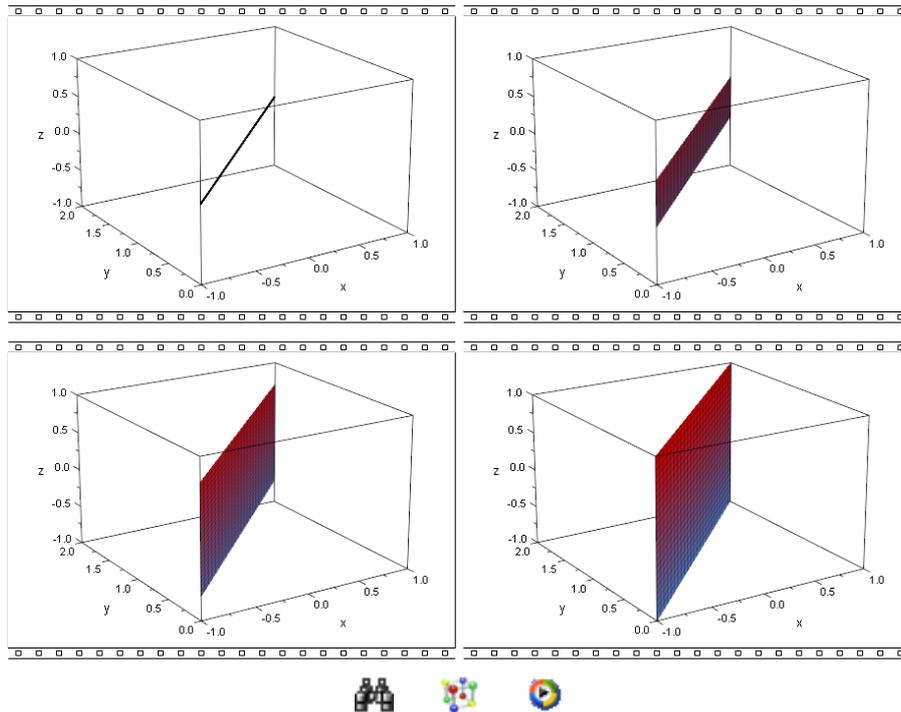


Abbildung 1.5: 2D-Geradengleichung  $y = x + 1$  im Raum liefert eine Ebene

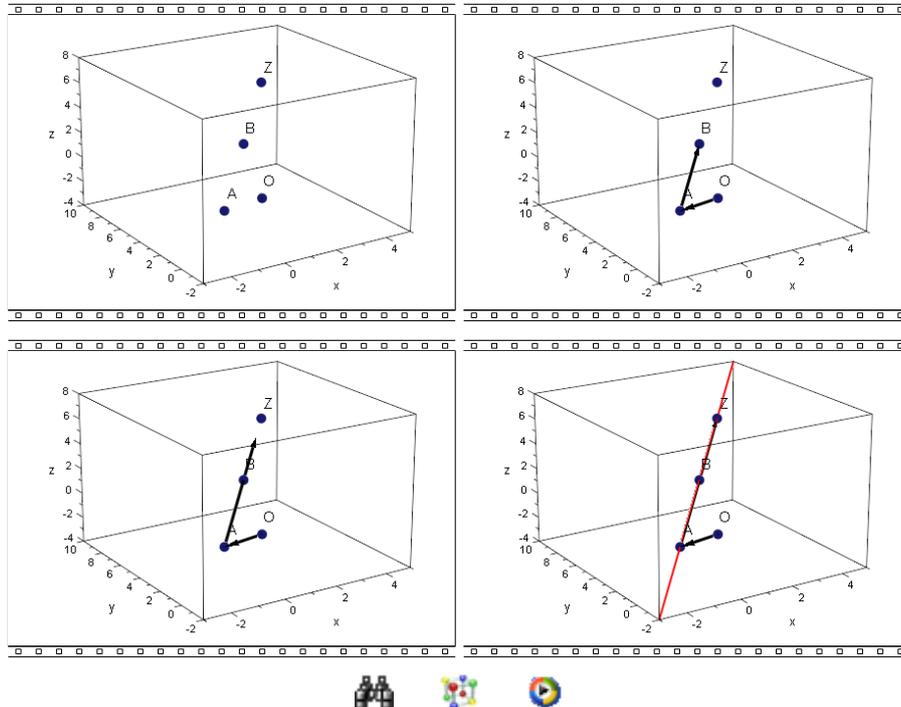


Abbildung 1.6: Wegbeschreibung im Dreidimensionalen

---

Modell zu entwickeln, um unsere *Wegbeschreibungen* in Form von mathematischen Gleichungen ausdrücken zu können. Der zentrale Bestandteil in den Abbildungen 1.3 und 1.6 sind die "Pfeile", denen wir zu folgen haben. Wir beginnen daher im nächsten Kapitel damit, "Pfeile" als mathematische Objekte (wir werden Sie *Vektoren* nennen) zu betrachten und "vernünftige Rechenregeln für Pfeile" zu definieren.

## 2 Vektoren und Vektorräume: Rechnen mit Pfeilen

In den Abbildungen 1.3 und 1.6 in Kapitel 1 hatten wir Geraden mit Hilfe von “Pfeilen” beschrieben. Wir wollen in diesem Kapitel “Pfeile” mathematisch definieren und zeigen, wie wir sinnvoll damit rechnen können. Mit ihrer Hilfe können wir dann später Geraden und auch Ebenen beschreiben zu können.

### 2.1 Technische-Vorbemerkungen zum Umgang mit MuPAD

Im folgenden unterscheiden wir immer zwei Ebenen: Die eine Ebene ist die mathematische Ebene, auf der wir rechnen. Die andere Ebene ist die visuelle Ebene, auf der wir mathematische Objekte von MuPAD darstellen lassen. MuPAD unterscheidet streng zwischen mathematischen und grafischen Objekten – im Prinzip genau so, wie wir es z.B. aus der Analysis gewohnt sind; auch dort unterscheiden wir zwischen Funktionsabbildungen (mathematische Ebene) und Funktionsgraphen (visuelle Ebene). Wir zum Zeichnen von Objekten immer so vor, dass wir uns aus der mathematischen Beschreibung des Objekts in MuPAD ein grafisches Objekt erstellen und uns dieses anschließend von MuPAD mit dem Befehl `plot` (“zeichne”) auf dem Bildschirm ausgeben lassen. Über dieses Buch hinaus bieten [3] und [4] leicht verständliche Einführungen in den Umgang mit MuPAD. Für technisch Interessierte ist zusätzlich [1] zu empfehlen. Weiteres Material zum Einsatz von MuPAD auf dem Gebiet der Analytischen Geometrie enthalten auch [5] und [6].

### 2.2 Vektoren in der Ebene: Der $\mathbb{R}^2$

#### Neue MuPAD-Funktionen

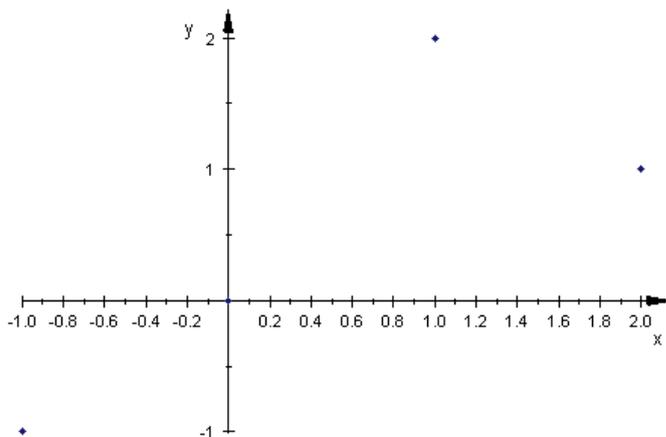
<code>plot</code>	Ausgabe von 2D- und 3D-Grafiken auf dem Bildschirm
<code>plot::Point2d</code>	2D-Punkte zeichnen
<code>matrix</code>	Definition von Vektoren
<code>plot::Arrow2d</code>	2D-Pfeile zeichnen
<code>bool</code>	Wahrheitsgehalt einer Aussage überprüfen
<code>expand</code>	Ausmultiplizieren von Termen
<code>Color</code>	Farbe (z.B. von Punkten oder Vektoren)
<code>RGB::Red</code>	rote Farbe
<code>LineWidth</code>	Linienstärke bei grafischen Objekten

Der Einfachheit halber beschränken wir unsere Betrachtungen zuerst auf den (bekannten) zweidimensionalen Fall und verallgemeinern diese dann in Abschnitt 2.3 auf den dreidimensionalen Fall. Rein anschaulich ist klar, dass “Pfeile” etwas anderes sind als Punkte. Diesen Unterschied müssen wir im folgenden herausarbeiten. Punkte im Koordinatensystem schreiben wir wie üblich in der Form  $P(x|y)$ , wobei  $x$  die  $x$ -Koordinate und  $y$  die  $y$ -Koordinate des Punktes  $P$  ist. Wir lassen uns die Punkte  $A(1|2)$ ,  $B(2|1)$ ,  $C(-1|-1)$  sowie den Koordinatenursprung  $U(0|0)$  von MuPAD in ein gemeinsames Koordinatensystem zeichnen. Dazu erzeugen wir die vier Punkte zunächst als einzelne grafische Objekte mit Hilfe der MuPAD-Funktion `plot::Point2d`:

- `A:= plot::Point2d( 1, 2); B:= plot::Point2d( 2, 1);`  
`C:= plot::Point2d(-1,-1); U:= plot::Point2d( 0, 0);`  
`plot::Point2d(1, 2)`  
`plot::Point2d(2, 1)`  
`plot::Point2d(-1, -1)`  
`plot::Point2d(0, 0)`

Jetzt zeichnen wir die Punkte mit dem Befehls `plot` alle in ein gemeinsames Koordinatensystem:

- `plot(A, B, C, U)`



Wir ordnen nun im folgenden jedem allgemeinen Punkt  $P(x|y)$  einen “Pfeil” vom Koordinatenursprung hin zu dem Punkt  $P(x|y)$  zu. Diesen Pfeil nennen wir den *Ortsvektor des Punktes*  $P(x|y)$  und schreiben ihn in der Form

$$\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

Ortsvektoren können wir in MuPAD mit dem Befehl `matrix` definieren

- `a:= matrix([ 1, 2]); b:= matrix([ 2, 1]);`  
`c:= matrix([-1,-1]); u:= matrix([ 0, 0]);`

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

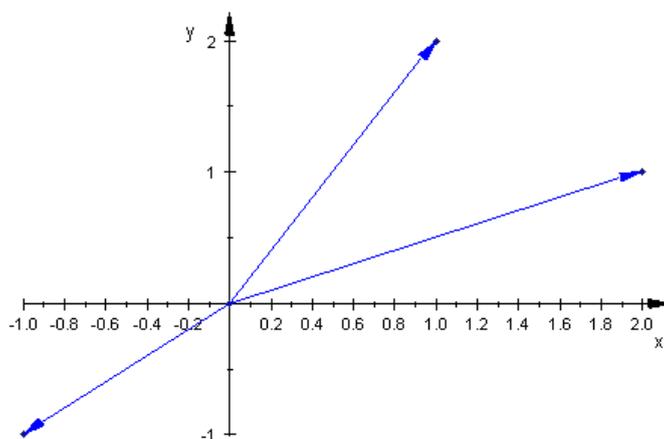
$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Damit haben wir die mathematische Darstellung von Ortsvektoren in MuPAD zur Hand. Mit Hilfe von `plot::Arrow2d` lassen sich die zugehörigen grafischen Pfeile definieren:

- `OrtsvektorA:= plot::Arrow2d(a):`  
`OrtsvektorB:= plot::Arrow2d(b):`  
`OrtsvektorC:= plot::Arrow2d(c):`  
`OrtsvektorU:= plot::Arrow2d(u):`

Dabei haben wir die Eingaben mit Doppelpunkten abgeschlossen. Auf diese Weise unterdrücken wir die MuPAD-Ausgaben, die uns hier keine neue Information liefern. Jetzt lassen sich die Punkte  $A$ ,  $B$ ,  $C$  und  $U$  sowie die zugehörigen Ortsvektoren ganz leicht in ein gemeinsames Koordinatensystem zeichnen:

- `plot(A, B, C, U,`  
`OrtsvektorA, OrtsvektorB, OrtsvektorC, OrtsvektorU)`



Fehlt dort nicht etwas? Der Ortsvektor des Ursprungs  $U(0|0)$  ist nicht zu sehen. Grund dafür ist die Tatsache, dass wir Ortsvektoren als "Pfeile" vom Koordinatenursprung hin zu einem entsprechenden Punkt definiert haben. Der Ortsvektor des Ursprungs verläuft also vom Punkt  $U(0|0)$  zu sich selbst und ist daher – korrekter Weise – nicht sichtbar.

Oben haben wir erwähnt, dass wir zwischen Punkten und Vektoren unterscheiden müssen. Diesen Unterschied machen wir uns an einem Beispiel klar: Der Punkt  $P(1|2)$  ist durch seine Koordinaten eindeutig beschrieben, d.h. insbesondere seine *Position* im Koordinatensystem ist festgelegt. Ebenso ist der *Ortsvektor*

$$\vec{p} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

eindeutig bestimmt als derjenige "Pfeil", der vom Ursprung hin zu dem Punkt  $P$  reicht. Diesen "Pfeil" können wir aber beliebig in der Ebene parallel verschieben, ohne dass sich seine Richtung oder seine Länge verändert. Wir erhalten rein geometrisch gesehen immer den gleichen Pfeil. Wir nennen daher jeden durch Verschiebung in der Ebene aus dem Ortsvektor entstehenden Pfeil einen *Repräsentanten des Vektors*  $\vec{p}$ . Geometrisch gesehen steht also der Vektor  $\vec{p}$  für die *Menge aller Pfeile mit gleicher Länge und gleicher Richtung wie der Ortsvektor des Punktes  $P$*  (siehe auch Abbildung 2.1).

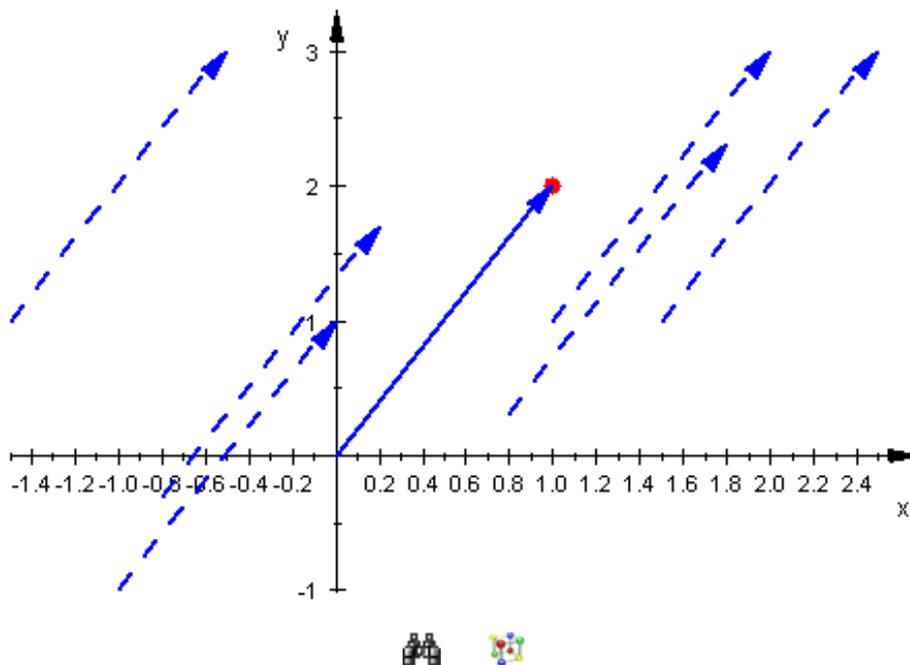


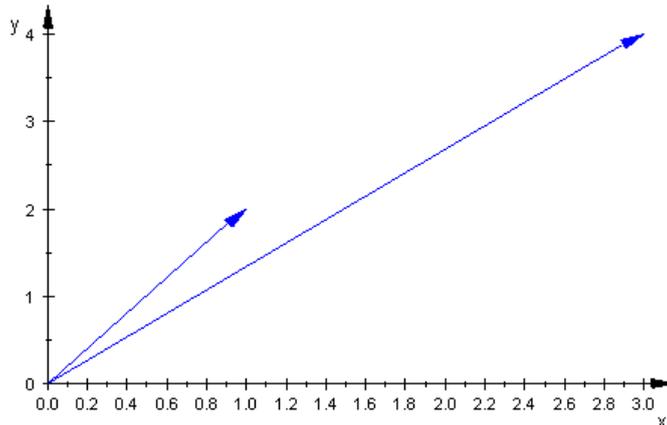
Abbildung 2.1: Ein Punkt und einige Repräsentanten des zugehörigen Vektors

Jetzt können wir Rechenregeln für Vektoren angeben. Sind also

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad \text{und} \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Vektoren, so wollen wir uns zuerst überlegen, wie wir  $\vec{a}$  und  $\vec{b}$  addieren können. Um eine sinnvolle Definition für die Addition zu finden, sollten wir uns überlegen, was geometrisch die Summe von  $\vec{a}$  und  $\vec{b}$  sein sollte. Betrachten wir als Beispiel die Ortsvektoren der beiden Punkte  $A(1|2)$  und  $B(3|4)$ :

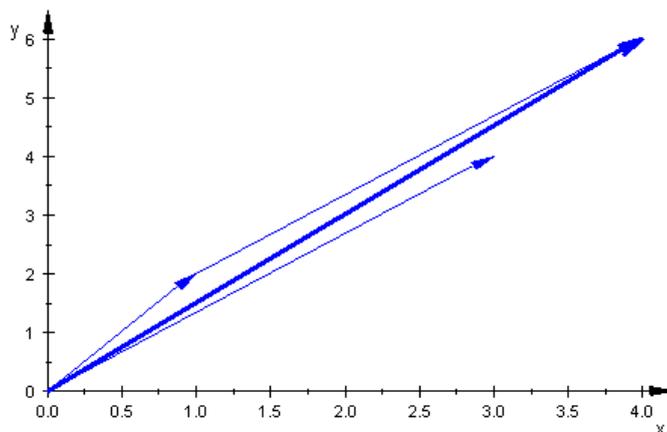
- `OrtsvektorA:= plot::Arrow2d(matrix([1, 2])):`  
`OrtsvektorB:= plot::Arrow2d(matrix([3, 4])):`  
`plot(OrtsvektorA, OrtsvektorB)`



Addition bedeutet frei aus dem Lateinischen übersetzt schlicht “zusammensetzen”. Geometrisch sollte sich die Summe der beiden Vektoren ergeben, indem man einfach einen Vektor an den anderen “ansetzt”.

Wir verschieben also den Ortsvektor von  $B(3|4)$  so, dass er an der Spitze des Ortsvektors von  $A(1|2)$  ansetzt und zeichnen den Vektor ein, der vom Ursprung hin zu der Pfeilspitze dieses Vektors reicht:

- `Hilfsvektor:= plot::Arrow2d(matrix([1, 2]),`  
`matrix([4, 6])):`  
`Summe:= plot::Arrow2d(matrix([4, 6]), LineWidth = 0.8):`  
`plot(OrtsvektorA, OrtsvektorB, Hilfsvektor, Summe)`



Den dicker gezeichneten Pfeil nennen wir also jetzt die *Summe der beiden Vektoren*  $\vec{a}$  und  $\vec{b}$ .

Dieser ist auch gleichzeitig der Ortsvektor des Punkte  $C(4|6)$ . Rein rechnerisch stellt sich jetzt also die Frage, wie der Ortsvektor

$$\vec{c} = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$$

sich aus den beiden Vektoren

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{und} \quad \vec{b} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

ergibt. Offensichtlich ergibt sich die erste Komponente von  $\vec{c}$ , indem man die beiden ersten Komponenten von  $\vec{a}$  und  $\vec{b}$  addiert; analog für die zweite Komponente von  $\vec{c}$  und die zweiten Komponenten von  $\vec{a}$  und  $\vec{b}$ . Insgesamt gilt also:

$$\vec{c} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 1+3 \\ 2+4 \end{pmatrix}.$$

Also macht es Sinn, die Summe zweier allgemeiner Vektoren

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad \text{und} \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

wie folgt zu definieren:

$$\vec{a} + \vec{b} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix}.$$

Wir können jetzt Vektoren addieren; MuPAD kann das natürlich auch und liefert uns die soeben hergeleitete Rechenregel "frei Haus", wenn wir nur allgemeine Vektoren definieren:

- `a:= matrix([a1, a2]): b:= matrix([b1, b2]):`

`a + b`

$$\begin{pmatrix} a1 + b1 \\ a2 + b2 \end{pmatrix}$$

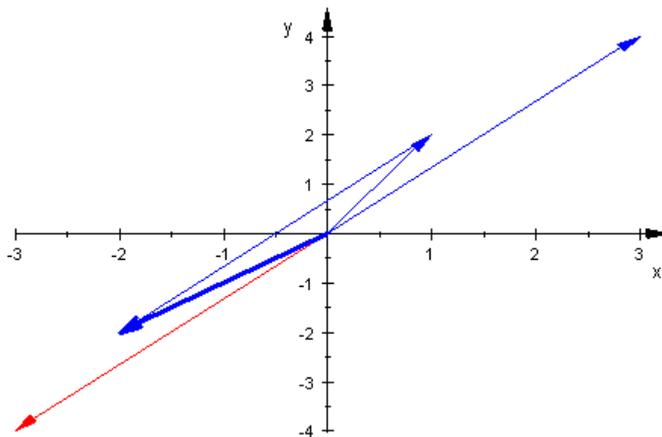
Wenn wir Vektoren addieren können, dann müssen wir uns als nächstes fragen, wie wir sie subtrahieren können. Rein rechnerisch ist das klar: Wenn wir  $\vec{a} + \vec{b}$  berechnen, indem wir die Komponenten von  $\vec{a}$  und  $\vec{b}$  addieren, dann berechnen wir  $\vec{a} - \vec{b}$ , indem wir diese voneinander abziehen. MuPAD liefert uns diese Rechenregel wieder auf Knopfdruck:

- `a - b`

$$\begin{pmatrix} a_1 - b_1 \\ a_2 - b_2 \end{pmatrix}$$

Was bedeutet die Subtraktion von Vektoren jetzt aber geometrisch? Wir können  $\vec{a} - \vec{b}$  auch in der Form  $\vec{a} + (-\vec{b})$  schreiben, d.h. wir addieren zu dem Vektor  $\vec{a}$  den so genannten *Gegenvektor* von  $\vec{b}$ . Unter dem Gegenvektor  $-\vec{b}$  von  $\vec{b}$  wollen wir einfach denjenigen Vektor verstehen, der sich ergibt, wenn wir die Richtung von  $\vec{b}$  umkehren. In unserem konkreten Beispiel ergibt sich (den Gegenvektor zeichnen wir mit Hilfe der Option `Color` in der Farbe `RGB::Red`):

- `Gegenvektor:= plot::Arrow2d(matrix([-3, -4]),  
Color = RGB::Red):`  
`Hilfsvektor:= plot::Arrow2d(matrix([1,2]),matrix([-2,-2])):`  
`Differenz:= plot::Arrow2d(matrix([-2,-2]), LineWidth = 0.8):`  
`plot(OrtsvektorA, OrtsvektorB, Gegenvektor, Hilfsvektor,  
Differenz)`



In der obigen Grafik ist der Gegenvektor von  $\vec{b}$  in roter Farbe eingezeichnet. Den Differenzvektor  $\vec{a} - \vec{b}$  haben wir dicker zeichnen lassen.

Sind die so definierte Addition und Subtraktion von Vektoren mathematisch sinnvoll? Wenn ja, dann sollten auch für Vektoren die von den reellen Zahlen gewohnten Rechenregeln gelten, die wir so als völlig selbstverständlich annehmen, d.h.

- Existenz des "Nullvektors":** Für den Vektor  $\vec{0}$ , dessen Komponenten nur aus Nullen bestehen und für alle Vektoren  $\vec{a}$  gilt:  $\vec{a} + \vec{0} = \vec{a}$ .
- Kommutativgesetz:**  $\vec{a} + \vec{b} = \vec{b} + \vec{a}$  für alle Vektoren  $\vec{a}$  und  $\vec{b}$ .
- Assoziativgesetz:**  $(\vec{a} + \vec{b}) + \vec{c} = \vec{a} + (\vec{b} + \vec{c})$  für alle Vektoren  $\vec{a}$ ,  $\vec{b}$  und  $\vec{c}$ .

Die Rechenregeln sind genau genommen alle im Detail zu beweisen, bevor wir sie benutzen können. Weil das Nachrechnen diese Eigenschaften recht langweilig sein kann,

lassen wir MuPAD für uns rechnen. Wir müssen MuPAD nur "sagen", was zu tun ist und das Ergebnis richtig interpretieren. Zuerst definieren wir uns drei allgemeine Vektoren sowie den Nullvektor:

- `a:= matrix([a1,a2]): b:= matrix([b1,b2]):`  
`c:= matrix([c1,c2]): nv:= matrix([0, 0]):`

Wir beweisen (i):

- `a + nv`  

$$\begin{pmatrix} a1 \\ a2 \end{pmatrix}$$

Dies ist offensichtlich wieder der Vektor  $\vec{a}$ , also gilt die Rechenregel  $\vec{a} + \vec{0} = \vec{a}$ . Zum Beweis von (ii) vergleichen wir einfach die Summen  $\vec{a} + \vec{b}$  und  $\vec{b} + \vec{a}$ :

- `a + b, b + a, bool(a + b = b + a)`  

$$\begin{pmatrix} a1 + b1 \\ a2 + b2 \end{pmatrix}, \begin{pmatrix} a1 + b1 \\ a2 + b2 \end{pmatrix}, \text{TRUE}$$

Wir sehen, dass  $\vec{a} + \vec{b}$  und  $\vec{b} + \vec{a}$  offensichtlich zu dem gleichen Vektor führt. Zusätzlich haben wir mit dem Befehl `bool` den Wahrheitsgehalt der Behauptung  $\vec{a} + \vec{b} = \vec{b} + \vec{a}$  abgefragt und MuPAD liefert uns das Ergebnis `TRUE`, d.h. die Aussage ist allgemein gültig. An dieser Stelle können wir den Wahrheitsgehalt natürlich so ablesen, d.h. wir hätten im Prinzip auf die Funktion `bool` verzichten können. Später werden wir aber noch den einen oder anderen komplizierteren Beweis mit MuPAD führen, so dass wir vielleicht nicht mehr auf einen Blick sehen, dass die entsprechende Aussage oder Gleichung all-gemeingültig ist. In diesem Kontext wird sich die Funktion `bool` als sehr nützlich erweisen.

Zum Beweis von (iii) verfahren wir ganz analog:

- `(a+b) + c, a + (b+c), bool((a+b) + c = a + (b+c))`  

$$\begin{pmatrix} a1 + b1 + c1 \\ a2 + b2 + c2 \end{pmatrix}, \begin{pmatrix} a1 + b1 + c1 \\ a2 + b2 + c2 \end{pmatrix}, \text{TRUE}$$

Fertig. Damit gelten also für die Addition von Vektoren genau die gleichen Regeln, die wir von der Addition gewöhnlicher reeller Zahlen gelten.

## Übungen

- 1.) Beweisen Sie  $\vec{a} + \vec{b} = \vec{b} + \vec{a}$  geometrisch mit Hilfe einer MuPAD-Grafik für die speziellen Vektoren

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{und} \quad \vec{b} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

2.) Beweisen Sie  $(\vec{a} + \vec{b}) + \vec{c} = \vec{a} + (\vec{b} + \vec{c})$  geometrisch mit Hilfe einer MuPAD-Grafik für die speziellen Vektoren

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad \text{und} \quad \vec{c} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}.$$

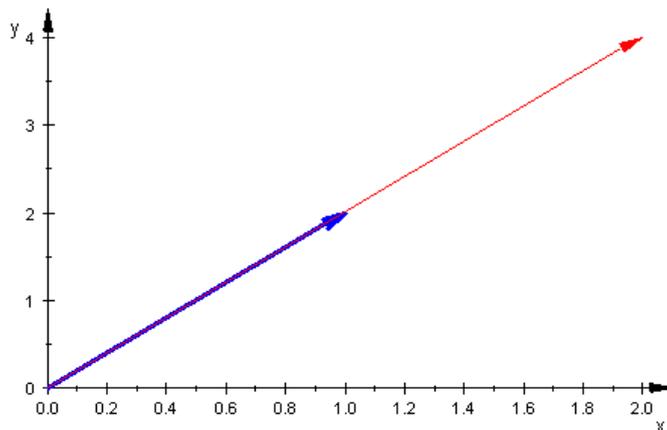
Wenn wir nun also Vektoren addieren können, dann stellt sich die Frage: Lässt sich auch eine sinnvolle Multiplikation definieren? Wie man zwei Vektoren miteinander auf sinnvolle Weise multiplizieren kann, werden wir später sehen. Hier überlegen wir uns jetzt, wie man einen Vektor mit einer reellen Zahl multiplizieren kann und welche geometrische Deutung diese Rechnung zulässt.

Als Beispiel schauen wir uns die beiden Vektoren

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{und} \quad \vec{b} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

an, die wir als Ortsvektoren der Punkte  $A(1|2)$  und  $B(2|4)$  auffassen und von MuPAD zeichnen lassen können:

```
• a:= matrix([1,2]): b:= matrix([2,4]):
  OrtsvektorA:= plot::Arrow2d(a, LineWidth = 0.8):
  OrtsvektorB:= plot::Arrow2d(b, Color = RGB::Red):
  plot(OrtsvektorA, OrtsvektorB)
```



Offensichtlich ist der rote Vektor – also der Ortsvektor des Punkte  $A$  – doppelt so lang wie der blaue Vektor. Vergleichen wir die Komponenten von  $\vec{a}$  und  $\vec{b}$ , so sehen wir, dass wir die Komponenten von  $\vec{a}$  nur mit dem Faktor 2 multiplizieren müssen, um  $\vec{b}$  zu erhalten.

Wir können also eine so genannte *skalare Multiplikation* von einem Vektor  $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$  mit dem *Skalar* (d.h. der reellen Zahl)  $k \in \mathbb{R}$  definieren über

$$k \cdot \vec{v} = k \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} k \cdot v_1 \\ k \cdot v_2 \end{pmatrix}.$$

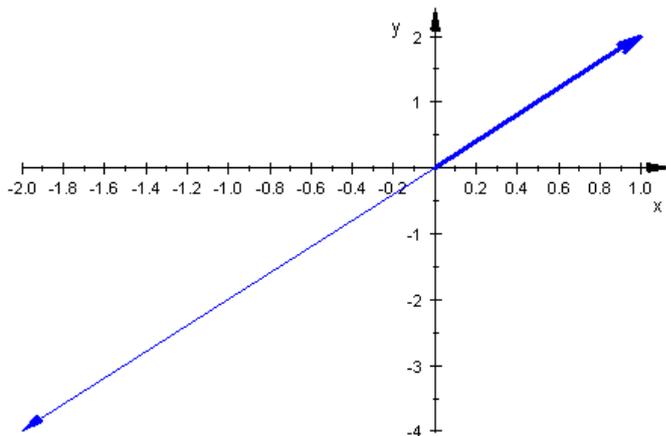
Wie schon bei der Addition von Vektoren liefert uns MuPAD diese "Rechenregel", wenn wir einen allgemeinen Skalar  $k$  und einen allgemeinen Vektor  $v$  definiere und  $k * v$  ausrechnen lassen:

- `v:= matrix([v1,v2]):`  
`k * v`  

$$\begin{pmatrix} kv1 \\ kv2 \end{pmatrix}$$

So weit, so gut. Was passiert nun geometrisch, wenn wir einen Vektor mit einer negativen Zahl multiplizieren? Wir schauen uns das anhand des Beispielvektors  $a$  an, den wir oben in MuPAD definiert hatten, und den wir mit dem Skalar  $-2$  multiplizieren:

- `c:= -2 * a:`  
`OrtsvektorC:= plot::Arrow2d(c):`  
`plot(OrtsvektorA, OrtsvektorC)`



Wie wir sehen, erhalten wir einen Vektor  $\vec{c}$ , der geometrisch zwar doppelt so lang wie der Vektor  $\vec{a}$  ist, aber die entgegengesetzte Richtung hat. Dies ist auch intuitiv verständlich, denn wir können schreiben

$$-2 \cdot \vec{a} = \begin{pmatrix} (-2) \cdot 1 \\ (-2) \cdot 2 \end{pmatrix} = \begin{pmatrix} -2 \\ -4 \end{pmatrix} = - \begin{pmatrix} 2 \\ 4 \end{pmatrix}.$$

Der Vektor  $-\begin{pmatrix} 2 \\ 4 \end{pmatrix}$  ist aber nichts anderes als der Gegenvektor des Vektors  $\vec{b}$  von oben. Wir erhalten also  $-2 \cdot \vec{a}$  geometrisch, indem wir den Vektor  $\vec{a}$  zunächst mit dem Faktor 2 strecken (d.h. seine Länge verdoppeln) und anschließend – wegen des Minuszeichens – seine Richtung umdrehen.

Auch für die Multiplikation eines Vektors mit einem Skalar gelten Rechenregeln, wie wir sie von dem Rechnen mit reellen Zahlen kennen:

- (i) **Distributivgesetz:**  $k \cdot (\vec{a} + \vec{b}) = k \cdot \vec{a} + k \cdot \vec{b}$  und  $(k + l) \cdot \vec{a} = k \cdot \vec{a} + l \cdot \vec{a}$  für alle  $k, l \in \mathbb{R}$  und alle Vektoren  $\vec{a}$  und  $\vec{b}$ .

(ii) **Assoziativgesetz:**  $(k \cdot l) \cdot \vec{a} = k \cdot (l \cdot \vec{a})$  für alle  $k, l \in \mathbb{R}$  und alle Vektoren  $\vec{a}$  und  $\vec{b}$ .

Den Beweis der Eigenschaften erledigen wir wieder mit MuPAD: Zunächst die Aussagen unter (i)

```

• a:= matrix([a1,a2]): b:= matrix([b1,b2]):
  k * (a+b), k*a + k*b, bool(k * (a+b) = k*a + k*b);
  (k+1) * a, k*a + l*a, bool((k+1) * a = k*a + l*a);
   $\begin{pmatrix} k(a1+b1) \\ k(a2+b2) \end{pmatrix}, \begin{pmatrix} a1k+b1k \\ a2k+b2k \end{pmatrix}, \text{FALSE}$ 
   $\begin{pmatrix} a1(k+l) \\ a2(k+l) \end{pmatrix}, \begin{pmatrix} a1k+a1l \\ a2k+a2l \end{pmatrix}, \text{FALSE}$ 

```

MuPAD liefert uns in beiden Fällen **FALSE**. Sind die Rechenregeln etwa gar nicht korrekt oder ist MuPAD zu dumm? Die Antwort ist "Nein": Die Rechenregeln sind korrekt und MuPAD ist nicht zu dumm. Aus Effizienzgründen führt **bool** einen direkten Vergleich der linken Seite mit der rechten Seite der Gleichung durch, ohne – wie es nötig wäre – die jeweiligen Komponenten der Vektoren auf der linken Seite auszumultiplizieren. Wir sollten MuPAD vor der Überprüfung auf Gleichheit dazu bringen, die Komponenten der Vektoren auf der linken Seite der Gleichungen auszumultiplizieren. Dies machen wir mit dem Befehl **expand**<sup>1</sup>:

```

• bool(expand(k * (a+b)) = k*a + k*b);
  bool(expand((k+1) * a) = k*a + l*a);
  TRUE

```

TRUE

Damit sind die Rechenregeln unter (i) bewiesen.

Für (ii) erhalten wir:

```

• bool(expand((k+l) * a) = k * (l*a))
  TRUE

```

Die Menge aller Vektoren mit zwei Komponenten und den in diesem Abschnitt definierten Rechenregeln bildet einen so genannten *Vektorraum*, den man den  $\mathbb{R}^2$  (gesprochen: R zwei) nennt. Wir sprechen daher im folgenden nicht mehr von der "Ebene" oder dem "Zweidimensionalen", sondern kurz und knapp vom  $\mathbb{R}^2$ .

<sup>1</sup>In MuPAD Pro 3.1 gibt es die Funktion **teste**, mit deren Hilfe man die Gleichheit der linken und der rechten Seite der jeweiligen Gleichung auch direkt nachweisen kann, ohne dass man extra auf die Funktion **expand** zurückgreifen muss; wenn auch aufwändiger, so schien es uns didaktisch klüger zu sein, das Ausmultiplizieren von MuPAD ausdrücklich mittels **expand** durchführen zu lassen, denn dieses ist genau das Vorgehen, was man wählt, wenn man die Gleichheit per Hand mit Papier und Bleistift beweist.

## 2.3 Vektoren im Raum: Der $\mathbb{R}^3$

### Neue MuPAD-Funktionen

<code>plot::Point3d</code>	3D-Punkte zeichnen
<code>plot::Arrow3d</code>	3D-Pfeile zeichnen
<code>RGB::Blue</code>	blaue Farbe
<code>RGB::Green</code>	grüne Farbe

Wir werden jetzt sehen, dass wir im Prinzip all das, was wir über den  $\mathbb{R}^2$  im vorhergehenden Abschnitt gelernt haben, unmittelbar auf den dreidimensionalen Raum – diesen werden wir später in Analogie zum vorhergehenden Abschnitt einfach als  $\mathbb{R}^3$  bezeichnen – übertragen können.

Wir müssen uns praktisch nur die Frage stellen, wie Punkte im Raum zu beschreiben sind. Im Gegensatz zu Punkten in einem zweidimensionalen Koordinatensystem müssen wir zur Angabe der Position eines Punktes im Raum *drei Koordinaten* –  $x$ -Koordinate,  $y$ -Koordinate und  $z$ -Koordinate genannt – angeben. Die  $x$ -Koordinate und die  $y$ -Koordinate entsprechen genau den Koordinaten eines Punktes im Zweidimensionalen. Die  $z$ -Koordinate kann man sich als "Höhe" vorstellen, in der der Punkte im Raum zu finden ist. Ein dreidimensionales Koordinatensystem hat also *drei Achsen*: Eine  $x$ -Achse, eine  $y$ -Achse und eine  $z$ -Achse (siehe auch Abbildung 2.2).

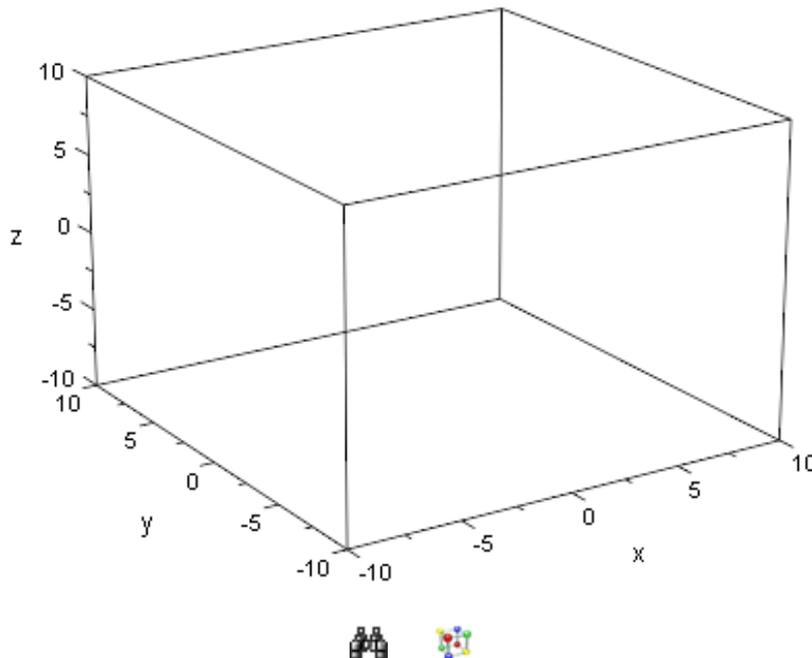
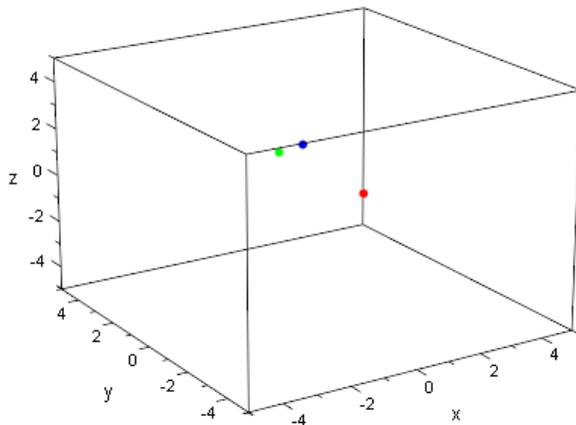


Abbildung 2.2: 3D-Koordinatensystem

Hat ein Punkt  $P$  im Raum die Koordinaten  $P(x|y|z)$ , so bedeutet dies, dass man

sich vom Koordinatenursprung  $(0|0|0)$  zuerst  $x$  Schritte in Richtung der  $x$ -Achse bewegt, dann  $y$  Schritte in Richtung der  $y$ -Achse und schließlich  $z$  Schritte in Richtung der  $z$ -Achse. Wir zeichnen mit MuPAD die drei Punkte  $A(1|2|1)$ ,  $B(-1|0|2)$  und  $C(-1|-4|2)$  mit Hilfe der Funktion `plot::Point3d` in unterschiedlichen Farben (`RGB::Blue` zeichnet den Punkt in blauer Farbe; `RGB::Green` zeichnet den Punkt in grüner Farbe) in ein dreidimensionales Koordinatensystem, dessen Achsenbeschriftungen auf allen Achsen von  $-5$  bis  $5$  reichen (in MuPAD erreicht man dies mit Hilfe des Attributs `ViewingBox`):

- `A:= plot::Point3d( 1, 2, 1, Color = RGB::Blue):`  
`B:= plot::Point3d(-1, 0, 2, Color = RGB::Green):`  
`C:= plot::Point3d(-1,-4, 2, Color = RGB::Red):`  
`plot(A,B,C, ViewingBox = [-5..5, -5..5, -5..5])`



Wie schon im vorhergehenden Abschnitt können wir jetzt jedem Punkt im Raum einen *Ortsvektor* zuordnen. Der Ortsvektor des Punktes  $P(x|y|z)$  ist geometrisch wieder genau derjenige "Pfeil", der im Koordinatenursprung  $(0|0|0)$  beginnt und dessen Spitze in dem Punkt  $P(x|y|z)$  endet. Wir schreiben den Ortsvektor in der Form

$$\vec{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

Zu unseren obigen drei Punkten  $A$ ,  $B$  und  $C$  können wir jetzt auch die Ortsvektoren in MuPAD definieren

- `a:= matrix([ 1, 2, 1]); b:= matrix([-1, 0, 2]);`  
`c:= matrix([-1,-4, 2]);`

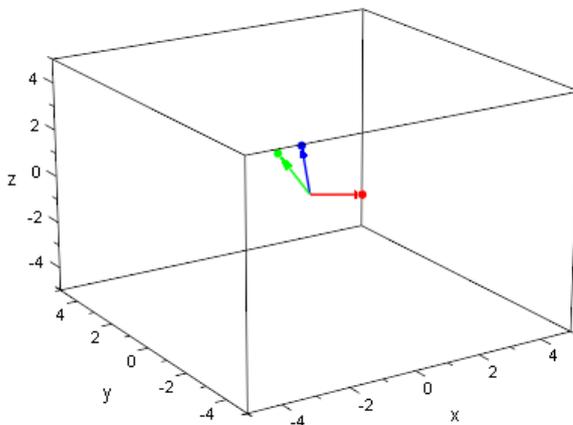
$$\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ -4 \\ 2 \end{pmatrix}$$

und diese anschließend mit den Punkten  $A$ ,  $B$  und  $C$  mit Hilfe von `plot::Arrow3d` in ein gemeinsames Koordinatensystem zeichnen:

- `OrtsvektorA:= plot::Arrow3d(a, Color = RGB::Blue):`  
`OrtsvektorB:= plot::Arrow3d(b, Color = RGB::Green):`  
`OrtsvektorC:= plot::Arrow3d(c, Color = RGB::Red):`  
`plot(A,B,C,OrtsvektorA,OrtsvektorB,OrtsvektorC,`  
`ViewingBox = [-5..5, -5..5, -5..5])`



Jeden der "Pfeile" können wir wieder beliebig parallel verschieben, ohne dass sich seine Richtung oder seine Länge verändert. Wir erhalten rein geometrisch gesehen immer den gleichen Pfeil. Wir nennen daher jeden durch Verschiebung aus dem Ortsvektor  $\vec{p}$  eines Punktes  $P(x|y|z)$  entstehenden Pfeil wieder einen *Repräsentanten des Vektors*  $\vec{p}$ . Geometrisch gesehen steht also der Vektor  $\vec{p}$  für die *Menge aller Pfeile mit gleicher Länge und gleicher Richtung wie der Ortsvektor des Punktes  $P$* . Abbildung 2.3 zeigt den Punkt  $A(1|2|1)$ , seinen Ortsvektor sowie diverse weitere Repräsentanten des Vektors.

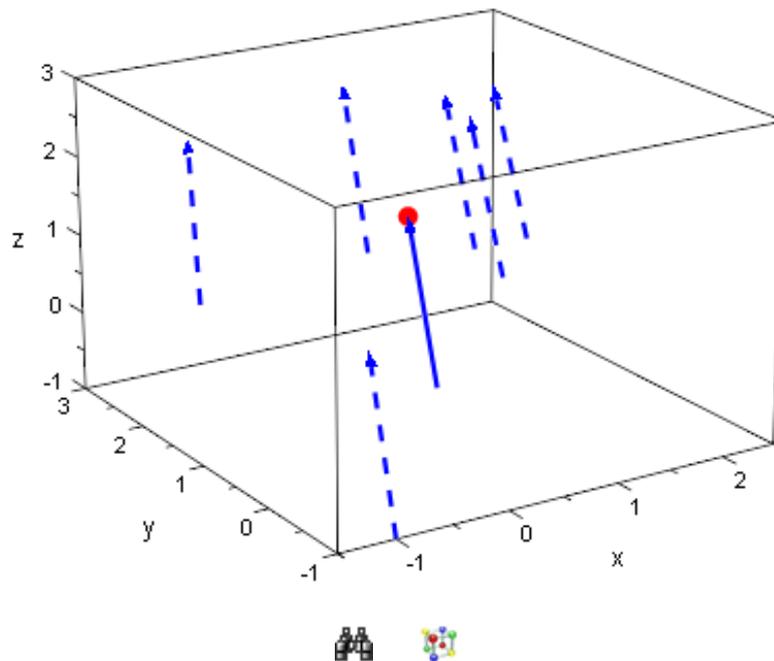


Abbildung 2.3: Der Punkt  $A(1|2|1)$ , sein Ortsvektor und weitere Repräsentanten

Für allgemeine Vektoren

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \text{und} \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

definieren wir die Addition bzw. Subtraktion ganz analog zur Addition bzw. Subtraktion von Vektoren im  $\mathbb{R}^2$ , die wir im vorhergehenden Abschnitt kennen gelernt hatten:

$$\vec{a} + \vec{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{pmatrix}$$

sowie

$$\vec{a} - \vec{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 - b_1 \\ a_2 - b_2 \\ a_3 - b_3 \end{pmatrix}.$$

Die Addition von Vektoren mit drei Komponenten erfolgt in MuPAD genau so, wie wir es von Vektoren des  $\mathbb{R}^2$  mit zwei Komponenten kennen:

- `a := matrix([a1, a2, a3]):`  
`b := matrix([b1, b2, b3]):`  
`a + b; a - b;`

$$\begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{pmatrix}$$

$$\begin{pmatrix} a_1 - b_1 \\ a_2 - b_2 \\ a_3 - b_3 \end{pmatrix}$$

Ferner definieren wir wieder eine *skalare Multiplikation*, d.h. die Multiplikation eines Vektors mit einer reellen Zahl  $k \in \mathbb{R}$ , über

$$k \cdot \vec{v} = k \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} k \cdot v_1 \\ k \cdot v_2 \\ k \cdot v_3 \end{pmatrix}.$$

Mit MuPAD ergibt sich auch hier das erwartete Ergebnis:

• `v:= matrix([v1,v2,v3]):`

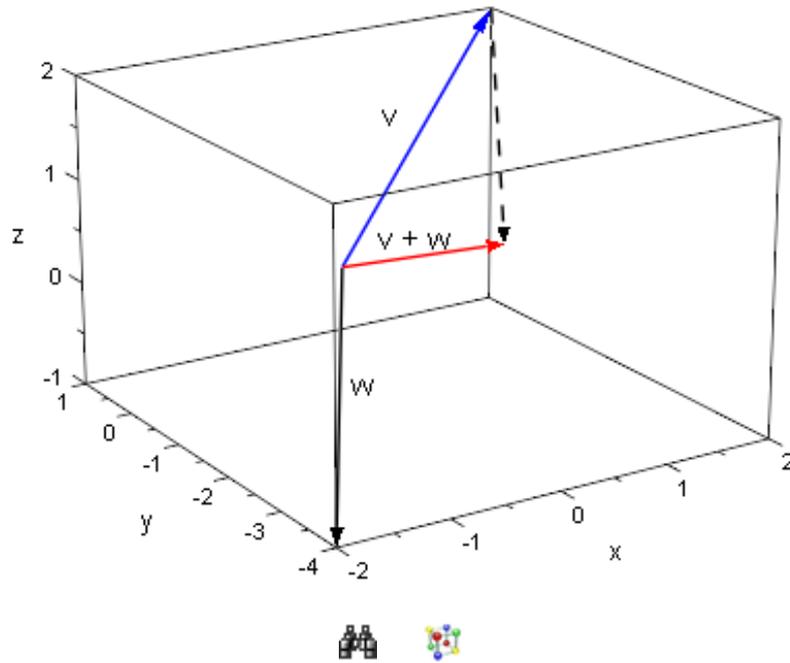
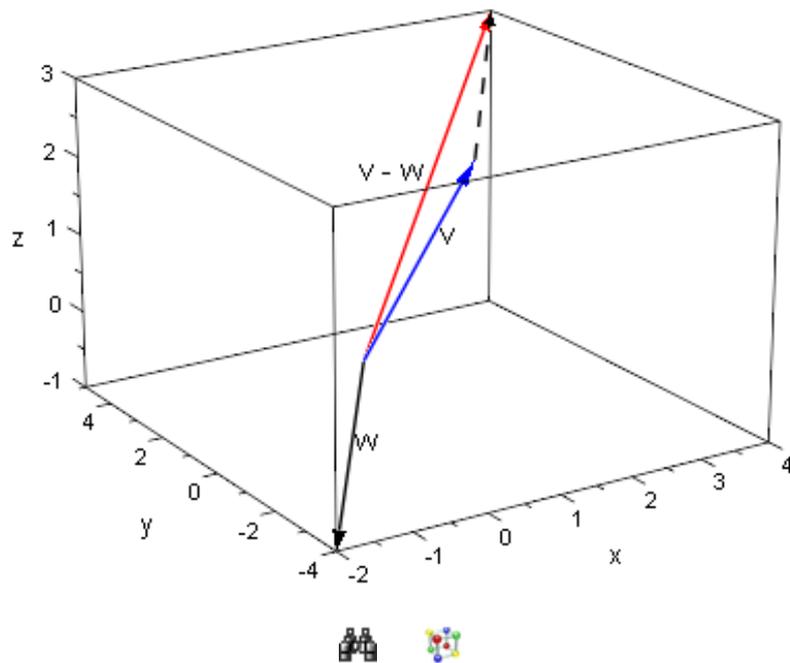
`k * v`

$$\begin{pmatrix} k v_1 \\ k v_2 \\ k v_3 \end{pmatrix}$$

Die geometrische Interpretation der Addition bzw. Subtraktion von Vektoren sowie der skalaren Multiplikation ist die gleiche, wie schon bei Vektoren des  $\mathbb{R}^2$ : Die Summe  $\vec{a} + \vec{b}$  erhält man geometrisch, indem man den Ortsvektor  $\vec{b}$  an die Spitze des Ortsvektors  $\vec{a}$  parallel verschiebt. Der Vektor  $\vec{a} + \vec{b}$  ist dann gegeben durch denjenigen Vektor, der im Ursprung beginnt und an der Spitze des parallel verschobenen Vektors  $\vec{b}$  endet. Zur Berechnung von  $\vec{a} - \vec{b}$  verfährt man analog, jedoch verschiebt man statt  $\vec{b}$  seinen Gegenvektor  $-\vec{b}$  parallel an die Spitze des Ortsvektors  $\vec{a}$ . Die Multiplikation  $k \cdot \vec{a}$  mit dem Skalar  $k \in \mathbb{R}$  bedeutet Streckung des Vektors  $\vec{a}$  mit dem Faktor  $k$ . Dabei kehrt sich für negatives  $k$  seine Richtung um. Die vier "Einzelsituationen" sind in Abbildung 2.4, 2.5, 2.6 und 2.7 dargestellt.

Für die Addition bzw. Subtraktion und die skalare Multiplikation von Vektoren mit drei Komponenten gelten die gleichen Rechenregeln wie wir sie schon für Vektoren des  $\mathbb{R}^2$  im vorhergehenden Abschnitt nachgerechnet hatten, d.h.

- (i) **Existenz des "Nullvektors"**: Für den Vektor  $\vec{0}$ , dessen Komponenten nur aus Nullen bestehen und für alle Vektoren  $\vec{a}$  gilt:  $\vec{a} + \vec{0} = \vec{a}$ .
- (ii) **Kommutativgesetz**:  $\vec{a} + \vec{b} = \vec{b} + \vec{a}$  für alle Vektoren  $\vec{a}$  und  $\vec{b}$ .
- (iii) **Assoziativgesetz**:  $(\vec{a} + \vec{b}) + \vec{c} = \vec{a} + (\vec{b} + \vec{c})$  für alle Vektoren  $\vec{a}$ ,  $\vec{b}$  und  $\vec{c}$ .

Abbildung 2.4: Addition zweier Vektoren im  $\mathbb{R}^3$ Abbildung 2.5: Subtraktion zweier Vektoren im  $\mathbb{R}^3$

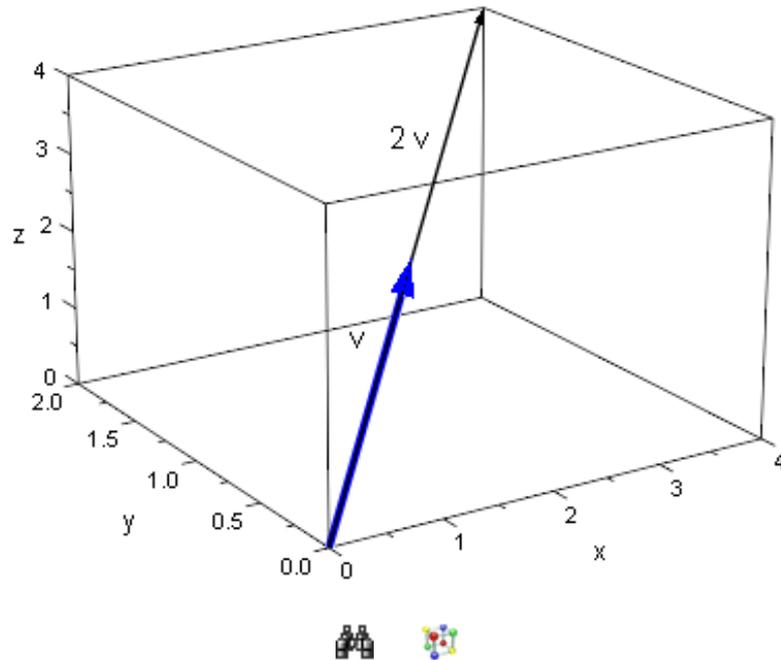


Abbildung 2.6: Multiplikation eines Vektors mit einem positiven Skalar

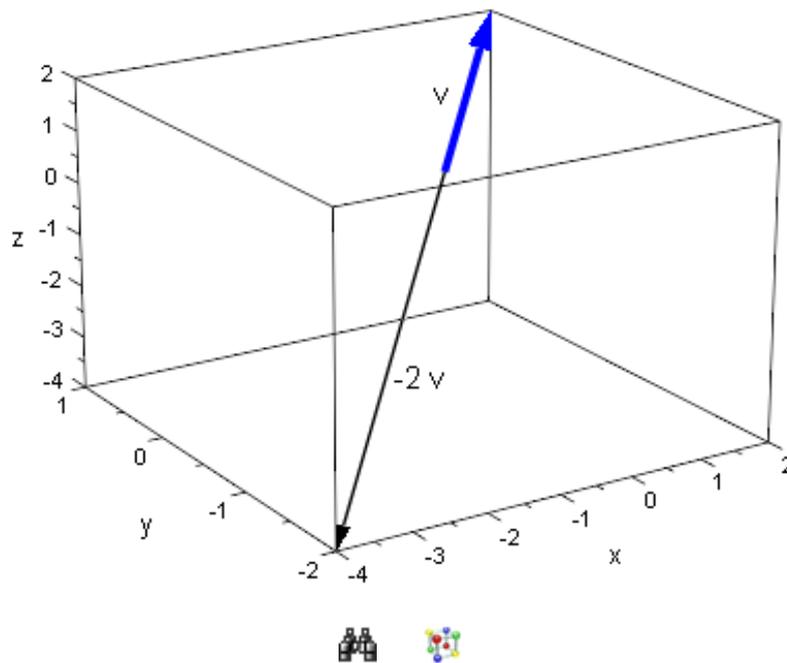


Abbildung 2.7: Multiplikation eines Vektors mit einem negativen Skalar

(iv) **Distributivgesetz:**  $k \cdot (\vec{a} + \vec{b}) = k \cdot \vec{a} + k \cdot \vec{b}$  und  $(k + l) \cdot \vec{a} = k \cdot \vec{a} + l \cdot \vec{a}$  für alle  $k, l \in \mathbb{R}$  und alle Vektoren  $\vec{a}$  und  $\vec{b}$ .

(v) **Assoziativgesetz:**  $(k \cdot l) \cdot \vec{a} = k \cdot (l \cdot \vec{a})$  für alle  $k, l \in \mathbb{R}$  und alle Vektoren  $\vec{a}$  und  $\vec{b}$ .

Die Menge aller Vektoren mit drei Komponenten bildet wieder einen Vektorraum, den wir den  $\mathbb{R}^3$  (gesprochen: R drei) nennen.

### Übungen

3.) Beweisen Sie (i) bis (v) für allgemeine Vektoren `matrix([a1,a2,a3])`, `matrix([b1,b2,b3])` und `matrix([c1,c2,c3])` sowie allgemeine Skalare `k` und `l` mit Hilfe von MuPAD.

## 2.4 Länge von Vektoren

### Neue MuPAD-Funktionen

<code>plot::Line2d</code>	Strecken in 2D
<code>sqrt</code>	Quadratwurzel einer Zahl
<code>v[i]</code>	<i>i</i> -te Komponente des Vektors <code>v</code>
<code>assume</code>	Eigenschaften an Variablen zuweisen
<code>plot::Line3d</code>	Strecken in 3D

Langsam aber sicher kommen wir zu den ersten geometrischen Anwendungen. In diesem Abschnitt wollen wir uns überlegen, wie man den Abstand zweier Punkte im Raum berechnen kann. Dazu bedarf es zunächst des Begriffs der *Länge eines Vektors*. Im  $\mathbb{R}^2$  können wir die Länge eines Vektors

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

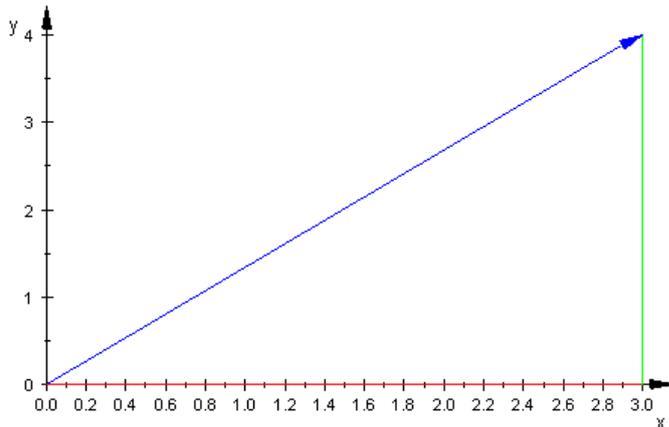
ganz einfach mit Hilfe des *Satzes von Pythagoras* berechnen. Betrachten wir das Beispiel des Vektors

- `v := matrix([3,4])`  

$$\begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

Wir stellen die Situation grafisch mit MuPAD dar (mit Hilfe der Funktion `plot::Line2d` lassen sich Strecken im Zweidimensionalen zeichnen):

- `OrtsvektorV := plot::Arrow2d(v):`  
`xAbschnitt := plot::Line2d([0,0], [3,0], Color = RGB::Red):`  
`yAbschnitt := plot::Line2d([3,0], [3,4], Color = RGB::Green):`  
`plot(OrtsvektorV, xAbschnitt, yAbschnitt)`



Der  $x$ -Abschnitt (in roter Farbe) und der  $y$ -Abschnitt (in grüner Farbe) bilden einen rechten Winkel. Wir können also die obige Figur als rechtwinkliges Dreieck ansehen, dessen Katheten durch den  $x$ -Abschnitt und den  $y$ -Abschnitt gegeben sind und dessen Hypotenuse der Vektor  $\vec{v}$  ist. Die Länge der Katheten beträgt 3 bzw. 4 Längeneinheiten, d.h. die Längen der Katheten sind durch die erste bzw. die zweite Komponente von  $\vec{v}$  gegeben. Damit beträgt die Länge von  $\vec{v}$ :

- `sqrt(v[1]^2 + v[2]^2)`  
5

Das Beispiel zeigt uns unmittelbar, wie wir die Länge eines allgemeinen Vektors

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

berechnen können – sie ist nach dem Satz von Pythagoras gegeben durch  $\sqrt{v_1^2 + v_2^2}$ . Die Kurzschreibweise für die Länge des Vektors  $\vec{v}$  ist im allgemeinen  $|\vec{v}|$  (gesprochen: Betrag von  $\vec{v}$ ). Man nennt  $|\vec{v}|$  auch die *Euklidische Norm* oder *2-Norm* des Vektors  $\vec{v}$ .

In MuPAD heißt die Funktion zur Berechnung der Länge eines Vektors `norm`<sup>2</sup>. Die Funktion `norm` erhält als zweites Argument stets die Zahl 2 (für 2-Norm). Bei der folgenden Berechnung weisen wir die Komponenten `v1` und `v2` zunächst explizit als reelle Zahlen aus<sup>3</sup>:

<sup>2</sup>Die Namensgebung `norm` für die “Länge” stammt aus der höheren Mathematik. Dort bezeichnet man die Länge eines mathematischen Objekts als seine “Norm”. Es gibt viele verschiedene Längenbegriffe in der Mathematik, so dass die Funktion `norm` stets ein weiteres Argument erhält, das angibt, welche Art von Längenbegriff einer Berechnung zugrunde gelegt werden soll. Im Fall der Länge eines Vektors im Sinne unserer Definition ist das nötige Zusatzargument für die Funktion `norm` die Zahl 2 für 2-Norm.

<sup>3</sup>Die Annahme, dass `v1` und `v2` reellwertig sind, ist notwendig, da MuPAD (wie auch andere Computeralgebra Systeme) stets über dem allgemeinsten Zahlbereich rechnet. Dieser ist durch die Menge der komplexen Zahlen gegeben, die man in der Schule i.a. nicht kennen lernt. Wird die Annahme `assume(v1, v2, Type::Real)` nicht getroffen, so taucht das “Konjugiert-Komplexe” von `v1` und `v2` bei der Berechnung der Länge auf.

- `assume({v1, v2}, Type::Real):`  
`v:= matrix([v1,v2]):`  
`norm(v, 2)`  
 $\sqrt{v_1^2 + v_2^2}$

Jetzt müssen wir unsere Beobachtungen für den zweidimensionalen Fall wieder für den  $\mathbb{R}^3$  verallgemeinern. Ein Vektor des  $\mathbb{R}^3$  ist stets von der Form

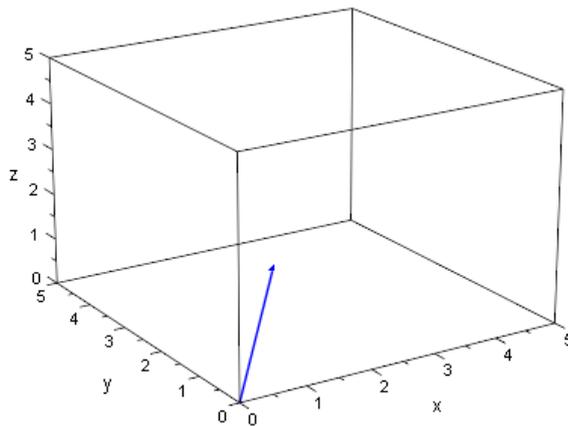
$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

Wenn wir die Länge dieses Vektors berechnen wollen, dann müssen wir im Prinzip des Satz des Pythagoras nur zweimal anwenden. Schauen wir uns zuerst wieder ein Beispiel an. Sicherlich können wir die Länge des Vektors

- `v:= matrix([3,4,0])`  
 $\begin{pmatrix} 3 \\ 4 \\ 0 \end{pmatrix}$

bestimmen, denn seine  $z$ -Komponente ist 0. Damit ist  $\mathbf{v}$  zwar ein Vektor im Raum, aber er liegt in der  $x$ - $y$ -Ebene, d.h. im Prinzip ist  $\mathbf{v}$  ein Vektor des  $\mathbb{R}^2$ :

- `OrtsvektorV:= plot::Arrow3d(v):`  
`plot(OrtsvektorV, ViewingBox = [0..5, 0..5, 0..5])`



Die Länge von  $\mathbf{v}$  ergibt sich also genau so, wie wir die Länge von Vektoren im  $\mathbb{R}^2$  bestimmt haben, d.h.:

- `sqrt(v[1]^2 + v[2]^2)`  
 $5$

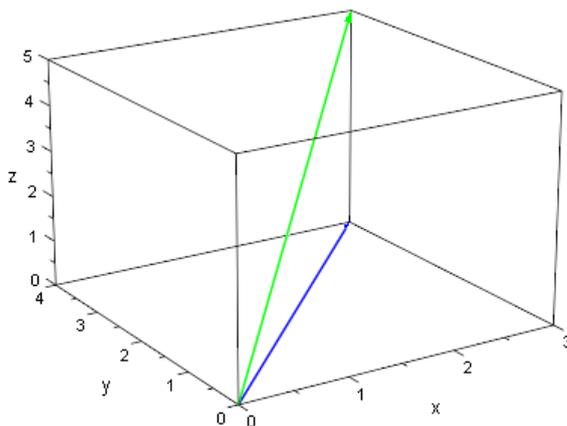
Ein kleines bisschen schwieriger wird die Berechnung der Länge eines Vektors, dessen  $x$ -,  $y$ - und  $z$ -Komponenten alle verschieden von 0 sind. Als Beispiel berechnen wir die Länge des Vektors

- `w:= matrix([3,4,5])`

$$\begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$$

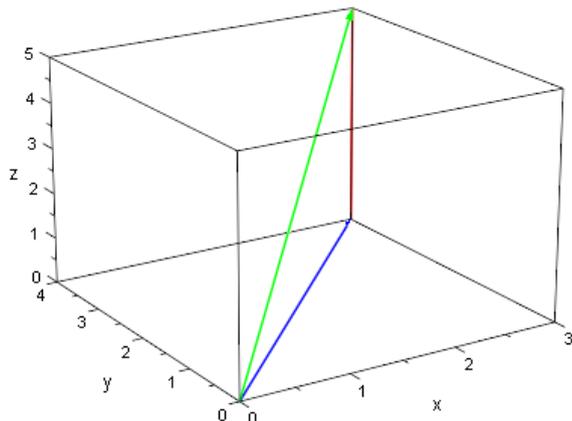
Wir stellen uns vor, der Vektor  $w$  werfe einen Schatten auf den Boden des Raums. Diesen Schatten erhalten wir, indem wir einfach die  $z$ -Komponente von  $w$  auf den Wert 0 setzen. Auf diese Weise erhalten wir genau den Vektor  $v$ , den wir oben schon in MuPAD definiert hatten.

- `OrtsvektorW:= plot::Arrow3d(w, Color = RGB::Green):`  
`plot(OrtsvektorV, OrtsvektorW)`



Verbinden wir nun die Spitzen der beiden eingezeichneten Vektoren, so ergibt sich wieder ein rechtwinkliges Dreieck, dessen eine Kathete durch den Vektor  $v$  und dessen andere Kathete durch die schwarze eingezeichnete Linie gegeben sind (Linien im Dreidimensionalen zeichnet man mit MuPAD über den Befehl `plot::Line3d`):

- `plot(OrtsvektorV, OrtsvektorW,`  
`plot::Line3d([3,4,0], [3,4,5], Color = RGB::Red))`



Die Länge der roten Strecke ist durch die  $z$ -Komponente des Vektors  $\vec{w}$  gegeben und beträgt daher 5 Längeneinheiten. Die Länge des Vektors  $\vec{w}$  (in blauer grüner Farbe eingezeichnet) berechnet sich also mit dem Satz des Pythagoras zu

- $\text{sqrt}(\text{sqrt}(3^2 + 4^2)^2 + 5^2)$   
 $5\sqrt{2}$

Unser konkretes Beispiel gibt unmittelbar Anlass zu der allgemeinen Formel zur Berechnung der Länge eines Vektors

$$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$$

im  $\mathbb{R}^3$ . Wir müssen uns dazu nur noch einmal genau anschauen, wie wir die Länge des konkreten Vektors  $\vec{w}$  mit MuPAD berechnet hatten. Sie ergab sich oben aus der Rechnung  $\text{sqrt}(\text{sqrt}(3^2 + 4^2)^2 + 5^2)$ <sup>4</sup>. Setzen wir für die konkreten Zahlen 3, 4 und 5, die durch die  $x$ -,  $y$ - und  $z$ -Komponenten des Vektors  $\vec{w}$  gegeben waren, allgemeine Komponenten  $w_1$ ,  $w_2$  und  $w_3$  ein, so erhalten wir als Länge eines Vektors im  $\mathbb{R}^3$ :

- $\text{sqrt}(\text{sqrt}(w_1^2 + w_2^2)^2 + w_3^2)$   
 $\sqrt{w_1^2 + w_2^2 + w_3^2}$

d.h. die Formel zur Berechnung der Länge eines Vektors im  $\mathbb{R}^3$  ist ganz analog zu der Formel, die wir zur Berechnung der Länge eines Vektors im  $\mathbb{R}^2$  gefunden hatten – wir müssen lediglich das Quadrat der  $z$ -Komponente mit unter die Wurzel schreiben. Kurz und knapp formuliert gilt also:

$$|\vec{w}| = \sqrt{w_1^2 + w_2^2 + w_3^2}.$$

<sup>4</sup> `sqrt` steht für "square root" und dient zur Berechnung der Quadratwurzel einer Zahl.

## Übungen

4.) Berechnen Sie die Länge der folgenden drei Vektoren des  $\mathbb{R}^2$ :

$$\vec{u} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \quad \vec{v} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

5.) Berechnen Sie die Länge der folgenden drei Vektoren des  $\mathbb{R}^3$ :

$$\vec{u} = \begin{pmatrix} -5 \\ -4 \\ 2 \end{pmatrix}, \quad \vec{v} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix}.$$

### 2.4.1 Abstand zweier Punkte

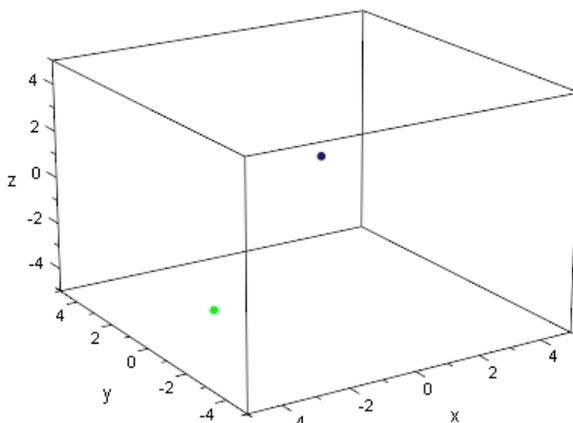
Wir können mit Hilfe des Längenbegriffs für Vektoren im  $\mathbb{R}^2$  und  $\mathbb{R}^3$  nun den Abstand zwischen zwei Punkten im Zweidimensionalen ebenso wie im Dreidimensionalen berechnen. Im folgenden beschäftigen wir uns nur mit dem Fall, dass der Abstand zweier Punkte  $P(p_1|p_2|p_3)$  und  $Q(q_1|q_2|q_3)$  im Raum berechnet werden soll. Die Abstandsberechnung zweier Punkte im Zweidimensionalen macht man im Prinzip genau so wie im dreidimensionalen Fall.

Gegeben seien die beiden Punkte  $P(1|1|1)$  und  $Q(-3|0|-4)$ , die wir zunächst mit MuPAD in ein gemeinsames Koordinatensystem zeichnen:

- ```

P:= plot::Point3d([1,1,1]):
Q:= plot::Point3d([-3,0,-4], Color = RGB::Green):
plot(P, Q, ViewingBox = [-5..5, -5..5, -5..5])

```

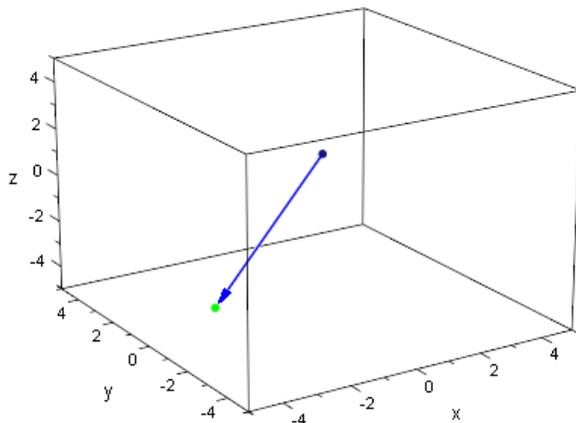


Wie kann man nun den Abstand von **P** und **Q** berechnen? Der Abstand der beiden Punkte entspricht offensichtlich genau der Länge des Vektors zwischen den beiden Punkte:

- ```

VektorPQ:= plot::Arrow3d([1,1,1], [-3,0,-4]):
plot(P, Q, VektorPQ, ViewingBox = [-5..5, -5..5, -5..5])

```



Wir müssen also nur die Länge des in blauer Farbe eingezeichneten Vektors bestimmen. Geometrisch ist der blaue Vektor genau der Differenzvektor der Vektoren

- `OrtsvektorP:= matrix([1,1,1]);`  
`OrtsvektorQ:= matrix([-3,0,-4])`

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

$$\begin{pmatrix} -3 \\ 0 \\ -4 \end{pmatrix}$$

Geometrisch ist der blaue Vektor ja nichts anderes, als der “Weg” von  $P$  nach  $Q$ . Man gelangt aber von  $P$  nach  $Q$ , indem man von  $P$  aus den zugehörigen Ortsvektor in entgegengesetzter Richtung zum Ursprung  $(0|0|0)$  zurück läuft und dann den Ortsvektor zum Punkt  $Q$  in Pfeilrichtung vom Ursprung  $(0|0|0)$  hin zum Punkt  $Q$  folgt (siehe auch Abbildung 2.8).

Damit ist also der Abstand von  $P$  und  $Q$  gegeben durch

- `norm(OrtsvektorP - OrtsvektorQ, 2)`

$$\sqrt{42}$$

Ganz allgemein können wir für die Berechnung des Abstands zweier allgemeiner Punkte  $P(p_1|p_2|p_3)$  und  $Q(q_1|q_2|q_3)$  mit MuPAD die folgende Formel angeben:

- `assume({p1,p2,p3,q1,q2,q3}, Type::Real):`  
`OrtsvektorP:= matrix([p1,p2,p3]):`  
`OrtsvektorQ:= matrix([q1,q2,q3]):`  
`AbstandPQ:= norm(OrtsvektorP - OrtsvektorQ, 2)`

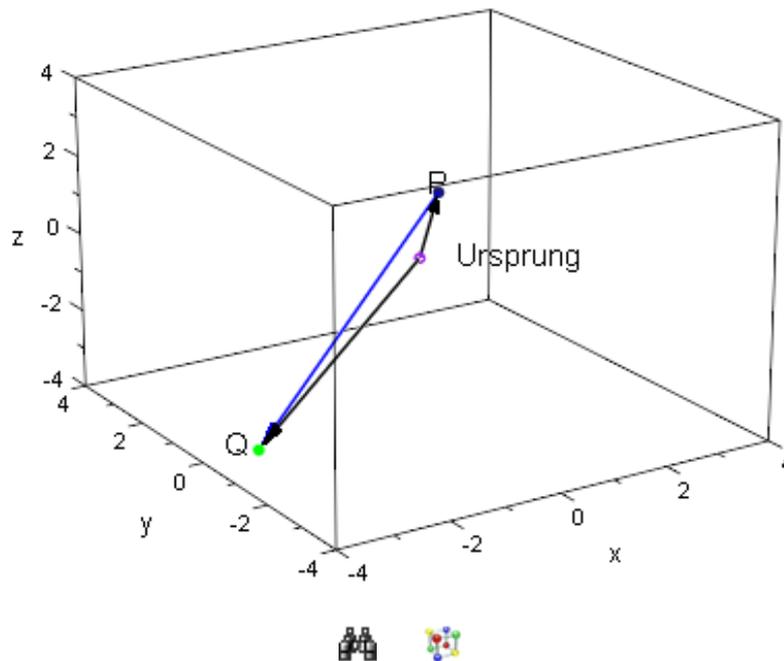


Abbildung 2.8: Der "Weg" von P nach Q

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}$$

### Übungen

- 6.) Zeichnen Sie die beiden Punkte  $P(1|5|-2)$  und  $Q(-1|1|-1)$  in gemeinsames Koordinatensystem. Zeichnen Sie den Vektor von  $P$  nach  $Q$  sowie die Ortsvektoren der Punkte  $P$  und  $Q$  mit in das Koordinatensystem. Berechnen Sie dann den Abstand der beiden Punkte.
- 7.) Finden Sie mit MuPAD eine Formel, mit deren Hilfe sich allgemein der Abstand eines Punkte  $P(p_1|p_2|p_3)$  mit  $p_1, p_2, p_3 \in \mathbb{R}$  vom Koordinatenursprung berechnen lässt.
- 8.) Finden Sie mit MuPAD eine Formel, mit der sich der Abstand zweier allgemeiner Punkte  $P(p_1|p_2)$  und  $Q(q_1|q_2)$  im  $\mathbb{R}^2$  mit  $p_1, p_2, q_1, q_2 \in \mathbb{R}$  berechnen lässt.

## 2.5 Standardskalarprodukt und Winkel zwischen Vektoren

### Neue MuPAD-Funktionen

<code>linalg::scalarProduct</code>	Standardskalarprodukt von zwei Vektoren
<code>linalg::angle</code>	Winkel zwischen zwei Vektoren
<code>float</code>	Näherungsweise Ausgabe eines exakten Ergebnisses auf 10 signifikante Stellen genau

Nachdem wir die Länge eines Vektors bestimmen können, können wir uns nun daran machen, Winkel zwischen zwei Vektoren zu berechnen. Um später die Formel zur Berechnung des Winkels zwischen zwei Vektoren  $\vec{v}$  und  $\vec{w}$  des  $\mathbb{R}^2$  oder des  $\mathbb{R}^3$  eleganter schreiben zu können, für wir das so genannte *Standardskalarprodukt* für Vektoren des  $\mathbb{R}^2$  oder des  $\mathbb{R}^3$  ein. Für

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad \text{und} \quad \vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$$

ist das (*Standard-*)Skalarprodukt  $\vec{v} * \vec{w}$  gegeben durch

$$\vec{v} * \vec{w} = v_1 \cdot w_1 + v_2 \cdot w_2 + v_3 \cdot w_3,$$

d.h. wir multiplizieren die  $x$ -Komponenten und die  $y$ -Komponenten von  $\vec{v}$  und  $\vec{w}$  miteinander und addieren die beiden Produkte (für den Fall von Vektoren des  $\mathbb{R}^2$  mit zwei Komponenten kann man in der obigen Formel die  $z$ -Komponenten  $v_3$  und  $w_3$  einfach weglassen und erhält die Definition des Skalarprodukts von Vektoren des  $\mathbb{R}^2$ ). MuPAD bietet die Funktion `linalg::scalarProduct` zur Berechnung des Skalarprodukts zweier Vektoren, die im Rahmen der Bibliothek `linalg` zur "Linearen Algebra" bereitsteht<sup>5</sup>:

- `assume({v1,v2,v3,w1,w2,w3}, Type::Real):`  
`v:= matrix([v1,v2,v3]):`  
`w:= matrix([w1,w2,w3]):`  
`linalg::scalarProduct(v,w)`

<sup>5</sup>Auch bei der Verwendung von `linalg::scalarProduct` ist es wichtig, dass bei Vektoren mit allgemeinen Komponenten – z.B. `matrix([v1,v2])` für allgemeine Bezeichner `v1` und `v2` – angenommen wird, dass diese reellwertig sind (dies ist möglich mit Hilfe der Funktion `assume`). Andernfalls geht MuPAD davon aus, dass die Komponenten auch komplexe Zahlen sein können. Allgemein wird in der Mathematik das Skalarprodukt zweier Vektoren  $\vec{v}$  und  $\vec{w}$  definiert als  $\sum_{i=1}^n v_i \cdot \bar{w}_i$ , wobei  $\vec{v}$  und  $\vec{w}$   $n$  Komponenten haben und der Querbalken über  $\vec{w}$  wieder das "Konjugiert-Komplexe" bezeichnet. Diese allgemeine Art und Weise Skalarprodukte zu definieren ist im Rahmen von Computeralgebra-Systemen üblich.

$$v_1 w_1 + v_2 w_2 + v_3 w_3$$

Wie oben erwähnt soll uns das Skalarprodukt helfen, eine Formel für den Winkel zwischen zwei Vektoren im  $\mathbb{R}^2$  bzw. im  $\mathbb{R}^3$  zu finden. Die folgenden Betrachtungen gelten sowohl für Vektoren des  $\mathbb{R}^2$  als auch für Vektoren des  $\mathbb{R}^3$ . Wir haben die folgende Ausgangssituation, die in Abbildung 2.9 dargestellt ist (der Einfachheit halber haben wir die Darstellung im Zweidimensionalen gewählt, damit die Idee zur Berechnung des Winkels zwischen den Vektoren  $\vec{v}$  und  $\vec{w}$  besser zum Vorschein kommt):

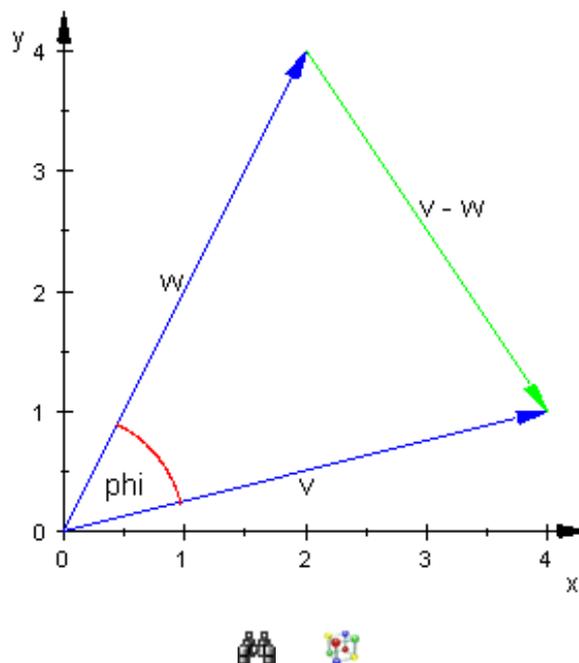


Abbildung 2.9: Der Winkel zwischen zwei Vektoren

Wir haben es also im Prinzip mit einem Dreieck zu tun, bei dem die die Längen der Seiten einfach durch die Längen der entsprechenden Vektoren, also durch  $|\vec{v}|$ ,  $|\vec{w}|$  und  $|\vec{v} - \vec{w}|$ , gegeben sind. Stellen wir uns auf den Standpunkt, dass  $\vec{v}$  und  $\vec{w}$  bekannt sind, dann können wir die Längen der Vektoren nach dem im vorhergehenden Abschnitt behandelten Verfahren berechnen. Also sind die Längen der Dreiecksseiten bekannt.

Jetzt müssen wir uns “nur noch” an den Mittelstufenunterricht erinnern, wo auch der „Kosinussatz“ im Rahmen der Trigonometrie behandelt wurde. Angewandt auf die obige Situation gilt:

$$|\vec{v} - \vec{w}|^2 = |\vec{v}|^2 + |\vec{w}|^2 - 2 \cdot |\vec{v}| \cdot |\vec{w}| \cdot \cos(\varphi).^6$$

<sup>6</sup>Wenn man in dieser Formel für  $\varphi 90^\circ$  bzw. im Bogenmaß  $\pi/2$  einsetzt, so gilt  $\cos(\pi/2) = 0$  und es bleibt die Gleichung  $|\vec{v} - \vec{w}|^2 = |\vec{v}|^2 + |\vec{w}|^2$  übrig, die wir alle als *Satz des Pythagoras* kennen.

Stellen wir die Gleichung nach  $\cos(\varphi)$  um, so erhalten wir

$$\cos(\varphi) = \frac{|\vec{v} - \vec{w}|^2 - |\vec{v}|^2 - |\vec{w}|^2}{-2 \cdot |\vec{v}| \cdot |\vec{w}|}.$$

Ab jetzt wird es unangenehm, wenn man per Hand weiter rechnet, denn wir erinnern uns, dass die Längen der Vektoren durch die Wurzelausdrücke  $|\vec{v}| = \sqrt{v_1^2 + v_2^2 + v_3^2}$ ,  $|\vec{w}| = \sqrt{w_1^2 + w_2^2 + w_3^2}$  bzw.  $|\vec{v} - \vec{w}| = \sqrt{(v_1 - w_1)^2 + (v_2 - w_2)^2 + (v_3 - w_3)^2}$  gegeben sind. Diese Ausdrücke müssen wir jetzt in die letzte Gleichung an den entsprechenden Stellen einsetzen. Die lästigen und zeitraubenden Umformungen lassen wir MuPAD erledigen, indem wir zuerst die rechte Seite der letzten Gleichung in MuPAD definieren

- `RechteSeite := (norm(v - w, 2)^2 - norm(v, 2)^2 - norm(w, 2)^2) / (-2 * norm(v, 2) * norm(w, 2))`

$$\frac{v_1^2 - (v_2 - w_2)^2 - (v_3 - w_3)^2 - (v_1 - w_1)^2 + v_2^2 + v_3^2 + w_1^2 + w_2^2 + w_3^2}{2 \sqrt{v_1^2 + v_2^2 + v_3^2} \sqrt{w_1^2 + w_2^2 + w_3^2}}$$

und sie anschließend mit Hilfe der Funktion `normal` vereinfachen lassen

- `normal(RechteSeite)`

$$\frac{v_1 w_1 + v_2 w_2 + v_3 w_3}{\sqrt{v_1^2 + v_2^2 + v_3^2} \sqrt{w_1^2 + w_2^2 + w_3^2}}$$

So weit, so gut! Damit haben wir also die Bestimmungsgleichung

$$\cos(\varphi) = \frac{v_1 w_1 + v_2 w_2 + v_3 w_3}{\sqrt{v_1^2 + v_2^2 + v_3^2} \sqrt{w_1^2 + w_2^2 + w_3^2}}$$

für den Winkel  $\varphi$  zwischen  $\vec{v}$  und  $\vec{w}$ . Schauen wir die rechte Seite scharf an, so entdecken wir die folgenden "bekannten Ausdrücke"

- `linalg::scalarProduct(v, w)`

$$v_1 w_1 + v_2 w_2 + v_3 w_3$$

im Zähler des Bruchs sowie das Produkt

- `norm(v, 2) * norm(w, 2)`

$$\sqrt{v_1^2 + v_2^2 + v_3^2} \sqrt{w_1^2 + w_2^2 + w_3^2}$$

im Nenner, d.h. unser Bruch besteht aus den "Einzelteilen"

$$\cos(\varphi) = \frac{\overbrace{v_1 w_1 + v_2 w_2 + v_3 w_3}^{=\vec{v} \cdot \vec{w}}}{\underbrace{\sqrt{v_1^2 + v_2^2 + v_3^2}}_{=|\vec{v}|} \underbrace{\sqrt{w_1^2 + w_2^2 + w_3^2}}_{=|\vec{w}|}}.$$

Also gilt: Der Winkel  $\varphi$  zwischen den zwei allgemeinen Vektoren  $\vec{v}$  und  $\vec{w}$  ist gegeben durch

$$\cos(\varphi) = \frac{\vec{v} * \vec{w}}{|\vec{v}| \cdot |\vec{w}|}.$$

MuPAD bietet im Rahmen der -Bibliothek auch eine Funktion zur Berechnung des Winkels zwischen zwei Vektoren. Sie heißt `linalg::angle` und erhält als Argumente stets zwei Vektoren. Der Winkel zwischen den beiden Vektoren

- `v:= matrix([1,2,1]):`  
`w:= matrix([-2,1,5]):`

kann daher mit MuPAD auch direkt über den Befehl

- `linalg::angle(v,w)`

$$\arccos\left(\frac{\sqrt{30}\sqrt{6}}{36}\right)$$

berechnet werden. Das Ergebnis zeigt einmal wieder ganz deutlich, dass MuPAD bei der Eingabe exakter Daten (die Komponenten der Vektoren sind nicht etwa Gleitkommazahlen) stets auch exakt weiter rechnet. Die Funktion `arccos` ist die "Umkehrfunktion" des Kosinus<sup>7</sup>. Um eine Näherung des Ergebnisses zu erhalten schreibt man einfach den Befehl `float` um den Aufruf der Funktion `linalg::angle` herum:

- `float(linalg::angle(v,w))`  
1.188903079

MuPAD liefert uns den Winkel im Bogenmaß. Um diesen Grad umzurechnen, müssen wir das obige Ergebnis mit 180 multiplizieren und durch  $\pi$  dividieren (die Zahl  $\pi$  ist in MuPAD als `PI` verfügbar):

- `float(180 * linalg::angle(v,w)/PI)`  
68.11912865

Der Winkel zwischen unseren beiden Vektoren `v` und `w` beträgt also etwa  $68^\circ$ .

## Übungen

- 9.) Berechnen Sie den Winkel zwischen den beiden Vektoren

$$\vec{v} = \begin{pmatrix} 2 \\ 4 \\ -2 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} 1 \\ -2 \\ 12 \end{pmatrix}.$$

<sup>7</sup>Die Ausgabe von MuPAD beschreibt im wesentlichen das, was Schülerinnen und Schüler rechnen, wenn Sie mit einem gewöhnlichen Taschenrechner den Winkel  $\varphi$  im Bogenmaß bestimmen; auf Taschenrechnern gibt es häufig die Taste  $\cos^{-1}$ . Diese Funktion wird dann auf den Ausdruck  $\frac{\sqrt{30}\sqrt{6}}{36}$  angewandt, der zuvor per Hand über die schulübliche Formel berechnet wurde.

- 10.) Beweisen Sie unter Zuhilfenahme von MuPAD für allgemeine Vektoren  $\vec{v}$  und  $\vec{w}$  des  $\mathbb{R}^3$  die "Kommutativität des Standardskalarprodukts":

$$\vec{v} * \vec{w} = \vec{w} * \vec{v}.$$

- 11.) Beweisen Sie unter Zuhilfenahme von MuPAD für allgemeine Vektoren  $\vec{v}$  und  $\vec{w}$  des  $\mathbb{R}^3$ , dass diese orthogonal sind genau dann, wenn ihr Skalarprodukt  $\vec{v} * \vec{w}$  gleich 0 ist (Zur Erinnerung:  $90^\circ$  entsprechen  $\pi/2$  im Bogenmaß!):

$$\vec{v} \perp \vec{w} \iff \vec{v} * \vec{w} = 0.$$

## 2.6 Kreuzprodukt von Vektoren

### Neue MuPAD-Funktionen

<code>solve</code>	Lösen von Gleichungen und Gleichungssystemen
<code>\$</code>	Folgenoperator
<code>in</code>	Operator in Verbindung mit dem Folgenoperator <code>\$</code> zum Durchlaufen der Elemente einer Menge oder Liste
<code>subs</code>	Einsetzen von Werten in Variablen
<code>linalg::crossProduct</code>	Kreuzprodukt von zwei Vektoren im $\mathbb{R}^3$

In diesem vorletzten Abschnitt des zweiten Kapitels sammeln wir das letzte nötige "Handwerkszeug", das uns noch fehlt, um endlich zu den uns interessierenden geometrischen Anwendungen zu kommen. Später werden wir immer wieder zu zwei gegebenen Vektoren  $\vec{v}$  und  $\vec{w}$  des  $\mathbb{R}^3$  einen Vektor finden müssen, der senkrecht auf beiden steht. Eine Möglichkeit, einen solchen Vektor zu finden, zeigt die letzte Übungsaufgabe des vorhergehenden Abschnitts auf. Dort war mit MuPAD zu zeigen, dass zwei Vektoren genau dann orthogonal sind (d.h. senkrecht aufeinander stehen), wenn für ihr Skalarprodukt gilt:  $\vec{v} * \vec{w} = 0$ . Dem Ansatz des letzten Kapitels folgend wollen wir einen Vektor finden, der orthogonal zu den beiden Vektoren

$$\vec{v} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad \text{und} \quad \vec{w} = \begin{pmatrix} -2 \\ 3 \\ -2 \end{pmatrix}$$

ist. Wir suchen also einen Vektor

- `n:= matrix([n1, n2, n3])`

$$\begin{pmatrix} n1 \\ n2 \\ n3 \end{pmatrix}$$

mit der Eigenschaft, dass für die Skalarprodukte mit den beiden gegebenen Vektoren  $\vec{v}$  und  $\vec{w}$  gilt:

- `assume({n1, n2, n3}, Type::Real):`  
`v:= matrix([1,2,1]):`  
`w:= matrix([-2,3,-2]):`  
`Gleichung1:= linalg::scalarProduct(v,n) = 0;`  
`Gleichung2:= linalg::scalarProduct(w,n) = 0;`  
 $n1 + 2n2 + n3 = 0$

$$3n2 - 2n1 - 2n3 = 0$$

Was wir erhalten ist also ein Gleichungssystem mit zwei Gleichungen, aber drei Unbekannten. Dieses lösen wir mit MuPAD mit Hilfe der Funktion `solve`<sup>8</sup>:

- `solve({Gleichung1, Gleichung2}, {n1, n2, n3})`  
`{[n1 = -n3, n2 = 0]}`

MuPAD teilt uns mit, dass  $n2 = 0$  und  $n1 = -n3$  gelten muss. Damit steht der Vektor

- `n:= matrix([-n3,0,n3])`

$$\begin{pmatrix} -n3 \\ 0 \\ n3 \end{pmatrix}$$

senkrecht auf unseren beiden speziellen Vektoren  $\vec{v}$  und  $\vec{w}$ . Wir überprüfen das zur Sicherheit noch einmal:

- `linalg::scalarProduct(v, n);`  
`linalg::scalarProduct(w, n);`  
 $0$   
 $0$

Die obige Lösung deutet an, dass der gesuchte und gefundene Vektor  $\vec{n}$  keinesfalls eindeutig bestimmt ist. Wir können für  $n3$  jede reelle Zahl einsetzen und erhalten stets

<sup>8</sup>Die Funktion `solve` ist "die" Funktion in MuPAD, mit der sich Gleichungen exakt lösen lassen. Will man ein Gleichungssystem mit Hilfe von `solve` lösen, so gibt man zunächst die Menge der Gleichungen und anschließend die Menge der Variablen an, nach denen die Gleichungen gelöst werden sollen.

ein Vektor, der auf  $\vec{v}$  und  $\vec{w}$  senkrecht steht. Das ist auch geometrisch einleuchtend, denn wir können  $\vec{n}$  in der Form schreiben:

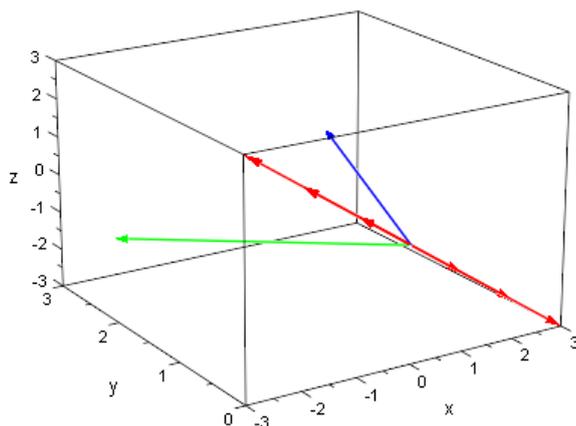
$$\vec{n} = \begin{pmatrix} -n3 \\ 0 \\ n3 \end{pmatrix} = n3 \cdot \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix},$$

d.h. je nach dem welchen Wert wir für  $n3$  einsetzen, wird der Vektor mit den Komponenten  $-1$ ,  $0$  und  $1$  entsprechend gestreckt bzw. zusätzlich in seiner Richtung umgekehrt. Zur Veranschaulichung zeichnen wir mit MuPAD die Vektoren  $\vec{v}$  und  $\vec{w}$  sowie den Vektor  $\vec{n}$  für einige Werte von  $n3$  in ein gemeinsames Koordinatensystem <sup>9</sup>:

```

• OrtsvektorV:= plot::Arrow3d(v, Color = RGB::Blue):
  OrtsvektorW:= plot::Arrow3d(w, Color = RGB::Green):
  OrtsvektorN:= plot::Arrow3d(subs(n, n3 = i),
                              Color = RGB::Red)
                              $ i in {-3,-2,-1,1,2,3};
plot(OrtsvektorV, OrtsvektorW, OrtsvektorN)

```



Die roten Vektoren sind sämtlich skalare Vielfach des Vektors mit den Komponenten  $-1$ ,  $0$  und  $1$  und stehen alle senkrecht sowohl auf dem Vektor  $\vec{v}$  (in blauer Farbe dargestellt) als auch auf dem Vektor  $\vec{w}$  (in grüner Farbe darstellt). Wie schon in den vorhergehenden Abschnitten wollen wir uns natürlich nicht damit zufrieden geben, für ein ganz spezielles Paar von zwei Vektoren  $\vec{v}$  und  $\vec{w}$  einen Vektor gefunden zu haben, der auf beiden senkrecht steht. Wir möchten natürlich am liebsten eine Formel oder Berechnungsvorschrift haben, die zu beliebigen verschiedenen Vektoren  $\vec{v}$  und  $\vec{w}$  stets einen Vektor liefert, der

<sup>9</sup>Hier benötigen wir zum Einsetzen ("Substituieren") der entsprechenden Werte  $-3, -2, -1, 1, 2, 3$  für  $n3$  in den Vektor  $\vec{n}$  den Befehl `subs`. `subs` erhält als erstes Argument den Ausdruck (in diesem Fall der Vektor  $\vec{n}$ ), in dem eine (oder auch mehrere) Variable durch einen Wert ersetzt werden, und als zweites Argument eine Gleichung, die angibt, durch welchen Wert die entsprechende Variable zu ersetzen ist. Im obigen Fall haben wir den Folgenoperator `$` in Kombination mit dem Befehl `in` verwendet, damit alle Werte der Menge  $\{-3, -2, -1, 1, 2, 3\}$  nacheinander für  $n3$  in den Vektor  $\vec{n}$  eingesetzt werden (es ist  $n3$  durch  $i$  zu ersetzen, wobei  $i$  alle Element der der Menge  $\{-3, -2, -1, 1, 2, 3\}$  durchläuft).

auf  $\vec{v}$  und  $\vec{w}$  senkrecht steht. Um eine solche Formel zu finden, müssen wir die obige Vorgehensweise nur für allgemeine Vektoren  $\vec{v}$  und  $\vec{w}$  des  $\mathbb{R}^3$  mit allgemeinen reellen Komponenten durchgehen. Zuerst müssen wir die allgemeinen Vektoren wieder in MuPAD definieren:

- `assume({v1,v2,v3,w1,w2,w3}, Type::Real):`  
`v:= matrix([v1,v2,v3]):`  
`w:= matrix([w1,w2,w3]):`

Wir suchen also einen Vektor

- `assume({n1,n2,n3}, Type::Real):`  
`n:= matrix([n1,n2,n3]):`

dessen Skalarprodukt sowohl mit  $\vec{v}$  als auch mit  $\vec{w}$  Null ist. Wie oben erhalten wir also ein zugegebenermaßen etwas unhandliches lineares Gleichungssystem in den Variablen  $n_1$ ,  $n_2$  und  $n_3$

- `Gleichung1:= linalg::scalarProduct(v,n) = 0;`  
`Gleichung2:= linalg::scalarProduct(w,n) = 0;`  
 $n_1 v_1 + n_2 v_2 + n_3 v_3 = 0$

$$n_1 w_1 + n_2 w_2 + n_3 w_3 = 0$$

das wir jetzt nach  $n_1$ ,  $n_2$  und  $n_3$  von MuPAD lösen lassen<sup>10</sup>(um später mit der Lösungsmenge weiterarbeiten zu können, weisen wir diese an die Variable **L** zu):

- `L:= solve({Gleichung1, Gleichung2}, {n1, n2, n3},`  
`IgnoreSpecialCases)`

$$\left\{ \left[ n_1 = \frac{n_3 v_2 w_3 - n_3 v_3 w_2}{v_1 w_2 - v_2 w_1}, n_2 = - \frac{n_3 v_1 w_3 - n_3 v_3 w_1}{v_1 w_2 - v_2 w_1} \right] \right\}$$

MuPAD liefert uns die Lösungen für  $n_1$  und  $n_2$  parametrisiert in Abhängigkeit von  $n_3$ , d.h. wir dürfen  $n_3$  frei vorwählen und können dann  $n_1$  und  $n_2$  gemäß dem vorgewählten Wert für  $n_3$  aus der Lösungsmenge ablesen. Der einfachste Wert, den wir für  $n_3$  wählen könnten, ist sicherlich Null. Allerdings folgt dann auch  $n_1 = 0$  und  $n_2 = 0$ , d.h.  $\vec{n}$  wäre der Nullvektor, der trivialerweise auf jedem Vektor senkrecht steht (das Skalarprodukt des Nullvektors mit einem beliebigen Vektor ist *immer* Null). Da wir aber trotzdem eine möglichst einfache Darstellung für die Komponenten des Vektors  $\vec{n}$  finden möchten, die von den Komponenten  $v_1$ ,  $v_2$  und  $v_3$  von  $\vec{v}$  sowie von den Komponenten  $w_1$ ,  $w_2$  und  $w_3$

<sup>10</sup>Die Zusatzoption `IgnoreSpecialCases` beim Aufruf der Funktion `solve` sollte hier unbedingt (!) verwendet werden. MuPAD berücksichtigt andernfalls bei der Lösung alle möglichen Fälle (Werden bestimmte Terme Null? Wenn ja, wie sieht die Lösung aus? Wie sieht die Lösung in dem Fall aus, wo sie nicht Null werden? etc.) Diese Fallunterscheidungen sind nützlich und mathematisch korrekt. Jedoch benötigen wir in dieser Situation nur die "allgemeinste" Lösung und verzichten auf alle Spezialfälle.

von  $\vec{w}$  abhängt, beobachten wir, dass beide Ausdrücke in der obigen Lösungsmenge für  $n_1$  und  $n_2$  im Zähler sowohl in jedem Summanden den Faktor  $n_3$  enthalten, als auch insgesamt den gleichen Nenner haben. Um eine möglichst einfache Darstellung für  $\vec{n}$  zu finden, wäre es auf jeden Fall wünschenswert, Brüche zu vermeiden. Wählen wir also für  $n_3$  den Wert  $v_1 \cdot w_2 - v_2 \cdot w_1$ , so kürzt sich in den Ausdrücken für  $n_1$  und  $n_2$  der Nenner heraus. Nachdem die Strategie klar ist, lassen wir MuPAD wieder die Rechenarbeit für uns erledigen: Wir setzen  $n_3$  zunächst auf den Wert  $v_1 \cdot w_2 - v_2 \cdot w_1$

- $n_3 := v_1 \cdot w_2 - v_2 \cdot w_1$ :  
und schauen uns die Lösungsmenge  $L$  erneut an:

- $L$ 

$$\left\{ \left[ \begin{aligned} n_1 &= \frac{v_2 w_3 (v_1 w_2 - v_2 w_1) - v_3 w_2 (v_1 w_2 - v_2 w_1)}{v_1 w_2 - v_2 w_1}, \\ n_2 &= - \frac{v_1 w_3 (v_1 w_2 - v_2 w_1) - v_3 w_1 (v_1 w_2 - v_2 w_1)}{v_1 w_2 - v_2 w_1} \end{aligned} \right] \right\}$$

MuPAD hat den Wert für  $n_3$  zwar korrekt eingesetzt, aber die Brüche noch nicht vollständig durchgekürzt<sup>11</sup>. Zum Kürzen verwenden wir wieder die Funktion `normal`, die wir einfach auf die Lösungsmenge  $L$  anwenden können:

- `normal(L)`  
 $\{[n_1 = v_2 w_3 - v_3 w_2, n_2 = v_3 w_1 - v_1 w_3]\}$

Damit lauten also die Lösungen für  $n_1$ ,  $n_2$  und  $n_3$ :

$$n_1 = v_2 \cdot w_3 - v_3 \cdot w_2 \quad n_2 = v_3 \cdot w_1 - v_1 \cdot w_3, \quad n_3 = v_1 \cdot w_2 - v_2 \cdot w_1,$$

d.h. der Vektor  $\vec{n}$  ist von der Form

$$\vec{n} = \begin{pmatrix} v_2 \cdot w_3 - v_3 \cdot w_2 \\ v_3 \cdot w_1 - v_1 \cdot w_3 \\ v_1 \cdot w_2 - v_2 \cdot w_1 \end{pmatrix}.$$

Den auf diese Weise aus den Vektoren  $\vec{v}$  und  $\vec{w}$  erhaltenen Vektor  $\vec{n}$  bezeichnet man als das *Kreuzprodukt* von  $\vec{v}$  und  $\vec{w}$  und schreibt auch

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \times \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} v_2 \cdot w_3 - v_3 \cdot w_2 \\ v_3 \cdot w_1 - v_1 \cdot w_3 \\ v_1 \cdot w_2 - v_2 \cdot w_1 \end{pmatrix}.$$

Mit MuPAD lässt sich das Kreuzprodukt zweier Vektoren auch direkt über den Befehl `linalg::crossProduct` berechnen. Im Falle unserer beiden allgemeinen Vektoren  $\vec{v}$  und  $\vec{w}$  erhalten wir so auch mit einem einzigen Befehl die Komponenten des Vektors  $\vec{n}$ , deren Darstellung wir in diesem Abschnitt hergeleitet haben:

<sup>11</sup>Aus Effizienzgründen wäre dieses Verhalten auch nicht wünschenswert, denn wenn wir noch viel komplexere Ausdrücke mit MuPAD bearbeiten, kann das Vereinfachen von Brüchen ein sehr aufwändiger Prozess sein. Vereinfachungen sollten also in dieser Situation nur dann vorgenommen werden, wenn sie vom Benutzer des Computeralgebra-Systems explizit gewünscht werden.

- `linalg::crossProduct(v,w)`

$$\begin{pmatrix} v_2 w_3 - v_3 w_2 \\ v_3 w_1 - v_1 w_3 \\ v_1 w_2 - v_2 w_1 \end{pmatrix}$$

Im  $\mathbb{R}^2$  gibt es im eigentlichen Sinne kein Kreuzprodukt. Hier beschränkt sich die Fragestellung darauf, zu *einem* gegebenen Vektor

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

des  $\mathbb{R}^2$  einen darauf senkrecht stehenden Vektor zu finden. Man kann dabei im Prinzip ganz analog zur obigen Vorgehensweise verfahren, nur dass alles viel einfacher ist, da man im Prinzip nur eine einzige Gleichung zu lösen hat. Das Auffinden eines zu einem allgemeinen Vektor im  $\mathbb{R}^2$  senkrechten Vektors ist Bestandteil der folgenden Übungen.

### Übungen

- 12.) Berechnen Sie mit Hilfe von `linalg::crossProduct` das Kreuzprodukt der beiden Vektoren

$$\vec{v} = \begin{pmatrix} 2 \\ 4 \\ -2 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} 1 \\ -2 \\ 12 \end{pmatrix}.$$

- 13.) Finden Sie zu dem allgemeinen Vektor

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

des  $\mathbb{R}^2$  einen Vektor  $\vec{n}$ , der darauf senkrecht steht. Der Vektor  $\vec{n}$  sollte nicht der Nullvektor sein, sofern  $\vec{v}$  nicht der Nullvektor ist (das wäre ja auch zu einfach)!

## 2.7 Erste elementare Anwendungen in der Geometrie

Wir können jetzt erste Anwendungen mit dem erarbeiteten mathematischen Handwerkszeug durchrechnen. Folgende Aufgabe wollen wir jetzt bearbeiten:

**Aufgabe:** Gegeben sind die drei Punkte  $A(2|0|-1)$ ,  $B(4|2|1)$  und  $C(-1|3|-2)$  im Raum, die ein Dreieck bilden.

- Zeichnen Sie  $A$ ,  $B$  und  $C$  in ein gemeinsames Koordinatensystem. Verbinden Sie die Punkte untereinander und überzeugen Sie sich, dass sie ein Dreieck bilden.
- Zeigen Sie, dass  $A$ ,  $B$  und  $C$  ein rechtwinkliges Dreieck im Raum bilden.

- c) Finden Sie einen Punkt  $X$ , so dass durch  $A, B, C$  und  $X$  ein Rechteck im Raum gegeben ist. Zeichnen Sie das Rechteck mit MuPAD.
- d) Berechnen Sie den Flächeninhalt des Rechtecks aus Teil c).

**Lösung:** a) Wir definieren die drei Punkte in MuPAD

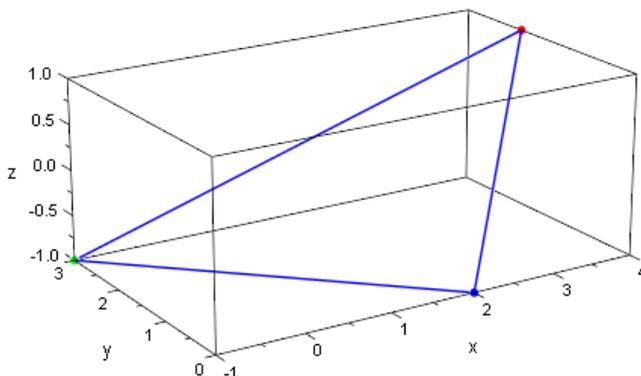
- `A:= plot::Point3d([ 2,0,-1], Color = RGB::Blue):`  
`B:= plot::Point3d([ 4,2, 1], Color = RGB::Red):`  
`C:= plot::Point3d([-1,3,-1], Color = RGB::Green):`

verbinden Sie mit Hilfe von `plot::Line3d` untereinander durch Linien

- `AB:= plot::Line3d([2,0,-1], [ 4,2, 1]):`  
`AC:= plot::Line3d([2,0,-1], [-1,3,-1]):`  
`BC:= plot::Line3d([4,2, 1], [-1,3,-1]):`

und zeichnen alle Objekte in ein gemeinsames Koordinatensystem <sup>12</sup>:

- `plot(A,B,C,AB,AC,BC,Scaling = Constrained)`



In MuPAD selbst kann die Grafik jetzt mit einem Doppelklick aktiviert und von allen Seiten genauer untersucht werden. Man überzeugt sich leicht, dass die drei Punkte ein Dreieck bilden.

<sup>12</sup> Mit Hilfe der Option `Scaling = Constrained` erreicht man, dass die Achsen gleich skaliert werden. Im allgemeinen wird von MuPAD die Skalierung so vorgenommen, dass alle relevanten Bestandteile einer Grafik möglichst gut sichtbar in das Auge des Betrachters gerückt werden. Die in MuPAD vorhandenen Heuristiken zur graphischen Darstellung sind an vielen Stellen sehr nützlich (man denke nur z.B. an Polstellen oder andere Singularitäten, die spezielle Kniffe bei der grafischen Darstellung erfordern). In unserem Fall führt die Heuristik bei der Skalierung des Koordinatensystems allerdings mitunter dazu, dass der gesuchte rechte Winkel nicht als solcher in der Grafik zu erkennen ist. Daher ist es sinnvoll, die Heuristiken zur Skalierung mit Hilfe der Option `Scaling = Constrained` zu deaktivieren

b) Die Grafik, die unter a) erstellt wurde, lässt vermuten, dass der rechte Winkel zwischen den Dreiecksseiten  $\overline{AB}$  und  $\overline{AC}$  zu finden ist. Wir müssen also gar nicht lange alle möglichen Innenwinkel "durchprobieren", sondern versuchen direkt nachzuweisen, dass der Innenwinkel zwischen  $\overline{AB}$  und  $\overline{AC}$  ein rechter Winkel ist. Wir können den Winkel zwischen zwei Vektoren berechnen und müssen also zuerst die Vektoren  $A$  nach  $B$  sowie von  $A$  nach  $C$  bestimmen. Zuerst definieren wir also in MuPAD die zugehörigen Ortsvektoren

- `OrtsvektorA:= matrix([ 2,0,-1]):`  
`OrtsvektorB:= matrix([ 4,2, 1]):`  
`OrtsvektorC:= matrix([-1,3,-1]):`

und berechnen dann den Vektor von  $A$  nach  $B$

- `VektorAB:= OrtsvektorB - OrtsvektorA`

$$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$

sowie den Vektor von  $A$  nach  $C$

- `VektorAC:= OrtsvektorC - OrtsvektorA`

$$\begin{pmatrix} -3 \\ 3 \\ 0 \end{pmatrix}$$

Um nachzuweisen, dass die Seiten  $\overline{AB}$  und  $\overline{AC}$  einen rechten Winkel einschließen können wir nun entweder mit `linalg::angle` den Winkel direkt berechnen oder einfach mittels `linalg::scalarProduct` das Skalarprodukt der beiden Vektoren bilden und prüfen, ob dieses Null wird. Wir zeigen, dass das Skalarprodukt Null wird:

- `linalg::scalarProduct(VektorAB, VektorAC)`

0

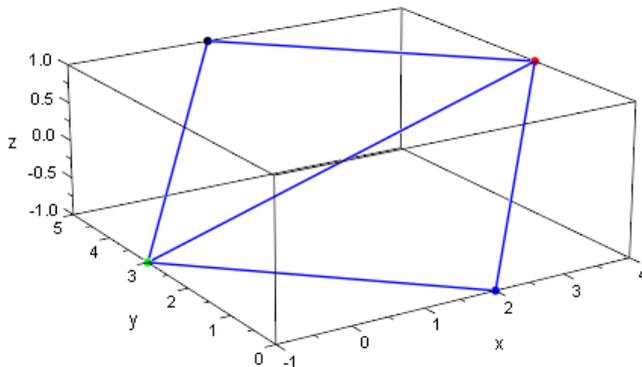
c) Wir erhalten eine Bestimmungsgleichung für den Ortsvektor des gesuchten Punkt  $X$ , wenn wir noch einmal einen genauen Blick auf die Darstellung des Dreiecks werfen, die wir unter a) mit MuPAD angefertigt hatten. Da  $\overline{AB}$  und  $\overline{AC}$  den rechten Winkel einschließen, sollte also für den Punkt  $X$  gelten: Die Vektoren von  $A$  nach  $B$  und von  $C$  nach  $X$  sollten gleich sein, d.h. die gleiche Länge und die gleiche Richtung haben. Wir erhalten also:

- `OrtsvektorX:= (OrtsvektorB - OrtsvektorA) + OrtsvektorC`

$$\begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix}$$

Zeichnen wir den zugehörigen Punkt  $D$  mit den Punkten  $A$ ,  $B$  und  $C$  sowie den Dreiecksseiten und zwei weiteren Linien von  $C$  nach  $X$  und von  $X$  nach  $B$  in ein gemeinsames Koordinatensystem, so erhalten wir:

- `X:= plot::Point3d([1,5,1], Color = RGB::Black):`  
`CX:= plot::Line3d([-1,3,-1],[1,5,1]):`  
`XB:= plot::Line3d([1,5,1],[4,2,1]):`  
`plot(A,B,C,X,AB,AC,BC,CX,XB,Scaling = Constrained)`



Die Hypotenuse des Dreiecks bildet nun wie erwartet eine Diagonale des Rechtecks.

d) Der Flächeninhalt des Rechtecks ergibt sich als Produkt "Länge  $\times$  Breite". Die Länge des Rechtecks sei im folgenden die Länge des Vektors **VektorAB** von  $A$  nach  $B$ . Die Breite des Rechtecks sei die Länge des Vektors **VektorAC** von  $A$  nach  $C$ . Die Länge eines Vektors können wir – wie gehabt – mit Hilfe des Befehls **norm** bestimmen. Anschließend multiplizieren wir die beiden Längen der Vektoren und erhalten so den Flächeninhalt des Rechtecks:

- `norm(VektorAB, 2) * norm(VektorAC, 2)`

$$6\sqrt{3}\sqrt{2}$$

## 3 Geraden und Ebenen

In diesem Kapitel kommen wir nun dazu, Geraden und Ebenen im Raum zu beschreiben, Verfahren kennen zu lernen, wie man wechselseitige Lagebeziehungen untersuchen und wie man all diese Situationen mit MuPAD visualisieren bzw. mit MuPAD gesuchte Lösungen berechnen kann. Wie schon im vorhergehenden Kapitel finden sich zu Beginn eines jeden Abschnitts eine Liste derjenigen MuPAD-Funktionen, die neu und vorher noch nicht aufgetaucht sind.

### 3.1 Beschreibung von Geraden

In der Motivation zu diesem Buch (siehe Abschnitt 1.2 in Kapitel 1) hatten wir Geraden in der Ebene sowie im Raum dadurch beschrieben, dass wir eine Art “Wegbeschreibung” angegeben haben. Diese Idee greifen wir jetzt wieder auf. Die in Abschnitt 1.2 in Form der Abbildungen 1.3 und 1.6 angedeuteten Wegbeschreibungen werden wir mathematisch korrekt als *Parameterdarstellungen* einer Geraden im  $\mathbb{R}^2$  bzw. im  $\mathbb{R}^3$  bezeichnen. Wir beginnen also zuerst damit, unsere “Wegbeschreibungen” in eine vernünftige mathematische Schreibweise zu überführen und machen uns im Anschluss daran noch einmal Gedanken, wie sich die aus dem Mittelstufenunterricht bekannten Geradengleichungen von den hier vorgestellten *Parameterformen* unterscheiden.

#### 3.1.1 Parameterform von Geraden im $\mathbb{R}^2$ und $\mathbb{R}^3$

##### Neue MuPAD-Funktionen

- `plot::Curve3d` Darstellung von Geraden (und allgemeiner Kurven) im Dreidimensionalen
- `plot::Curve2d` Darstellung von Geraden (und allgemeiner Kurven) im Zweidimensionalen

Das schöne an den in Kapitel 2 erarbeiteten Grundlagen zu Vektoren im  $\mathbb{R}^2$  und  $\mathbb{R}^3$  ist, dass wir jetzt eine Darstellungsform für Geraden angeben können, die rein formelmäßig sowohl für den zweidimensionalen wie auch für den dreidimensionalen Fall gilt; im ersten Fall haben die Vektoren zwei Komponenten (eine  $x$ - und eine  $y$ -Komponente) und im dreidimensionalen Fall haben sie eben drei Komponenten (eine  $x$ - eine  $y$ - und zusätzlich eine  $z$ -Komponente).

Wie wir in Abschnitt 1.2 bereits angemerkt hatten, ist eine Gerade (sowohl im  $\mathbb{R}^2$  als auch im  $\mathbb{R}^3$ ) durch zwei (verschiedene) Punkte  $A$  und  $B$  eindeutig bestimmt. Wir haben

in Kapitel 2 zu Anfang jedem Punkt einen Ortsvektor zugeordnet. Die Ortsvektoren der Punkte  $A$  und  $B$  bezeichnen wir mit  $\vec{a}$  und  $\vec{b}$ . Wenn wir also die Gerade durch  $A$  und  $B$  beschreiben sollten, so erinnern wir uns an unsere Wegbeschreibung:

1. Fahre z.B. vom Startpunkt  $\vec{0}$  aus mit dem Fahrzeug in Richtung  $A$ .
2. Fahre dann von  $A$  die Straße entlang in Richtung  $B$  bis Du den allgemeinen Punkt  $X$  auf der Geraden erreicht hast.

Formulieren wir also die informelle Wegbeschreibung mathematisch: Wir wollen einen beliebigen Punkt  $X$  auf der durch  $A$  und  $B$  beschriebenen Geraden erreichen können. Um die Vektorrechnung anwenden zu können, sprechen wir nicht mehr von den Punkten, sondern von ihren Ortsvektoren. Für den Ortsvektor eines beliebigen Punktes  $X$  auf der Geraden durch  $A$  und  $B$  gilt:

$$\vec{x} = \vec{a} + k \cdot \overrightarrow{AB} = \vec{a} + k \cdot (\vec{b} - \vec{a}),$$

wobei  $k$  eine reelle Zahl ist. Mit anderen Worten: Wir erreichen jeden Punkt der Geraden, indem wir zunächst dem Vektor  $\vec{a}$  vom Nullpunkt auf die Gerade folgen und dann den Differenzvektor  $\vec{b} - \vec{a}$  als Richtungsvektor so weit mit dem skalaren Faktor  $k$  verlängern oder verkürzen, dass wir jeden Punkt  $X$  auf der Geraden erreichen (hier ist zu bedanken, dass wir, wenn wir für  $k$  eine negative Zahl wählen, natürlich auch in den "entgegengesetzte Richtung" entlang der Geraden laufen können). Den Vektor  $\vec{a}$  nennen wir einen *Stützvektor* der Geraden und den Vektor  $\overrightarrow{AB} = \vec{b} - \vec{a}$  einen *Richtungsvektor* der Geraden. Die Gleichung  $\vec{x} = \vec{a} + k \cdot (\vec{b} - \vec{a})$  nennen wir eine *Parameterform* der Geraden. Die gesamte Vorgehensweise ist in Abbildung 3.1 in vier Einzelbildern einer MuPAD-Animation dargestellt:

Die folgende Aufgabenstellung können wir jetzt bereits bearbeiten.

**Aufgabe:** Bestimmen Sie eine Parameterform derjenigen Geraden, die durch die Punkte  $A(1|2|1)$  und  $B(-2|-1|3)$  verläuft. Zeichnen Sie die Gerade zusammen mit den Punkten  $A$  und  $B$  mit MuPAD in ein gemeinsames Koordinatensystem.

**Lösung:** Als Stützvektor der Geraden wählen wir den Ortsvektor des Punktes  $A$  (man kann natürlich auch den Ortsvektor von  $B$  wählen; hier sieht man, dass Parameterformen einer Geraden natürlich *nicht eindeutig bestimmt sind*). Der Richtungsvektor ergibt sich dann wieder als Differenzvektor des Ortsvektors von  $A$  und des Ortsvektors von  $B$ :

- `OrtsvektorA:= matrix([ 1, 2,1]):`  
`OrtsvektorB:= matrix([-2,-1,3]):`  
`Gerade:= OrtsvektorA + k * (OrtsvektorB - OrtsvektorA)`  

$$\begin{pmatrix} 1 - 3k \\ 2 - 3k \\ 2k + 1 \end{pmatrix}$$

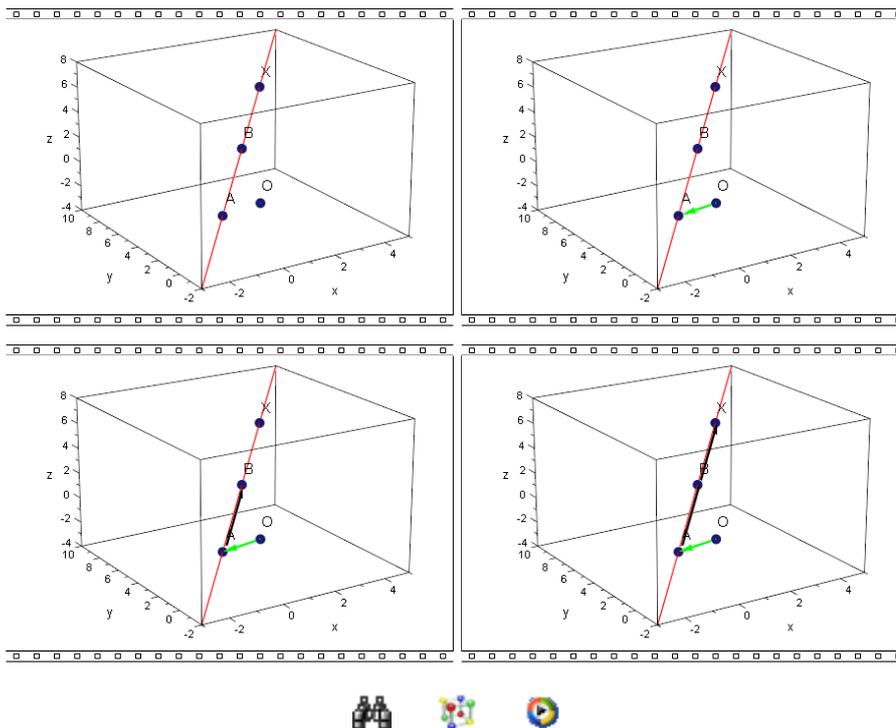


Abbildung 3.1: Parameterform einer Geraden

Damit ist die Geradengleichung bestimmt<sup>1</sup>.

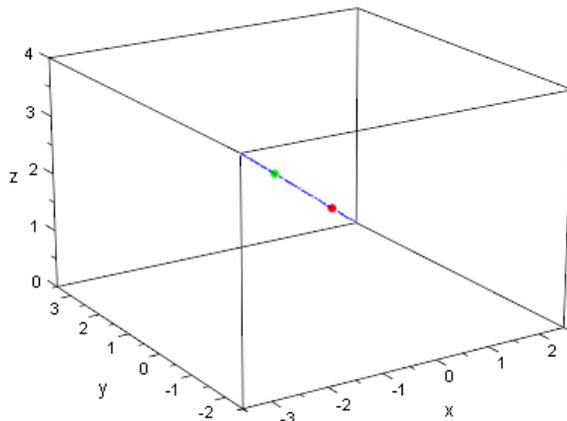
Geraden im Raum lassen sich mit MuPAD z.B. über den Befehl `plot::Curve3d` darstellen<sup>2</sup>. Die Punkte  $A$  und  $B$  zeichnen wir wie gewohnt mit Hilfe von `plot::Point3d`<sup>3</sup>:

- `A:= plot::Point3d(OrtsvektorA, Color = RGB::Red):`  
`B:= plot::Point3d(OrtsvektorB, Color = RGB::Green):`  
`G:= plot::Curve3d(Gerade, k = -1/2..3/2):`  
`plot(A, B, G)`

<sup>1</sup>MuPAD fasst dabei stets den Stützvektor und das  $k$ -Fache des Richtungsvektors stets zu einem Vektor zusammen. Mit etwas künstlichem Aufwand lässt sich dieses Verhalten umgehen. Jedoch widerspricht es im Prinzip der Natur eines Computeralgebra System, das u.a. dazu dient "Dinge auszurechnen": MuPAD rechnet und fasst daher die Summe der Vektoren zu einem Vektor zusammen.

<sup>2</sup>Man kann theoretisch natürlich auch `plot::Line3d` verwenden, jedoch hat `plot::Curve3d` hier einfach den Vorteil, dass wir dieser MuPAD-Funktion Parameterform direkt als Argument übergeben können und nur noch einen Laufbereich für den Skalar  $k$  angeben müssen.

<sup>3</sup>Auch dieser Funktion können wir, wenn wir das möchten, direkt den Ortsvektor eines Punktes als Argument übergeben und müssen die Koordinaten nicht explizit als Einträge einer Liste angeben



Für Geraden im  $\mathbb{R}^2$  geht man ganz analog vor. Hier verwendet man nur statt der Funktion `plot::Curve3d` die Funktion `plot::Curve2d`.

### Übungen

- 14.) Bestimmen Sie eine Parameterform der Geraden im  $\mathbb{R}^2$  durch die beiden Punkte  $A(1|-2)$  und  $B(6|1)$ . Zeichnen Sie die Punkte mit Hilfe von `plot::Point2d` sowie die Gerade mit Hilfe von `plot::Curve2d` in unterschiedlichen Farben in ein gemeinsames Koordinatensystem.
- 15.) Bestimmen Sie eine Gleichung der Geraden durch den Punkt  $A(2|-1|-1)$  und den Koordinatenursprung. Zeichnen Sie den Koordinatenursprung sowie den Punkt  $A$  und die berechnete Gerade in ein gemeinsames Koordinatensystem.

### 3.1.2 Koordinatenform von Geraden im $\mathbb{R}^2$

#### Neue MuPAD-Funktionen

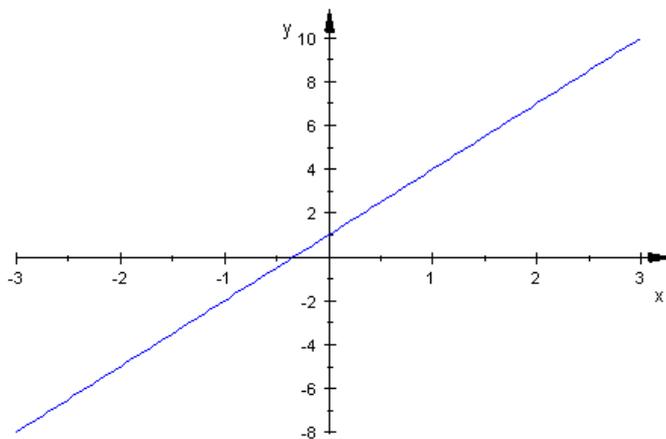
<code>plot::Function2d</code>	Darstellung von Funktionen im Dreidimensionalen
<code>LineWidth</code>	Einstellung der Linienstärke eines grafischen Objekts

Die natürliche Frage, die sich jetzt stellt, ist: Wo ist die Verbindung zwischen der aus der Analysis bekannten Geradengleichung der Form  $y = m \cdot x + b$ , wobei  $m$  die Steigung der Geraden und  $b$  ihr  $y$ -Achsenabschnitt ist? In der Sprache der Analytischen Geometrie bezeichnet man eine Gleichung der Form  $y = m \cdot x + b$  als *Koordinatenform einer Geraden im  $\mathbb{R}^2$* . Die Bezeichnung kommt daher, weil man in die Geradengleichung direkt Koordinaten eines Punkte (also Werte für  $x$  und  $y$ ) einsetzen kann. Bei den im vorhergehenden Abschnitt kennen gelernten Parameterformen von Geraden  $\vec{x} = \vec{a} + k\vec{u}$ , wobei  $\vec{a}$  der Stützvektor der Geraden und  $\vec{u}$  ihr Richtungsvektor ist, gibt es nur einen *Parameter* – nämlich  $k$  – in den wir etwas einsetzen können.

Wie kommt man nun von einer Koordinatenform einer Geraden im  $\mathbb{R}^2$  zu einer Parameterform und umgekehrt?

Wir wollen uns zuerst überlegen, wie wir von einer Koordinatenform  $y = m \cdot x + b$  zu einer Parameterform gelangen. Eine mögliche Vorgehensweise besteht darin, einfach zwei Punkte  $A$  und  $B$  auf der Geraden zu bestimmen (durch geeignete Wahl von  $x$  und  $y$ , so dass die Geradengleichung  $y = m \cdot x + b$  erfüllt ist). Mit Hilfe der beiden Punkte bzw. ihren Ortsvektoren  $\vec{a}$  und  $\vec{b}$  kann dann direkt eine Parameterform der Geraden über  $\vec{x} = \vec{a} + k \cdot (\vec{b} - \vec{a})$  angegeben werden. Wir schauen uns ein Beispiel an: Gegeben sei z.B. die Gerade  $y = 3 \cdot x + 1$ . Diese können wir zur Veranschaulichung mit Hilfe von `plot::Function2d` mit MuPAD darstellen lassen:

- `g:= plot::Function2d(3*x + 1, x = -3..3):`  
`plot(g)`



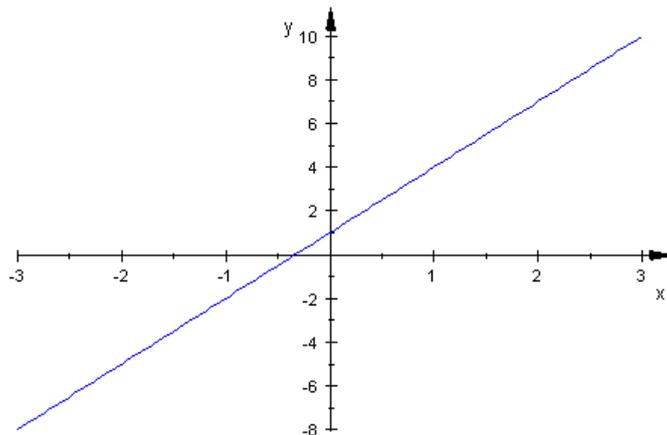
Die Punkte  $A(0|1)$  und  $B(1|4)$  liegen auf der Geraden. Also ist eine Parameterform der Geraden gegeben durch:

- `Stuetzvektor:= matrix([0,1]):`  
`Richtungsvektor:= matrix([1,4]) - matrix([0,1]):`  
`ParameterForm:= Stuetzvektor + k * Richtungsvektor`

$$\begin{pmatrix} k \\ 3k + 1 \end{pmatrix}$$

Zeichnen wir diese mit Hilfe von `plot::Curve2d` zusammen mit der Funktion  $g$  in ein gemeinsames Koordinatensystem so erhalten wir:

- `g2:= plot::Curve2d(ParameterForm, k = -3..3):`  
`plot(g, g2)`



Wir sehen nur eine Gerade, d.h. die beiden Darstellungen (Koordinatenform und Parameterform) beschreiben in der Tat die gleiche Gerade.

Wie erhalten wir aber aus einer Parameterform einer Geraden eine Koordinatenform? Hier gibt es – wie so oft – mehrere Wege, die zum Ziel führen. Einen dieser Wege wollen wir hier kennen lernen. Es mag nicht der einfachste Weg sein, aber wir wählen ihn bewusst, denn die im folgenden vorgestellte Vorgehensweise lässt sich leicht auf den dreidimensionalen Fall verallgemeinern und kann dann dort zur Berechnung Koordinatenformen einer Ebene verwendet werden.

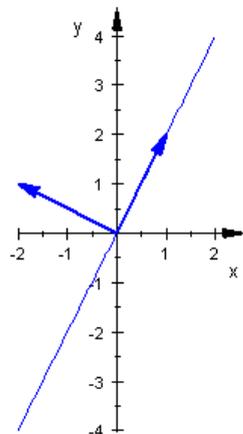
Betrachten wir als Beispiel die Gerade

$$g : \vec{x} = \begin{pmatrix} -1 \\ -2 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Zu dem Vektor Richtungsvektor  $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$  berechnen wir einen so genannten *Normalenvektor*, d.h. einen Vektor  $\vec{n}$ , der orthogonal zu dem Richtungsvektor der Geraden ist. Der Normalenvektor ist nicht eindeutig bestimmt, denn man kann einen Vektor, der auf dem Richtungsvektor der Geraden senkrecht steht, immer mit einem beliebigen Skalar  $l \in \mathbb{R}$  multiplizieren und erhält stets wieder einen Vektor, der orthogonal zu dem Richtungsvektor ist. Eine mögliche Vorgehensweise zum Auffinden eines Kandidaten für den Vektor  $\vec{n}$  besteht einfach darin, die Komponenten von  $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$  zu vertauschen und anschließend bei einer der Komponenten das Vorzeichen um zudrehen. So ist also z.B. der Vektor  $\vec{n} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$  orthogonal zu dem Richtungsvektor der Geraden (man kann hier das Skalarprodukt der beiden Vektoren sofort im Kopf ausrechnen und sich vergewissern, dass das korrekt ist; wer nicht im Kopf rechnen möchte, kann natürlich auch die MuPAD-Funktion `linalg::scalarProduct` zur Berechnung des Skalarprodukts benutzen). Zur Veranschaulichung stellen wir die Gerade sowie ihren Richtungsvektor und den Nor-

malenvektor mit MuPAD dar<sup>4</sup>:

- `a:= matrix([-1,-2]):`  
`b:= matrix([1,2]):`  
`n:= matrix([-2,1]):`  
`ParameterForm:= a + k * b:`  
`Gerade:= plot::Curve2d(ParameterForm, k = -1..3):`  
`Richtungsvektor:= plot::Arrow2d(b, LineWidth = 0.6):`  
`Normalenvektor:= plot::Arrow2d(n, LineWidth = 0.6):`  
`plot(Gerade, Richtungsvektor, Normalenvektor,`  
`Scaling = Constrained)`



Weil der Vektor  $\vec{n}$  senkrecht auf dem Richtungsvektor der Geraden steht, steht er auch senkrecht auf jedem skalaren Vielfachen  $k \cdot \vec{b}$  des Richtungsvektors  $\vec{b}$ , d.h. es gilt:

$$0 = \vec{n} * (k \cdot \vec{b}) = \vec{n} * \underbrace{(\vec{x} - \vec{a})}_{=k \cdot \vec{b}},$$

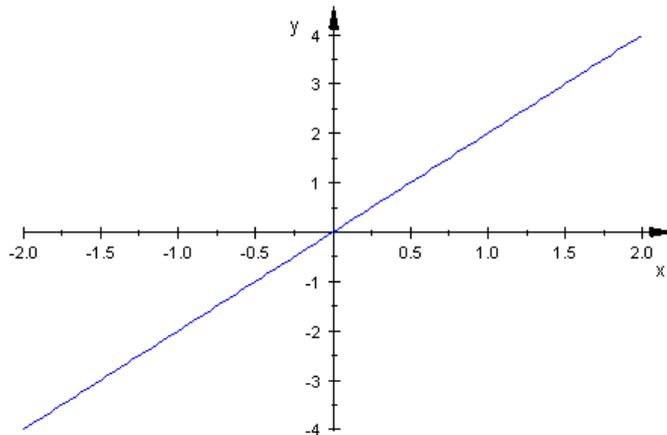
wobei  $\vec{x}$  der Ortsvektor eines beliebigen Punktes der Geraden aus ihrer Parameterform ist. Die Gleichung  $\vec{n} * (\vec{x} - \vec{a}) = 0$  ist, wenn man sie ausmultipliziert aber von der folgenden Form (die Komponenten des Vektors  $\vec{x}$  bezeichnen wir im folgenden mit  $x_1$  und  $x_2$ ):

- `assume({x1, x2}, Type::Real):`  
`linalg::scalarProduct(n, matrix([x1,x2]) - a) = 0`  
 $x_2 - 2x_1 = 0$

Bedenken wir jetzt noch, dass  $x_1$  der  $x$ -Koordinate eines Punktes in der Ebene und  $x_2$  der  $y$ -Koordinate eines Punktes in der Ebene entspricht und stellen die obige Gleichung nach  $x_2$  um, so erhalten wir die Geradengleichung  $x_2 = 2 \cdot x_1$  bzw.  $y = 2 \cdot x$ . Zeichnen wir diese Gerade mit Hilfe von `plot::Function2d` mit der oben in Parameterform gegebenen Gerade `Gerade` in ein gemeinsames Koordinatensystem, so sehen wir, dass die beiden Geraden übereinstimmen:

<sup>4</sup>Mit Hilfe des Attributs `LineWidth` kann man die Linienstärke eines grafischen Objekts in cm angeben.

- `Gerade2:= plot::Function2d(2*x, x = -2..2):  
plot(Gerade, Gerade2)`



Das hier kennen gelernte Verfahren zur Bestimmung der Koordinatenform einer Geraden aus einer gegebenen Parameterform mit Hilfe eines Normalenvektors  $\vec{n}$  wird uns später im Kontext von Ebenen wieder begegnen. Für Geraden im  $\mathbb{R}^3$  gibt es wie in Abbildung 1.5 angedeutet *keine Koordinatenform*.

### Übungen

- 16.) Berechnen Sie eine Parameterform der durch  $y = -2 \cdot x - 5$  gegebenen Geraden. Stellen Sie beiden Geraden grafisch mit MuPAD dar. Vergewissern Sie sich anhand einer Grafik, bei der Sie beide Geraden in ein gemeinsames Koordinatensystem zeichnen, dass die von Ihnen berechnete Parameterform korrekt ist.
- 17.) Berechnen Sie eine Koordinatenform der Gerade

$$g : \vec{x} = \begin{pmatrix} 2 \\ 5 \end{pmatrix} + k \cdot \begin{pmatrix} -1 \\ 3 \end{pmatrix}.$$

Stellen Sie beiden Geraden grafisch mit MuPAD dar. Vergewissern Sie sich anhand einer Grafik, bei der Sie beide Geraden in ein gemeinsames Koordinatensystem zeichnen, dass die von Ihnen berechnete Parameterform korrekt ist.

## 3.2 Lagebeziehung von Geraden und Punkten im $\mathbb{R}^2$ und $\mathbb{R}^3$

In diesem Abschnitt wollen wir uns den möglichen Lagebeziehungen zwischen Punkten und Geraden widmen. Sicherlich gibt es nur zwei mögliche Lagebeziehungen, für einen Punkt und eine Gerade: Entweder der Punkt liegt auf der Geraden oder er liegt nicht auf der Geraden (dies gilt natürlich sowohl für Punkte und Geraden im  $\mathbb{R}^2$  wie auch im  $\mathbb{R}^3$ ). Wir beschränken uns im folgenden auf den dreidimensionalen Fall. Als Beispiel

betrachten wir die Gerade

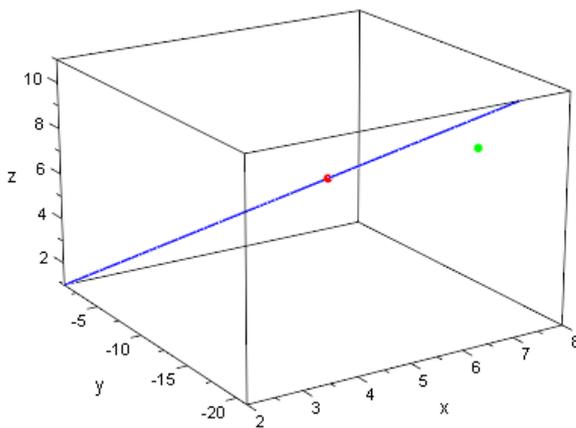
$$g : \vec{x} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -4 \\ 2 \end{pmatrix}$$

sowie die Punkte  $P(5 | -13 | 7)$  und  $Q(8 | -27 | 17)$ . Liegt  $P$  bzw.  $Q$  auf der Geraden  $g$ ? Wir verschaffen uns zunächst einen Eindruck, indem wir die beiden Punkte zusammen mit der Geraden in ein Koordinatensystem zeichnen:

- ```

Parameterform:= matrix([2,-1,1]) + k * matrix([1,-4,2]):
OrtsvektorP:= matrix([5,-13,7]):
OrtsvektorQ:= matrix([8,-27,17]):
g:= plot::Curve3d(Parameterform, k = 0..5):
P:= plot::Point3d(OrtsvektorP, Color = RGB::Red):
Q:= plot::Point3d(OrtsvektorQ, Color = RGB::Green):
plot(g, P, Q)

```



die obige Grafik suggeriert, dass der Punkt  $P$  (in roter Farbe eingezeichnet) auf der Geraden liegt, der Punkt  $Q$  (in grüner Farbe eingezeichnet) dagegen nicht. Derartige Grafiken sind sehr hilfreich, wenn es darum geht, Vermutungen aufzustellen, die können aber auch täuschen. Wir wollen daher explizit nachrechnen, dass  $P$  auf  $g$  und  $Q$  nicht auf  $g$  liegt. Wenn ein Punkt auf einer Geraden liegt, dann muss es einen Wert für  $k$  geben, derart, dass, wenn wir diesen in die Parameterform der Geraden einsetzen, wir den Ortsvektor des Punktes erhalten. Im Falle des Punktes  $P$  und – vorausgesetzt, er liegt wirklich auf der Geraden  $g$  und nicht etwa knapp daneben – müsste es also einen Wert  $k$  geben mit

$$\begin{pmatrix} 5 \\ -13 \\ 7 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -4 \\ 2 \end{pmatrix},$$

wobei uns die ersten Komponenten der Vektoren auf der linken und rechten Seite eine Gleichung für  $k$  liefern ebenso wie auch die zweite und die dritte Komponente der

rechten und linken Seite. Wir erhalten also praktisch ein Gleichungssystem bestehend aus drei Gleichungen mit nur einer Unbekannten (nämlich  $k$ ). Ist das Gleichungssystem lösbar, d.h. es gibt *einen Wert* für  $k$ , so dass alle drei Gleichungen simultan erfüllt sind, so liegt der betreffende Punkt auf der Geraden (und andernfalls liegt er nicht auf der Geraden).

Wir stellen das obige Gleichungssystem bestehend aus den drei Gleichungen für  $k$  auf

- $G11 := \{ \text{OrtsvektorP}[1] = \text{Parameterform}[1], \\ \text{OrtsvektorP}[2] = \text{Parameterform}[2], \\ \text{OrtsvektorP}[3] = \text{Parameterform}[3] \}$   
 $\{5 = k + 2, 7 = 2k + 1, -13 = -4k - 1\}$

und lösen es mit Hilfe von der MuPAD-Funktion `solve`:

- $\text{solve}(G11, \{k\})$   
 $\{[k = 3]\}$

MuPAD teilt uns mit, dass das Gleichungssystem  $G11$  lösbar ist und dass  $k = 3$  die Lösung ist. Setzen wir den Wert  $k = 3$  in die Parameterform der Geraden  $g$  ein<sup>5</sup>, so müsste sich jetzt also der Ortsvektor des Punktes  $P$  ergeben:

- $\text{subs}(\text{Parameterform}, k = 3)$   

$$\begin{pmatrix} 5 \\ -13 \\ 7 \end{pmatrix}$$

Unsere Vermutung, die wir oben anhand der Darstellung der Geraden  $g$  und des Punktes  $P$  geäußert hatten, war also korrekt. Nun zu dem Punkt  $Q$ , von dem wir anhand der Grafik vermutet hatten, dass er nicht auf der Geraden  $g$  liegt. Wir stellen das entsprechende Gleichungssystem für  $Q$  auf:

- $G12 := \{ \text{OrtsvektorQ}[1] = \text{Parameterform}[1], \\ \text{OrtsvektorQ}[2] = \text{Parameterform}[2], \\ \text{OrtsvektorQ}[3] = \text{Parameterform}[3] \}$   
 $\{8 = k + 2, 7 = 2k + 1, -13 = -4k - 1\}$

und lösen es mit Hilfe von der MuPAD-Funktion `solve`:

- $\text{solve}(G12, \{k\})$

<sup>5</sup>Zum Einsetzen ("Substituieren") eines Wertes für einen Parameter in die Parameterform einer Geraden verwenden wir wieder den Befehl `subs`.

}

Das Gleichungssystem besitzt offensichtlich keine Lösung, d.h. auch hier war unsere Vermutung richtig:  $Q$  liegt nicht auf der Geraden  $g$ .

### Übungen

- 18.) Welcher der Punkte  $P(0|7|-3)$ ,  $Q(-2|1|6)$ ,  $R(-2|1|7)$  und  $S(13/4|-6|7/2)$  liegt auf der Geraden

$$g : \vec{x} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -4 \\ 2 \end{pmatrix} ?$$

## 3.3 Lagebeziehung zweier Geraden

Im Zweidimensionalen gibt es drei mögliche Lagebeziehungen zwischen zwei Geraden: Entweder sie sind *parallel*, sie *schneiden sich*, oder sie sind *identisch*. Wichtig ist, dass man auch an den trivialen Fall denkt, dass zwei Geraden identisch sind. Wenn wir diese mit MuPAD zeichnen lassen, so wird die Grafik natürlich sofort suggerieren, dass wir zweimal die gleiche Gerade gezeichnet haben. Allerdings haben wir in Abschnitt 3.1.1 u.a. gesehen, dass die Parameterform einer Geraden nicht eindeutig ist. Wir könnten also durchaus zwei verschiedene Parameterformen ein und derselben Geraden angeben, die rein äußerlich den Eindruck erwecken, dass sie unterschiedliche Geraden beschreiben.

Im Dreidimensionalen kommt noch eine vierte mögliche Lagebeziehung zwischen zwei Geraden hinzu: Zwei Geraden im Raum können *windschief* zueinander liegen. Was es anschaulich bedeutet, dass zwei Geraden windschief zueinander liegen, kann man sich z.B. wie folgt klarmachen: Man betrachtet einfach zwei sich schneidende Geraden, die z.B. in der  $x$ - $y$ -Ebene liegen<sup>6</sup>. Dann verschiebt man eine der beiden Geraden parallel zur  $x$ - $y$ -Ebene nach oben. Das so entstehende Geradenpaar hat zwar unterschiedliche Richtung wie bei zwei sich schneidenden Geraden, aber dadurch dass beide auf zwei verschiedenen "Höhenlevels" verlaufen, schneiden sie sich nicht (und sind aufgrund der verschiedenen Richtungen auch weder parallel noch identisch). Abbildung 3.2 zeigt eine MuPAD-Animation, die die Verschiebung der einen Geraden im Raum hervorragend visualisiert.

Alle vier möglichen Lagebeziehungen zweier Geraden im Raum sind in Abbildung 3.3 noch einmal zusammengefasst.

Wir arbeiten jetzt je zwei der vier Situationen aus Abbildung 3.3 gleichzeitig ab und beginnen mit dem Fall zweier paralleler oder identischer Geraden.

<sup>6</sup>Wenn wir im Prinzip zum jetzigen Zeitpunkt noch nicht wissen können, was mathematisch mit einer Ebene gemeint ist, so sollte hier doch die rein intuitive Vorstellung dessen, was eine Ebene ist, genügen, um der Erklärung folgen zu können

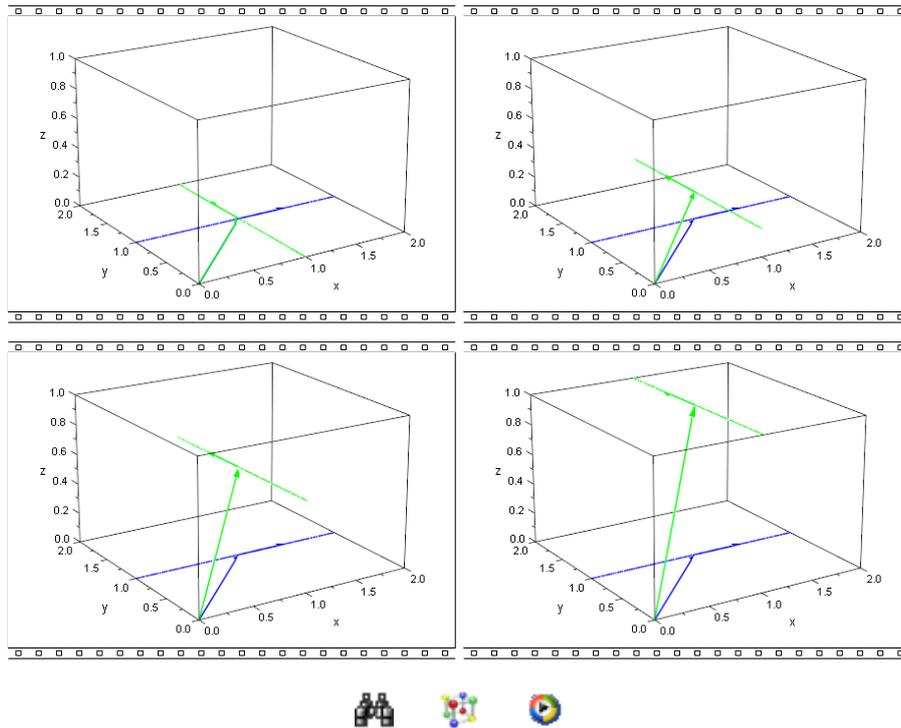


Abbildung 3.2: Von sich schneidenden Geraden hin zu windschiefen Geraden

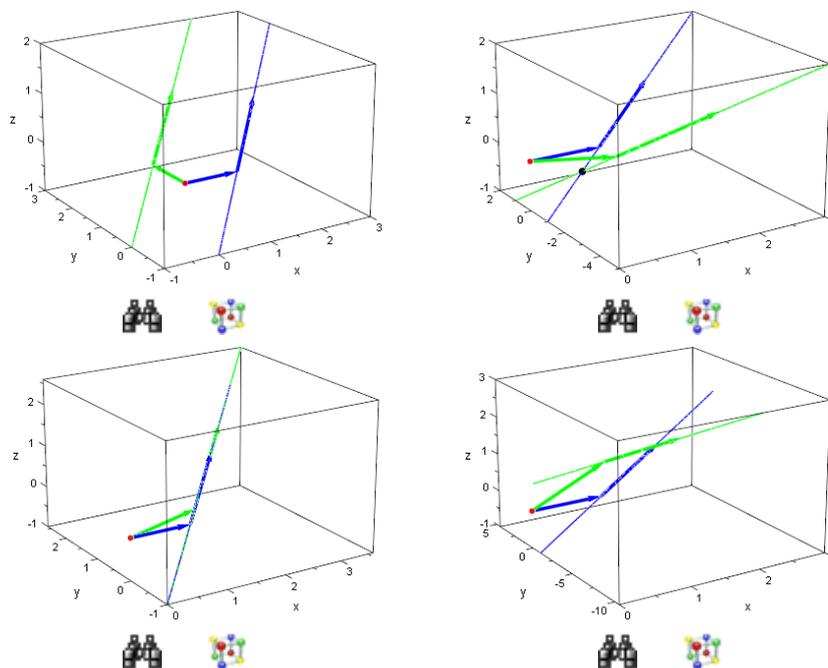


Abbildung 3.3: Mögliche Lagebeziehungen zweier Geraden im Raum

### 3.3.1 Parallele und identische Geraden

Gegeben seien die beiden Geraden

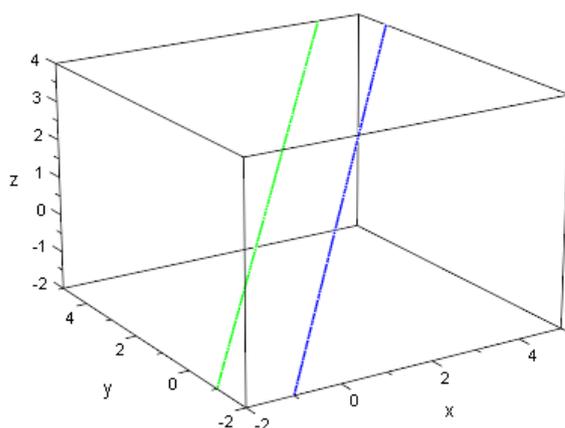
$$g: \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{und} \quad h: \vec{x} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix},$$

deren Parameterform wir zuerst in MuPAD definieren und sie anschließend dann beide in einem gemeinsamen Koordinatensystem darstellen lassen:

- ```

ParameterformG:= matrix([1,0,0]) + k * matrix([1,1,1]):
ParameterformH:= matrix([0,1,0]) + l * matrix([2,2,2]):
g:= plot::Curve3d(ParameterformG, k = -2..4,
                  Color = RGB::Blue):
h:= plot::Curve3d(ParameterformH, l = -1..2,
                  Color = RGB::Green):
plot(g,h)

```



Die Grafik deutet uns an, dass die Geraden in der Tat parallel zueinander verlaufen. Wir müssen jetzt eine mathematische Argumentation finden, die einwandfrei zeigt, dass sie in der Tat wirklich parallel sind. Was bedeutet die Parallelität der beiden Geraden? Nun, wenn sie parallel sind, dann müssen beide die gleiche oder genau die entgegengesetzte "Richtung" haben<sup>7</sup>. Ihre "Richtung" wird durch ihren Richtungsvektor

bestimmt, also durch den Vektor  $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  im Fall der Geraden  $g$  und durch den Vektor

$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$  im Fall der Geraden  $h$ . Was bedeutet es mathematisch, dass die beiden

Geraden die gleiche bzw. genau die entgegengesetzte Richtung haben? Die Antwort ist: Ihre Richtungsvektoren müssen die gleiche bzw. genau die entgegengesetzte Richtung

<sup>7</sup>Sofern man von einer Richtung sprechen kann; wir appellieren hier an die Intuition.

haben. Offensichtlich ist der Richtungsvektor der Geraden  $h$  ein skalares Vielfaches des Richtungsvektors der Geraden  $g$  (man muss den Richtungsvektor von  $g$  nur mit 2 multiplizieren und erhält so den Richtungsvektor der Geraden  $h$ ). Wenn ein Vektor ein skalares Vielfaches eines anderen Vektors ist, so haben beide Vektoren die gleiche oder genau die entgegengesetzte Richtung.

Für den Fall, dass es einmal nicht so leicht zu sehen ist, dass die Richtungsvektoren zweier paralleler Geraden skalare Vielfache voneinander sind, können wir dies natürlich auch mit MuPAD nachrechnen, indem wir ein entsprechendes Gleichungssystem aufstellen<sup>8</sup>. Dieses (zugegebenermaßen triviale) Gleichungssystem lösen wir mit Hilfe der MuPAD-Funktion `solve`:

```
• G1:= k * matrix([1,1,1]) - matrix([2,2,2]):
  solve({G1[1] = 0, G1[2] = 0, G1[3] = 0}, {k})
  {[k = 2]}
```

So weit, so gut! Nun sind aber sicherlich auch die Richtungsvektoren zweier identischer Geraden skalare Vielfache voneinander<sup>9</sup>. Um sicherzugehen, dass  $g$  und  $h$  nicht identisch sind, müssen wir aber nur prüfen, ob sie keine gemeinsamen Punkte haben. Genauer: Wir müssen im Prinzip nur nachweisen, dass es einen Punkt auf  $g$  gibt, der nicht auf  $h$  liegt. Ist dies der Fall, so sind die beiden Geraden parallel. Andernfalls sind sie identisch. Ein Punkt, der sicherlich auf der Geraden  $g$  liegt, ist der Punkt  $P(1|0|0)$ , also der Punkt, dessen Ortsvektor der Stützvektor Stützvektor der Geraden  $g$  ist. Wir prüfen, ob dieser Punkt auf der Geraden  $h$  liegt:

```
• p:= matrix([1,0,0]):
  G1:= {p[1] = ParameterformH[1],
        p[2] = ParameterformH[2],
        p[3] = ParameterformH[3]}:
  solve(G1, {1})
  {}
```

Das Gleichungssystem hat keine Lösung, d.h. die beiden Geraden sind in der Tat parallel. Die folgenden beiden Geraden sind identisch:

$$g : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{und} \quad h : \vec{x} = \begin{pmatrix} -1 \\ -2 \\ -2 \end{pmatrix} + l \cdot \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix},$$

<sup>8</sup>Die Berechnung lässt sich hier eleganter durchführen, wenn wir die Differenz des  $k$ -fachen des ersten Vektor und des zweiten Vektors gleich Null setzen.

<sup>9</sup>Wenn man so will, so sind zwei identische Geraden praktisch ein Spezialfall des allgemeinen Falls zweier paralleler Geraden. In der Literatur spricht man häufig von *echt parallelen Geraden*, wenn diese parallel aber nicht identisch sind. Wir werden hier im folgenden mit dem Begriff *parallel* in diesem Sinne immer *echt parallele* Geraden meinen.

denn ihre Richtungsvektoren sind wieder Vielfache voneinander und der Punkt  $P(1|0|0)$  liegt sowohl auf der Geraden  $g$  (es ist der Punkt, dessen Ortsvektor der Antragsvektor der Geraden  $g$  ist) als auch auf der Geraden  $h$ :

```

• p:= matrix([1,0,0]):
  ParameterformH:= matrix([-1,-2,-2]) + 1 * matrix([2,2,2]):
  Gl:= {p[1] = ParameterformH[1],
        p[2] = ParameterformH[2],
        p[3] = ParameterformH[3]}:
  solve(Gl, {1})
  {[l = 1]}

```

### Übungen

- 19.) Zeichnen Sie die beiden oben zuletzt definierten Geraden  $g$  und  $h$  in ein gemeinsames Koordinatensystem und vergewissern Sie sich, dass diese identisch sind.
- 20.) Untersuchen Sie die folgenden drei Geraden im  $\mathbb{R}^3$  auf ihre gegenseitige Lage zueinander:

$$\begin{aligned}
 g_1: \vec{x} &= \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix} + k \cdot \begin{pmatrix} 3 \\ -2 \\ 2 \end{pmatrix}, \\
 g_2: \vec{x} &= \begin{pmatrix} -4 \\ 5 \\ -6 \end{pmatrix} + l \cdot \begin{pmatrix} -6 \\ 4 \\ -4 \end{pmatrix}, \\
 g_3: \vec{x} &= \begin{pmatrix} -2 \\ 3 \\ -4 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ -2/3 \\ 2/3 \end{pmatrix}.
 \end{aligned}$$

Zeichnen Sie dazu alle drei Geraden in ein gemeinsames Koordinatensystem und stellen Sie eine Vermutung über die gegenseitige Lage auf. Weisen Sie Ihre Vermutung dann rechnerisch nach.

### 3.3.2 Windschiefe und sich schneidende Geraden

Im letzten Abschnitt haben wir an Beispielen gesehen, wie wir rechnerisch überprüfen können, ob zwei gegebene Geraden parallel oder identisch sind. Hier werden wir uns ein Verfahren überlegen, wie wir rechnerisch nachweisen können, dass sich zwei Geraden schneiden oder windschief zueinander im Raum liegen. Als Beispiel betrachten wir die beiden Geraden

$$g: \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{und} \quad h: \vec{x} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}.$$

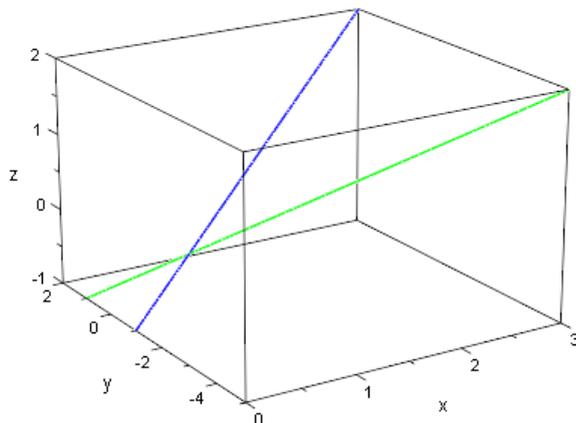
Wir definieren die beiden Parameterformen in MuPAD und zeichnen die zugehörigen Geraden anschließend in ein gemeinsames Koordinatensystem:

```

• ParameterformG:= matrix([1,0,0]) + k * matrix([1,1,1]):
  ParameterformH:= matrix([1,-1,0]) + l * matrix([1,-2,1]):
  g:= plot::Curve3d(ParameterformG, k = -1..2,
                    Color = RGB::Blue):
  h:= plot::Curve3d(ParameterformH, l = -1..2,
                    Color = RGB::Green):

plot(g,h)

```



Anhand der Grafik drängt sich die Vermutung auf, dass sich  $g$  und  $h$  in einem gemeinsamen Punkt scheiden. Potentielle gemeinsame Punkte zweier Geraden berechnet man im Prinzip genauso, wie man Schnittstellen von Funktionen in der Analysis bestimmt: Man setzt einfach die Parameterformen der beiden Geraden gleich und löst das entstehende Gleichungssystem in zwei Unbekannten.

Bevor wir aber auf die beschriebene Weise den potentiellen Schnittpunkt berechnen, wollen wir uns zuerst noch die Frage stellen: Können wir den Geradengleichungen auch ohne der mit MuPAD erzeugten Grafik ansehen, dass sie zumindest nicht parallel und auch nicht identisch sein können?

Die Antwort ist "Ja". Wie schon im letzten Abschnitt, wo wir am Anfang unser Augenmerk auf die beiden Richtungsvektoren der gegebenen Geraden gerichtet haben, liefern

uns diese auch hier die nötige Information. Die beiden Richtungsvektoren  $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  und

$\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$  sind nämlich *keine skalaren Vielfachen* voneinander. Das können wir mit MuPAD auch wieder rechnerisch nachweisen, indem wir zeigen, dass das entsprechende Gleichungssystem keine Lösung besitzt<sup>10</sup>:

<sup>10</sup>Die Berechnung lässt sich hier eleganter durchführen, wenn wir die Differenz des  $k$ -fachen des ersten Vektor und des zweiten Vektors gleich Null setzen.

- `G1:= k * matrix([1,1,1]) - matrix([1,-2,1]):  
solve({G1[1] = 0, G1[2] = 0, G1[3] = 0}, {k})  
{}`

Damit haben also  $g$  und  $h$  unterschiedliche “Richtungen” und können weder parallel noch identisch sein. Die Richtungsvektoren liefern uns also vorab immer eine Möglichkeit zu entscheiden, ob die Geraden parallel oder identisch bzw. windschief oder schneidend sind. Hier wissen wir jetzt, dass sich  $g$  und  $h$  entweder in einem Punkt schneiden oder dass die beiden Geraden windschief sind.

Jetzt verfahren wir wie oben angedeutet: Wir setzen die beiden Parameterformen gleich und versuchen den gemeinsamen Schnittpunkt zu berechnen. Gelingt dies, so haben wir unsere Vermutung von oben nachgewiesen. Gelingt dies nicht, so haben  $g$  und  $h$  keinen einzigen gemeinsamen Punkt und müssen – da sie unterschiedliche “Richtungen” haben – windschief sein:

- `G1:= {ParameterformG[1] = ParameterformH[1],  
ParameterformG[2] = ParameterformH[2],  
ParameterformG[3] = ParameterformH[3]}:  
solve(G1, {k,l})  
{ [ k = -1/3, l = -1/3 ] }`

Das Gleichungssystem hat eine Lösung, d.h. die Geraden  $g$  und  $h$  schneiden sich in der Tat. Wir müssen jetzt nur noch entweder den für  $k$  erhaltenen Wert in die Parameterform der Geraden  $g$  einsetzen oder den für  $l$  erhaltenen Wert in die Parameterform der Geraden  $h$ <sup>11</sup>. Wir setzen den für  $k$  berechneten Wert von  $-1/3$  in die Parameterform der Geraden  $g$  ein und erhaltenen so den Ortsvektor des Schnittpunktes  $S$ <sup>12</sup>:

- `OrtsvektorS:= subs(ParameterformG, k = -1/3)`  

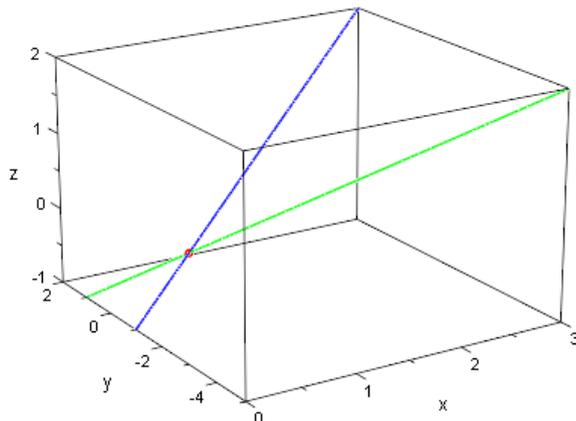
$$\begin{pmatrix} \frac{2}{3} \\ -\frac{1}{3} \\ -\frac{1}{3} \end{pmatrix}$$

Zur Veranschaulichung der Situation und zur visuellen Kontrolle unserer Rechnung zeichnen wir noch einmal die beiden Geraden zusammen mit ihrem Schnittpunkt in gemeinsames Koordinatensystem:

- `S:= plot::Point3d(OrtsvektorS, Color = RGB::Red):  
plot(g,h,S)`

<sup>11</sup>In unserem Beispiel haben  $k$  und  $l$  beiden den gleichen Wert, was natürlich nicht unbedingt immer der Fall sein muss.

<sup>12</sup>Zum Einsetzen (“Substituieren”) eines Wertes für einen Parameter in die Parameterform einer Geraden verwenden wir wieder den Befehl `subs`.



Im Gegensatz zu den obigen Geraden  $g$  und  $h$  ist das folgende Paar von Geraden

$$g : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{und} \quad h : \vec{x} = \begin{pmatrix} 1 \\ -1/2 \\ 1 \end{pmatrix} + l \cdot \begin{pmatrix} 1/2 \\ -5 \\ 1 \end{pmatrix}.$$

windschief. Die Richtungsvektoren sind offensichtlich keine skalaren Vielfachen voneinander<sup>13</sup>. Wir wissen also bereits jetzt: Entweder  $g$  und  $h$  schneiden sich oder sie sind windschief. Um wirklich nachzuweisen, dass  $g$  und  $h$  windschief ist, genügt es, zu zeigen, dass sie keinen gemeinsamen Punkt haben. Setzen wir die Parameterformen gleich und versuchen wir mit MuPAD einen Lösung des Gleichungssystems zu finden, so zeigt uns MuPAD:

```

• ParameterformG:= matrix([1, 0,0]) + k * matrix([1,1,1]):
  ParameterformH:= matrix([1,-1/2,1]) + l * matrix([1/2,-5,1]):
  G1:= {ParameterformG[1] = ParameterformH[1],
        ParameterformG[2] = ParameterformH[2],
        ParameterformG[3] = ParameterformH[3]}:
  solve(G1, {k,l})
  {}

```

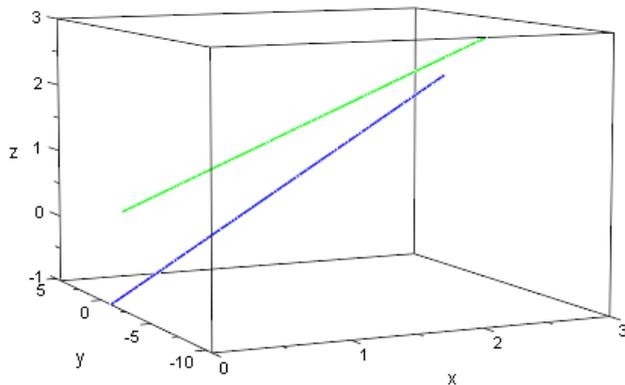
Das Gleichungssystem hat keine Lösung – die beiden Geraden sind also windschief. Zum Anschluss stellen wir – wie gewohnt – die Situation natürlich noch einmal grafisch dar:

```

• g:= plot::Curve3d(ParameterformG, k = -1..2,
                    Color = RGB::Blue):
  h:= plot::Curve3d(ParameterformH, l = -1..2,
                    Color = RGB::Green):
  plot(g,h)

```

<sup>13</sup>Wir ersparen es uns hier, diese triviale Rechnung wieder mit MuPAD durchzuführen. Wer sich noch ein wenig unsicher fühlt, der sollte noch einmal nach schauen, wie wir oben für das Paar der sich schneidenden Geraden vorgegangen sind und die Rechnung noch einmal selbst mit MuPAD durchführen.



### Übungen

21.) Untersuchen Sie die folgenden drei Geraden im  $\mathbb{R}^3$  auf ihre gegenseitige Lage zueinander:

$$g_1 : \vec{x} = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix} + k \cdot \begin{pmatrix} 3 \\ -2 \\ 2 \end{pmatrix},$$

$$g_2 : \vec{x} = \begin{pmatrix} -4 \\ 5 \\ -6 \end{pmatrix} + l \cdot \begin{pmatrix} -2 \\ 1 \\ -4 \end{pmatrix},$$

$$g_3 : \vec{x} = \begin{pmatrix} -2 \\ 3 \\ -4 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}.$$

Zeichnen Sie dazu alle drei Geraden in ein gemeinsames Koordinatensystem und stellen Sie eine Vermutung über die gegenseitige Lage auf. Weisen Sie Ihre Vermutung dann rechnerisch nach.

### 3.3.3 Zusammenfassung: Lagebeziehung zweier Geraden im Raum

Wir fassen die Ergebnisse der letzten beiden Unterabschnitte zur Lagebeziehung zweier Geraden zusammen: Wenn man die Parameterformen  $g : \vec{x} = \vec{a} + k \cdot \vec{u}$  und  $h : \vec{x} = \vec{b} + l \cdot \vec{v}$  zweier Geraden im  $\mathbb{R}^3$  gegeben hat, so kann man zur Untersuchung der gegenseitigen Lagebeziehung wie folgt vorgehen:

- 1) Prüfe, ob die Richtungsvektoren  $\vec{u}$  und  $\vec{v}$  skalare Vielfache voneinander sind.
  - 1.1) Sind  $\vec{u}$  und  $\vec{v}$  skalare Vielfache voneinander, so sind  $g$  und  $h$  entweder parallel oder identisch.

- 1.2) Sind  $\vec{u}$  und  $\vec{v}$  keine skalaren Vielfachen voneinander, so sind  $g$  und  $h$  entweder windschief oder sie schneiden sich in einem gemeinsamen (Schnitt)Punkt.
- 2) Löse das Gleichungssystem " $g = h$ ", d.h. das Gleichungssystem in zwei Variablen  $k$  und  $l$ , dass sich aus den drei Komponenten der Gleichung  $\vec{a} + k \cdot \vec{u} = \vec{b} + l \cdot \vec{v}$  ergibt.
- 2.1) Hat das Gleichungssystem keine Lösung, so sind die Geraden im Fall der Gültigkeit von 1.1) parallel und im Fall der Gültigkeit von 1.2) windschief.
- 2.2) Hat das Gleichungssystem genau eine Lösung, so schneiden sich die beiden Geraden. Den Schnittpunkt erhält man, indem man entweder den für  $k$  erhaltenen Wert in die Parameterform von  $g$  oder den für  $l$  erhaltenen Wert in die Parameterform von  $h$  einsetzt.
- 2.3) Hat das Gleichungssystem mehr also nur eine Lösung, so sind die beiden Geraden identisch.

### Übungen

- 22.) Untersuchen Sie die folgenden drei Geraden im  $\mathbb{R}^3$  auf ihre gegenseitige Lage zueinander:

$$g_1 : \vec{x} = \begin{pmatrix} -1 \\ 3 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix},$$

$$g_2 : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} + l \cdot \begin{pmatrix} 2 \\ 3 \\ 6 \end{pmatrix},$$

$$g_3 : \vec{x} = \begin{pmatrix} 3 \\ 3 \\ 5 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Zeichnen Sie dazu alle drei Geraden in ein gemeinsames Koordinatensystem und stellen Sie eine Vermutung über die gegenseitige Lage auf. Weisen Sie Ihre Vermutung dann rechnerisch nach.

### 3.4 Beschreibung von Ebenen

In den vorhergehenden Abschnitten haben wir uns mit der Beschreibung von Geraden im Raum beschäftigt und Lagebeziehungen von Punkten und Geraden sowie Lagebeziehungen von zweier Geraden untersucht. Ähnlich werden wir jetzt für Ebenen im

Raum vorgehen. Wir überlegen uns zunächst, wie man Ebenen sinnvoll mathematisch beschreiben kann (hier werden wir ähnlich wie bei Geraden im  $\mathbb{R}^2$  zwei Varianten kennen lernen – Parameterformen und Koordinatenformen von Ebenen). Dann untersuchen wir Lagebeziehungen von Punkten und Ebenen, Geraden und Ebenen sowie die möglichen Lagebeziehungen zwischen zwei Ebenen.

### 3.4.1 Parameterform von Ebenen

#### Neue MuPAD-Funktionen

`plot::Surface` Darstellung von Ebenen im Raum

Eine Ebene im Raum können wir uns wie eine Art unendlich weit ausgedehnte “Tischplatte” vorstellen. Um eine Darstellungsform für Ebenen zu erhalten, fragen wir uns, wie viele Tischbeine ein Tisch braucht, damit er aufgestellt werden kann und wie dieselben unter dem Tisch montiert werden müssen, damit er nicht umkippen kann.

Die Frage nach der Anzahl der Tischbeine ist leicht zu beantworten. Wir brauchen mindestens drei Tischbeine. Andernfalls fällt der Tisch in jedem Fall um. Gehen wir also davon aus, dass wir drei Tischbeine zur Verfügung haben: Wie müssen wir diese anbringen, damit der Tisch stehen kann? Wichtig ist im Prinzip nur, dass die Tischbeine nicht entlang einer geraden Linie an der Tischplatte angebracht werden, denn das hätte den gleichen Effekt, als wenn wir nur zwei Tischbeine verwenden und der Tisch könnte nicht stehen.

Die drei Tischbeine, die nötig sind, um einen Tisch zum Stehen zu bringen, entsprechen einfach drei verschiedenen Punkten  $A$ ,  $B$  und  $C$  im  $\mathbb{R}^3$ . Eine Ebene sollte sich also mit Hilfe von *drei vorgegebenen Punkten*, die natürlich alle verschieden sein müssen und nicht alle drei auf einer Geraden liegen dürfen, beschreiben lassen.

So weit, so gut! Wie bei Geraden müssen wir uns überlegen, wie wir einen beliebigen Punkt  $X$  auf der Ebene erreichen können. Ganz analog zur Vorgehensweise bei der Parameterform von Geraden starten wir beim Nullpunkt. Einen der Ortsvektoren der gegebenen Punkte  $A$ ,  $B$  und  $C$  – z.B. den Ortsvektor  $\vec{a}$  des Punktes  $A$  – wählen wir als Stützvektor für die Ebene. Diesen Vektor müssen wir zuerst “entlang fliegen”, um auf die Ebene zu gelangen. Ebenso wie bei der Parameterdarstellung von Geraden bilden wir den Differenzvektor  $\vec{b} - \vec{a}$  der beiden Ortsvektoren der Punkte  $A$  und  $B$  (dieser liegt dann in der Ebene, die wir beschreiben möchten und verläuft vom Punkt  $A$  zum Punkt  $B$ ). Zusätzlich bilden wir aber auch noch den Differenzvektor  $\vec{c} - \vec{a}$  (auch dieser liegt in der Ebene, die wir beschreiben wollen und verläuft entsprechend vom Punkt  $A$  zum Punkt  $C$ ). Die Vektoren  $\vec{b} - \vec{a}$  und  $\vec{c} - \vec{a}$  haben unter den gegebenen Voraussetzungen unterschiedliche Richtung. Man nennt sie die *Spannvektoren* der Ebene. Nun können wir einen beliebigen Punkt  $X$  auf der Ebene erreichen, indem wir zunächst den Vektor  $\vec{a}$

entlang fliegen und so auf die Ebene gelangen. Anschließend wählen wir ein geeignetes Vielfaches, mit dem wir die beiden Spannvektoren "verlängern" und setzen dann den einen Spannvektor an den anderen an. Ein Bild sagt hier mir als tausend Worte und deshalb veranschaulicht Abbildung 3.4 die beschriebene Vorgehensweise.

Mathematisch erhalten wir damit die folgende Darstellung für eine Ebene  $F$ , die durch die drei Punkte  $A$ ,  $B$  und  $C$  gegeben ist:

$$F : \vec{x} = \vec{a} + k \cdot (\vec{b} - \vec{a}) + l \cdot (\vec{c} - \vec{a}),$$

wobei wieder  $k$  und  $l$  reelle Zahlen sind.

Wir schauen uns ein konkretes Beispiel an: Gesucht ist eine Parameterform derjenigen Ebene  $F$ , die durch die Punkte  $A(1|0|1)$ ,  $B(-1|1|2)$  und  $C(0|1|1)$  gegeben ist. Zuerst definieren wir die Ortsvektoren der Punkte  $A$ ,  $B$  und  $C$  in MuPAD:

- `OrtsvektorA:= matrix([ 1,0,1]):`
- `OrtsvektorB:= matrix([-1,1,2]):`
- `OrtsvektorC:= matrix([ 0,1,1]):`

Die Parameterform der Ebene  $F$  können wir dann wie schon von der Parameterform von Geraden gewohnt in MuPAD eingeben<sup>14</sup>:

- `ParameterformF:= OrtsvektorA +`  
`k * (OrtsvektorB - OrtsvektorA) +`  
`l * (OrtsvektorC - OrtsvektorA)`

$$\begin{pmatrix} 1 - l - 2k \\ k + l \\ k + 1 \end{pmatrix}$$

Die Ebene lässt sich zusammen mit den drei Punkten mit Hilfe des Befehls `plot::Surface` in ein gemeinsames Koordinatensystem zeichnen<sup>15</sup>:

- `F:= plot::Surface(ParameterformF, k = -1..2, l = -1..2):`
- `A:= plot::Point3d(OrtsvektorA, Color = RGB::Blue):`
- `B:= plot::Point3d(OrtsvektorB, Color = RGB::Red):`
- `C:= plot::Point3d(OrtsvektorC, Color = RGB::Green):`
- `plot(F,A,B,C)`

<sup>14</sup>MuPAD fasst dabei stets den Stützvektor und die Vielfachen der Spannvektoren wie schon bei der Parameterform von Geraden zu einem Vektor zusammen. Mit etwas künstlichem Aufwand lässt sich dieses Verhalten umgehen. Jedoch widerspricht es im Prinzip der Natur eines Computeralgebra System, das u.a. dazu dient "Dinge aus zurechnen": MuPAD rechnet und fasst daher die Summe der Vektoren zu einem Vektor zusammen.

<sup>15</sup>Dabei sind für die beiden Parameter  $k$  und  $l$  aus der Parameterform der Ebene jeweils Bereiche anzugeben, die die Parameter durchlaufen sollen (ganz analog zur Vorgehensweise bei der Darstellung von Geraden mit Hilfe von `plot::Curve2d` oder `plot::Curve3d`).

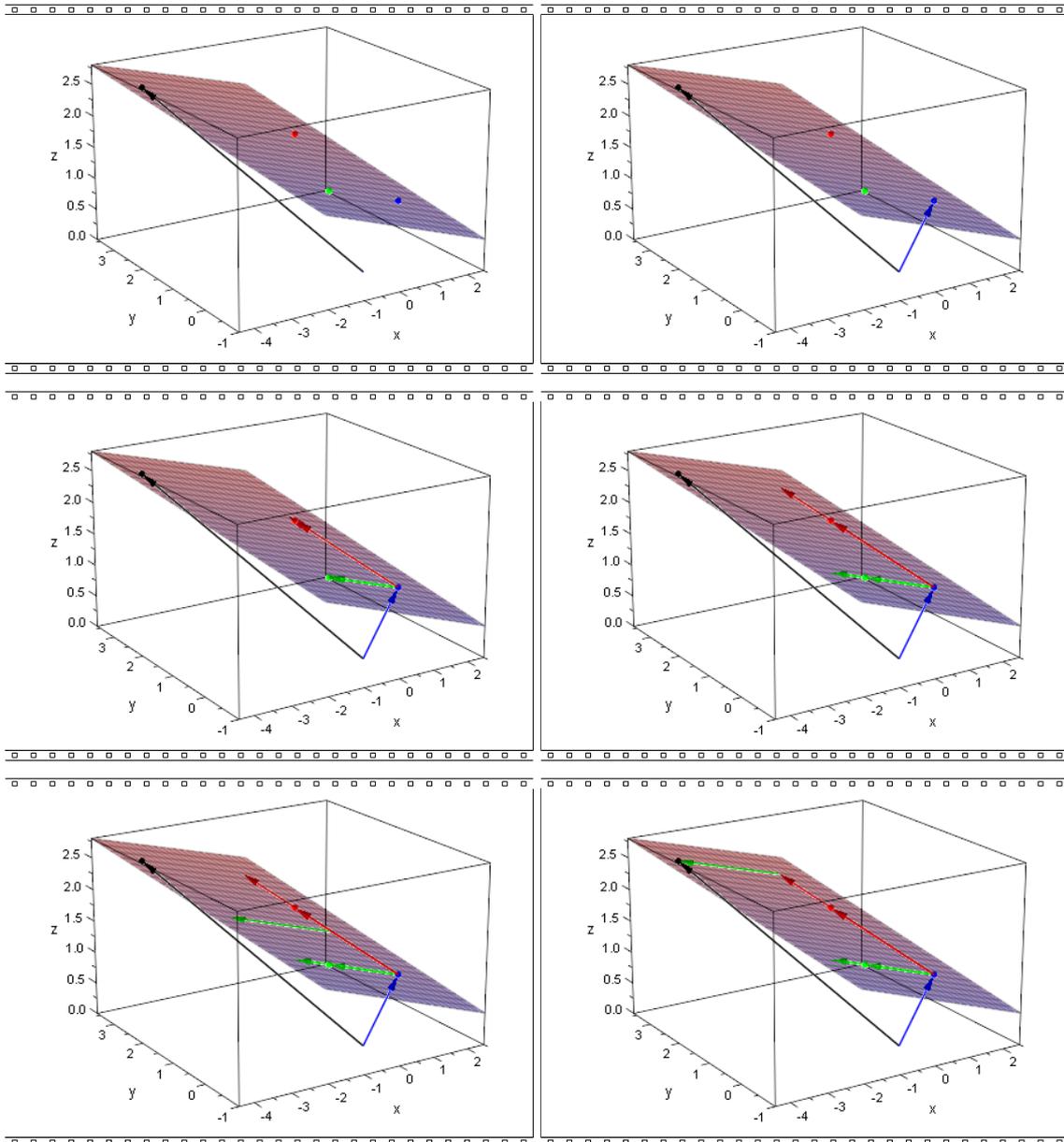
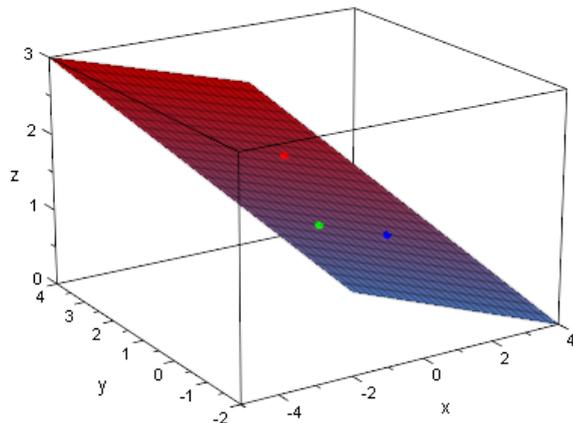
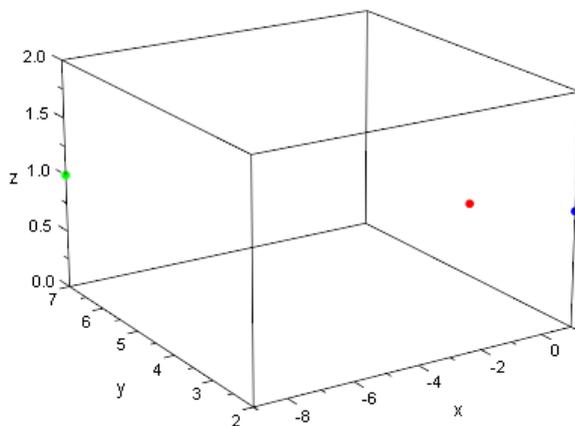


Abbildung 3.4: Parameterform einer Ebene



Die folgenden drei Punkte  $A(1|2|1)$ ,  $B(-1|3|1)$  und  $C(-9|7|1)$  definieren *keine Ebene* im Raum. Warum? Nun, schauen wir uns ihre Lage im Koordinatensystem zunächst einmal mit Hilfe von MuPAD an<sup>16</sup>:

- `OrtsvektorA:= matrix([ 1,2,1]):`
- `OrtsvektorB:= matrix([-1,3,1]):`
- `OrtsvektorC:= matrix([-9,7,1]):`
- `A:= plot::Point3d(OrtsvektorA, Color = RGB::Blue):`
- `B:= plot::Point3d(OrtsvektorB, Color = RGB::Red):`
- `C:= plot::Point3d(OrtsvektorC, Color = RGB::Green):`
- `plot(A,B,C)`



Die Grafik vermittelt zunächst nicht den Eindruck, dass die drei Punkte keine Ebene bestimmen. Mit einem Doppelklick auf die Grafik kann diese in MuPAD aktiviert werden und man kann die gesamte grafische Szene drehen. Tut man das, so erhält man z.B. eine Sicht auf die Szene wie in [Abbildung 3.5](#).

<sup>16</sup>Wir definieren zuerst die Ortsvektoren der drei Punkte und dann die zugehörigen grafischen Objekte mit Hilfe von `plot::Point3d`, auch wenn dies hier ein wenig umständlich wirkt (man könnte natürlich darauf verzichten, die Ortsvektoren vorher explizit zu definieren, wenn es nur darum geht, die Punkte zu zeichnen). Weiter unten wollen wir aber mit den Ortsvektoren rechnen können.

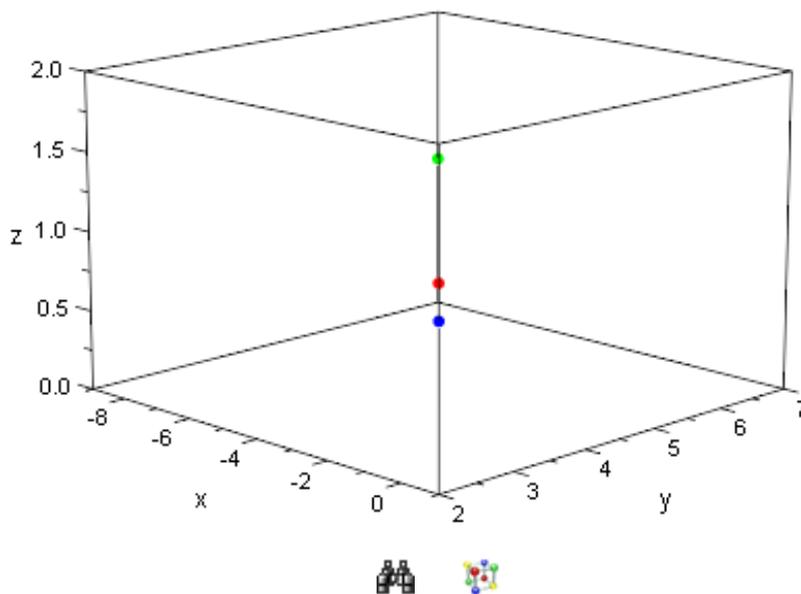


Abbildung 3.5: Drei Punkte auf einer Geraden im Raum

Abbildung 3.5 lässt vermuten, dass die drei Punkte  $A$ ,  $B$  und  $C$  auf einer Geraden liegen. Wenn die Punkte  $A$ ,  $B$  und  $C$  eine Ebene bestimmen würden, dann dürfen die Spannvektoren der Ebene

- Ortsvektor $B$  - Ortsvektor $A$ ,  
Ortsvektor $C$  - Ortsvektor $A$

$$\begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -10 \\ 5 \\ 0 \end{pmatrix}$$

weder die gleiche noch genau die entgegengesetzte Richtung haben. Dies ist aber nicht der Fall, denn offensichtlich ist der zweite Vektor genau das Fünffache des ersten Vektors. Die drei Punkte liegen in der Tat auf einer Geraden und bestimmen daher keine Ebene.

### Übungen

- 23.) Bestimmen Sie – falls möglich – eine Parameterform derjenigen Ebene, die durch die Punkte  $A(1|1|2)$ ,  $B(1|0|0)$  und  $C(-1|0|1)$  gegeben ist. Stellen Sie sowohl die drei Punkte als auch (falls möglich) die dadurch bestimmte Ebene in einem gemeinsamen Koordinatensystem dar.
- 24.) Bestimmen Sie – falls möglich – eine Parameterform derjenigen Ebene, die durch die Punkte  $A(1|1|1)$ ,  $B(1|3|1)$  und  $C(1|14|1)$  gegeben ist. Stellen Sie sowohl die

drei Punkte als auch (falls möglich) die dadurch bestimmte Ebene in einem gemeinsamen Koordinatensystem dar.

### 3.4.2 Koordinatenform von Ebenen

Ähnlich wie bei Geraden im  $\mathbb{R}^2$  gibt es auch für Ebenen im  $\mathbb{R}^3$  eine Koordinatenform, d.h. eine die Ebene bestimmende Gleichung, in der direkt die Koordinaten  $x$ ,  $y$  und  $z$  eines beliebigen Punktes  $X(x|y|z)$  auf der Ebene vorkommen. In Abschnitt 3.1.2 hatten wir ein Verfahren kennen gelernt, wie wir aus einer gegebenen Parameterform einer Geraden im  $\mathbb{R}^2$  eine Koordinatenform bestimmen konnten. Der wesentliche Trick bestand darin, einen so genannten *Normalenvektor*  $\vec{n}$  zu finden, der senkrecht auf dem Richtungsvektor der Geraden stand. Ganz analog können wir zum Auffinden der Koordinatenform einer gegebenen Ebene

$$F : \vec{x} = \vec{a} + k \cdot \vec{u} + l \cdot \vec{v}$$

vorgehen. Wir müssen hier nur einen Vektor  $\vec{n}$  finden, der senkrecht auf beiden Spannvektoren  $\vec{u}$  und  $\vec{v}$  der Ebene  $F$  steht. Das klingt kompliziert, aber im Prinzip können wir das schon.

Wir haben in Abschnitt 2.6 das *Kreuzprodukt zweier Vektoren* des  $\mathbb{R}^3$  hergeleitet. Das Kreuzprodukt von  $\vec{u}$  und  $\vec{v}$  ist wieder ein Vektor des  $\mathbb{R}^3$ , der genau die gewünschte Eigenschaft hat (er ist orthogonal sowohl zu  $\vec{u}$  als auch zu  $\vec{v}$ ).

Ist also  $\vec{n}$  das Kreuzprodukt der beiden Vektoren  $\vec{u}$  und  $\vec{v}$ , so gilt für das Skalarprodukt von  $\vec{n}$  mit  $\vec{u}$  sowie mit  $\vec{v}$ :

$$\vec{n} * \vec{u} = 0 \quad \text{und} \quad \vec{n} * \vec{v} = 0.$$

Damit gilt aber auch  $\vec{n} * (k \cdot \vec{u} + l \cdot \vec{v}) = 0$ , d.h.

$$0 = \vec{n} * \underbrace{(k \cdot \vec{u} + l \cdot \vec{v})}_{=\vec{x}-\vec{a}} = \vec{n} * (\vec{x} - \vec{a}).$$

Die Gleichung  $\vec{n} * (\vec{x} - \vec{a}) = 0$  ist genau das, was man als eine *Koordinatenform* der Ebene  $F$  bezeichnet.

Schauen wir uns ein Beispiel an. Im letzten Abschnitt hatten wir zu Beginn eine Parameterform der Ebene  $F$  berechnet, die durch die Punkte  $A(1|0|1)$ ,  $B(-1|1|2)$  und  $C(0|1|1)$  bestimmt war:

```

• OrtsvektorA:= matrix([ 1,0,1]):
  OrtsvektorB:= matrix([-1,1,2]):
  OrtsvektorC:= matrix([ 0,1,1]):
  u:= OrtsvektorB - OrtsvektorA:
  v:= OrtsvektorC - OrtsvektorA:
  ParameterformF:= OrtsvektorA + k * u + l * v

```

$$\begin{pmatrix} 1 - l - 2k \\ k + l \\ k + 1 \end{pmatrix}$$

Wir wollen jetzt eine Normalenform dieser Ebene bestimmen: Einen Normalenvektor erhalten über das Kreuzprodukt der Spannvektoren der Ebene, also<sup>17</sup>:

- `n:= linalg::crossProduct(u,v)`

$$\begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

Damit ist also eine Koordinatenform der Ebene  $F$  gegeben durch das Skalarprodukt<sup>18</sup>:

- `assume({x,y,z}, Type::Real):`  
`OrtsvektorX:= matrix([x,y,z]):`  
`KForm:= linalg::scalarProduct(n,`  
`OrtsvektorX - OrtsvektorA) = 0`  
 $2 - y - z - x = 0$

Wir können mit MuPAD natürlich auch Ebenen in Koordinatenform grafisch darstellen lassen<sup>19</sup>. Dazu lösen wir die Koordinatenform `KForm` z.B. nach `z` auf, damit wir die Ebene mit Hilfe von `plot::Surface` zeichnen lassen können<sup>20</sup>.

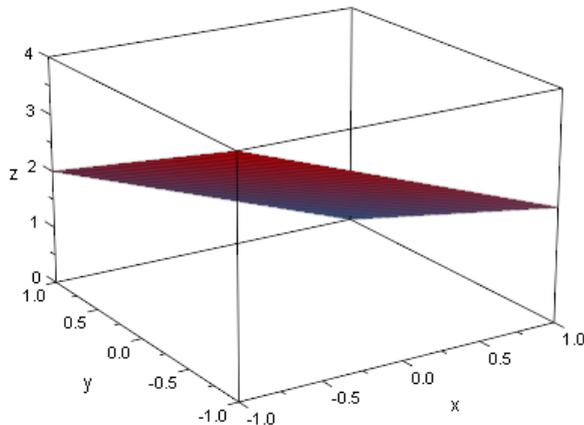
<sup>17</sup>Das Kreuzprodukt zweier Vektoren hatten wir mit dem Befehl `linalg::crossProduct` berechnet.

<sup>18</sup>Das Skalarprodukt zweier Vektoren hatten wir mit dem Befehl `linalg::scalarProduct` berechnet.

<sup>19</sup>In MuPAD kann man Ebenen auf mindestens drei Arten zeichnen lassen. Die erste Möglichkeit besteht in der Verwendung der Funktion `plot::Surface`, die wir auch hier verwenden. Diese hat den Vorteil, dass man bei der grafischen Darstellung echte Lichtreflexe auf der Ebene ausmachen kann. Gleiches ist auch der Fall, wenn man die Funktion `plot::Implicit3d` verwendet (nähere Details zur Verwendung dieser MuPAD-Funktion finden sich in der Online-Dokumentation). Die Funktion `plot::Implicit3d` hat den Vorteil, dass man die Koordinatenform direkt in der oben gegebenen Form übergeben kann. Allerdings ist `plot::Implicit3d` im Prinzip zur Darstellung allgemeiner impliziter Flächen gedacht und ist daher auch über viel kompliziertere Algorithmen intern in MuPAD realisiert. Daher sollte man – aus reiner Effizienz – eher auf `plot::Implicit3d` zur Darstellung von Ebenen verzichten. Die dritte Möglichkeit zur Darstellung von Ebenen bietet die Funktion `plot::Parallelogram3d`, die jedoch zur Darstellung von Ebenen in Koordinatenform nicht gut geeignet ist, denn man muss die Koordinatenform in diesem Fall zunächst erst wieder in eine Parameterform umrechnen (nähere Details zu `plot::Parallelogram3d` finden sich ebenfalls in der Online-Dokumentation zu MuPAD). Andererseits ist `plot::Parallelogram3d` aber sehr effizient, was die Verwendung dieser Funktion insbesondere im Kontext von Animationen wiederum interessant macht. Der Benutzer sollte also hier die für ihn günstigste Darstellungsmöglichkeit selbst auswählen.

<sup>20</sup>Das Auflösen der Koordinatenform nach einer Variablen mag ein wenig umständlich erscheinen. Es ist nötig, da sich mit `plot::Surface` im Prinzip eigentlich nur Parameterformen von Ebenen zeichnen lassen. Uns erschien es sinnvoll, die aus dem vorhergehenden Abschnitt bereits bekannte Funktion zur Darstellung von Ebenen auch hier weiter zu verwenden und für die Darstellung von Ebenen in Koordinatenform nicht eine weitere neue MuPAD-Funktion zu “beanspruchen”.

- `F:= plot::Surface([x, y, solve(KForm, z)[1]],  
x = -1..1, y = -1..1):  
plot(F)`



Zum Anschluss dieses Abschnitts wollen wir noch kurz überlegen, wie man aus einer Koordinatenform einer gegebenen Ebenen eine Parameterform berechnen kann. Das ist aber recht einfach. Wir nehmen uns dazu einfach die Koordinatenform der Ebene, in unserem Beispiel also die Gleichung

- `KForm`  
 $2 - y - z - x = 0$

und lösen diese nach einer der vorkommenden Variablen, also z.B. nach  $z$ , auf:

- `solve(KForm, z)`  
 $\{2 - y - x\}$

Wenn wir also Werte für  $x$  und  $y$  vorgeben, so ist der Wert von  $z$  durch die Gleichung  $z = 2 - y - x$  festgelegt. Schreiben wir die drei Koordinaten als Komponenten des zugehörigen Ortsvektors, so erhalten wir:

$$\begin{pmatrix} x \\ y \\ 2 - y - x \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} + x \cdot \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} + y \cdot \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

Wenn wir jetzt noch  $x$  durch  $k$  und  $y$  durch  $l$  ersetzen, so haben wir also eine Parameterform in der uns bekannten Gestalt bestimmt.

Später werden wir sehen, dass es in der Regel angenehmer ist, mit Ebenen in Koordinatenform zu arbeiten<sup>21</sup>. Daher werden wir fast immer so vorgehen, dass wir bei drei

<sup>21</sup>Dies ist insbesondere dann der Fall, wenn man lästige Rechenaufgaben wie das Lösen von Gleichungssystemen per Hand erledigen muss und nicht an ein Computeralgebra-System wie MuPAD delegieren kann.

gegebenen Punkten zwar eine Ebenengleichung zunächst in Parameterform aufstellen, diese dann aber mehr oder weniger unmittelbar in Koordinatenform überführen.

### Übungen

25.) Bestimmen Sie für die folgenden Ebenen je eine Koordinatenform:

$$F : \vec{x} = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 1 \\ -2 \end{pmatrix},$$

$$G : \vec{x} = \begin{pmatrix} -3 \\ -1 \\ 2 \end{pmatrix} + k \cdot \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 1 \\ 1 \\ -5 \end{pmatrix}.$$

Stellen Sie mit MuPAD jede Ebene einzeln in einem 3D-Koordinatensystem dar.

## 3.5 Lagebeziehung von Ebenen und Punkten

Wie schon in Abschnitt 3.2, wo wir die möglichen Lagebeziehungen von Geraden und Punkten untersucht hatten, untersuchen wir zuerst die möglichen Lagebeziehungen von Ebenen und Punkten, bevor wir zu interessanteren Problemstellungen kommen. Haben wir eine Ebene  $F$  und einen Punkt  $P$  im  $\mathbb{R}^3$  gegeben, so gibt es zwei Möglichkeiten: Entweder der Punkt  $P$  liegt auf der Ebene  $F$  oder der Punkt liegt nicht auf der Ebene  $F$ . Wir wollen uns im folgenden zunächst überlegen, wie wir prüfen können, ob der gegebene Punkt  $P$  auf der Ebene  $F$  liegt, wenn  $F$  in Parameterform gegeben ist. Ist

$$F : \vec{x} = \vec{a} + k \cdot \vec{u} + l \cdot \vec{v}$$

eine Parameterform der Ebene  $F$ , so liegt  $P$  auf  $F$ , wenn es spezielle reelle Zahlen  $k$  und  $l$  gibt, so dass für den Ortsvektor des Punktes  $P$  gilt

$$\vec{p} = \vec{a} + k \cdot \vec{u} + l \cdot \vec{v}.$$

Letztere Gleichung liefert uns ein Gleichungssystem mit drei Gleichungen in den beiden Unbekannten  $k$  und  $l$ , die es zu bestimmen gilt. Wir betrachten ein konkretes Beispiel. Seien  $P(-3 | -7/2 | -2)$  und  $Q(7|3|5)$  gegeben sowie die Ebene  $F$  in Parameterform

$$F : \vec{x} = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 1 \\ -2 \end{pmatrix}.$$

Wir definieren zunächst die Ortsvektoren von  $P$  und  $Q$  sowie die Parameterform der Ebene  $F$  in MuPAD:

- ```

p:= matrix([-3,-7/2,-2]):
q:= matrix([ 7, 3, 5]):
ParameterformF:=      matrix([1,-2, 1]) +
                      k * matrix([2, 1, 1]) +
                      l * matrix([0, 1,-2]):

```

Wir müssen jetzt das Gleichungssystem lösen, dass durch die Gleichung

$$\begin{pmatrix} -3 \\ -7/2 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 1 \\ -2 \end{pmatrix}$$

gegeben ist. Die Lösung des Gleichungssystems erhalten wir dann wie üblich mit Hilfe der Funktion `solve`:

- ```

G1:= {p[1] = ParameterformF[1],
      p[2] = ParameterformF[2],
      p[3] = ParameterformF[3]}:
solve(G1, {k,l})

```

$$\left\{ \left[ k = -2, l = \frac{1}{2} \right] \right\}$$

Das Gleichungssystem ist lösbar, d.h. der Punkt  $P$  liegt auf der Ebene  $F$ . Wenn wir in die Parameterform der Ebene  $F$  für  $k$  den Wert  $-2$  und für  $l$  den Wert  $-1/2$  einsetzen, so erhalten wir genau den Ortsvektor des Punktes  $P$ . Der Punkt  $Q$  liegt dagegen nicht auf der Ebene  $F$ , denn das entsprechende Gleichungssystem ist für  $Q$  nicht lösbar:

- ```

G1:= {q[1] = ParameterformF[1],
      q[2] = ParameterformF[2],
      q[3] = ParameterformF[3]}:
solve(G1, {k,l})

```

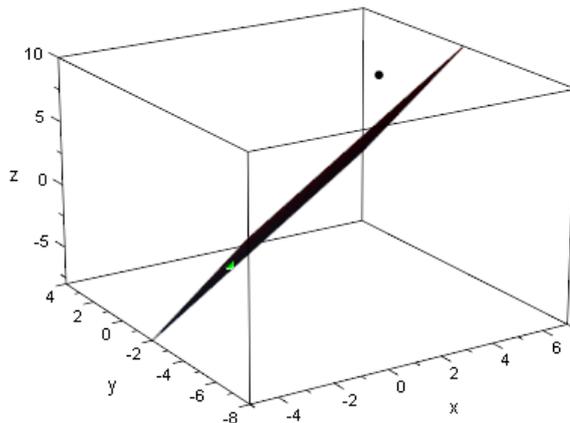
$$\{ \}$$

Bevor wir zu dem Fall kommen, wo die Ebene  $F$  in Koordinatenform gegeben ist, stellen wir die Ebene  $F$  zusammen mit den Punkten  $P$  und  $Q$  in einem gemeinsamen Koordinatensystem dar:

- ```

P:= plot::Point3d(p, Color = RGB::Green):
Q:= plot::Point3d(q, Color = RGB::Black):
F:= plot::Surface(ParameterformF, k = -3..3, l = -3..3):
plot(P,Q,F)

```



Eine Koordinatenform der Ebene  $F$  erhalten wir nach dem in Abschnitt 3.4.2 kennen gelerntem Verfahren wie folgt:

- ```
n:= linalg::crossProduct(matrix([2, 1, 1]),
                           matrix([0, 1,-2]));
assume({x,y,z}, Type::Real):
KForm:= linalg::scalarProduct(n,
                               matrix([x,y,z]) - matrix([1,-2, 1])) = 0
4y - 3x + 2z + 9 = 0
```

Haben wir eine Koordinatenform der Ebene  $F$  vorliegen, so müssen wir kein Gleichungssystem mehr lösen, um zu prüfen ob ein gegebener Punkt auf der Ebene liegt. Wir müssen nur die Koordinaten des jeweiligen Punktes für  $x$ ,  $y$  und  $z$  in die Koordinatenform der Ebene einsetzen und prüfen, ob die Gleichung erfüllt ist<sup>2223</sup>:

- ```
bool(subs(KForm, x = p[1], y = p[2], z = p[3]));
bool(subs(KForm, x = q[1], y = q[2], z = q[3]))
TRUE
```

FALSE

Die erste Gleichung ist erfüllt, d.h.  $P$  liegt auf der Ebene  $F$ . Die zweite Gleichung ist nicht erfüllt, d.h.  $Q$  liegt nicht auf der Ebene  $F$ .

## Übungen

- 26.) Welche Lagebeziehung haben die drei Punkte  $A(6|5|2)$ ,  $B(1|-2|3)$  und  $C(10|6|11)$  zu der Ebene

$$F: \vec{x} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + l \cdot \begin{pmatrix} 3 \\ 4 \\ -2 \end{pmatrix}?$$

<sup>22</sup>Wir verwenden hier wieder die Funktion `bool`, um zu prüfen, ob die entsprechenden Gleichungen erfüllt sind.

<sup>23</sup>Zum Einsetzen der Werte für  $x$ ,  $y$  und  $z$  verwenden wir wie gewohnt den Befehl `subs`.

27.) Welche Lagebeziehung haben die drei Punkte  $A(1|1|1)$ ,  $B(2|1|-2)$  und  $C(-6|7/12|1/3)$  zu der Ebene

$$F : 2 \cdot x - 4 \cdot y + z - 2 = 0?$$

28.) Berechnen Sie  $k$ , so dass der Punkt  $A(1|2|4 \cdot k)$  auf der Ebene

$$F : -5 \cdot x + y - 6 \cdot z - 1 = 0$$

liegt.

### 3.6 Lagebeziehung von Gerade und Ebene

Für die Lagebeziehung einer Geraden  $g$  und einer Ebene  $F$  im Raum gibt es drei Möglichkeiten: Die Gerade kann die Ebene in einem Punkt durchstoßen, sie kann parallel zur Ebene verlaufen oder sie kann in der Ebene liegen. Abbildung 3.6 zeigt die drei möglichen Situationen.

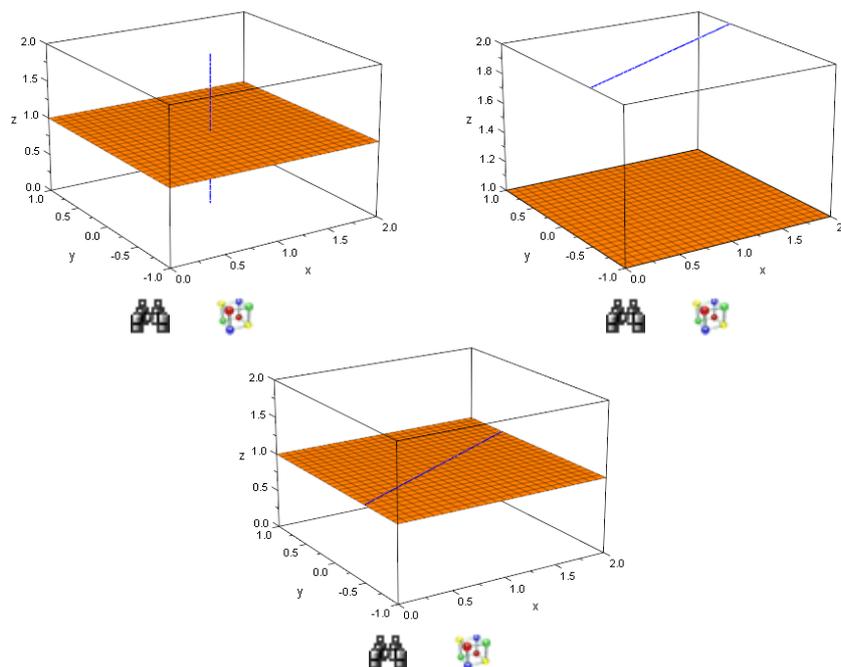


Abbildung 3.6: Lagebeziehungen von Gerade und Ebene

Wie im vorhergehenden Abschnitt müssen wir zwischen den beiden Fällen unterscheiden, ob die Ebene  $F$  in Parameterform oder in Koordinatenform gegeben ist.

### 3.6.1 Ebene in Parameterform

Wir beginnen wir mit dem Fall, dass  $F$  in Parameterform

$$F : \vec{x} = \vec{a} + k \cdot \vec{u} + l \cdot \vec{v}$$

gegeben ist. Da für Geraden im  $\mathbb{R}^3$  keine Koordinatenform existiert (vergleiche Abschnitt 3.1.2), liegt  $g$  in jedem Fall in Parameterform vor:

$$g : \vec{x} = \vec{a} + r \cdot \vec{w}.$$

Wenn wir die Lagebeziehung von  $F$  und  $g$  bestimmen wollen, dann können wir immer so vorgehen, dass wir ihre Parameterformen gleichsetzen und auf diese Weise versuchen, gemeinsame Punkte zu berechnen. Dies führt immer auf ein Gleichungssystem bestehend aus drei Gleichungen mit drei Unbekannten (den Parameter  $k$ ,  $l$  und  $r$  in unserer Bezeichnungsweise). Wir müssen dann drei mögliche Fälle unterscheiden:

- 1) Wenn das Gleichungssystem genau eine Lösung hat, so liefert uns diese die Werte für  $k$ ,  $l$  und  $r$ , die wir nur noch in die Ebenengleichung bzw. in die Geradengleichung einsetzen müssen, um den Schnittpunkt von  $g$  und  $F$  zu berechnen.
- 2) Wenn das Gleichungssystem keine Lösung hat, so haben  $g$  und  $F$  keine gemeinsamen Punkte und  $g$  muss folglich parallel zu  $F$  verlaufen.
- 3) Wenn das Gleichungssystem unendlich viele Lösungen besitzt<sup>24</sup>, so haben  $F$  und  $g$  unendlich viele gemeinsame Punkte, d.h.  $g$  muss in  $F$  liegen.

Für jeden der drei möglichen Fälle betrachten wir jetzt ein Beispiel. Seien also zunächst

$$F : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad g : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Wir definieren die Parameterformen in MuPAD

- `ParameterformF:= matrix([1,0,1]) + k * matrix([1,0,0])  
+ l * matrix([0,1,0]):`  
`ParameterformG:= matrix([1,0,1]) + r * matrix([0,0,1]):`

und lösen das Gleichungssystem, dass sich ergibt, wenn wir die Parameterform von  $g$  und die Parameterform von  $F$  gleichsetzen:

- `solve({ParameterformF[1] = ParameterformG[1],  
ParameterformF[2] = ParameterformG[2],  
ParameterformF[3] = ParameterformG[3]},  
{k,l,r})`

<sup>24</sup>Man sagt auch, dass die Lösungsmenge in diesem Fall *allgemein gültig* ist.

$$\{[k = 0, l = 0, r = 0]\}$$

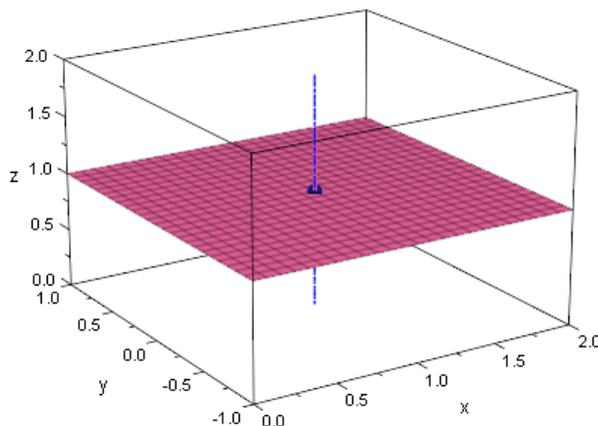
Das Gleichungssystem hat also genau eine Lösung, d.h. die Gerade  $g$  schneidet die Ebene  $F$  in einem Punkt  $S$ , dessen Ortsvektor wir erhalten, indem wir entweder die Werte für  $k$  und  $l$  in die Ebenengleichung oder den Wert für  $r$  in die Geradengleichungen einsetzen. Wir setzen  $r = 0$  in die Geradengleichung ein und erhalten:

- `OrtsvektorS := subs(ParameterformG, r = 0)`

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

Der Schnittpunkt  $S$  hat also die Koordinaten  $S(1|0|1)$ <sup>25</sup>. Zeichnen wir mit MuPAD die Ebene  $F$ , die Gerade  $g$  sowie den Punkt  $S$  in ein gemeinsames Koordinatensystem, so erhalten wir<sup>26</sup>:

- `F := plot::Surface(ParameterformF, k = -1..1, l = -1..1):`  
`g := plot::Curve3d(ParameterformG, r = -1..1):`  
`S := plot::Point3d(OrtsvektorS, PointSize = 4):`  
`plot(F,g,S)`



Als nächstes betrachten wir ein Beispiel für eine Gerade, die parallel zu einer Ebene verläuft. Wir übernehmen die Ebene  $F$  bzw. ihre Parameterdarstellung von oben und betrachten statt der oben definierten Gerade  $g$  die neue Gerade

$$g : \vec{x} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} + r \cdot \begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix}.$$

<sup>25</sup>Was wenig überraschend ist, denn der Ortsvektor dieses Punktes wird sowohl in der Parameterform der Ebene  $F$  als auch in der Parameterform der Gerade  $g$  als Stützvektor verwendet.

<sup>26</sup>Mit Hilfe des Attributs `PointSize` haben wir die Größe des Schnittpunktes stark erhöht, so dass er in der Grafik einfach besser zu erkennen ist.

Wir gehen genauso wie oben vor, d.h. wir versuchen wieder über Gleichsetzen der Parameterformen von  $F$  und  $g$  bzw. Lösen des dadurch gegebenen linearen Gleichungssystems gemeinsame Punkte von  $g$  und  $F$  zu berechnen. Die Parameterform von  $F$  hatten wir oben bereits in MuPAD definiert, so dass wir nur die Parameterform der Geraden  $g$  neu in MuPAD definieren müssen:

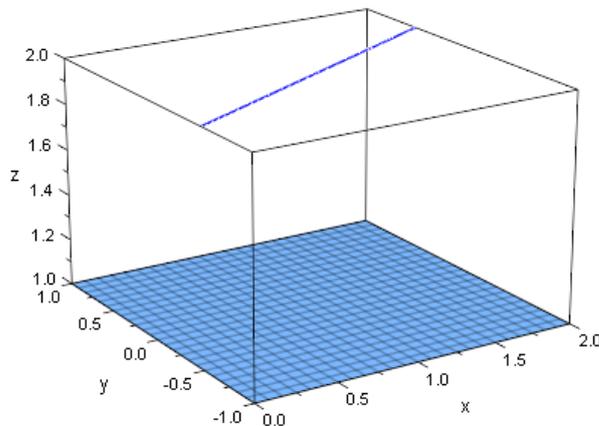
- `ParameterformG:= matrix([3,1,2]) + r * matrix([-2,-1,0]):`

Ganz analog zur obigen Vorgehensweise lösen wir das lineare Gleichungssystem mit Hilfe der MuPAD-Funktion `solve`:

- `solve({ParameterformF[1] = ParameterformG[1],  
ParameterformF[2] = ParameterformG[2],  
ParameterformF[3] = ParameterformG[3]},  
{k,l,r})`  
`}`

Die Lösungsmenge ist leer, d.h.  $g$  und  $F$  haben keine gemeinsamen Punkte. Damit muss  $g$  parallel zu  $F$  verlaufen. Grafisch erhalten wir<sup>27</sup>:

- `g:= plot::Curve3d(ParameterformG, r = 1/2..3/2):`  
`plot(F,g)`



Jetzt fehlt uns noch ein Beispiel für eine Gerade, die in einer Ebene liegt. Wir übernehmen wieder die Ebene  $F$  aus den vorhergehenden beiden Beispielen und wählen eine neue Gerade  $g$  mit der Parameterform<sup>28</sup>

$$g : \vec{x} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix}.$$

<sup>27</sup>Die Ebene  $F$  hatten wir oben bereits gezeichnet und das zugehörige grafische Objekt, das mittels `plot::Surface` erzeugt wurde, in der Variablen `F` abgespeichert.

<sup>28</sup>Hier ist Vorsicht geboten: Die Geradengleichung der Gerade  $g$  sieht fast genau so aus, wie die Gleichung der Gerade, die wir oben betrachtet hatten. Sie unterscheidet sich nur in der dritten Komponente des Stützvektors.

Wir definieren die Parameterform der Gerade  $g$  wieder in MuPAD und lösen sofort das nötige Gleichungssystem, dessen Lösungsmenge uns Aufschluss über die Lagebeziehung von  $g$  und  $F$  liefert:

- ```

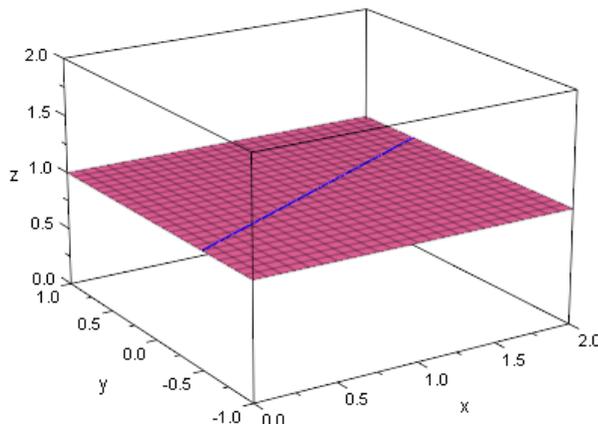
ParameterformG:= matrix([3,1,1]) + r * matrix([-2,-1,0]):
solve({ParameterformF[1] = ParameterformG[1],
      ParameterformF[2] = ParameterformG[2],
      ParameterformF[3] = ParameterformG[3]},
      {k,l,r})
      
```

$$\{[k = 2 - 2r, l = 1 - r]\}$$

Die Lösungen für  $k$  und  $l$  wurden von MuPAD in Abhängigkeit des Parameters  $r$  ausgedrückt, d.h. wir können einen beliebigen Wert für  $r$  wählen und erhalten mit  $k = 2 - 2 \cdot r$  sowie  $l = 1 - r$  stets eine Lösung des Gleichungssystem. Das Gleichungssystem hat also unendlich viele Lösungen und  $g$  liegt folglich in der Ebene  $F$ . Zur Veranschaulichung lassen wir auch die die Gerade und die Ebene in einem gemeinsamen Koordinatensystem darstellen:

- ```

g:= plot::Curve3d(ParameterformG, r = 1/2..3/2):
plot(F,g)
      
```



### 3.6.2 Ebene in Koordinatenform

Zum Anschluss dieses Abschnitts müssen wir uns noch überlegen, wie wir die Lagebeziehung einer Geraden  $g$  und einer Ebene  $F$  bestimmen, wenn die Ebene in Koordinatenform gegeben ist. Die Vorgehensweise hier ist allerdings viel einfacher. Angenommen, die Gerade liegt in der Ebene. Ihre Parameterform ist ganz allgemein von der Gestalt

$$g : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + r \cdot \begin{pmatrix} w_1 \\ -w_2 \\ w_3 \end{pmatrix}$$

und die Koordinatenform der Ebene  $F$  von der Gestalt

$$b_1 \cdot x + b_2 \cdot y + b_3 \cdot z + c = 0.$$

Wenn nun  $g$  in der Ebene  $F$  liegt, so liegt *jeder Punkt* der Geraden in  $F$ , d.h. alle Punkte mit den Koordinaten  $x = a_1 + r \cdot w_1$ ,  $y = a_2 + r \cdot w_2$  und  $z = a_3 + r \cdot w_3$  müssen die Ebenengleichung erfüllen. Wenn wir also diese Koordinaten in Koordinatenform einsetzen, so müsste – in dem Fall  $g$  in der Tat in der Ebene  $F$  liegt – die entstehende Gleichung unendlich viele Lösungen haben. Es sind aber theoretisch noch zwei weitere Fälle möglich: Die Gleichung ist nur für einen speziellen Wert von  $r$  erfüllt oder sie ist für keinen Wert von  $r$  erfüllt. Wenn die entstehende Gleichung für genau einen Wert von  $r$  erfüllt ist, so können wir diesen berechnen, indem wir die Koordinaten eines beliebigen Punktes auf der Geraden in die Koordinatenform der Ebene einsetzen. Den Wert für  $r$  brauchen wir dann nur noch in die Geradengleichung einsetzen und erhalten so die Koordinaten des Schnittpunktes der Geraden mit der Ebene. Wenn dagegen die Gleichung für keinen Wert von  $r$  erfüllt ist, dann haben  $F$  und  $g$  keinen gemeinsamen Punkt, d.h.  $g$  muss parallel zu  $F$  verlaufen.

Wir schauen uns wieder entsprechende Beispiele an. Gegeben sei die Ebene  $F$  mit der folgenden Koordinatenform

- `KForm:= 2*x + 4*y + z + 1 = 0:`

sowie die drei Geraden in Parameterform

- `g1:= matrix([2,4, 0]) + k * matrix([2,4, 1]):`  
`g2:= matrix([0,1,-1]) + l * matrix([1,0,-2]):`  
`g3:= matrix([0,0,-1]) + r * matrix([0,1,-4]):`

Die Gerade `g1` schneidet die Ebene  $F$ , denn setzen wir die Komponenten der Parameterform von `g1` in die Koordinatenform `KForm` der Ebene  $F$  ein, so erhalten wir eine Gleichung

- `G1:= subs(KForm, x = g1[1],`  
`y = g1[2],`  
`z = g1[3])`

$$21k + 21 = 0$$

die wir mit Hilfe von `solve` nach `k` auflösen können:

- `solve(G1, k)`  
`{-1}`

Die Gleichung hat genau eine Lösung, d.h. die Gerade `g1` schneidet die Ebene  $F$ . Den Ortsvektor des Schnittpunktes erhalten wir, wenn wir den Wert  $k = -1$  in die Parameterform der Geraden einsetzen:

- `subs(g1, k = -1)`

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

Nun zu der Lagebeziehung unserer Ebene  $F$  zu der zweiten Geraden  $g2$ . Auch hier setzen wir die Komponenten der Parameterform von  $g1$  in die Koordinatenform `KForm` der Ebene  $F$  ein:

- `subs(KForm, x = g2[1],  
y = g2[2],  
z = g2[3])`

$$4 = 0$$

Offensichtlich führt das Verfahren in dieser Situation auf eine Gleichung, die keine Lösung besitzt. Damit verläuft  $g2$  parallel zur Ebene  $F$ .

Schließlich untersuchen wir noch die Lage von  $g3$  zu  $F$ :

- `subs(KForm, x = g3[1],  
y = g3[2],  
z = g3[3])`

$$0 = 0$$

Diese Gleichung ist also immer erfüllt, d.h. jeder Punkt der Geraden  $g3$  liegt auch in der Ebene  $F$ .

## Übungen

29.) Untersuchen Sie die Lagebeziehung der Geraden

$$g1 : \vec{x} = \begin{pmatrix} 1 \\ 3 \\ 7 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ 0 \\ -4 \end{pmatrix}, \quad g2 : \vec{x} = \begin{pmatrix} 3 \\ 5 \\ 8 \end{pmatrix} + s \cdot \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix},$$

$$g3 : \vec{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + t \cdot \begin{pmatrix} 2 \\ 0 \\ -4 \end{pmatrix},$$

zu der Ebene

$$F : \vec{x} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} + k \cdot \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} + l \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

30.) Zeichnen Sie die Ebene  $F$  sowie die drei Geraden  $g1$ ,  $g2$  und  $g3$  aus der vorhergehenden Aufgaben mit MuPAD in ein gemeinsames Koordinatensystem. Verwenden Sie für die Geraden unterschiedliche Farben, damit man sie besser auseinander halten kann.

31.) Untersuchen Sie die Lagebeziehung der Geraden

$$h1 : \vec{x} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \quad h2 : \vec{x} = \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix},$$

$$h3 : \vec{x} = \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix} + t \cdot \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix},$$

zu der Ebene

$$G : x + y + z + 1 = 0.$$

32.) Zeichnen Sie die Ebene  $G$  sowie die drei Geraden  $h1$ ,  $h2$  und  $h3$  aus der vorhergehenden Aufgaben mit MuPAD in ein gemeinsames Koordinatensystem. Verwenden Sie für die Geraden unterschiedliche Farben, damit man sie besser auseinander halten kann.

### 3.7 Lagebeziehung zweier Ebenen

Ähnlich wie bei einer Geraden und einer Ebene gibt es auch zwischen zwei Ebenen drei mögliche Lagebeziehungen: Zwei Ebenen können parallel zueinander verlaufen, sie können identisch sein oder sich schneiden. Wenn sich zwei Ebenen schneiden, dann ist das Schattenbild eine Gerade. Die drei möglichen Lagebeziehungen zweier Ebenen sind in Abbildung 3.7 dargestellt.

Unser Vorgehen zur Untersuchung der gegenseitigen Lage zweier Ebenen zueinander muss sich wieder danach richten, in welcher Form die Ebenen gegeben sind: Liegen beide Ebenen in Parameterform vor oder liegt eine Ebene in Parameterform vor und die andere in Koordinaten oder liegen etwa beide Ebenen in Koordinatenform vor? Je nach dem, welche Situation gegeben ist, ist die Vorgehensweise zur Untersuchung der gegenseitigen Lage ein wenig anders.

#### 3.7.1 Beide Ebenen in Parameterform

Wir beginnen damit, Lagebeziehungen zweier Ebenen zu untersuchen, die beide in Parameterform vorliegen. Sei  $F$  die Ebene mit der Parameterform

$$F : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

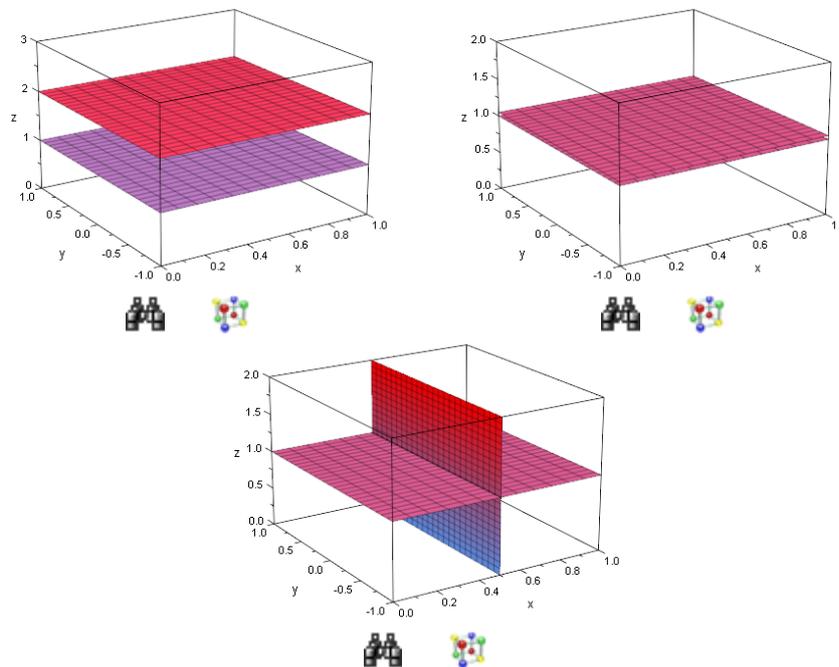


Abbildung 3.7: Lagebeziehungen zwischen zwei Ebenen

Zu dieser Ebene geben wir uns jetzt weiteren Ebenen  $G$  vor und untersuchen die Lagebeziehungen von  $F$  und  $G$ . Sei zuerst

$$G : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Um die gegenseitige von  $F$  und  $G$  zu bestimmen, können wir prinzipiell immer versuchen, gemeinsame Punkte von  $F$  und  $G$  zu berechnen (ganz analog zu der Vorgehensweise zur Untersuchung der Lagebeziehung einer Geraden zu einer Ebene). Wir definieren die Parameterformen in MuPAD:

```

• ParameterformF:=      matrix([1, 0, 1]) +
                        k * matrix([1,-1, 0]) +
                        l * matrix([1, 1, 0]):
  ParameterformG:=      matrix([1, 0, 2]) +
                        r * matrix([1, 0, 0]) +
                        s * matrix([0, 1, 0]):

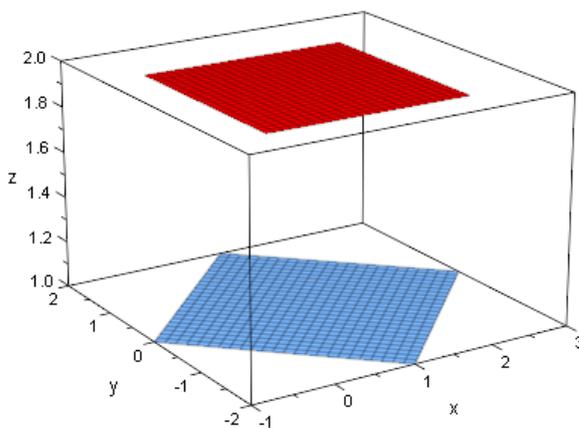
```

Zum Auffinden gemeinsamer Punkte müssen wir die beiden Parameterformen wie gewohnt gleichsetzen und das zugehörige Gleichungssystem lösen. In unserer Situation haben wir zwar – wie üblich – drei Gleichungen, aber vier Unbekannte  $k$ ,  $l$ ,  $r$  und  $s$ . Wir lösen das Gleichungssystem nach  $k$ ,  $l$ ,  $r$  und  $s$ :

- `solve({ParameterformF[1] = ParameterformG[1],  
ParameterformF[2] = ParameterformG[2],  
ParameterformF[3] = ParameterformG[3]},  
{k,l,r,s})`  
`}`

Das Gleichungssystem hat keine Lösung, d.h.  $F$  und  $G$  sind parallel. Zur Veranschaulichung zeichnen wir beide Ebenen in ein gemeinsames Koordinatensystem:

- `F:= plot::Surface(ParameterformF, k = -1..1, l = -1..1):`  
`G:= plot::Surface(ParameterformG, r = -1.5..1, s = -1.5..1):`  
`plot(F,G)`



Nun ändern wir die Definition der Ebene  $G$  ab und betrachten statt der oben angegebenen Parameterform die neue Parameterform

$$G : \vec{x} = \begin{pmatrix} -1 \\ 6 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Ganz analog zur obigen Vorgehensweise untersuchen wir die Lagebeziehung der Ebene  $F$  und der neuen Ebene  $G$ , indem wir zunächst die Parameterform von  $G$  in MuPAD definieren<sup>29</sup>

- `ParameterformG:= matrix([-1,-6, 1]) +`  
`r * matrix([ 1, 0, 0]) +`  
`s * matrix([ 0, 1, 0]):`

und anschließend das Gleichungssystem lösen, das sich ergibt, wenn wir die Parameterformen von  $F$  und  $G$  gleichsetzen:

<sup>29</sup>Die Parameterform von  $F$  definieren wir hier nicht erneut; weiter oben haben wir ihre Definition in MuPAD bereits angegeben.

- `solve({ParameterformF[1] = ParameterformG[1],  
ParameterformF[2] = ParameterformG[2],  
ParameterformF[3] = ParameterformG[3]},  
{k,l,r,s})`  

$$\left\{ \left[ k = \frac{r}{2} - \frac{s}{2} + 2, l = \frac{r}{2} + \frac{s}{2} - 4 \right] \right\}$$

Dieses Mal liefert uns MuPAD eine Lösungsmenge, die nicht die leere Menge ist. Wir ist die Ausgabe aber zu interpretieren? Nun, die Lösungen für die Parameter  $k$  und  $l$  werden *beide* in Abhängigkeit von  $r$  und  $s$  angegeben. Wenn wir also einen beliebigen Punkt  $P$  auf der Ebene  $G$  wählen, dann gibt es zu diesem Punkt immer Werte für  $r$  und  $s$  derart, dass sich der Ortsvektor von  $P$  ergibt, sofern wir  $r$  und  $s$  in die Parameterform der Ebene  $G$  einsetzen. Die obige Lösungsmenge sagt uns dann, wie wir bei gegebenen Werten für  $r$  und  $s$  die Werte für  $k$  und  $l$  wählen müssen, damit wir den Punkt  $P$  auch auf der Ebene  $F$  erreichen.

Wir machen uns das an einem konkreten Beispiel klar: Wir setzen für  $r$  den Wert  $7/2$  und für  $s$  den Wert  $-6/5$  in die Ebenengleichung von  $G$  ein und erhalten so den Ortsvektor eines Punktes  $P$  auf  $G$ :

- `OrtsvektorP := subs(ParameterformG, r = 7/2,  
s = -6/5)`

$$\begin{pmatrix} \frac{5}{2} \\ -\frac{36}{5} \\ 1 \end{pmatrix}$$

Die von uns berechnete Lösungsmenge sagt uns nun, dass wir für  $k$  den Wert  $r/2 - s/2 + 2 = 87/20$  und für  $l$  den Wert  $r/2 + s/2 - 4 = -57/20$  wählen sollen. Setzen wir diese Werte in die Parameterform von  $F$  ein, so erhalten wir in der Tat den Ortsvektor des Punktes  $P$ :

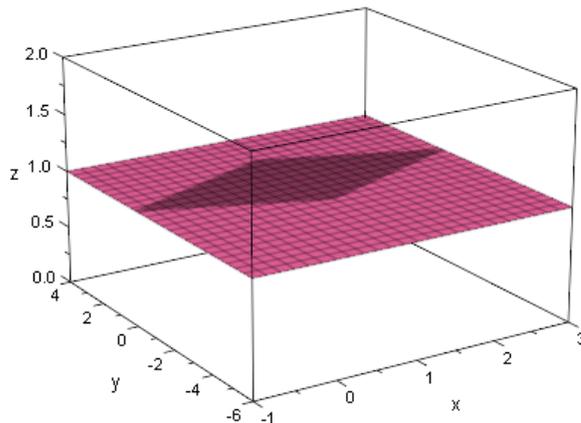
- `subs(ParameterformG, r = 87/20, s = -6/5)`

$$\begin{pmatrix} \frac{5}{2} \\ -\frac{36}{5} \\ 1 \end{pmatrix}$$

Da dies für jeden beliebigen Punkt  $P$  auf  $G$  gilt, müssen die Ebenen  $F$  und  $G$  identisch sein. Anschaulich können wir uns die Situation leicht wie folgt mit MuPAD visualisieren<sup>30</sup>:

- `G := plot::Surface(ParameterformG, r = 0..4, s = 0..10):  
plot(F,G)`

<sup>30</sup>Das grafische Objekt zu der Parameterform der Ebene  $F$  hatten wir oben schon definiert und der Variablen  $F$  zugewiesen.



Es bleibt noch der Fall zweier sich schneidender Ebenen zu untersuchen. Wir betrachten dazu die Ebene  $F$  wie oben und die Ebene

$$G: \vec{x} = \begin{pmatrix} 1/2 \\ 0 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Wir definieren ihre Parameterform in MuPAD und lösen im gleichen Schritt das Gleichungssystem, das sich ergibt, wenn wir die Parameterform der oben definierten Ebene  $F$  und die der Ebene  $G$  gleichsetzen:

- ```

ParameterformG:= matrix([1/2, 0, 1]) +
                  r * matrix([ 0, 1, 0]) +
                  s * matrix([ 0, 0, 1]):
solve({ParameterformF[1] = ParameterformG[1],
      ParameterformF[2] = ParameterformG[2],
      ParameterformF[3] = ParameterformG[3]},
      {k,l,r,s})

```

$$\left\{ \left[ k = -\frac{r}{2} - \frac{1}{4}, l = \frac{r}{2} - \frac{1}{4}, s = 0 \right] \right\}$$

Auch hier liefert uns MuPAD eine Lösungsmenge, die nicht die leere Menge ist. Im Gegensatz zu dem vorhergehenden Beispiel, wo die beiden Ebenen identisch waren, sind hier die Lösungen für  $k$ ,  $l$  und  $s$  (also für drei der vier Parameter) gegeben und diese hängen jeweils nur von *höchstens einem* Parameter ab (nämlich von  $r$ ). Im Fall der Lösungsmenge bei den identischen Ebenen im vorhergehenden Beispiel konnten wir für beliebige Werte von  $r$  und  $s$  stets Wert für  $k$  und  $l$  finden, so dass ein beliebiger Punkt auf  $G$  auch auf  $F$  liegt. Diese Freiheit haben wir hier nicht, denn der Parameter  $s$  muss den Wert 0 haben. Damit können wir nur noch den Wert von  $r$  frei wählen, was gleichbedeutend damit ist, dass wir uns entlang einer Geraden auf der Ebene  $G$  bewegen.

Setzen wir den Wert  $s = 0$  in die Parameterform der Ebene  $G$  ein, so erhalten wir

- ```

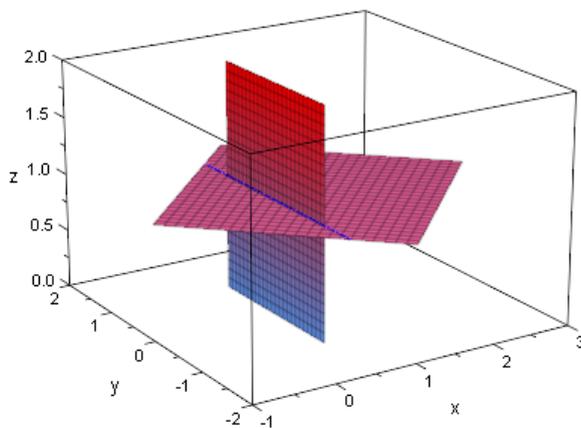
ParameterformS:= subs(ParameterformG, s = 0)

```

$$\begin{pmatrix} \frac{1}{2} \\ r \\ 1 \end{pmatrix}$$

also eine Parameterform einer Geraden. Alle Punkte auf dieser Geraden liegen sowohl auf der Ebene  $G$  als auch – nach der gleichen Argumentation wie oben für den Fall identischer Ebenen – auf der Ebene  $F$ . Die Ebenen  $F$  und  $G$  schneiden sich in dieser Geraden. Wir zeichnen die beiden Ebenen  $G$  und  $G$  sowie die Schnittgerade in ein gemeinsames Koordinatensystem:

- `S:= plot::Curve3d(ParameterformS, r = -1.5..1.5):`  
`G:= plot::Surface(ParameterformG, r = -1..1, s = -1..1):`  
`plot(F,G,S)`



### 3.7.2 Eine Ebene in Koordinatenform, eine Ebene in Parameterform

Nun kommen wir zu dem Fall, dass eine der Ebenen in Koordinatenform gegeben ist und die andere Ebene in Parameterform vorliegt. Sei die Ebene  $F$  in Koordinatenform gegeben durch

$$F : x - 2 \cdot y + z + 2 = 0$$

und die Ebene  $G$  in Parameterform gegeben durch

$$G : \vec{x} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}.$$

Im folgenden zeigen wir nur noch, wie man die Lagebeziehung der Ebenen rechnerisch bestimmen kann und verzichten auf die Visualisierungen, die wir im vorhergehenden Abschnitt bereits vorgenommen hatten.

Wir definieren die Koordinatenform der Ebene  $F$  und die Parameterform der Ebene  $G$  wie gewohnt in MuPAD:

- `KForm:= x - 2*y + z + 2 = 0:`  
`ParameterformG:= matrix([1,3,1]) +`  
`k * matrix([2,1,0]) +`  
`l * matrix([0,1,2]):`

Wir erinnern uns an Abschnitt 3.6, wo wir die Lagebeziehungen von Geraden und Ebenen untersucht hatten. Wenn die Ebene in Koordinatenform gegeben war, so habe wir die durch die Parameterform der Geraden gegebenen Koordinaten einfach in die Koordinatenform der Ebene eingesetzt, um etwaige gemeinsame Punkte von Gerade und Ebene auffinden zu können. Ganz analog gehen wir auch vor, wenn wir zwei Ebenen gegeben haben, von denen eine in Koordinatenform und die andere in Parameterform vorliegt:

- `G1:= subs(KForm, x = ParameterformG[1],`  
`y = ParameterformG[2],`  
`z = ParameterformG[3])`  

$$-2 = 0$$

Diese Gleichung kann offensichtlich nicht erfüllt werden, d.h.  $F$  und  $G$  haben keinen einzigen gemeinsamen Punkt und müssen folglich parallel sein.

Wir übernehmen wieder die Definition der Ebene  $F$  in Koordinaten und betrachten statt der oben definierten Ebene  $G$  in Parameterform die neue Ebene

$$G: \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}.$$

Definition der Parameterform in MuPAD und anschließendes Einsetzen in die Koordinatenform der Ebene  $F$  ergibt:

- `ParameterformG:= matrix([1,2,1]) +`  
`k * matrix([2,1,0]) +`  
`l * matrix([0,1,2]):`  
`G1:= subs(KForm, x = ParameterformG[1],`  
`y = ParameterformG[2],`  
`z = ParameterformG[3])`  

$$0 = 0$$

Diese Gleichung ist immer erfüllt, d.h.  $F$  und  $G$  haben unendlich viele gemeinsame Punkt. Die beiden Ebenen sind also identisch.

Schließlich schauen wir uns noch die Situation an, wenn sich die beiden Ebenen schneiden. Wir übernehmen wieder die Definition von  $F$  in Koordinatenform und be-

trachten statt der obigen Ebene  $G$  die neue Ebene

$$G: \vec{x} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}.$$

Wir definieren die Parameterform in MuPAD, setzen diese wieder in die Koordinatenform ein und erhalten damit die folgende Gleichung:

```

• ParameterformG:=      matrix([ 1, 3, 1]) +
                        k * matrix([-1,-1, 1]) +
                        l * matrix([ 0, 1, 2]):
  Gl:= subs(KForm, x = ParameterformG[1],
            y = ParameterformG[2],
            z = ParameterformG[3])

```

$$2k - 2 = 0$$

Diese Gleichung besitzt eine nun genau eine Lösung, nämlich

```

• solve(Gl, k)
  {1}

```

Den Wert  $k = 1$  müssen wir jetzt nur noch in die Parameterform der Ebene  $G$  einsetzen und wir erhalten die Gleichung der Schnittgeraden von  $F$  und  $G$ :

```

• subs(ParameterformG, k = 1)

```

$$\begin{pmatrix} 0 \\ l + 2 \\ 2l + 2 \end{pmatrix}$$

### 3.7.3 Beide Ebenen in Koordinatenform

Als letztes gilt es noch die Lagebeziehungen für zwei in Koordinatenform gegebenen Ebenen zu bestimmen. Seien  $F$  und  $G$  Ebenen mit den Koordinatenformen

$$F: x - 2 \cdot y + z + 2 = 0 \quad \text{und} \quad G: 2 \cdot x - 4 \cdot y + 2 \cdot z + 8 = 0.$$

Wenn wir die Lage von  $F$  und  $G$  bestimmen wollen, dann können wir auch hier wieder einfach den Ansatz machen, gemeinsame Punkte zu berechnen. Gemeinsame Punkte von  $F$  und  $G$  erfüllen sowohl die Gleichung, die durch die Koordinatenform von  $F$  gegeben ist, als auch die Gleichung, die durch die Koordinatenform von  $G$  gegeben ist. Wir können daher die beiden Koordinatenformen einfach als lineares Gleichungssystem mit zwei Gleichungen in den drei Unbekannten  $x$ ,  $y$  und  $z$  auffassen. Dieses lässt sich in MuPAD wie gewohnt mit Hilfe der Funktion `solve` lösen:

```

• KFormF:= x - 2*y + z + 2 = 0:
  KFormG:= 2*x - 4*y + 2*z + 8 = 0:
  solve({KFormF, KFormG}, {x,y,z})
  {}

```

Das Gleichungssystem besitzt keine Lösung, d.h. die beiden Ebenen  $F$  und  $G$  haben keine gemeinsamen Punkte und sind damit echt parallel.

Betrachten wir statt der oben definierten Ebene  $G$  die neue Ebene

$$G : 2 \cdot x - 4 \cdot y + 2 \cdot z + 4 = 0,$$

so stellen wir fest:

```

• KFormG:= 2*x - 4*y + 2*z + 4 = 0:
  solve({KFormF, KFormG}, {x,y,z})
  {[x = 2*y - z - 2]}

```

Die von MuPAD ausgegebene Gleichung ist aber gerade die Koordinatenform der Ebene  $F$ , d.h. alle gemeinsamen Punkte von  $F$  und  $G$  liegen auf der Ebene  $F$  – die beiden Ebenen sind also identisch.

Schließlich betrachten wir als letztes die Lage der Ebene

$$G : -3 \cdot x + 2 \cdot y - z + 2 = 0,$$

zu unserer Ebene  $F$ :

```

• KFormG:= -3*x + 2*y - z + 2 = 0:
  solve({KFormF, KFormG}, {x,y,z})
  {[x = 2, y = z/2 + 2]}

```

Wie ist nun diese Lösungsmenge zu interpretieren? Wir können den Parameter  $z$  beliebig wählen und die obige Lösungsmenge gibt uns die Werte für  $x$  (unabhängig von  $z$  immer konstant 2) und  $y$ . Wählen wir für  $z$  also einen beliebigen Wert  $k$ , so erhalten wir:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ \frac{k}{2} + 2 \\ k \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 0 \\ \frac{1}{2} \\ 1 \end{pmatrix}$$

Letzteres ist aber genau die Parameterform einer Geraden. Diese Gerade ist die Schnittgerade von  $F$  und  $G$ .

**Übungen**

33.) Bestimmen Sie die Lagebeziehungen der folgenden Ebenen zueinander.

$$F1 : \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} -3 \\ -1 \\ 1 \end{pmatrix} + l \cdot \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix},$$

$$F2 : \vec{x} = \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix} + r \cdot \begin{pmatrix} 6 \\ 2 \\ -2 \end{pmatrix} + s \cdot \begin{pmatrix} -2 \\ -6 \\ -4 \end{pmatrix},$$

$$F3 : -5 \cdot x + 7 \cdot y - 8 \cdot z - 1 = 0,$$

$$F4 : 8 \cdot x - 5 \cdot z = 0.$$

Zeichnen Sie je zwei der Ebenen anschließend in ein gemeinsames Koordinatensystem.

**3.8 Koordinatenachsen und Koordinatenebenen**

Auch die Achsen eines Koordinatensystems lassen sich als Geraden beschreiben. Wir beschränken uns hier auf den dreidimensionalen Fall – im Zweidimensionalen geht alles ganz analog. Die  $x$ -Achse zeichnet sich dadurch aus, dass die  $y$ - und  $z$ -Koordinaten aller Punkte, die darauf liegen, Null sind, d.h. sie sind von der Form  $(k|0|0)$  für ein  $k \in \mathbb{R}$ . Aus der Beobachtung kann man dann sofort eine Parameterdarstellung der  $x$ -Achse ableiten:

$$gx : \vec{x} = \begin{pmatrix} k \\ 0 \\ 0 \end{pmatrix} = k \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Damit haben wir implizit direkt den Nullvektor als Stützvektor der Geraden angenommen, denn wir können die obige Parameterdarstellung auch in der Form

$$gx : \vec{x} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

schreiben. Ganz analog erhält man die Parameterdarstellungen der  $y$ - und der  $z$ -Koordinatenachse in der Form

$$gy : \vec{x} = l \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{und} \quad gz : \vec{x} = m \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Diese Darstellungen der Koordinatenachsen zu kennen ist immer dann wichtig, wenn es darum geht, die Schnittpunkte einer gegebenen Geraden oder einer gegebenen

Ebene mit den Koordinatenachsen zu bestimmen<sup>31</sup>.

Neben Koordinatenachsen behandelt man im Mathematikunterricht auch häufig die Koordinatenebenen im Dreidimensionalen. Unter Koordinatenebenen versteht man diejenigen Ebenen, die von je zwei der Koordinatenachsen aufgespannt werden. Abbildung 3.8 zeigt die drei Koordinatenebenen im Raum.

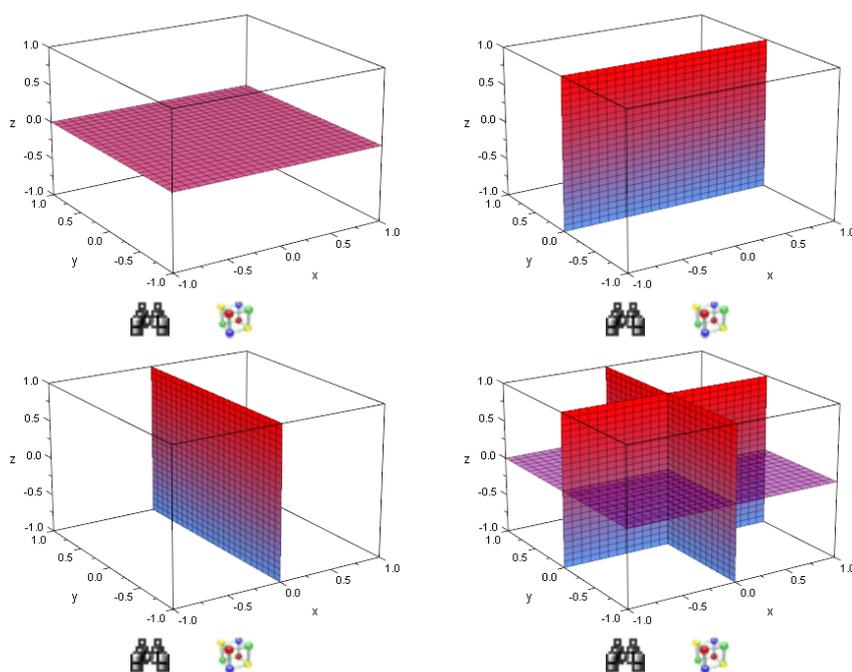


Abbildung 3.8: Die drei Koordinatenebenen

Die Koordinatenebene, die durch die  $x$ - und die  $y$ -Koordinatenachse aufgespannt wird, bezeichnen wir als  $x$ - $y$ -Ebene. Eine Parameterform dieser Ebene ist gegeben durch

$$Kxy: \vec{x} = k \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Eine Koordinatenform der Ebene zu finden ist nicht schwer: Alle Punkte auf der  $x$ - $y$ -Ebene haben die Eigenschaft, dass ihre  $z$ -Koordinate Null ist, d.h.  $z = 0$  ist eine Koordinatenform der  $x$ - $y$ -Ebene. Ebenso findet man die Parameterform der  $x$ - $z$ -Ebene und der  $y$ - $z$ -Ebene, die durch die  $x$ - und  $y$ -Koordinatenachse bzw. durch die  $y$ - und die  $z$ -

<sup>31</sup>In der Literatur werden Schnittpunkte mit den Koordinatenachsen im Kontext der Analytischen Geometrie auch häufig als *Spurpunkte* bezeichnet. Schnittgeraden von Ebenen mit den Koordinatenebenen bezeichnet man auch als *Spurgeraden*.

Koordinatenachse aufgespannt werden:

$$Kxz : \vec{x} = k \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{und} \quad Kyz : \vec{x} = k \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + l \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Eine Koordinatenform der  $x$ - $z$ -Ebene ist folglich  $y = 0$  und eine Koordinatenform der  $y$ - $z$ -Ebene ist  $x = 0$ .

### 3.9 Schnittwinkelberechnung

In den vorhergehenden Abschnitten haben wir uns überlegt, wie wir Schnitte von Geraden und Ebenen berechnen können. In diesem Abschnitt wollen wir uns überlegen, wie wir den Schnittwinkel von zwei sich schneidenden Geraden, zwei sich schneidenden Ebenen oder einer Gerade und einer Ebene, die einander schneiden, berechnen können.

#### 3.9.1 Schnittwinkel zweier Geraden

Um den Schnittwinkel zweier sich schneidender Geraden zu berechnen, ist die zentrale Beobachtung die, dass ihr Schnittwinkel durch den Winkel zwischen ihren beiden Richtungsvektoren gegeben ist (siehe Abbildung 3.9).

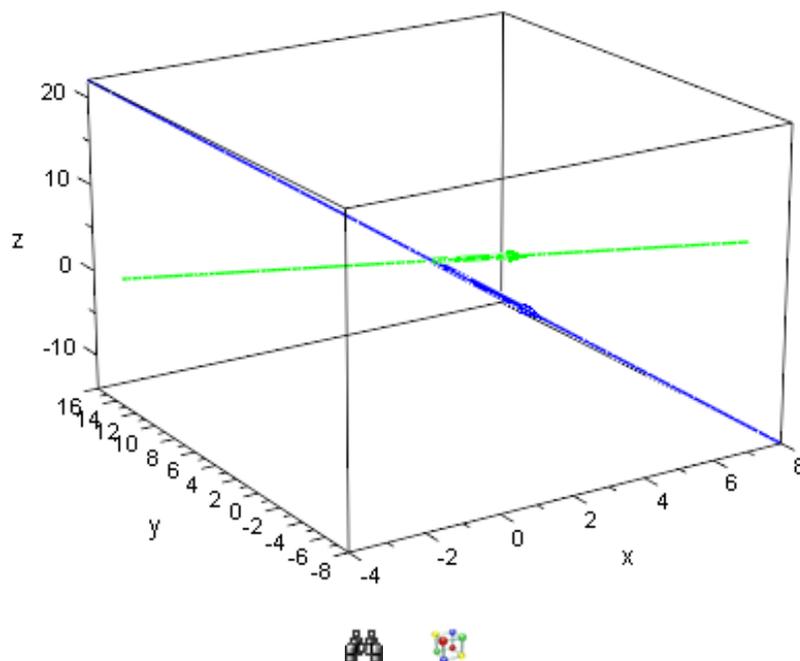


Abbildung 3.9: Zwei sich schneidende Geraden mit Richtungsvektoren

Wie man den Winkel zwischen zwei Vektoren berechnen kann, haben wir ihn Abschnitt 2.5 gesehen. Bezeichnen wir die Richtungsvektoren zweier sich schneidender Geraden mit  $\vec{u}$  und  $\vec{v}$ , so ist der Winkel zwischen den beiden Vektoren und damit der Schnittwinkel der beiden Geraden implizit gegeben durch

$$\cos(\varphi) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|}.$$

Schauen wir uns ein Beispiel an. Die beiden Geraden

$$g: \vec{x} = \begin{pmatrix} 2 \\ 4 \\ 4 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ -4 \\ -6 \end{pmatrix}, \quad h: \vec{x} = \begin{pmatrix} -2 \\ 10 \\ 2 \end{pmatrix} + l \cdot \begin{pmatrix} -2 \\ 3 \\ -1 \end{pmatrix}$$

schneiden sich<sup>32</sup>. Der Schnittwinkel zwischen den beiden Geraden ergibt sich als Winkel zwischen ihren Richtungsvektoren mit Hilfe der MuPAD-Funktion `linalg::angle`<sup>33</sup>:

- `RichtungsvektorG:= matrix([ 2,-4,-6]):`  
`RichtungsvektorH:= matrix([-2, 3,-1]):`  
`phi:= linalg::angle(RichtungsvektorG,`  
`RichtungsvektorH)`

$$\pi - \arccos\left(\frac{5\sqrt{56}\sqrt{14}}{392}\right)$$

Auf 10 signifikante Stellen genau lautet das Ergebnis

- `float(phi)`  
1.936003548

sowie umgerechnet in das Gradmass:

- `float(phi * 180 / PI)`  
110.9248324

## Übungen

- 34.) Weisen Sie nach, dass sich die beiden Geraden  $g$  und  $h$ , die im Rahmen dieses Abschnitts auf Seite 101 definiert wurden, tatsächlich schneiden. Zeichnen Sie anschließend beide Geraden in ein gemeinsames Koordinatensystem.

<sup>32</sup>Diese Behauptung ist in den Übungen am Ende dieses Abschnitts nachzuweisen.

<sup>33</sup>Wir erinnern an dieser Stelle daran, dass MuPAD den gewünschten Winkel im Bogenmaß und in exakter Form angibt. Mit Hilfe von `float` können wir uns eine Näherung auf bis auf 10 signifikante Stellen berechnen lassen. Die Umrechnung in das Gradmass nehmen wir hier von Hand vor – man kann sich aber auch leicht die Umrechnungsformel in MuPAD definieren und diese an dafür benutzen.

35.) Zeigen Sie, dass sich die folgenden beiden Geraden schneiden:

$$g : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}, \quad h : \vec{x} = \begin{pmatrix} 3 \\ -2 \\ 6 \end{pmatrix} + l \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

Berechnen Sie den Schnittwinkel zwischen den folgenden beiden Geraden. Zeichnen Sie anschließend  $g$  und  $h$  in ein gemeinsames Koordinatensystem.

### 3.9.2 Schnittwinkel von Gerade und Ebene

Um den Schnittwinkel zwischen einer Geraden und einer Ebene zu berechnen, projizieren wir die Gerade in die Ebene. Dann ist der Schnittwinkel zwischen der Geraden und der Ebene genau der Schnittwinkel zwischen der Geraden und orthogonalen Projektion in die Ebene. Abbildung 3.10 zeigt die Situation. Dort ist auch ein Normalenvektor der Ebene in schwarzer eingezeichnet.

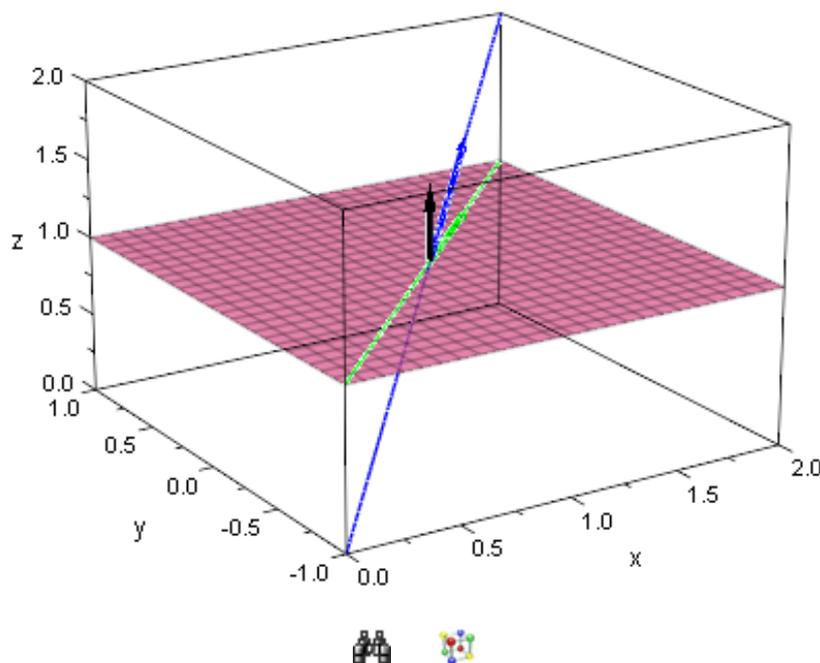


Abbildung 3.10: Schnittwinkel von Gerade und Ebene

Prinzipiell könnte man zur Berechnung des Schnittwinkels von Gerade und Ebene wirklich erst die Projektion der Geraden in die Ebenen berechnen und anschließend den Schnittwinkel zwischen Gerade und Ebene als Schnittwinkel der Geraden mit ihren Projektion in die Ebene berechnen. Dieses Verfahren ist aber unnötig kompliziert. Einfacher wird es, wenn man sich daran erinnert, dass der Normalenvektor immer senkrecht auf der Ebene steht, d.h. dass er mit jedem Vektor in der Ebene einen rechten



31.48215411

**Übungen**

36.) Weisen Sie nach, dass die auf Seite 103 definierte Gerade  $g$  die Ebene  $F$  schneidet. Zeichnen Sie anschließend die Gerade und die Ebene in ein gemeinsames Koordinatensystem.

37.) Zeigen Sie, dass sich

$$F : \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} + l \cdot \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

und

$$g : \vec{x} = \begin{pmatrix} 3 \\ -2 \\ 6 \end{pmatrix} + m \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

schneiden. Berechnen Sie den Schnittwinkel, indem Sie zunächst mit Hilfe von `linalg::crossProduct` einen Normalenvektor der Ebene  $F$  bestimmen und dann wie in diesem Abschnitt beschrieben vorgehen. Zeichnen Sie anschließend die Ebene  $F$  und die Gerade  $g$  in ein gemeinsames Koordinatensystem.

**3.9.3 Schnittwinkel zweier Ebenen**

Den Schnittwinkel zweier Ebenen erhalten wir, indem wir den Winkel zwischen ihren Normalenvektoren berechnen. Abbildung 3.11 zeigt die Situation.

Als Beispiel betrachten wir die beiden Ebenen

$$F : z - 1 = 0 \quad \text{und} \quad G : x - z = 0.$$

$F$  und  $G$  schneiden sich<sup>36</sup>. Die Normalenvektoren lesen wir an den Koordinatenformen der Ebenen ab und berechnen den Winkel `phi` zwischen ihnen mit Hilfe von `linalg::angle`:

```
• NormalenvektorF:= matrix([0,0, 1]):
  NormalenvektorG:= matrix([1,0,-1]):
  phi:= linalg::angle(NormalenvektorF,
                     NormalenvektorG)
```

$$\frac{3\pi}{4}$$

Auf 10 signifikante Stellen bzw. umgerechnet in das Gradmass erhalten wir damit:

<sup>36</sup>Diese Behauptung ist in den Übungen am Ende dieses Abschnitts nachzuweisen.

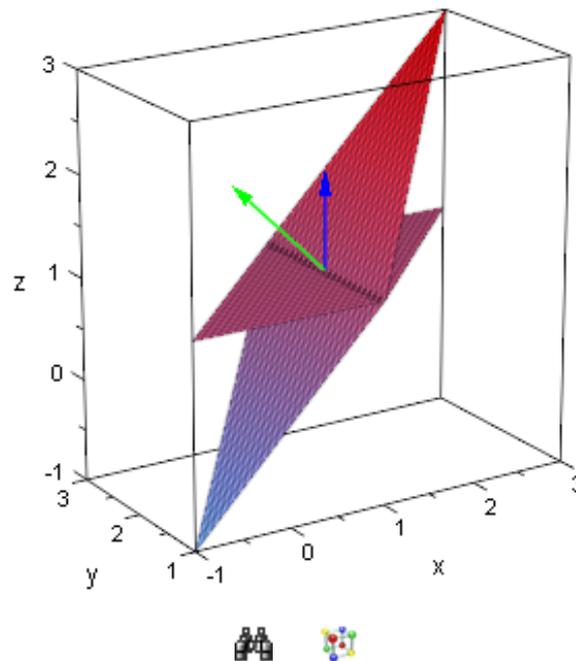


Abbildung 3.11: Schnittwinkel von zwei Ebenen

- `float(phi);`  
`float(phi * 180 / PI)`  
 2.35619449  
 135.0

### Übungen

- 38.) Weisen Sie nach, dass die auf Seite 104 definierten Ebenen  $F$  und  $G$  aneinander schneiden. Zeichnen Sie anschließend die beiden Ebenen in ein gemeinsames Koordinatensystem.
- 39.) Zeigen Sie, dass sich

$$F: \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 4 \\ 2 \\ 4 \end{pmatrix} + l \cdot \begin{pmatrix} -2 \\ -2 \\ -1 \end{pmatrix},$$

$$G: \vec{x} = \begin{pmatrix} 3 \\ -2 \\ 6 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} + s \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

schneiden. Berechnen Sie den Schnittwinkel, indem Sie zunächst mit Hilfe von `linalg::crossProduct` die Normalenvektoren der Ebenen bestimmen und

dann wie in diesem Abschnitt beschrieben vorgehen. Zeichnen Sie anschließend die beiden Ebenen in ein gemeinsames Koordinatensystem.

### 3.10 Abstandsprobleme

Das letzte größere Thema, dem wir uns in diesem Kapitel widmen wollen, besteht in der Berechnung von Abständen. Auch hier beschränken wir uns auf Betrachtungen im Dreidimensionalen. Den Abstand eines Punktes von einer Geraden kann man im Fall des  $\mathbb{R}^2$  im Prinzip ganz analog zu der Vorgehensweise berechnen, die wir unten beschreiben<sup>37</sup>.

Des Weiteren verzichten wir hier im Fall von Ebenen auf die Verwendung der so genannten *Hesseschen Normalform* und führen die Abstandsberechnung ausschließlich auf ganz elementarer (konstruktiver) Ebene durch.

#### 3.10.1 Abstand eines Punktes von einer Geraden im $\mathbb{R}^3$

Wir gehen im folgenden davon aus, dass wir eine Gerade  $g : \vec{x} = \vec{a} + k \cdot \vec{v}$  und einen Punkt  $P$  gegeben haben, der nicht auf der Geraden liegt<sup>38</sup>. Der Abstand des Punktes  $P$  zu der Geraden  $g$  soll bestimmt werden. Zur Berechnung des Abstands müssen wir im Prinzip das Lot  $l$  vom Punkt  $P$  auf die Gerade  $g$  fällen. Stellen wir uns das Lot als Gerade vor, so besteht ein möglicher Ansatz darin, eine Lotgerade zu finden, deren Stützvektor durch den Ortsvektor des Punktes  $P$  gegeben ist und deren Richtungsvektor orthogonal zu dem Richtungsvektor der gegebenen Geraden  $g$  ist. Anschließend bestimmen wir den Schnittpunkt der Lotgeraden mit der Geraden  $g$ . Ist der Schnittpunkt  $L$ , diesen Punkt nennt man auch *Lotfußpunkt*, bestimmt, so ist der Abstand des Punktes  $P$  von der Geraden  $g$  gegeben durch die Länge des Vektors  $\overrightarrow{PL}$ .

Das klingt sehr viel versprechend, doch gibt es einen nicht so ganz einfachen Aspekt in unserer Überlegung: Das Auffinden eines "richtigen" Vektors, der orthogonal zu dem Richtungsvektor der Geraden  $g$  ist und als Richtungsvektor für die Lotgerade  $l$  fungieren kann. Abbildung 3.12 zeigt einige zu einer gegebenen Geraden orthogonale Vektoren.

Wie finden wir den "richtigen" Vektor heraus? Wir stellen uns einfach vor, wir hätten den Lotfußpunkt  $L$  auf der Geraden  $g$  bereits gefunden. Dann muss der Vektor  $\overrightarrow{LP}$  orthogonal zu dem Richtungsvektor  $\vec{u}$  der Geraden  $g$  sein. Abbildung 3.13 zeigt die Situation. Mathematisch gilt also, dass das Skalarprodukt der beiden Vektoren  $\overrightarrow{LP}$  und  $\vec{u}$  gleich Null sein muss. Wir erhalten also die Gleichung

$$\overrightarrow{LP} * \vec{u} = 0,$$

<sup>37</sup>Lediglich die alternative Vorgehensweise im Folgeabschnitt über die Definition einer Hilfsebene kann so im  $\mathbb{R}^2$  nicht durchgeführt werden. Im  $\mathbb{R}^2$  wird aus der Hilfsebene eine Hilfsgerade (eine so genannte *Normale*).

<sup>38</sup>Wenn der Punkt  $P$  auf der Geraden liegt, dann ist der Abstand von  $P$  zu  $g$  zwar trivialerweise Null.

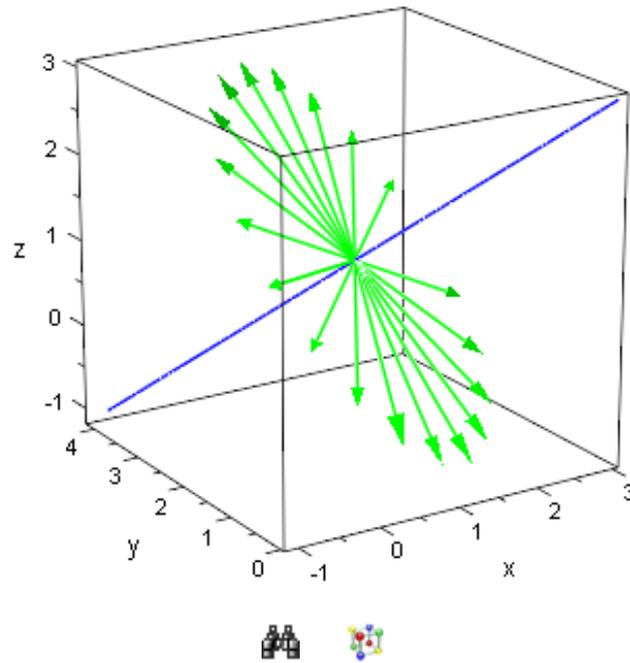
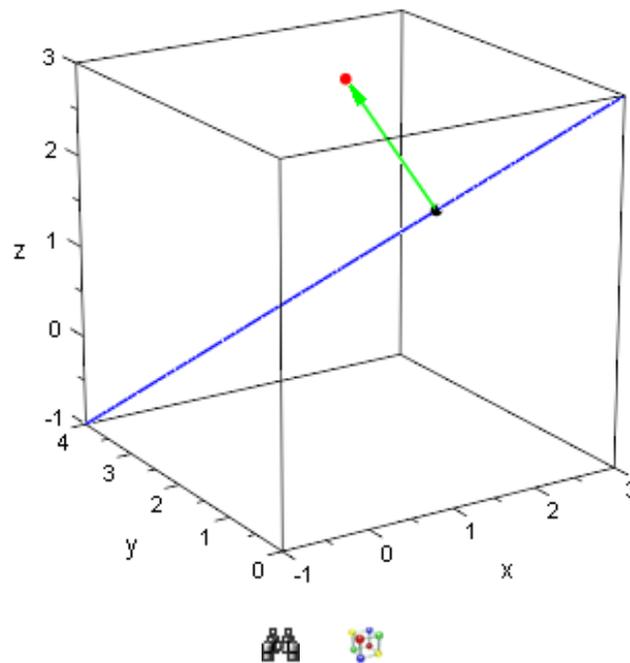


Abbildung 3.12: Einige zu einer Geraden orthogonale Vektoren

Abbildung 3.13: Vektor vom Lotfußpunkt hin zum Punkt  $P$

die es zu lösen gilt. Wäre der Punkt  $L$  völlig unbekannt, so wäre die Bestimmung von  $L$  nicht möglich ( $L$  hat drei Koordinaten, d.h. wir hätten eine Gleichung mit drei Variablen zu lösen – die Lösung wäre also in der Regel nicht eindeutig). Da der Punkt  $L$  aber auf der Geraden liegt, gibt es ein  $k \in \mathbb{R}$ , so dass für seinen Ortsvektor  $\vec{l}$  gilt:

$$\vec{l} = \vec{a} + k \cdot \vec{u}.$$

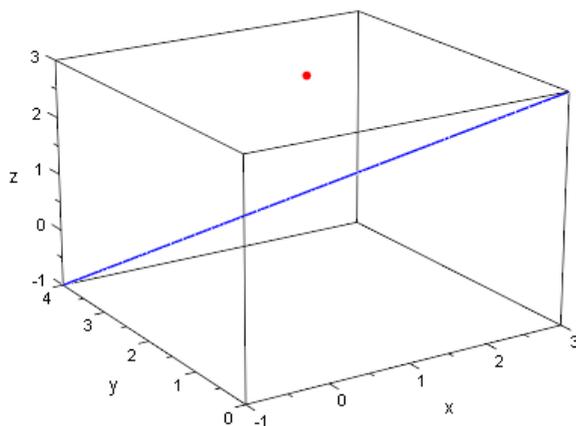
Da wir  $\vec{a}$  und  $\vec{u}$  als bekannt voraussetzen können, taucht also nur noch *eine Unbekannte* (nämlich  $k$ ) in den Komponenten von  $\vec{l}$  auf. Setzen wir diese Darstellung für  $\vec{l}$  in die Gleichung  $\overrightarrow{LP} \cdot \vec{u} = 0$  ein (wobei wir wieder  $\vec{u}$  und auch  $P$  als gegeben voraussetzen können), so erhalten wir eine lineare Gleichung für  $k$ , die sich leicht lösen lässt. Anschließend setzen wir den für  $k$  erhaltenen Wert in die Parameterform der Geraden  $g$  ein und erhalten so den Ortsvektor  $\vec{l}$  des Lotfußpunktes. Dann müssen wir nur noch die Länge von  $\overrightarrow{LP}$  berechnen und haben damit den Abstand des Punktes  $P$  zur Geraden  $g$  berechnet.

Schauen wir uns ein Beispiel an: Gegeben sei der Punkt  $P(1|2|3)$  und die Gerade

$$g : \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}.$$

Zuerst zeichnen wir den Punkt mit Hilfe von `plot::Point3d` und die Gerade mit Hilfe von `plot::Curve3d` in ein gemeinsames Koordinatensystem:

- `ParameterformG:= matrix([1,2,1]) + k*matrix([1,-1,1]):`  
`OrtsvektorP:= matrix([1,2,3]):`  
`P:= plot::Point3d(OrtsvektorP, Color = RGB::Red):`  
`g:= plot::Curve3d(ParameterformG, k = -2..2):`  
`plot(P,g)`



Da der Lotfußpunkt  $L$  auf der Geraden liegt, ist sein Ortsvektor von der Gestalt

- `OrtsvektorL:= ParameterformG`

$$\begin{pmatrix} k+1 \\ 2-k \\ k+1 \end{pmatrix}$$

für einen zu bestimmenden Wert für  $k$ . Wir hatten uns oben überlegt, dass die Orthogonalitätsbedingung uns die folgende lineare Gleichung in  $k$  liefert<sup>39</sup>:

- `G1:= linalg::scalarProduct(OrtsvektorP - OrtsvektorL, matrix([1,-1,1])) = 0`  
 $2 - 3k = 0$

Das Auflösen der Gleichung nach  $k$  erledigt wie immer `solve` für uns:

- `solve(G1, k)`

$$\left\{ \frac{2}{3} \right\}$$

Damit ist der Ortsvektor des Lotfußpunktes von  $P$  aus auf die Gerade  $g$  gegeben durch<sup>40</sup>:

- `OrtsvektorL:= subs(ParameterformG, k = 2/3)`

$$\begin{pmatrix} \frac{5}{3} \\ \frac{4}{3} \\ \frac{5}{3} \end{pmatrix}$$

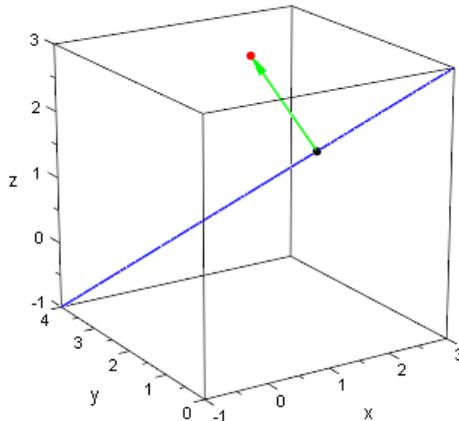
Wir zeichnen den Lotfußpunkt  $L$  (in schwarzer Farbe `RGB::Black`), den Vektor vom Lotfußpunkt hin zum Punkt  $P$  (mit Hilfe von `plot::Arrow3d` in grüner Farbe `RGB::Green`) sowie den Punkt  $P$  selbst und die Gerade  $g$  zur Veranschaulichung in ein gemeinsames Koordinatensystem<sup>41</sup>:

- `L:= plot::Point3d(OrtsvektorL, Color = RGB::Black):`  
`VektorLP:= plot::Arrow3d(OrtsvektorL, OrtsvektorP,`  
`Color = RGB::Green):`  
`plot(L, VektorLP, g, P, Scaling = Constrained)`

<sup>39</sup>Das Skalarprodukt berechnen wir wie gewohnt mit Hilfe von `linalg::scalarProduct`. Wir nehmen dabei wieder  $k$  mit Hilfe von `assume` explizit als reelle Zahl an.

<sup>40</sup>Das Einsetzen eines Wertes für  $k$  in die Parameterform der Geraden führen wir wir üblich mit Hilfe des Befehls `subs` durch.

<sup>41</sup>Mit Hilfe der Option `Scaling = Constrained` erreicht man, dass die Achsen gleich skaliert werden. Dies ist hier wichtig, damit der rechte Winkel zwischen dem Vektor  $\overrightarrow{LP}$  und der Geraden  $g$  auch als rechter Winkel erscheint (siehe auch Fußnote 12, Seite 48).



Nachdem der Lotfußpunkt  $L$  bestimmt ist, müssen wir nur noch die Länge des Vektors  $\overrightarrow{LP}$  bestimmen<sup>42</sup> und erhalten so den Abstand des Punktes  $P$  zur Geraden  $g$ :

- `d:= norm(OrtsvektorP - OrtsvektorL, 2)`  

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

Auf 10 signifikante Stelle gerundet ergibt sich damit das Endergebnis

- `float(d)`  
1.632993162

Gerade in der Analytischen Geometrie führen häufig mehrere Wege zum Ziel. So kann der Abstand des Punkte  $P(1|2|3)$  von der Geraden  $g$  mit

$$g : \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}.$$

auch über eine Hilfsebene  $H$  berechnet werden. Man geht dazu wie folgt vor: Man bestimmt diejenige Ebene, die den Punkt  $P$  als Stützvektor und den Richtungsvektor der Geraden  $g$  als Normalenvektor besitzt (siehe auch Abbildung 3.14).

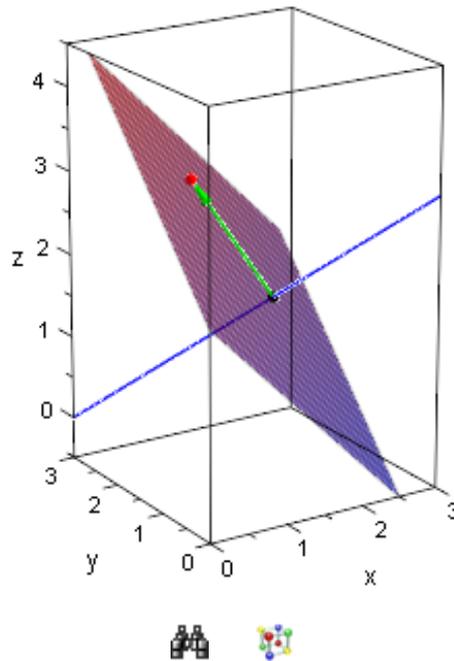
Der Schnittpunkt von Gerade und Ebene ist dann der Lotfußpunkt  $L$  und wir können den Abstand von  $P$  und  $g$  wie oben als Länge des Vektors  $\overrightarrow{LP}$  berechnen. In unserem Beispiel ist damit eine Gleichung dieser Hilfsebene  $H$  gegeben durch

- `assume({x,y,z}, Type::Real):`  
`KForm:= linalg::scalarProduct(matrix([1,-1,1]),`  
`matrix([x,y,z]) - matrix([1,2,3])) = 0`  

$$x - y + z - 2 = 0$$

Der Lotfußpunkt  $L$  ergibt sich als Schnittpunkt der Ebenen mit der Geraden  $g$ , indem wir die Komponenten der Parameterform von  $g$  in die Koordinatenform der Ebene einsetzen:

<sup>42</sup>Hier kommt wieder die Funktion `norm` zur Berechnung des Länge eines Vektors zum Einsatz.

Abbildung 3.14: Hilfsebene  $H$  orthogonal zu  $g$  durch  $P$ 

- `G1 := subs(KForm, x = ParameterformG[1],  
y = ParameterformG[2],  
z = ParameterformG[3])`

$$3k - 2 = 0$$

Als Lösung erhalten wir – wie oben bereits gesehen – den Wert  $k = 2/3$ , der auf den Lotfußpunkt  $L(5/3|4/3|5/3)$  führt, wenn wir ihn in die Parameterform der Geraden  $g$  einsetzen. Der Abstand von  $P$  zu  $g$  ergibt sich dann wieder als Länge des Vektors  $\overrightarrow{LP}$ .

### Übungen

40.) Gegeben sei der Punkt  $P(3|1|1)$  sowie die Gerade

$$g: \vec{x} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}.$$

Zeigen Sie, dass  $P$  nicht auf  $g$  liegt. Zeichnen Sie anschließend  $P$  und  $g$  in ein gemeinsames Koordinatensystem.

Berechnen Sie dann den Abstand von  $P$  zu  $g$  nach beiden in diesem Abschnitt vorgestellten Methoden (einmal über die Bedingung, dass das Skalarprodukt Null wird und einmal durch Bestimmung einer geeigneten Hilfsebene).

Zeichnen Sie zum Abschluss die Hilfsebene, den Punkt und die Gerade in ein gemeinsames Koordinatensystem.

### 3.10.2 Abstand zweier paralleler Geraden im $\mathbb{R}^3$

Nach dem, was wir uns im vorhergehenden Abschnitt überlegt haben, ist es nun nicht mehr schwer, den Abstand zweier paralleler Geraden zu berechnen. Wissen wir, dass die beiden Geraden

$$g : \vec{x} = \vec{a} + k \cdot \vec{u} \quad \text{und} \quad h : \vec{x} = \vec{b} + r \cdot \vec{v}$$

parallel sind, so hat jeder Punkt auf der Geraden  $h$  den gleichen Abstand von der Geraden  $g$ . Wir können uns also einfach irgendeinen Punkt auf  $h$  aussuchen und nach einer der beiden im vorhergehenden Abschnitt behandelten Methoden den Abstand dieses Punktes zur Geraden  $g$  berechnen. In jedem Fall liegt der Punkt  $B$  mit Ortsvektor  $\vec{b}$  auf  $h$ , d.h. der Punkt  $B$  ist die einfachste Wahl, die wir treffen können.

Betrachten wir ein Beispiel. Die beiden Geraden

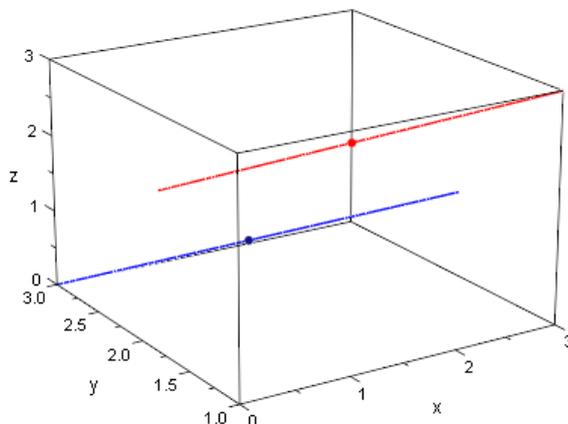
$$g : \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \quad \text{und} \quad h : \vec{x} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

sind parallel<sup>43</sup>. Wir definieren beide Geraden in MuPAD und zeichnen Sie zusammen mit ihren Antragspunkten in ein gemeinsames Koordinatensystem:

```

• ParameterformG:= matrix([1,2,1]) + k*matrix([1,-1,1]):
  ParameterformH:= matrix([2,2,2]) + r*matrix([1,-1,1]):
  OrtsvektorA:= matrix([1,2,1]):
  OrtsvektorB:= matrix([2,2,2]):
  A:= plot::Point3d(OrtsvektorA):
  B:= plot::Point3d(OrtsvektorB, Color = RGB::Red):
  g:= plot::Curve3d(ParameterformG, k = -1..1):
  h:= plot::Curve3d(ParameterformH, r = -1..1,
                    Color = RGB::Red):
plot(A, B, g, h)

```



<sup>43</sup>Diese Behauptung ist in den Übungsaufgaben am Ende dieses Abschnitts nachzuweisen.

Der Abstand von  $g$  und  $h$  ist also der Abstand des Punktes  $P(2|2|2)$  (Stützpunkt der Geraden  $h$ ) zu der Geraden  $g$ . Diesen Abstand berechnen wir wie im vorhergehenden Abschnitt. Für den Ortsvektor des Lotfußpunktes  $L$  muss gelten:

- `OrtsvektorL := ParameterformG`

$$\begin{pmatrix} k + 1 \\ 2 - k \\ k + 1 \end{pmatrix}$$

für einen zu bestimmenden Wert für  $k$ . Die Orthogonalitätsbedingung lautet:

- `G1 := linalg::scalarProduct(OrtsvektorB - OrtsvektorL, matrix([1,-1,1])) = 0`  
 $2 - 3k = 0$

Das Auflösen der Gleichung nach  $k$  erledigt wie immer `solve` für uns:

- `solve(G1, k)`

$$\left\{ \begin{array}{l} 2 \\ 3 \end{array} \right\}$$

Damit ist der Ortsvektor des Lotfußpunktes von  $P$  aus auf die Gerade  $g$  gegeben durch <sup>44</sup>:

- `OrtsvektorL := subs(ParameterformG, k = 2/3)`

$$\begin{pmatrix} \frac{5}{3} \\ \frac{4}{3} \\ \frac{5}{3} \end{pmatrix}$$

Dies ist im übrigen genau der Lotfußpunkt, den wir auch im vorhergehenden Abschnitt schon erhalten hatten.

## Übungen

41.) Zeigen Sie rechnerisch, dass die beiden Geraden  $g$  und  $h$ , die zu Beginn dieses Abschnitts definiert wurden, in der Tat parallel sind.

42.) Gegeben seien die beiden Geraden

$$g : \vec{x} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \quad \text{und} \quad h : \vec{x} = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} + r \cdot \begin{pmatrix} -2 \\ 1 \\ -3 \end{pmatrix}.$$

Zeigen Sie, dass  $g$  und  $h$  parallel sind und berechnen Sie ihren Abstand. Zeichnen Sie anschließend die beiden Geraden sowie den von Ihnen gewählten Lotvektor mit den Lotfußpunkten in ein gemeinsames Koordinatensystem.

<sup>44</sup>Das Einsetzen eines Wertes für  $k$  in die Parameterform der Geraden führen wir wir üblich mit Hilfe des Befehls `subs` durch.

### 3.10.3 Abstand windschiefer Geraden im $\mathbb{R}^3$

Nicht ganz so einfach wie bei einem gegebenen Punkt den Abstand von einer Geraden zu bestimmen oder auch den Abstand zweier paralleler Geraden zu berechnen ist es, wenn die beiden Geraden windschief sind. Seien

$$g : \vec{x} = \vec{a} + k \cdot \vec{u} \quad \text{und} \quad h : \vec{x} = \vec{b} + r \cdot \vec{v}$$

windschief. Um den Abstand von  $g$  und  $h$  zu berechnen, müssen wir im Prinzip natürlich auch wieder den Abstand eines Punktes auf  $h$  von der Geraden  $g$  (oder umgekehrt) bestimmen. Im Gegensatz zu parallelen Geraden können wir uns diesen Punkt auf der Geraden  $h$  aber nicht frei auswählen. Es gibt auf jeder der beiden windschiefen Geraden genau einen Punkt, der von dem jeweils anderen Punkt auf der zweiten Geraden minimalen Abstand hat. Wir bezeichnen den entsprechenden Punkt auf  $g$  mit  $L1$  und den entsprechenden Punkt auf  $h$  mit  $L2$ . Wir stellen uns jetzt einfach vor, wir hätten die Punkte  $L1$  bzw.  $L2$  bereits berechnet. Wie schon bei der Bestimmung des Abstands eines Punktes von einer Geraden, muss dann der Vektor  $\overrightarrow{L1L2}$  senkrecht auf dem Richtungsvektor der Geraden  $g$  stehen, d.h.

$$\overrightarrow{L1L2} * \vec{u} = 0.$$

Ebenso muss  $\overrightarrow{L1L2}$  aber auch senkrecht auf dem Richtungsvektor der Geraden  $h$  stehen, d.h.

$$\overrightarrow{L1L2} * \vec{v} = 0.$$

Weil  $L1$  ein Punkt der Geraden  $g$  ist und  $L2$  ein Punkt der Geraden  $h$  ist, muss es Werte für  $k$  und  $r$  geben derart, dass für die Ortsvektoren von  $L1$  bzw.  $L2$  gilt:

$$\vec{l1} = \vec{a} + k \cdot \vec{u} \quad \text{und} \quad \vec{l2} = \vec{b} + r \cdot \vec{v}.$$

Die gesamte Situation ist in Abbildung 3.15 veranschaulicht.

Lässt sich das Problem nun lösen? Die Antwort ist: Ja, denn die Komponenten von  $\overrightarrow{L1L2} = \vec{l2} - \vec{l1}$  hängen zwar jetzt von zwei Parametern ab (nämlich von  $k$  und von  $r$ ), aber wir haben auch zwei Bedingungen (nämlich  $\overrightarrow{L1L2} * \vec{u} = 0$  und  $\overrightarrow{L1L2} * \vec{v} = 0$ ). Diese beiden Bedingungen liefern uns ein Gleichungssystem mit zwei Gleichungen und zwei Unbekannten, dass, sofern die beiden Geraden in der Tat windschief sind, auch immer eine eindeutige Lösung besitzt. Ist das Gleichungssystem gelöst, so müssen wir die Werte für  $k$  und  $r$  nur noch in die Parameterformen der Geraden  $g$  bzw.  $h$  einsetzen und erhalten so die Ortsvektoren der Punkte  $L1$  und  $L2$  (Abbildung 3.15 zeigt diese Punkte). Der Abstand von  $g$  und  $h$  ist dann entsprechend durch die Länge des Vektors  $\overrightarrow{L1L2}$  gegeben (den Vektor  $\overrightarrow{L1L2}$  haben wir in grüner Farbe in Abbildung 3.15 mit eingezeichnet).

Schauen wir uns ein konkretes Beispiel an: Die beiden Geraden

$$g : \vec{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \quad \text{und} \quad h : \vec{x} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} + r \cdot \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}.$$

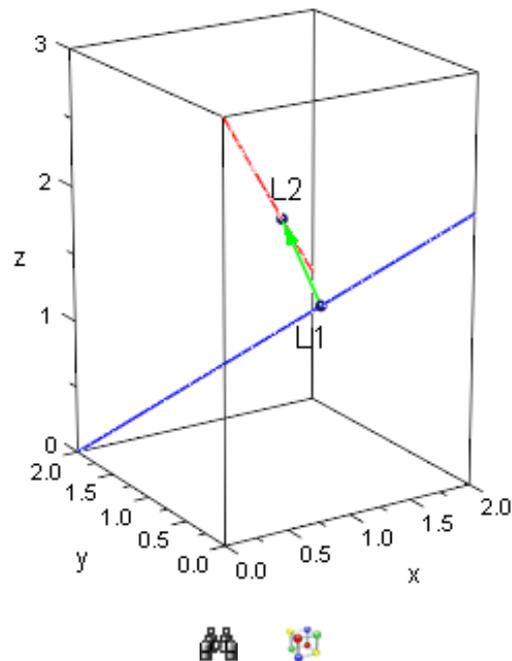


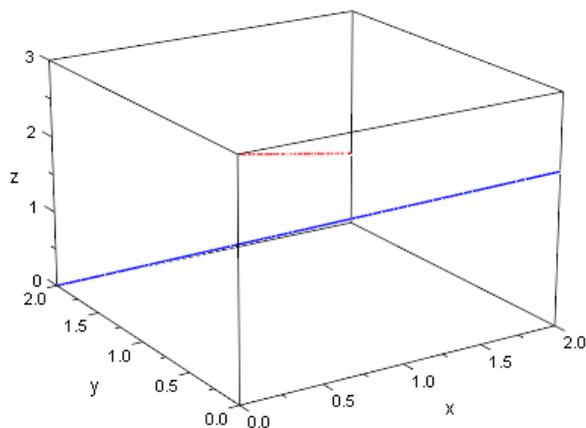
Abbildung 3.15: Abstand zweier windschiefer Geraden

sind windschief<sup>45</sup>. Wir definieren ihre Parameterformen in MuPAD und zeichnen beide Geraden zunächst in ein gemeinsames Koordinatensystem:

- ```

ParameterformG:= matrix([1,1,1]) + k * matrix([1,-1,1]):
ParameterformH:= matrix([1,1,2]) + r * matrix([-1,-1,1]):
g:= plot::Curve3d(ParameterformG, k = -1..1):
h:= plot::Curve3d(ParameterformH, r = -1..1,
                  Color = RGB::Red):
plot(g,h)

```



<sup>45</sup>Diese Behauptung ist in den Übungsaufgaben am Ende dieses Abschnitts nachzuweisen.

Da die Lotfußpunkte  $L1$  und  $L2$ , die wir nun finden müssen, nach unseren obigen Ausführungen Punkte der Geraden  $g$  bzw.  $h$  sind, gilt:

- `OrtsvektorL1:= ParameterformG;`  
`OrtsvektorL2:= ParameterformH;`

$$\begin{pmatrix} k+1 \\ 1-k \\ k+1 \end{pmatrix}$$

$$\begin{pmatrix} 1-r \\ 1-r \\ r+2 \end{pmatrix}$$

Die Orthogonalität des Vektors  $\overrightarrow{L1L2} = \vec{l2} - \vec{l1}$  zu den beiden Richtungsvektoren der Geraden führt auf die beiden linearen Gleichungen:

- `G11:= linalg::scalarProduct(`  
`OrtsvektorL2 - OrtsvektorL1,`  
`matrix([1,-1,1])) = 0;`  
`G12:= linalg::scalarProduct(`  
`OrtsvektorL2 - OrtsvektorL1,`  
`matrix([-1,-1,1])) = 0`

$$r - 3k + 1 = 0$$

$$3r - k + 1 = 0$$

Wir lösen das Gleichungssystem mit Hilfe der MuPAD-Funktion `solve`:

- `solve({G11, G12}, {k,r})`

$$\left\{ \left[ k = \frac{1}{4}, r = \frac{-1}{4} \right] \right\}$$

Jetzt müssen wir nur noch die Werte  $k = 1/4$  und  $r = -1/4$  in die Parameterform der Geraden  $g$  bzw.  $h$  einsetzen und erhalten so die Ortsvektoren der Punkte  $L1$  bzw.  $L2$ :

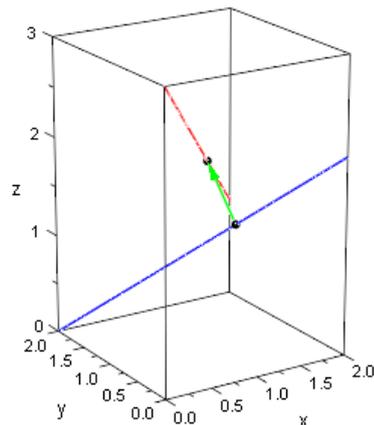
- `OrtsvektorL1:= subs(ParameterformG, k = 1/4);`  
`OrtsvektorL2:= subs(ParameterformH, r = -1/4)`

$$\begin{pmatrix} \frac{5}{4} \\ \frac{3}{4} \\ \frac{5}{4} \\ \frac{5}{4} \end{pmatrix}$$

$$\begin{pmatrix} \frac{5}{4} \\ \frac{5}{4} \\ \frac{7}{4} \\ \frac{5}{4} \end{pmatrix}$$

Zeichnen wir die Punkt  $L1$  und  $L2$  sowie den Vektor  $\overrightarrow{L1L2}$  zusammen mit den beiden Geraden  $g$  und  $h$  in ein gemeinsames Koordinatensystem, so ergibt sich<sup>46</sup>:

- `L1:= plot::Point3d(OrtsvektorL1, Color = RGB::Black):`  
`L2:= plot::Point3d(OrtsvektorL2, Color = RGB::Black):`  
`VektorL1L2:= plot::Arrow3d(OrtsvektorL1, OrtsvektorL2,`  
`Color = RGB::Green):`  
`plot(g, h, L1, L2, VektorL1L2, Scaling = Constrained)`



Der Abstand von  $g$  und  $h$  entspricht jetzt also der Länge des Vektors  $\overrightarrow{L1L2}$  (oben in grüner Farbe eingezeichnet):

- `d:= norm(OrtsvektorL2 - OrtsvektorL1, 2)`

$$\frac{\sqrt{2}}{2}$$

Auf 10 signifikante Stellen gerundet ergibt das:

- `float(d)`  
0.7071067812

<sup>46</sup>Wir verwenden wieder die Option `Scaling = Constrained`, damit alle Achsen gleich skaliert werden und rechte Winkel auch als solche erkennbar sind.

Wie schon bei der Abstandsberechnung eines Punktes von einer Geraden, wollen wir auch noch eine alternative Möglichkeit vorstellen, mit der sich der Abstand zweier windschiefer Geraden berechnen lässt. Auch hier definiert man eine Hilfsebene  $H$ . Die Idee ist die folgende: Wir bestimmen eine Ebene  $H$ , die die Gerade  $h$  enthält und parallel zu der Geraden  $g$  verläuft. Haben wir eine solche Ebene gefunden, so müssen wir nur noch den Abstand der Geraden  $g$  von der Ebene  $H$  berechnen. Wie man allerdings den Abstand einer Geraden von einer Ebene berechnet, werden wir erst ein wenig später diskutieren.

### Übungen

43.) Zeigen Sie rechnerisch, dass die beiden Geraden  $g$  und  $h$ , die zu Beginn dieses Abschnitts definiert wurden, in der Tat windschief sind.

44.) Gegeben seien die beiden Geraden

$$g : \vec{x} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \quad \text{und} \quad h : \vec{x} = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}.$$

Zeigen Sie, dass  $g$  und  $h$  windschief sind und berechnen Sie ihren Abstand. Zeichnen Sie anschließend beide Geraden mit den beiden Lotfußpunkten und deren Verbindungsvektor in ein gemeinsames Koordinatensystem.

#### 3.10.4 Abstand eines Punktes von einer Ebene im $\mathbb{R}^3$

Wir können bisher den Abstand eines Punktes von einer Geraden, den Abstand zweier paralleler Geraden und den Abstand zweier windschiefer Geraden berechnen. Wie steht es nun mit dem Abstand eines Punktes oder einer Geraden von einer Ebene? In diesem Abschnitt wollen wir uns zunächst überlegen, wie man den Abstand eines Punktes von einer Ebene berechnen kann. Den Abstand einer Geraden von einer Ebene zu berechnen ist leicht, wenn wir dies beherrschen.

Gegeben sei also eine Ebene  $F$  und ein Punkt  $P$ , der nicht auf der Ebene liegt (andernfalls ist der Abstand des Punktes zur Ebene eh Null und es gibt gar nichts zu berechnen). Zur Berechnung des Abstands von  $P$  zu  $F$  müssen wir das Lot von  $P$  auf  $F$  bestimmen. Die Situation ist in Abbildung 3.16 veranschaulicht.

Der in Abbildung 3.16 in grüner Farbe eingezeichnete Lotvektor vom Lotfußpunkt auf der Ebene (diesen wollen wir im folgenden mit  $L$  bezeichnen) hin zu dem Punkt  $P$  steht senkrecht auf der Ebene  $F$ . Er ist also nichts weiter als ein Normalenvektor der Ebene  $F$ . Wir können uns daher einen beliebigen Normalenvektor  $\vec{n}$  der Ebene  $F$  wählen und die Gerade durch den Punkt  $P$  mit Richtungsvektor  $\vec{n}$  bestimmen. Anschließend bestimmen wir den Schnittpunkt dieser so genannten Lotgeraden mit der Ebene  $F$  und erhalten so den Lotfußpunkt  $L$  auf der Ebene  $F$ . Der Abstand von  $P$  zu  $F$  ist dann

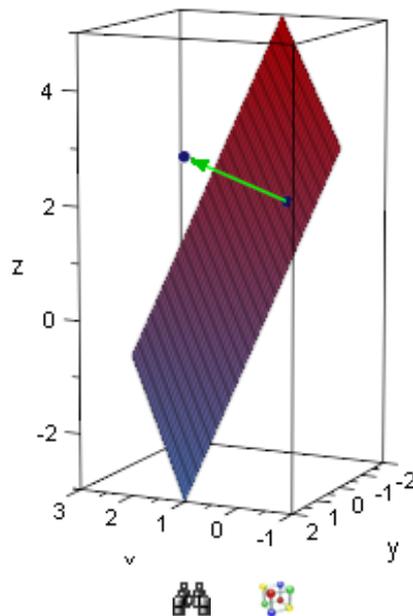


Abbildung 3.16: Abstand eines Punktes von einer Ebene

gegeben durch die Länge des Vektors  $\overrightarrow{LP}$  (in grüner Farbe in Abbildung 3.16 zu sehen).

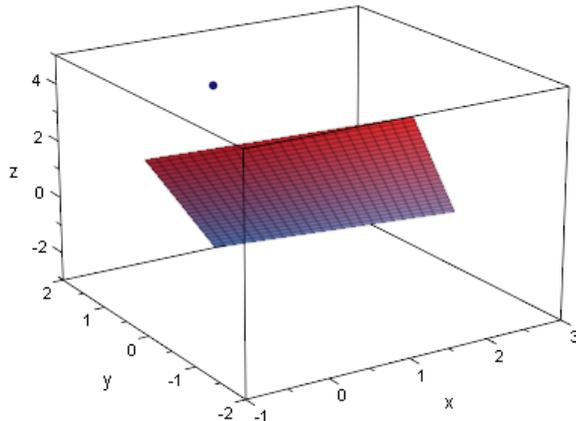
Betrachten wir z.B. die Ebene  $F$  in Parameterform

$$F := \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix} + s \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

sowie den Punkt  $P(1|2|3)$ , der nicht auf der Ebene  $F$  liegt<sup>47</sup>. Wir definieren die Parameterform der Ebene  $F$  in MuPAD und zeichnen die Ebene und den Punkt  $P$  in ein gemeinsames Koordinatensystem:

- ```
ParameterformF:= matrix([1,0,1]) + r*matrix([1,1,-3])
+ s*matrix([1,-1,1]):
OrtsvektorP:= matrix([1,2,3]):
F:= plot::Surface(ParameterformF, r = -1..1, s = -1..1):
P:= plot::Point3d(OrtsvektorP):
plot(F,P)
```

<sup>47</sup>Diese Behauptung ist in den Übungsaufgaben am Ende dieses Abschnitts nachzuweisen.



Als erstes bestimmen wir einen Normalenvektor  $\vec{n}$  der Ebene  $F$  mit Hilfe von `linalg::crossProduct` als Kreuzprodukt der beiden Spannvektoren der Ebene:

- `n:= linalg::crossProduct(matrix([1,1,-3]),  
matrix([1,-1,1]))`

$$\begin{pmatrix} -2 \\ -4 \\ -2 \end{pmatrix}$$

Dann stellen wir eine Parameterform der Lotgeraden  $l$  durch  $P$  mit Richtungsvektor  $\vec{n}$  auf

- `ParameterformL:= OrtsvektorP + k*n`

$$\begin{pmatrix} 1 - 2k \\ 2 - 4k \\ 3 - 2k \end{pmatrix}$$

und bestimmen den Schnittpunkt mit der Ebene  $F$ , indem wir die Parameterform der Ebene  $F$  und die Parameterform der Gerade  $l$  gleichsetzen und das entstehende Gleichungssystem `G1` mit Hilfe von `solve` nach den Parametern  $r$ ,  $s$  und  $k$  lösen:

- `G1:= {ParameterformF[1] = ParameterformL[1],  
ParameterformF[2] = ParameterformL[2],  
ParameterformF[3] = ParameterformL[3]}:  
solve(G1, {r,s,k})`

$$\left\{ \left[ k = \frac{1}{2}, r = \frac{-1}{2}, s = \frac{-1}{2} \right] \right\}$$

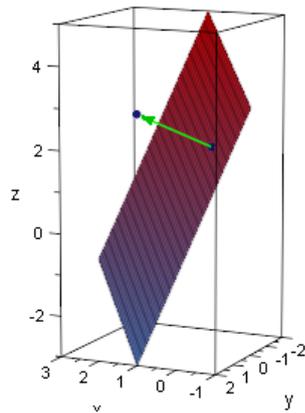
Setzen wir die Werte  $r = -1/2$  und  $s = -1/2$  in die Parameterform der Ebene  $F$  ein, so erhalten wir die Ortsvektor des Lotfußpunktes  $L$  auf der Ebene  $F$ :

- `OrtsvektorL:= subs(ParameterformF, r = -1/2, s = -1/2)`

$$\begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

Zur Veranschaulichung unserer bisherigen Ergebnisse zeichnen wir den Punkt  $P$ , die Ebene  $F$  sowie den Lotfußpunkt  $L$  und den Vektor  $\overrightarrow{LP}$  in ein gemeinsames Koordinatensystem <sup>48</sup>:

- `L:= plot::Point3d(OrtsvektorL):`  
`VektorLP:= plot::Arrow3d(OrtsvektorL, OrtsvektorP,`  
`Color = RGB::Green):`  
`plot(F, P, L, VektorLP, Scaling = Constrained)`



Der Abstand des Punktes  $P$  zu der Ebene  $F$  entspricht nun der Länge des Vektors  $\overrightarrow{LP}$ , also:

- `d:= norm(OrtsvektorP - OrtsvektorL, 2)`  
 $\sqrt{6}$

bzw. wieder auf 10 signifikante Stellen gerundet:

- `float(d)`  
2.449489743

Liegt die Ebene  $F$  in Koordinatenform  $a \cdot x + b \cdot y + c \cdot z + d = 0$  vor, so entsprechen die Koeffizienten  $a$ ,  $b$  und  $c$  von  $x$ ,  $y$  und  $z$  den Komponenten eines Normalenvektors der Ebene  $F$ . Man kann dann die Lotgerade durch den Punkt, dessen Abstand zur

<sup>48</sup>Wir haben die Grafik leicht gedreht, damit man den Lotvektor  $\overrightarrow{LP}$  besser sehen kann. Des weiteren wurde wieder die Option `Scaling = Constrained` verwendet, damit die Achsen gleich skaliert werden und der rechte Winkel zwischen der Ebene und dem Lotvektor auch als solcher in der Grafik erscheint.

Ebene  $F$  berechnet werden soll, ganz genau so berechnen, wie wir es oben im Fall der Ebene in Parameterform getan haben. Lediglich der Lotfußpunkt als Schnittpunkt der Lotgerade mit der Ebene muss dann auf andere Weise berechnet werden. Wie man aber den Schnittpunkt einer Geraden mit einer Ebene in Koordinatenform berechnet, haben wir bereits in Abschnitt 3.6.2 behandelt. Damit können wir auch den Abstand eines beliebigen Punktes von einer Ebene in Koordinatenform berechnen.

### Übungen

- 45.) Zeigen Sie rechnerisch, dass der oben definierte Punkt  $P$  nicht auf der oben definierten Ebene  $F$  liegt.
- 46.) Gegeben sei der Punkt  $P(1|3|1)$  und die Ebene  $F$  in Parameterform

$$F : \vec{x} = \begin{pmatrix} 1 \\ 3 \\ -1 \end{pmatrix} + r \cdot \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} + s \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

Zeigen Sie, dass  $P$  nicht auf  $F$  liegt. Bestimmen Sie den Abstand des Punktes  $P$  von der Ebene  $F$ . Zeichnen Sie anschließend sowohl den Punkt  $P$  als auch die Ebene  $F$  zusammen mit dem entsprechenden Lotfußpunkt und dem Lotvektor in ein gemeinsames Koordinatensystem.

### 3.10.5 Abstand einer Gerade von einer Ebene im $\mathbb{R}^3$

Den Abstand einer Gerade zu einer Ebene zu bestimmen ist jetzt nach dem, was wir im vorhergehenden Abschnitt erarbeitet haben, sehr leicht. Sei  $F$  eine Ebene (in Koordinatenform oder in Parameterform),  $\vec{n}$  wieder ein Normalenvektor der Ebene sowie  $g : \vec{x} = \vec{a} + k \cdot \vec{u}$  eine Gerade. Wir setzen im folgenden voraus, dass  $g$  parallel zu der Ebene  $F$  verläuft. Wenn  $g$  in der Ebene  $F$  liegt, so ist der Abstand von  $g$  zu  $F$  eh Null und es ist nichts zu berechnen. Wenn die Gerade  $g$  die Ebene  $F$  schneidet, ist der Abstand ebenfalls Null (der Abstand der Gerade zur Ebene ist der minimale Abstand eines Punktes der Geraden zur Ebene – also im Fall, dass  $g$  die Ebene  $F$  scheidet, der Abstand des Schnittpunktes von der Ebene).

Wenn  $g$  aber parallel zu der Ebene  $F$  verläuft, dann hat jeder Punkt auf  $g$  den gleichen Abstand zur Ebene  $F$  und wir können uns einen beliebigen Punkt auf  $g$  auswählen und nach der im vorhergehenden Abschnitt kennen gelernten Methode den Abstand dieses Punktes zur Ebene  $F$  berechnen. Am einfachsten macht man sich selbst das Leben, wenn man den Punkt  $A$  auf  $g$  wählt, also den Punkt des Ortsvektor der Stützvektor der Geraden  $g$  ist.

Schauen wir uns ein Beispiel an: Die Gerade  $g$  gegeben durch

$$g : \vec{x} = \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$



und bestimmen dann die Lotgerade  $l$  mit Richtungsvektor  $\mathbf{n}$ , die durch den Punkt  $P(1| -2| -2)$ :

- `ParameterformL := OrtsvektorP + m*n:`

Schließlich bestimmen wir den Schnittpunkt der Lotgeraden mit der Ebene  $F$ , indem wir die Parameterform der Ebene und die Parameterform der Geraden gleichsetzen und das entstehenden lineare Gleichungssystem nach  $k$ ,  $r$  und  $s$  lösen:

- `G1 := {ParameterformF[1] = ParameterformL[1],  
ParameterformF[2] = ParameterformL[2],  
ParameterformF[3] = ParameterformL[3]}:`  
`solve(G1, {r,s,m})`  
 $\left\{ \left[ m = \frac{14}{17}, r = \frac{-2}{17}, s = \frac{-19}{17} \right] \right\}$

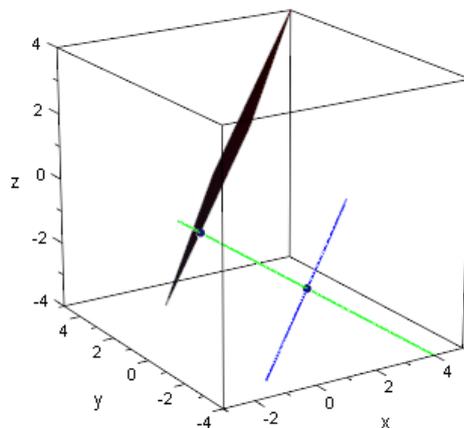
Setzen wir die Werte  $r = -2/17$  und  $s = -19/17$  in die Parameterform der Ebene  $F$  ein, so erhalten wir den Ortsvektor des Lotfußpunktes  $L$  auf der Ebene  $F$ :

- `OrtsvektorL := subs(ParameterformF, r = -2/17,  
s = -19/17)`

$$\begin{pmatrix} -\frac{25}{17} \\ -\frac{6}{17} \\ -\frac{6}{17} \end{pmatrix}$$

Zur Veranschaulichung zeichnen wir den Lotfußpunkt und die Lotgerade mit der Geraden  $g$ , dem Punkt  $P$  sowie der Ebene  $F$  in ein gemeinsames Koordinatensystem:

- `L := plot::Point3d(OrtsvektorL):`  
`Lotgerade := plot::Curve3d(ParameterformL, m = -1..1,  
Color = RGB::Green):`  
`plot(P, g, F, L, Lotgerade,  
Scaling = Constrained)`



Der Abstand von  $F$  und  $g$  ist also jetzt durch die Länge des Lotvektors  $\overrightarrow{LP}$  gegeben:

- `d := norm(OrtsvektorP - OrtsvektorL, 2)`

$$\frac{14\sqrt{17}}{17}$$

bzw. auf 10 signifikante Stellen gerundet ergibt sich:

- `float(d)`  
3.395498751

## Übungen

- 47.) Zeigen Sie rechnerisch, dass die oben definierte Gerade  $g$  parallel zu der Ebene  $F$  verläuft.
- 48.) Gegeben seien die Gerade  $g$  und die Ebene  $F$  in Parameterform

$$g : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ -1 \\ -3 \end{pmatrix}$$

und

$$F : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} + s \cdot \begin{pmatrix} -2 \\ 1 \\ 3 \end{pmatrix}.$$

Zeigen Sie, dass  $g$  parallel zu  $F$  verläuft. Bestimmen Sie den Abstand der Geraden  $g$  von der Ebene  $F$ . Zeichnen Sie anschließend sowohl die Ebene  $F$  als auch die Gerade  $g$  sowie den Lotfußpunkt und die Lotgerade in ein gemeinsames Koordinatensystem.

### 3.10.6 Noch einmal Abstand zweier windschiefer Geraden

Wir kommen in diesem Abschnitt noch einmal auf die Abstandsbestimmung bei zwei windschiefen Geraden zurück. Seien also

$$g : \vec{x} = \vec{a} + r \cdot \vec{u} \quad \text{und} \quad h : \vec{x} = \vec{b} + s \cdot \vec{v}$$

windschief zueinander. Dann können wir den Abstand von  $g$  und  $h$  auch mit Hilfe einer Ebene  $H$  bestimmen, die die Gerade  $h$  enthält und parallel zur Geraden  $g$  im Raum liegt. Wenn wir eine solche Ebene gefunden haben, dann berechnen wir einfach den Abstand der Ebene  $F$  zu der Geraden  $g$  auf genau die Art und Weise, die wir im vorhergehenden Abschnitt kennen gelernt haben. Die Situation ist in Abbildung 3.17 veranschaulicht.

Wie findet man eine Gleichung der Ebene  $F$ , die  $h$  enthält und parallel zu  $g$  verläuft? Nichts leichter als das: Wir übernehmen einfach die Parameterform der Geraden  $h$  (damit ist sichergestellt, dass die Ebene, die wir definieren, in jedem Fall die Gerade  $h$

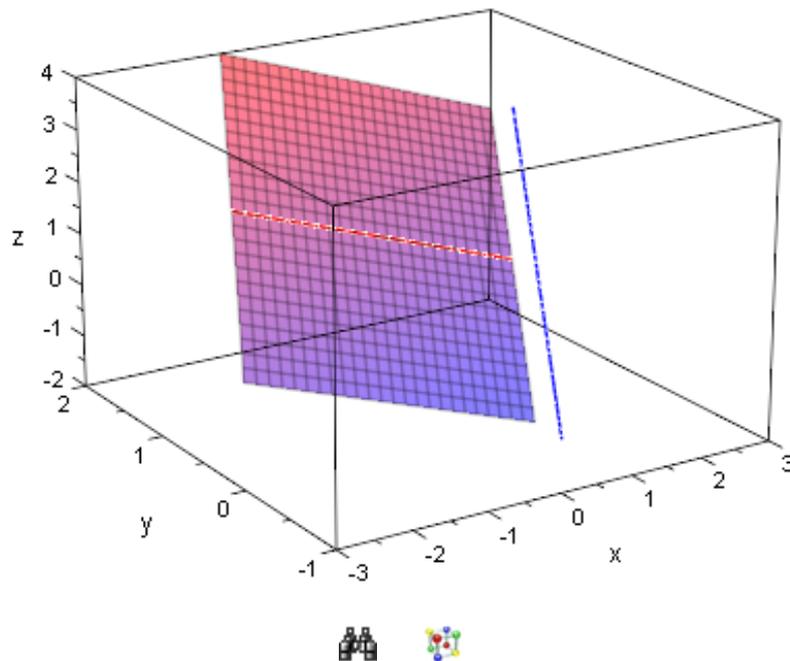


Abbildung 3.17: Alternative Abstandsbestimmung windschiefer Geraden

enthält) und ergänzen diese um den Richtungsvektor der Geraden  $g$ . Die allgemeine Gleichung der Ebene  $F$  lautet also

$$F: \vec{x} = \vec{b} + p \cdot \vec{v} + q \cdot u.$$

Schauen wir uns ein konkretes Beispiel an: Die beiden Geraden

$$g: \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{und} \quad h: \vec{x} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + s \cdot \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$$

sind windschief<sup>50</sup>. Wollen wir den Abstand von  $g$  und  $h$  bestimmen, so können wir also wie folgt vorgehen: Wir definieren zunächst die Parameterformen der beiden Geraden in MuPAD

- `ParameterformG:= matrix([1,0,1]) + r * matrix([1,1,2]):`  
`ParameterformH:= matrix([0,1,1]) + s * matrix([-2,0,1]):`

und stellen anschließend die Parameterform der Ebene  $F$  auf, die die Gerade  $h$  enthält und parallel zu Geraden  $g$  im Raum liegt<sup>51</sup>:

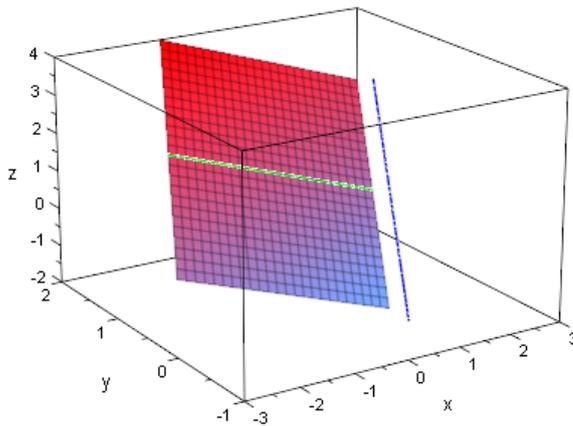
<sup>50</sup>Diese Behauptung ist in den Übungsaufgaben am Ende dieses Abschnitts nachzuweisen.

<sup>51</sup>Um später "Namenskonflikte" bei den vorkommenden Parametern zu vermeiden, definieren wir die Parameterform der Ebene  $F$  explizit nicht mit Hilfe der Variablen `ParameterformG` und `ParameterformH`.

- `ParameterformF:= matrix([0,1,1]) + p*matrix([-2,0,1])  
+ q*matrix([1,1,2]):`

Bevor wir mit der Abstandsberechnung fortfahren, stellen wir die Situation grafisch dar:

- `g:= plot::Curve3d(ParameterformG, r = -1..1):`  
`h:= plot::Curve3d(ParameterformH, s = -1..1,`  
`Color = RGB::Green):`  
`F:= plot::Surface(ParameterformF, p = -1..1,`  
`q = -1..1):`  
`plot(g, h, F)`



Damit ist der Abstand von  $g$  und  $h$  gleich dem Abstand der Geraden  $g$  von der Ebene  $F$ . Diesen Abstand bestimmen wir jetzt genau so, wie wir es uns im letzten Abschnitt überlegt haben, d.h. wir bestimmen den Abstand des Stützpunktes  $(1|0|1)$  auf der Geraden  $g$  von der Ebene  $F$ . Zuerst müssen wir wieder einen Normalenvektor  $\vec{n}$  der Ebene  $F$  berechnen und mit Hilfe dieses Vektors die Lotgerade  $l$  durch den Punkt  $P$  mit Richtungsvektor  $n$  sowie deren Schnittpunkt mit der Ebene  $F$  (dies lieferte uns ja den Lotfußpunkt auf der Ebene):

- `n:= linalg::crossProduct(matrix([-2,0,1]),  
matrix([ 1,1,2]))`

$$\begin{pmatrix} -1 \\ 5 \\ -2 \end{pmatrix}$$

Eine Parameterform der Lotgerade  $l$  lautet also:

- `ParameterformL:= matrix([1,0,1]) + m*n:`

$$\begin{pmatrix} 1 - m \\ 5m \\ 1 - 2m \end{pmatrix}$$

Schließlich bestimmen wir den Schnittpunkt der Lotgeraden mit der Ebene  $F$ , indem wir die Parameterform der Ebene und die Parameterform der Geraden gleichsetzen und das entstehenden lineare Gleichungssystem nach  $p$ ,  $q$  und  $m$  lösen:

- $G1 := \{ \text{ParameterformF}[1] = \text{ParameterformL}[1],$   
 $\text{ParameterformF}[2] = \text{ParameterformL}[2],$   
 $\text{ParameterformF}[3] = \text{ParameterformL}[3] \}:$   
 $\text{solve}(G1, \{p, q, m\})$   
 $\left\{ \left[ m = \frac{1}{5}, p = \frac{-2}{5}, q = 0 \right] \right\}$

Setzen wir die Werte  $p = -2/5$  und  $q = 0$  in die Parameterform der Ebene  $F$  ein, so erhalten wir den Ortsvektor des Lotfußpunktes  $L$  auf der Ebene  $F$ :

- $\text{OrtsvektorL} := \text{subs}(\text{ParameterformF}, p = -2/5,$   
 $q = 0)$   
 $\begin{pmatrix} \frac{4}{5} \\ 1 \\ \frac{3}{5} \end{pmatrix}$

Der Abstand von  $F$  und  $g$  ist also jetzt durch die Länge des Lotvektors  $\overrightarrow{LP}$  gegeben:

- $d := \text{norm}(\text{matrix}([1, 0, 1]) - \text{OrtsvektorL}, 2)$   
 $\frac{\sqrt{6} \sqrt{5}}{5}$

bzw. auf 10 signifikante Stellen gerundet ergibt sich:

- $\text{float}(d)$   
1.095445115

## Übungen

49.) Zeigen Sie rechnerisch, dass die oben definierten Gerade  $g$  und  $h$  windschief zueinander im Raum liegen.

50.) Gegeben seien die Geraden  $g$  und  $h$  in Parameterform

$$g : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ -1 \\ -3 \end{pmatrix}$$

und

$$h : \vec{x} = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix} + s \cdot \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

---

Zeigen Sie, dass  $g$  und  $h$  windschief zueinander sind. Bestimmen Sie den Abstand von  $g$  und  $h$  nach der in diesem Abschnitt beschriebenen Methode mit Hilfe einer Ebene, die eine Gerade enthält und parallel zu der anderen Geraden im Raum liegt. Zeichnen Sie die beiden Geraden und die von Ihnen bestimmte Ebene in ein gemeinsames Koordinatensystem.

## 4 Kreise und Kugeln

In Kapitel 3 haben wir Geraden und Ebenen betrachtet, also lineare Objekte (die Koordinatenform einer Ebene enthält die Variablen  $x$ ,  $y$  und  $z$  alle in linearer Form; es taucht z.B. nie das Quadrat einer der Variablen in der Koordinatenform einer Ebene auf). Hier kommen wir jetzt zu einfachen nicht-linearen Gleichungen, mit deren Hilfe sich Kreise im  $\mathbb{R}^2$  bzw. Kugeln im  $\mathbb{R}^3$  beschreiben lassen.

Das mathematische Handwerkszeug, das zur Beschreibung von Kreisen und Kugeln benötigt wird, haben wir uns bereits zum größten Teil in Kapitel 2 angeeignet. Schon im letzten Kapitel haben wir gesehen, dass die Anzahl der neu zu lernenden MuPAD-Funktionen von Abschnitt zu Abschnitt weniger wird. In diesem Kapitel werden wir im Prinzip nur wenige weitere MuPAD-Funktionen benötigen, die bisher noch nicht vor gekommen sind, darunter die Funktion `plot::Circle2d` zur Darstellung von Kreisen im  $\mathbb{R}^2$  sowie die Funktion `plot::Sphere` zur Darstellung von Kugeln im  $\mathbb{R}^3$ .

### 4.1 Beschreibung von Kreisen im $\mathbb{R}^2$

#### Neue MuPAD-Funktionen

<code>plot::Circle2d</code>	Darstellung von Kreisen im Zweidimensionalen
<code>lhs</code>	Zugriff auf die linke Seite einer Gleichung
<code>rhs</code>	Zugriff auf die rechte Seite einer Gleichung

Bevor wir zu Kugeln im Raum kommen, überlegen wir uns zuerst den einfacheren Fall, nämlich wie man Kreise im Zweidimensionalen beschreiben kann. Die Gleichung, die einen Kreis beschreibt, kann keine Funktionsgleichung sein, wie wir sie aus dem Analysisunterricht kennen, denn bei der einer Funktion wird jedem  $x$ -Wert eindeutig ein Funktionswert zugeordnet. Das Beispiel des Kreises um den Koordinatenursprung in Abbildung 4.1 zeigt, dass diese Eigenschaft bei Kreisen nicht gegeben ist.

Die zentrale Idee zur Beschreibung eines Kreises um einen Mittelpunkt  $M$  mit Radius  $r$  ist die Beobachtung, dass natürlich jeder Punkt auf dem Kreisrand den Abstand  $r$  vom Mittelpunkt  $M$  hat. Zeichnet man die  $x$ -Koordinate und die  $y$ -Koordinate eines Punktes  $P$  auf der Kreislinie als Strecken ein, so entsteht ein rechtwinkliges Dreieck. Abbildung 4.2 zeigt die Situation für einen Kreis, dessen Mittelpunkt  $M$  der Koordinatenursprung  $(0|0)$  ist.

Dabei bilden also  $x$  und  $y$  die Katheten und  $r$  die Hypotenuse des rechtwinkligen

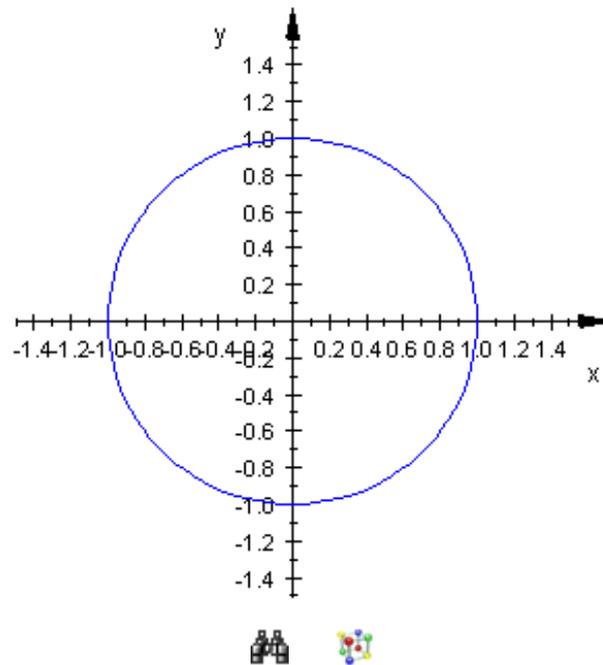
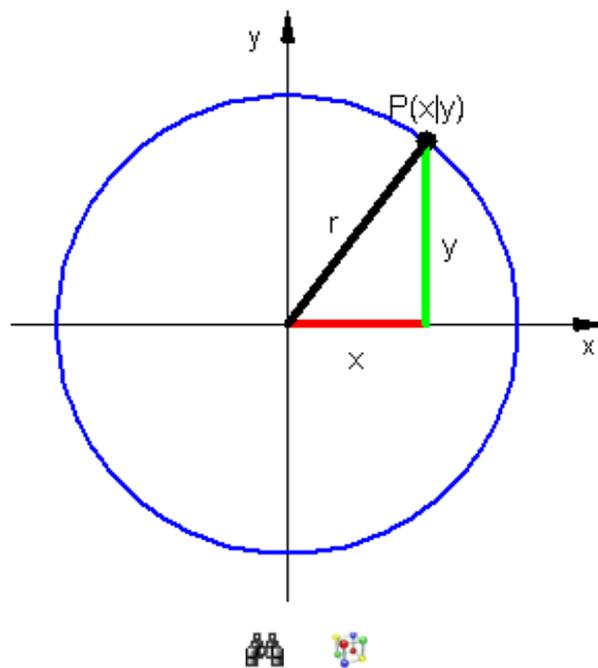
Abbildung 4.1: Kreis um den Koordinatenursprung mit Radius  $r = 1$ 

Abbildung 4.2: Beschreibung eines Kreises mit Mittelpunkt im Ursprung

Dreiecks. Nach dem *Satz des Pythagoras* muss also gelten:

$$x^2 + y^2 = r^2 \quad \text{bzw.} \quad x^2 + y^2 - r^2 = 0.$$

Damit sind wir schon fast am Ziel. Diese quadratische Gleichung in  $x$  und  $y$  beschreibt für ein gegebenes  $r$  einen beliebigen Kreis mit Radius  $r$  um den Ursprung. Wir müssen uns jetzt nur noch überlegen, wie die Gleichung eines Kreises zu lauten hat, dessen Mittelpunkt nicht im Koordinatenursprung  $(0|0)$  liegt. Die Situation ist in [Abbildung 4.3](#) dargestellt, wobei mit  $m_x$  und  $m_y$  die  $x$ - bzw.  $y$ -Koordinate des Mittelpunktes  $M$  gemeint sind.

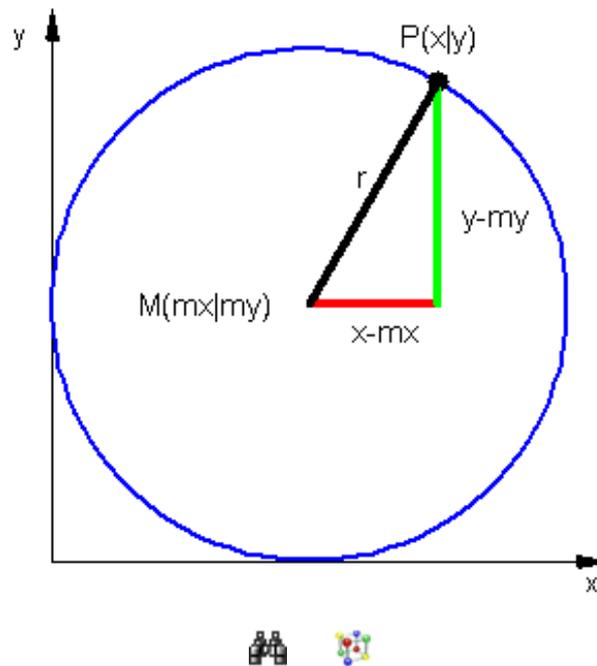


Abbildung 4.3: Beschreibung eines Kreises mit beliebigem Mittelpunkt

Wenn also der Mittelpunkt des Kreises nicht der Koordinatenursprung ist, so müssen wir lediglich die Länge der Katheten anders berechnen, d.h. wir müssen die Differenz der  $x$ -Koordinate des Punktes  $P$  auf der Kreislinie und der  $x$ -Koordinate des Mittelpunktes  $M$  berechnen und erhalten so die Länge der roten Kathete in [Abbildung 4.3](#). Ganz analog erhalten wir die Länge der grünen Kathete, indem wir die Differenz der  $y$ -Koordinate des Punkte  $P$  auf der Kreislinie und der  $y$ -Koordinate des Mittelpunktes  $M$  berechnen. Jetzt können wir wieder den *Satz des Pythagoras* verwenden, um die allgemeine Kreisgleichung eines Kreises mit Radius  $r$  und Mittelpunkt  $M(m_x|m_y)$  anzugeben:

$$(x - m_x)^2 + (y - m_y)^2 = r^2 \quad \text{bzw.} \quad (x - m_x)^2 + (y - m_y)^2 - r^2 = 0.$$

Setzen wir in diese allgemeine Kreisgleichung gedanklich für  $m_x$  den Wert 0 und für  $m_y$  den Wert 0 ein, so erhalten wir genau die Gleichung  $x^2 + y^2 - r^2 = 0$ , also die

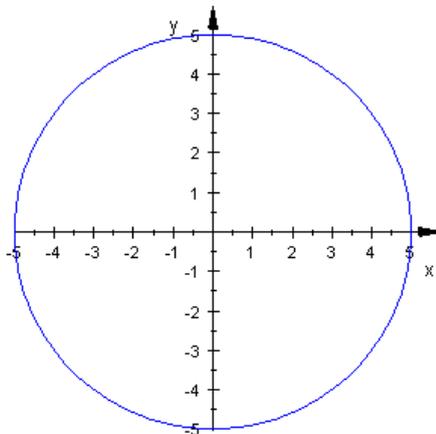
Kreisgleichung, die wir uns zuvor für Kreise mit Radius  $r$  um den Koordinatenursprung überlegt hatten.

Betrachten wir einige Beispiele: Der Kreis um den Koordinatenursprung mit Radius  $r = 5$  ist durch die Kreisgleichung

- $r := 5$ :  
Kreisgleichung:  $x^2 + y^2 = r^2$   
 $x^2 + y^2 = 25$

beschrieben. Wir können den Kreis mit Hilfe der Funktion `plot::Circle2d` darstellen lassen. Der Funktion `plot::Circle2d` müssen wir dazu den Radius des Kreises sowie – falls vom Ursprung verschieden – seinen Mittelpunkt:

- `plot(plot::Circle2d(r))`

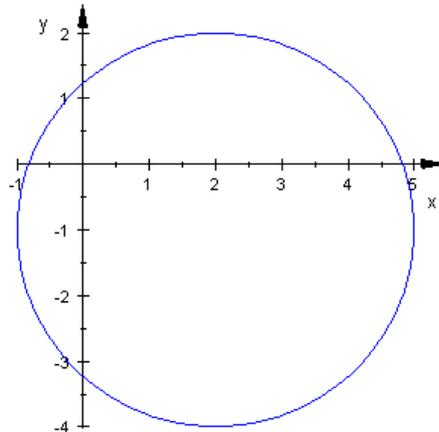


Der Kreis mit Radius  $r = 3$  und Mittelpunkt  $M(2 | -1)$  hat die Kreisgleichung

- $r := 3$ :  
Kreisgleichung:  $(x-2)^2 + (y-(-1))^2 = r^2$   
 $(x-2)^2 + (y+1)^2 = 9$

Unter Angabe des Radius und des Mittelpunktes können wir den Kreis dann von MuPAD darstellen lassen. Der Mittelpunkt kann dabei entweder als zweielementige Liste oder auch als Vektor mit zwei Komponenten angegeben werden. Da die Vektorschreibweise näher an den bisher gewohnten Schreibweisen ist, wählen wir diese:

- `plot(plot::Circle2d(r, matrix([2,-1])))`



Bei der obigen Kreisgleichung sehen wir schön, dass MuPAD die Klammern nicht ausmultipliziert. Wollen wir explizit die Klammern ausrechnen lassen, so müssen wir den Befehl `expand` auf die Kreisgleichung anwenden:

- `expand(Kreisgleichung)`

$$2y - 4x + x^2 + y^2 + 5 = 9$$

Allerdings können wir an dieser Gleichung in ausmultiplizierter Form keine wirklichen Informationen mehr ablesen: Welchen Radius hat der Kreis, den die Gleichung beschreibt? Wo liegt sein Mittelpunkt?

Um das herauszufinden, müssen wir die Gleichung  $2 \cdot y - 4 \cdot x + x^2 + y^2 + 5 = 9$  wieder in die Form  $(x - mx)^2 + (y - my)^2 = r^2$  überführen<sup>1</sup>.

Wir definieren uns also zuerst die ausmultiplizierte Gleichung in MuPAD:

- `Kreisgleichung:= 2*y - 4*x + x^2 + y^2 + 5 = 9`

$$2y - 4x + x^2 + y^2 + 5 = 9$$

Nun müssen wir alle Summanden auf der linken Seite der Gleichung<sup>2</sup> abgreifen, die  $x$  bzw.  $y$  enthalten, und darauf die quadratische Ergänzung durchführen:

- `LinkeSeite:= lhs(Kreisgleichung)`

$$2y - 4x + x^2 + y^2 + 5$$

Die Summanden, die  $x$  bzw.  $y$  enthalten, wählen wir mit dem Befehl `select` in Verbindung mit dem Befehl `has` wie folgt aus:

- `SummandX:= select(LinkeSeite, has, x);`  
`SummandY:= select(LinkeSeite, has, y)`

<sup>1</sup>Üblicherweise macht man das per Hand mit Hilfe quadratischer Ergänzung. Genau das wollen wir im folgenden mit MuPAD "simulieren".

<sup>2</sup>Auf die linke Seite einer Gleichung greift man in MuPAD mit dem Befehl `lhs` für "left hand side" zu. Entsprechend kann man mit Hilfe von `rhs` auf die rechte Seite einer Gleichung zugreifen.

$$x^2 - 4x$$

$$2y + y^2$$

Wir führen jetzt die quadratische Ergänzung zuerst auf dem Summanden `SummandX` durch, der  $x$  enthält. Quadratische Ergänzung bedeutet nichts anderes als eine reelle Zahl  $mx$  zu finden, so dass gilt:

$$\text{SummandX} + mx^2 = (x - mx)^2.$$

Diese Gleichung können wir aber in MuPAD direkt so eingeben und nach  $mx$  auflösen lassen<sup>3</sup>. Wir schließen dabei nur den Fall  $x = 0$  aus, um die triviale Lösung, dass  $mx$  jede beliebige reelle Zahl sein kann, auszuschließen<sup>4</sup>:

- `assume(x <> 0):`  
`solve(SummandX + mx^2 = (x - mx)^2, mx)`  
`{2}`

Die  $x$ -Koordinate des Mittelpunkts des Kreise ist also  $mx = 2$  (was wir hier natürlich im Prinzip schon wussten, weil wir den Kreis oben über seinen Mittelpunkt und seinen Radius definiert hatten). Die  $y$ -Koordinate  $my$  des Kreismittelpunkts muss also ganz analog die Gleichung

$$\text{SummandY} + my^2 = (y - my)^2.$$

erfüllen:

- `assume(y <> 0):`  
`solve(SummandY + my^2 = (y - my)^2, my)`  
`{-1}`

Die  $y$ -Koordinate des Kreismittelpunkts ist also  $my = -1$ , d.h. der Kreis hat insgesamt den Mittelpunkt  $M(2 | -1)$ . Wir haben also korrekt gerechnet. Nun müssen wir noch den Radius des Kreises bestimmen. Wegen

- `expand((x - 2)^2 + (y + 1)^2)`  
 $2y - 4x + x^2 + y^2 + 5$

zeigt Vergleich dieses Ausdrucks mit der Kreisgleichung

- **Kreisgleichung**

<sup>3</sup>Die Lösung der quadratischen Gleichung können wir wie gewohnt mit Hilfe von `solve` bestimmen.

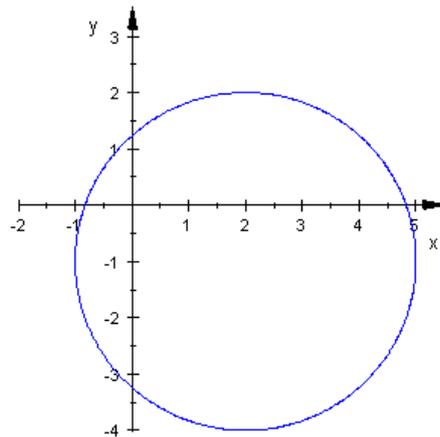
<sup>4</sup>Mit Hilfe von `assume` können Annahmen an Variablen getroffen werden. Auf diese Weise lassen sich auch Variablen als reell oder etwa positiv deklarieren.

$$2y - 4x + x^2 + y^2 + 5 = 9$$

dass  $r^2$  gleich 9 ist, also  $r = 3$  (für diese Berechnung müssen wir MuPAD nun wirklich nicht mehr bemühen). Damit haben wir unsere ursprünglichen Angaben zum Mittelpunkt und zum Radius des Kreises vollständig mit Hilfe quadratischer Ergänzung aus der ausmultiplizierten Form der Kreisgleichung wiederherstellen können.

Allgemein sollte es natürlich auch eine Möglichkeit geben, Kreise darstellen zu lassen, die mit Hilfe einer vollständig ausmultiplizierten Kreisgleichung angegeben sind, ohne zuvor explizit deren Mittelpunkt und Radius mit Hilfe quadratischer Ergänzung zu ermitteln. In MuPAD heißt die Funktion, die das für uns erledigt `plot::Implicit2d`<sup>5</sup>:

- `plot(plot::Implicit2d(Kreisgleichung, x = -2..5, y = -4..3), Scaling = Constrained)`



## Übungen

51.) Bestimmen Sie die Kreisgleichung derjenigen Kreise mit Radius  $r$  und Mittelpunkt  $M$ .

- $r = 1, M(0|0)$
- $r = 2, M(-2|1)$

<sup>5</sup>Der Funktion `plot::Implicit2d` können beliebige Gleichungen in zwei Variablen übergeben werden. Die Funktion stellt dann im Prinzip die Lösungsmenge der Gleichung dar, ohne dass diese vom Benutzer in Form einer Funktionsgleichung aufgelöst werden muss. Daher kommt auch der Name "implicit" für "implizit" definierte Kurven im Zweidimensionalen. Bei Verwendung der Funktion `plot::Implicit2d` ist es zwingend notwendig, Bereiche für  $x$  und  $y$  zu spezifizieren, die durchlaufen werden sollen. Bei ungünstiger Wahl von  $x$  und  $y$  kann es daher schon einmal passieren, dass man nur einen halben Kreis gezeichnet bekommt. Man muss dann die Bereiche für  $x$  und  $y$  entsprechend anpassen, damit man den gesamten Kreis einsehen kann. Die Option `Scaling = Constrained` ist hier zusätzlich notwendig, damit die Achsen gleich skaliert werden und der Kreis nicht etwa augenscheinlich den Eindruck einer Ellipse erweckt.

- $r = 3, M(2| - 2)$

Zeichnen Sie die Kreise mit MuPAD in ein gemeinsames Koordinatensystem. Wählen Sie für die einzelnen Kreise unterschiedliche Farben.

52.) Bestimmen Sie den Mittelpunkt und den Radius der Kreise, die durch die Gleichungen

- $8 \cdot y - 6 \cdot x + x^2 + y^2 = 0$
- $8 \cdot x - 4 \cdot y + x^2 + y^2 + 18 = 0$

gegeben sind. Ermitteln Sie den Mittelpunkt einmal zeichnerisch durch Verwendung der Funktion `plot::Implicit2d` und einmal rechnerisch mittels quadratischer Ergänzung.

## 4.2 Lagebeziehung von Punkten und Kreisen im $\mathbb{R}^2$

In diesem Abschnitt wollen wir ganz kurz die Lagebeziehung von Punkten und Kreisen in der Ebene bestimmen. Gegeben ein Kreis und ein Punkt in der Ebene, so liegt der Punkt entweder in dem Kreis, auf dem Kreisrand oder außerhalb des Kreises. Liegt ein Punkt auf dem Kreisrand, so müssen seine Koordinaten die Kreisgleichung erfüllen, d.h. hat  $P$  die Koordinaten  $P(p_1|p_2)$  und ist  $(x - mx)^2 + (y - my)^2 = r^2$  eine Gleichung des Kreises mit Mittelpunkt  $M(mx|my)$  und Radius  $r$ , so muss gelten:  $(p_1 - mx)^2 + (p_2 - my)^2 = r^2$ .

Vom Standpunkt der Vektorrechnung können wir die allgemeine Kreisgleichung auch als Abstandsberechnung des Punktes  $P$  vom Mittelpunkt des Kreises betrachten. Es gilt nämlich:

$$(p_1 - mx)^2 + (p_2 - my)^2 = r^2 \iff \left| \begin{pmatrix} mx \\ my \end{pmatrix} - \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \right|^2 = r^2.$$

Wenn also  $P$  den Abstand  $r$  vom Mittelpunkt  $M$  des Kreises hat, so liegt  $P$  auf der Kreislinie. Diese Sichtweise ist recht nützlich, um einzusehen, wann der Punkt  $P$  im Innern und wann er außerhalb des Kreises liegt. Gilt nämlich  $(p_1 - mx)^2 + (p_2 - my)^2 < r^2$ , so liegt der Punkt  $P$  im Innern des Kreises, denn der Abstand des Punktes  $P$  vom Mittelpunkt  $M$  des Kreises ist echt kleiner als der Radius  $r$ . Im Falle  $(p_1 - mx)^2 + (p_2 - my)^2 > r^2$  liegt  $P$  außerhalb des Kreises. Diese Idee greifen wir in Abschnitt ?? wieder auf, um Kugeln zu beschreiben, und wird dort auch noch ein wenig ausführlicher erklärt, als es zum jetzigen Zeitpunkt hier der Fall ist.

Die drei verschiedenen Lagebeziehungen zwischen Punkten und Kreisen sind in Abbildung 4.4 veranschaulicht.

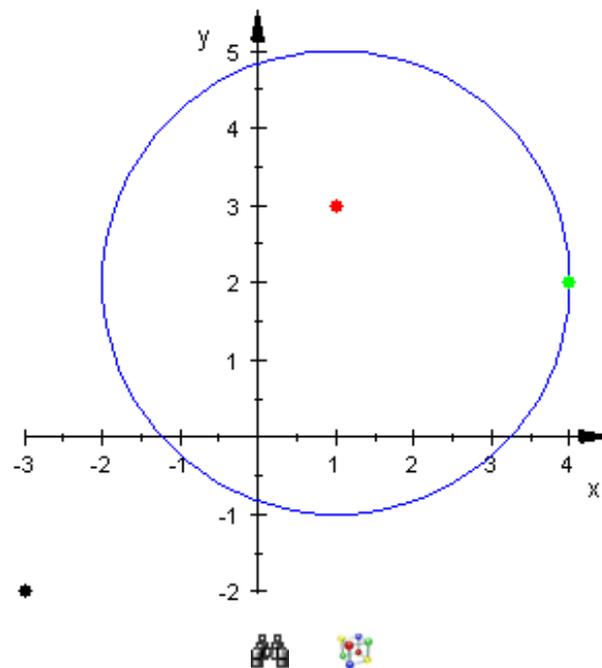


Abbildung 4.4: Lagebeziehungen von Punkten und Kreisen

Betrachten wir einige Beispiele. Gegeben seien der Kreis mit Radius  $r = 3$  mit Mittelpunkt  $M(1|2)$  sowie die drei Punkte  $A(1|3)$ ,  $B(4|2)$  und  $C(-3|-2)$ . Um die Lagebeziehung der Punkte mit MuPAD zu untersuchen, definieren wir zunächst die Kreisgleichung:

- $K := (x - 1)^2 + (y - 2)^2 = 3^2$   
 $(x - 1)^2 + (y - 2)^2 = 9$

Mit Hilfe des Befehls `subs` können wir für  $x$  und  $y$  die Koordinaten des jeweiligen Punktes einsetzen:

- `subs(K, x = 1, y = 3)`  
 $1 = 9$

Die Gleichung ist offensichtlich nicht erfüllt, d.h.  $A$  liegt nicht auf dem Kreisrand. Wegen  $1 < 9$  folgt aber, dass  $A$  im Innern des Kreises liegt. Für die Lage des Punktes  $B$  zum Kreis  $K$  erhalten wir:

- `subs(K, x = 4, y = 2)`  
 $9 = 9$

Die Koordinaten des Punktes  $A$  erfüllen also die Kreisgleichung, d.h.  $A$  liegt auf dem Rand des Kreises. Der Punkt  $C$  liegt außerhalb des Kreises, denn seine Koordinaten erfüllen die Kreisgleichung nicht und  $32 > 9$ :

- `subs(K, x = -3, y = -2)`

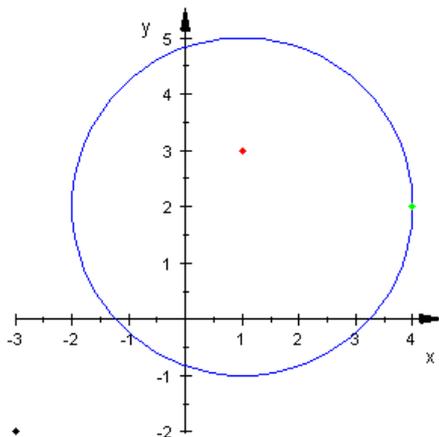
$$32 = 9$$

Zum Anschluss zeichnen wir den Kreis  $K$  sowie die drei Punkte  $A$ ,  $B$  und  $C$  noch in ein gemeinsames Koordinatensystem:

```

• Kreis:= plot::Circle2d(3, matrix([2,1])):
  A:= plot::Point2d(matrix([ 1, 3]), Color = RGB::Red):
  B:= plot::Point2d(matrix([ 4, 2]), Color = RGB::Green):
  C:= plot::Point2d(matrix([-3,-2]), Color = RGB::Black):
  plot(Kreis, A, B, C)

```



### Übungen

53.) Bestimmen Sie die Lage der Punkte  $A(-3|-2)$ ,  $B(-1|1)$  und  $C(2|2)$  zu dem Kreis  $K$  mit Radius  $r = 2$  um den Koordinatenursprung.

54.) Gegeben ist der Kreis mit der Gleichung

$$K : (x - 2)^2 + (y + 1)^2 = 25$$

sowie der allgemeine Punkt  $P(6|r)$ . Bestimmen Sie alle Werte für  $r$  derart, dass  $P$  auf dem Kreisrand liegt.

## 4.3 Lagebeziehung von Geraden und Kreisen im $\mathbb{R}^2$

Eine Gerade kann einen Kreis entweder in zwei Punkten schneiden<sup>6</sup>, sie kann den Kreis in genau einem Punkt berühren<sup>7</sup> oder außerhalb an dem Kreis vorbei laufen, ohne mit der Kreislinie einen gemeinsamen Punkt zu haben<sup>8</sup>. Die drei verschiedenen Lagebeziehungen sind in Abbildung 4.5 veranschaulicht.

<sup>6</sup>Eine solche Gerade nennt man dann eine *Sekante* des Kreises.

<sup>7</sup>Eine solche Gerade nennt man dann eine *Tangente* an den Kreis und den Punkt  $B$ , in dem sie den Kreis berührt, den *Drehpunkt*.

<sup>8</sup>Eine solche Gerade nennt man dann eine *Passante*.

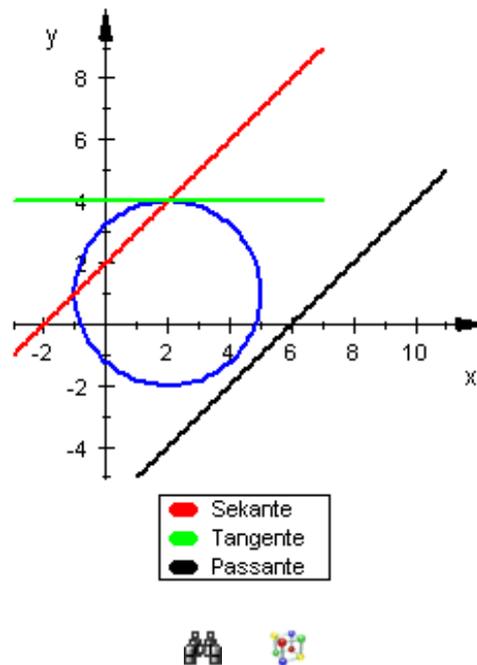


Abbildung 4.5: Lagebeziehungen von Kreisen und Geraden

Da es für Geraden im  $\mathbb{R}^2$  sowohl eine Parameterform als auch eine Koordinatenform gibt (siehe auch Abschnitt 3.1.2), müssen wir im folgenden zunächst unterscheiden, in welcher Form uns die Gerade gegeben ist.

### 4.3.1 Geraden in Koordinatenform

Gegeben sei ein Kreis  $K$  mit Radius  $r$  und Mittelpunkt  $M(mx|my)$  sowie eine Gerade  $g$  in Koordinatenform. Um die gegenseitige Lage der Geraden und des Kreises zu untersuchen, können wir wieder so vorgehen, dass wir versuchen, gemeinsame Punkte zu berechnen. Gibt es zwei gemeinsame Punkte, so ist die Gerade eine Sekante des Kreises. Gibt es genau einen gemeinsamen Punkt, so berührt die Gerade den Kreis in demselben und ist folglich eine Tangente. Hat die Gerade gar keinen Punkt mit dem Kreis gemeinsam, so handelt es sich um eine Passante.

Wir betrachten wieder Beispiele, um uns die Vorgehensweise zu veranschaulichen. Der Kreis  $K$  habe seinen Mittelpunkt bei  $M(2|1)$  und den Radius  $r = 3$ . Ferner seien die drei Geraden

$$g1 : y = x - 2, \quad g2 : y = 4, \quad g3 : y = x - 6$$

gegeben. Wir definieren zunächst die Gleichung des Kreises  $K$  sowie die Koordinatenformen der Geraden in MuPAD<sup>9</sup>:

<sup>9</sup>Hier setzen wir mittels `assume(x,y, Type::Real)` voraus, dass  $x$  und  $y$  nur reelle Werte annehmen dürfen.

- `assume({x,y}, Type::Real):`  
`K:= (x - 2)^2 + (y - 1)^2 = 3^2:`  
`KFormG1:= y = x - 2:`  
`KFormG2:= y = 4:`  
`KFormG3:= y = x - 6:`

Dann setzen wir nacheinander jede der Geradengleichungen in die Kreisgleichung ein<sup>10</sup>. Wir erhalten dann in allen drei Fällen eine Gleichung in  $x$  und können diese mit Hilfe von `solve` lösen:

- `G1:= subs(K, KFormG1)`

$$(x - 2)^2 + (x - 3)^2 = 9$$

- `L:= solve(G1, x)`

$$\left\{ \frac{5}{2} - \frac{\sqrt{17}}{2}, \frac{\sqrt{17}}{2} + \frac{5}{2} \right\}$$

Im Fall der Geraden  $g_1$  gibt es also zwei gemeinsame Punkte mit dem Kreis  $K$ . Die  $x$ -Koordinaten dieser beiden Punkte haben wir soeben berechnet. Die  $y$ -Koordinaten erhalten wir, indem wir die obigen Werte entweder in die Geradengleichungen oder in die Kreisgleichung einsetzen und die verbleibende Gleichung dann nach  $y$  auflösen. Der Einfachheit halber setzen wir die berechneten Werte in die Geradengleichung ein<sup>11</sup>:

- `subs(KFormG1, x = L[1]);`  
`subs(KFormG1, x = L[2])`

$$y = \frac{1}{2} - \frac{\sqrt{17}}{2}$$

$$y = \frac{\sqrt{17}}{2} + \frac{1}{2}$$

Wie liegt nun die Gerade  $g_2$  zu dem Kreis  $K$ ? Wir verfahren ganz analog zur obigen Vorgehensweise:

- `G1:= subs(K, KFormG2)`

$$(x - 2)^2 + 9 = 9$$

<sup>10</sup>Hierfür benötigen wir wieder den Befehl `subs`. Da  $g_1$ ,  $g_2$  und  $g_3$  bereits Gleichungen sind, können wir einfach `subs(K, g1)`, `subs(K, g2)` bzw. `subs(K, g3)` schreiben.

<sup>11</sup>Wir haben die Lösungsmenge an die Variable `L` zugewiesen. Auf diese Weise können wir jetzt über `L[1]` auf das erste Element der Menge und mit `L[2]` auf das zweite Element der Menge zugreifen. Mit Hilfe von `float(L)` können dann auf einen Schlag auch Näherungswerte für die exakten Schnittstellen berechnet werden.

- `L:= solve(G1, x)`  
`{2}`

In diesem Fall haben der Kreis und die Gerade also genau einen Punkt gemeinsam, d.h. die Gerade  $g_2$  ist eine Tangente an den Kreis. Die  $y$ -Koordinate des Schnittpunktes ist in diesem Fall klar, da die Gerade  $g_2$  die Koordinatenform  $y = 4$  hat, d.h. alle Punkte auf  $g_2$  (also insbesondere auch der Berührungspunkt auf dem Kreis  $K$ ) haben die  $y$ -Koordinate 4.

Die Gerade  $g_3$  ist eine Passante des Kreises  $K$ , denn Sie hat mit ihm keinen gemeinsamen Punkt:

- `G1:= subs(K, KFormG3)`  
 $(x - 2)^2 + (x - 7)^2 = 9$
- `L:= solve(G1, x)`  
`{}`

### 4.3.2 Geraden in Parameterform

Sind die Geraden in Parameterform gegeben, so müssen wir ein kleines bisschen anders vorgehen. Die Gleichung eines Kreises mit Mittelpunkt  $M(mx|my)$  und Radius  $r$  ist wie üblich in der Form

$$(x - mx)^2 + (y - my)^2 = r^2$$

gegeben. Eine Gerade  $g$  in Parameterform im  $\mathbb{R}^2$  ist ganz allgemein von der Form

$$g : \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + k \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

Um die Lage von  $g$  zu  $K$  zu bestimmen, versuchen wir wieder gemeinsame Punkte von  $K$  und  $g$  zu berechnen. Dazu entnehmen wir der Parameterform der Geraden, dass gilt:

$$x = a_1 + k \cdot v_1 \quad \text{und} \quad y = a_2 + k \cdot v_2.$$

Diese beiden Gleichungen für  $x$  und  $y$  setzen wir dann in die Kreisgleichung ein und erhalten so eine quadratische Gleichung in  $k$ . Diese quadratische Gleichung kann dann wieder keine reelle Lösung, eine reelle Lösung oder zwei reelle Lösungen haben (je nach dem, ob es sich bei der Geraden um eine Passante, eine Tangente oder eine Sekante handelt). Haben wir einen oder mehrere Werte für  $k$  berechnen können, so müssen wir diese nur noch in die Parameterform der Geraden  $g$  einsetzen und finden so die  $x$ - und die  $y$ -Koordinate der Schnittpunkte bzw. des Berührungspunktes.

Als Beispiel betrachten wir wieder den Kreis  $K$  mit Radius  $r = 3$  und Mittelpunkt  $M(2|1)$ . Ferner seien die folgenden drei Geraden in Parameterform gegeben:

$$g1 : \vec{x} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + k \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad g2 : \vec{x} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$g3 : \vec{x} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Wir definieren die Kreisgleichung des Kreises  $K$  sowie die Parameterformen der drei Geraden in MuPAD<sup>12</sup>:

- `assume(k, Type::Real):`  
`K:= (x - 2)^2 + (y - 1)^2 = 3^2:`  
`ParameterformG1:= matrix([2, 4]) + k*matrix([-1,1]):`  
`ParameterformG2:= matrix([2,-2]) + k*matrix([ 1,0]):`  
`ParameterformG3:= matrix([6, 0]) + k*matrix([ 1,1]):`

Beginnen wir also damit, die Lage der Geraden  $g1$  zum Kreis  $K$  zu bestimmen. Wir setzen wie oben beschrieben die Komponenten der Parameterform in die Kreisgleichung ein:

- `G1:= subs(K, x = ParameterformG1[1],`  
`y = ParameterformG1[2])`  
 $(k + 3)^2 + k^2 = 9$

Die erhaltene Gleichung `G1` gilt es jetzt nach  $k$  aufzulösen:

- `L:= solve(G1,k)`  
`{-3,0}`

Die Gleichung hat zwei Lösungen und die Gerade  $g1$  ist damit eine Passante. Wir setzen die berechneten Werte nun für  $k$  in Parameterform der Geraden  $g1$  ein und erhalten somit die Ortsvektoren der beiden Schnittpunkte:

- `OrtsvektorS1:= subs(ParameterformG1, k = L[1]);`  
`OrtsvektorS2:= subs(ParameterformG1, k = L[2])`  
 $\begin{pmatrix} 5 \\ 1 \end{pmatrix}$   
 $\begin{pmatrix} 2 \\ 4 \end{pmatrix}$

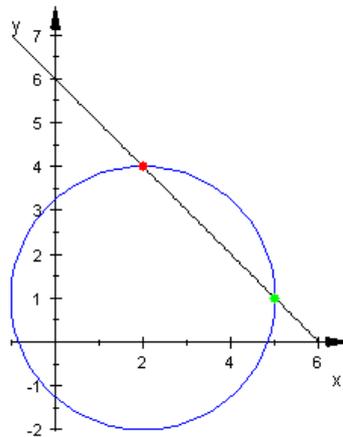
<sup>12</sup>Hier weisen wir mit Hilfe von `assume(k, Type::Real)` den Parameter  $k$  explizit als reelle Zahl aus.

Jetzt können wir den Kreis, die Gerade  $g_1$  sowie die beiden Schnittpunkte<sup>13</sup> in ein gemeinsames Koordinatensystem zeichnen:

```

• Kreis:= plot::Circle2d(3, matrix([2,1])):
  g1:= plot::Curve2d(ParameterformG1, k = -4..3,
                    Color = RGB::Black):
  S1:= plot::Point2d(OrtsvektorS1, Color = RGB::Green,
                    PointSize = 2):
  S2:= plot::Point2d(OrtsvektorS2, Color = RGB::Red,
                    PointSize = 2):
plot(Kreis, g1, S1, S2)

```



Nun zu der Geraden  $g_2$ : Wie schon im Fall der Geraden  $g_1$  setzen wir die Komponenten der Parameterform in die Kreisgleichung des Kreises  $K$  ein und erhalten eine quadratische Gleichung in  $k$ :

```

• G1:= subs(K, x = ParameterformG2[1],
           y = ParameterformG2[2])

```

$$k^2 + 9 = 9$$

Die erhaltene Gleichung  $G_1$  gilt es jetzt nach  $k$  aufzulösen:

```

• L:= solve(G1,k)
  {0}

```

Die Gleichung hat genau eine Lösung und die Gerade  $g_2$  ist damit eine Tangente an den Kreis. Wir setzen den berechneten Wert für  $k$  in Parameterform der Geraden  $g_2$  ein und erhalten somit den Ortsvektor des Berührungspunktes  $B$ :

```

• OrtsvektorB:= subs(ParameterformG2, k = L[1])

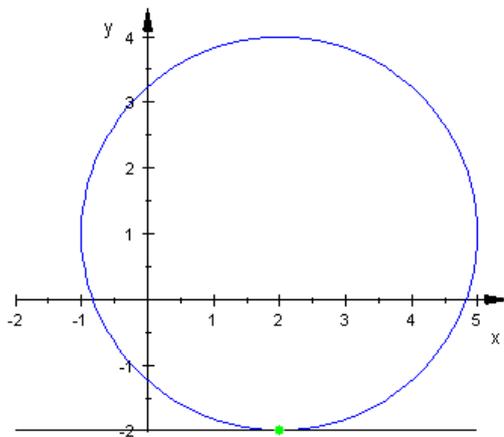
```

<sup>13</sup>Mit Hilfe des Attributs `PointSize` setzen wir hier die Größe der Schnittpunkte herauf, damit sie besser in der Zeichnung zu erkennen sind.

$$\begin{pmatrix} 2 \\ -2 \end{pmatrix}$$

Jetzt können wir den Kreis, die Gerade  $g_2$  sowie den Berührungspunkt  $B$  in ein gemeinsames Koordinatensystem zeichnen<sup>14</sup>:

- `g2:= plot::Curve2d(ParameterformG2, k = -4..3,  
Color = RGB::Black):`  
`B:= plot::Point2d(OrtsvektorB, Color = RGB::Green,  
PointSize = 2):`  
`plot(Kreis, g2, B):`



Schließlich stellt sich die Gerade  $g_3$  als Passante heraus, denn die Gleichung, die sich durch Einsetzen der Komponenten der Parameterform der Geraden  $g_3$  in die Kreisgleichung ergibt, hat keine reelle Lösung:

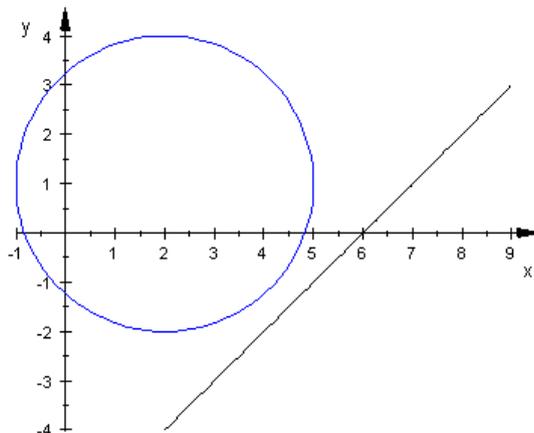
- `G1:= subs(K, x = ParameterformG3[1],  
y = ParameterformG3[2])`  
 $(k - 1)^2 + (k + 4)^2 = 9$

- `L:= solve(G1,k)`  
`{}`

Auch für diesen Fall veranschaulichen wir die Situation mit MuPAD:

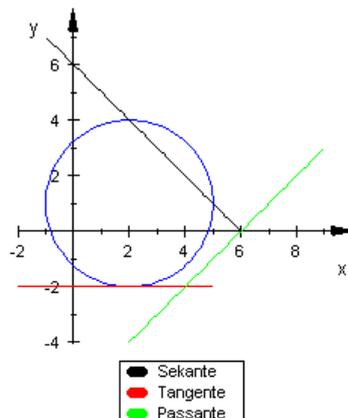
- `g3:= plot::Curve2d(ParameterformG3, k = -4..3,  
Color = RGB::Black):`  
`plot(Kreis, g3):`

<sup>14</sup>Den Kreis hatten wir oben bereits mit Hilfe von `plot::Circle2d` definiert. Wir verwenden diese Definition im folgenden weiter ohne den Kreis jeweils neu zu definieren.



Zum Anschluss dieses Abschnitts zeichnen wir noch einmal alle drei Geraden  $g_1$ ,  $g_2$  und  $g_3$  zusammen mit dem Kreis  $K$  in ein gemeinsames Koordinatensystem. Um die Geraden besser auseinander halten zu können (und zwar nicht nur aufgrund unterschiedlicher Färbung), versehen wir unsere Zeichnung mit Hilfe der Attribute `LegendText` sowie `LegendVisible` mit einer Legende<sup>15</sup>:

- `g1::LegendText:= "Sekante":`
- `g1::Color:= RGB::Black:`
- `g2::LegendText:= "Tangente":`
- `g2::Color:= RGB::Red:`
- `g3::LegendText:= "Passante":`
- `g3::Color:= RGB::Green:`
- `plot(Kreis, g1, g2, g3, LegendVisible)`



<sup>15</sup>Hat man in MuPAD ein grafisches Objekt definiert und in einer Variable `Variablenname` abgespeichert, so verhält sich dieses Objekt wie aus den objektorientierten Konzepten der Informatik gewohnt: Es hat Eigenschaften und mit Hilfe von `Variablenname::Eigenschaft` kann auf diese zugegriffen bzw. die entsprechende Eigenschaft abgeändert werden. Dieses Konzept ist einzigartig in MuPAD und so in keiner der derzeit aktuell Versionen der anderen verbreiteten Computeralgebra-System verfügbar.

### Übungen

55.) Definieren Sie die Gleichung des Kreises mit Mittelpunkt  $M(-2|3)$  und Radius  $r = 4$  und bestimmen Sie die Lage der folgenden drei Geraden zu dem Kreis:

$$g1 : \vec{x} = \begin{pmatrix} -2 \\ 7 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$g2 : \vec{x} = \begin{pmatrix} 4 \\ -2 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

$$g3 : \vec{x} = \begin{pmatrix} -2 \\ 2 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Berechnen Sie – wenn möglich – Schnitt- und Berührungspunkte und zeichnen Sie für jede Gerade den Kreis, die Gerade selbst sowie etwaige Schnitt- und Berührungspunkte in ein gemeinsames Koordinatensystem.

## 4.4 Beschreibung von Kugeln

Zur Herleitung einer Gleichung für Kugeln im Raum gibt es – ebenso wie bei Kreisen im Zweidimensionalen – zwei leicht verschiedene Ansätze, die auf das gleiche Resultat führen. In Abschnitt hatten wir die Punkte auf einem Kreis als diejenigen charakterisiert, die alle vom Mittelpunkt des Kreises den Abstand  $r$  haben, wobei  $r$  der Radius des betreffenden Kreises ist. Diese Sichtweise lässt sich sofort ins Dreidimensionale verallgemeinern und führt unmittelbar auf die Gleichung einer Kugel (sofern der Mittelpunkt und ihr Radius vorgegeben sind).

Im Detail: Gegeben sind uns ein Mittelpunkt  $M(mx|my|mz)$  im Raum sowie ein Radius  $r$ . Alle Punkte  $X(x|y|z)$ , die auf der durch  $M$  und  $r$  definierten Kugel liegen, haben den Abstand  $r$  von dem Punkt  $M$ <sup>16</sup>. Mit anderen Worten: Die Differenzvektoren zwischen dem Ortsvektor des Kugelmittelpunkts und einem jedem Ortsvektor eines Punktes der Oberfläche der Kugel haben alle die gleiche Länge (siehe Abbildung 4.6).

Mathematisch bedeutet dies aber nichts anderes als

$$\left| \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} mx \\ my \\ mz \end{pmatrix} \right| = r.$$

Da sich der Betrag eines Vektors berechnet, indem man seine Komponenten quadriert, über alle Komponenten aufsummiert und anschließend die Wurzel zieht, taucht in der obigen Formel eine Wurzel auf, wenn man sie vollständig ausrechnet. Um nicht ständig

<sup>16</sup>Wäre der Abstand kleiner, so lägen sie im Innern der Kugel, wäre er größer, so lägen sie außerhalb der Kugel; diese Tatsache werden wir im Folgeabschnitt bei der Bestimmung der Lagebeziehung von Punkten und Kugeln wieder aufgreifen.

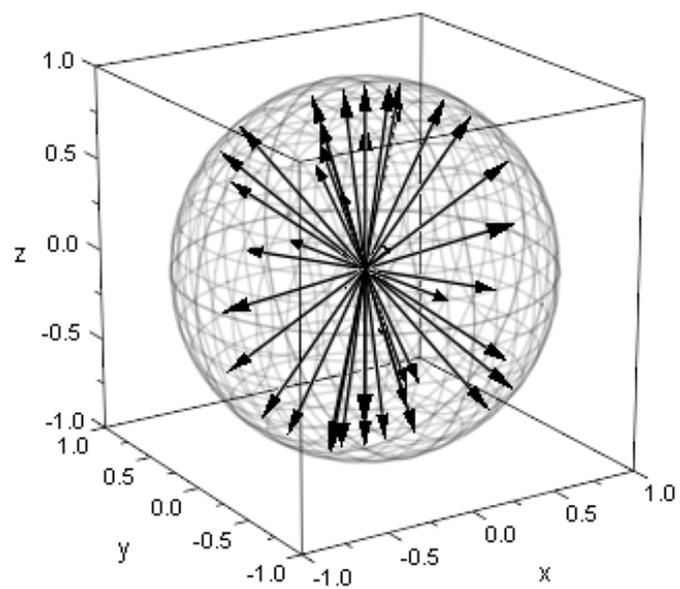
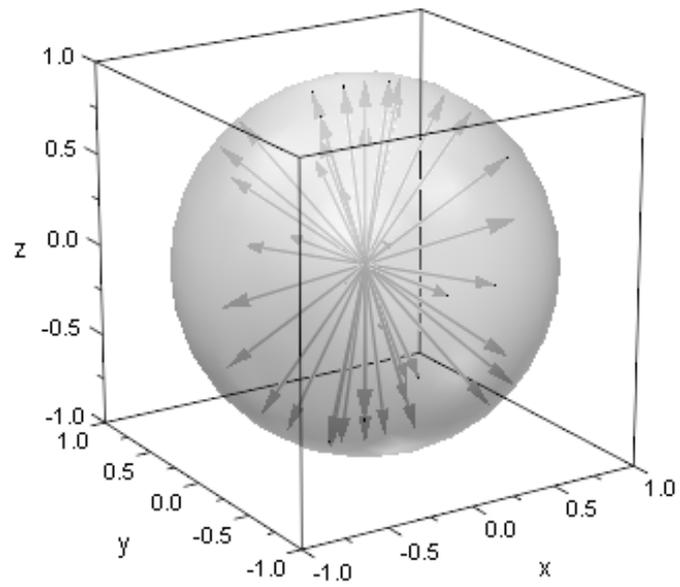


Abbildung 4.6: Beschreibung von Kugeln

mit Wurzeln rechnen zu müssen, macht man es sich einfacher, indem man die obige Gleichung einfach quadriert:

$$\left| \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} mx \\ my \\ mz \end{pmatrix} \right|^2 = r^2.$$

Rechnen wir diese Gleichung aus, so erhalten wir die *allgemeine Kugelgleichung* in der Form

$$(x - mx)^2 + (y - my)^2 + (z - mz)^2 = r^2,$$

also im Prinzip einfach eine um einen Summanden erweiterte Kreisgleichung<sup>17</sup>.

Die Gleichung der Kugel mit Radius  $r = 2$  und Mittelpunkt  $M(-1|2|-3)$  lautet also

- $K := (x + 1)^2 + (y - 2)^2 + (z + 3)^2 = 2^2$   
 $(x + 1)^2 + (y - 2)^2 + (z + 3)^2 = 4$

Auch hier multipliziert MuPAD die Gleichung nicht aus. Wir können dies wieder mit Hilfe des Befehls `expand` erreichen:

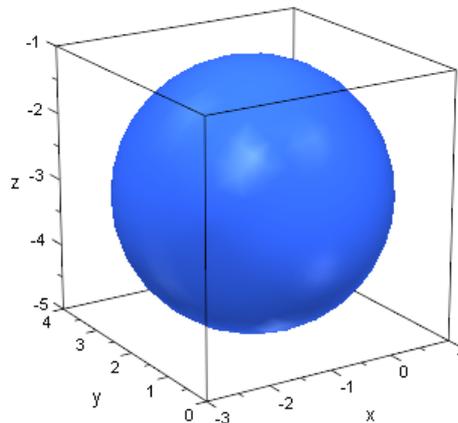
- `expand(K)`  
 $2x - 4y + 6z + x^2 + y^2 + z^2 + 14 = 4$

Wie schon bei Kreisen im  $\mathbb{R}^2$  können wir der Gleichung einer Kugel, sofern sie in vollständig ausmultiplizierter Form wie oben vorliegt, nicht mehr so leicht entnehmen, welchen Radius und welchen Mittelpunkt die zugehörige Kugel hat. Um eine solche Gleichung in die ursprüngliche Form  $(x - mx)^2 + (y - my)^2 + (z - mz)^2 = r^2$  zu überführen, müssen wir wieder auf quadratische Ergänzung zurückgreifen. Das Verfahren haben wir bei Kreisen aber bereits diskutiert (siehe Seite 134 folgende, Abschnitt 4.1) und gehen daher hier nicht weiter darauf ein. Im Fall von Kugeln muss man statt einer zweifachen einfach eine dreifache quadratische Ergänzung durchführen (einmal für die Summanden, die  $x$  enthalten, einmal für die Summanden die  $y$  enthalten und einmal für die Summanden, die  $z$  enthalten).

Um Kugeln mit MuPAD darzustellen, kann man entweder auf den Befehl `plot::Sphere` zurückgreifen oder man bedient sich des Befehls `plot::Implicit3d`. Verwendet man `plot::Sphere`, so muss der Radius der Kugel und der Ortsvektor ihres Mittelpunktes angegeben werden. In unserem Beispiel von oben ergibt sich damit:

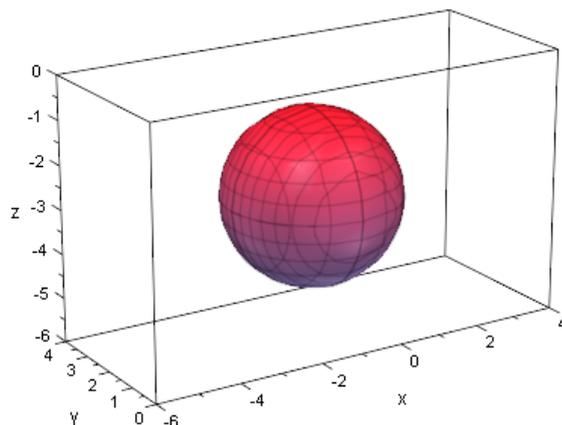
- `plot(plot::Sphere(2, matrix([-1,2,-3])))`

<sup>17</sup> $(x - mx)^2 + (y - my)^2 = r^2$  entspricht einem Zylinder im Raum, denn  $(x - mx)^2 + (y - my)^2 = r^2$  beschreibt einen Kreis mit Mittelpunkt  $M(mx|my)$  in der Ebene und die  $z$ -Koordinate wird durch die Gleichung an keine Bedingung gebunden, d.h. der Kreis setzt sich nach "oben" und nach "unten" von der  $x$ - $y$ -Ebene unendlich zu einem Zylinder fort.



Die Funktion `plot::Implicit3d` dagegen erwartet eine Kreisgleichung<sup>18</sup>:

- `plot(plot::Implicit3d(K, x = -6..4,  
y = 0..4,  
z = -6..0),  
Scaling = Constrained)`



## Übungen

56.) Definieren Sie die Gleichungen der Kugeln mit Radius  $r$  und Mittelpunkt  $M$  in MuPAD:

- $r = 3, M(0|0|0)$
- $r = 4, M(-2|3|-6)$

<sup>18</sup>`plot::Implicit3d` kann ganz allgemein zur Darstellung von implizit definierten Flächen im Raum verwendet werden; die Algorithmen, die zur Visualisierung verwendet werden, sind wesentlich komplizierter, weshalb `plot::Implicit3d` bei weitem nicht so effizient ist, wie die Funktion `plot::Sphere`. Ferner müssen bei der Verwendung von `plot::Implicit3d` Bereiche für die in der Kreisgleichung vorkommenden Variablen  $x$ ,  $y$  und  $z$  angegeben werden und man sollte die Option `Scaling = Constrained` verwenden, damit die Achsen gleich skaliert werden und die Kugel nicht verzerrt erscheint.

- $r = 5, M(1|5| - 3)$

Zeichnen Sie jede Kugel mit MuPAD einmal mit Hilfe von `plot::Sphere` und einmal mit Hilfe von `plot::Implicit3d`.

57.) Welchen Radius und welchen Mittelpunkt hat die Kugel mit Gleichung

$$10 \cdot y - 8 \cdot x - 6 \cdot z + x^2 + y^2 + z^2 + 50 = 25?$$

TIPP: Benutzen Sie quadratische Ergänzung und gehen Sie nach dem in Abschnitt 4.1 beschriebenen Schema vor.

## 4.5 Lagebeziehung von Punkten und Kugeln

### Neue MuPAD-Funktionen

`RGB::Grey` graue Farbe  
`FillColor` Füllfarbe flächiger Objekte

Ähnlich wie wir im  $\mathbb{R}^2$  in Abschnitt 4.4 bei der Lagebeziehung von Kreisen und Punkten die drei Fälle unterschieden hatten, ob ein Punkt auf einem Kreis, in seinem Innern oder außerhalb des Kreises liegt, wollen wir auch im Fall von Kugeln und Punkten im Raum vorgehen. In vorhergehenden Abschnitt 4.4 haben wir die Menge der Punkte  $X(x|y|z)$  auf der Oberfläche einer Kugel  $K$  mit Mittelpunkt  $M(mx|my| mz)$  und Radius  $r$  durch die "quadierte Abstandsgleichung"

$$\left| \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} mx \\ my \\ mz \end{pmatrix} \right|^2 = r^2$$

charakterisiert. Diese Gleichung war äquivalent zu der allgemeinen Kugelgleichung

$$(x - mx)^2 + (y - my)^2 + (z - mz)^2 = r^2.$$

Setzt man also in die linke Seite der Kreisgleichung für  $x$ ,  $y$  und  $z$  die Koordinaten eines Punktes  $P(p_1|p_2|p_3)$  eines Punktes im Raum ein, so ist durch  $(p_1 - mx)^2 + (p_2 - my)^2 + (p_3 - mz)^2$  das Quadrat des Abstands des Punktes  $P$  von dem Mittelpunkt  $M$  der Kugel  $K$  gegeben<sup>19</sup>. Wollen wir die Lage des Punktes  $P$  zu der Kugel  $K$  bestimmen, so müssen wir also nur die folgenden drei Fälle unterscheiden:

- 1) Gilt  $(p_1 - mx)^2 + (p_2 - my)^2 + (p_3 - mz)^2 = r^2$ , so liegt  $P$  auf der Kugeloberfläche der Kugel  $K$ .

<sup>19</sup>Diesen Satz lasse man sich auf der Zunge zergehen.

2) Gilt  $(p_1 - mx)^2 + (p_2 - my)^2 + (p_3 - mz)^2 < r^2$ , so liegt  $P$  im Innern der Kugel  $K$ .

3) Gilt  $(p_1 - mx)^2 + (p_2 - my)^2 + (p_3 - mz)^2 > r^2$ , so liegt  $P$  außerhalb der Kugel  $K$ .

Betrachten wir ein Beispiel: Gegeben seien die Kugel  $K$  mit Mittelpunkt  $M(-1|2|1)$  und Radius  $r = 2$  sowie die Punkte  $A(-2|1|1)$ ,  $B(-2|1|\sqrt{2} + 1)$  und  $C(-2|1|3)$ . Wir wollen die Lagebeziehung der Punkte zu der Kugel bestimmen. Zuerst definieren wir die Kugelgleichung in MuPAD:

- $K := (x + 1)^2 + (y - 2)^2 + (z - 1)^2 = 2^2$   
 $(x + 1)^2 + (y - 2)^2 + (z - 1)^2 = 4$

Jetzt setzen wir nacheinander für  $x$ ,  $y$  und  $z$  die Koordinaten der Punkte  $A$ ,  $B$  und  $C$  in die Kreisgleichung ein: Der Punkt  $A$  hat die Koordinaten  $A(-2|1|1)$ , also:

- $\text{subs}(K, x = -2, y = 1, z = 1)$   
 $2 = 4$

Diese Gleichung ist nicht erfüllbar, d.h.  $A$  liegt nicht auf der Kugeloberfläche. Wegen  $2 < 4$  können wir an der Gleichung ablesen, dass  $A$  im Innern der Kugel  $K$  liegt. Der Punkt  $B$  hat die Koordinaten  $B(-2|1|\sqrt{2} + 1)$ , also<sup>20</sup>:

- $\text{subs}(K, x = -2, y = 1, z = 1 + \text{sqrt}(2))$   
 $4 = 4$

Die Koordinaten von  $B$  erfüllen die Kugelgleichung, d.h.  $B$  liegt auf der Kugeloberfläche. Der Punkt  $C(-2|1|3)$  dagegen liegt außerhalb der Kugel, wie wir ebenfalls durch Einsetzen seiner Koordinaten in die Kugelgleichung feststellen:

- $\text{subs}(K, x = -2, y = 1, z = 3)$   
 $6 = 4$

Die Gleichung ist nicht erfüllt und wegen  $6 > 4$  hat  $C$  einen Abstand vom Mittelpunkt der Kugel, der echt größer ist also der Radius der Kugel.

Zum Anschluss zeichnen wir die Kugel  $K$  sowie die drei Punkte  $A$ ,  $B$  und  $C$  noch in ein gemeinsames Koordinatensystem <sup>21</sup>:

- $Kugel := \text{plot}::\text{Sphere}(2, \text{matrix}([-1, 2, 1]),$   
 $\text{FillColor} = \text{RGB}::\text{Grey}.[0.5]):$   
 $A := \text{plot}::\text{Point3d}(\text{matrix}([-2, 1, 1]),$

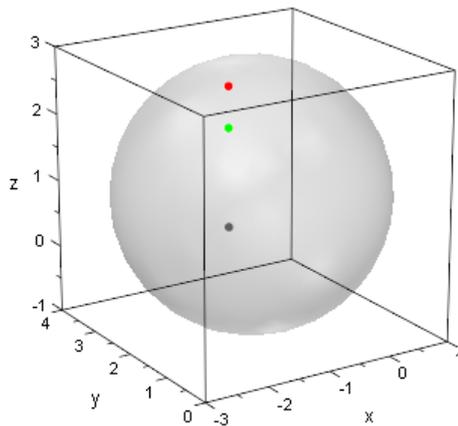
<sup>20</sup> `sqrt` steht für "square root" und dient zur Berechnung der Quadratwurzel. Alternativ kann natürlich  $2^{(1/2)}$  statt `sqrt(2)` verwendet werden.

<sup>21</sup> Mit Hilfe von `FillColor` beeinflussen wir die Füllfarbe der Kugel. Wir wählen hier die Farbe `RGB::Grey`, damit die Kugel in grauer Farbe dargestellt wird. Darüber hinaus fügen wir mit Hilfe des Punktes `.` die Liste `[0.5]` an die Farbe an, so dass die Kugel mit einer Durchsichtigkeit von 50% gezeichnet wird.

```

Color = RGB::Black):
B:= plot::Point3d(matrix([-2,1,sqrt(2)+1]),
Color = RGB::Green):
C:= plot::Point3d(matrix([-2,1,3]),
Color = RGB::Red):
plot(A,B,C,Kugel)

```



Wir haben hier die Grafik mit einem Doppelklick in MuPAD aktiviert und leicht gedreht, damit die Lage der Punkte besser sichtbar wird.

### Übungen

- 58.) Untersuchen Sie die Lage der Punkte  $A(2|-3|\sqrt{23}+2)$ ,  $B(2|-3|6)$  und  $C(2|-3|7)$  zu der Kugel  $K$  mit Radius  $r = 5$  und Mittelpunkt  $M(3|-4|2)$ . Zeichnen Sie anschließend die Kugel und die drei Punkte in ein gemeinsames Koordinatensystem. Verwenden Sie Grau als Füllfarbe für die Kugel und machen Sie sie zu 60% durchsichtig.
- 59.) Gegeben ist die Kugel  $K$  mit Mittelpunkt  $M(1|-1|3)$  und Radius  $r = 4$ . Bestimmen Sie alle Werte für  $k$  derart, dass die Punkte  $P(2|-1|k)$  auf der Kugeloberfläche liegen.

## 4.6 Lagebeziehung von Geraden und Kugeln

Wir untersuchen jetzt die möglichen Lagebeziehungen von Geraden und Kugeln im  $\mathbb{R}^3$ . Es gibt drei verschiedene Möglichkeiten, wie eine Kugel und eine Gerade im Raum zueinander liegen können: Die Gerade kann die Kugeloberfläche in zwei Punkten durchstoßen<sup>22</sup>, sie kann die Kugel in genau einem Punkt auf der Oberfläche berühren<sup>23</sup> oder läuft

<sup>22</sup>Eine solche Gerade nennt man wie schon bei Kreisen eine *Sekante*.

<sup>23</sup>Eine solche Gerade nennt man dann eine *Tangente* an die Kugel und den Punkt, in dem sie die Kugeloberfläche berührt, den *Berührungspunkt*.

außerhalb an der Kugel vorbei, ohne mit der Kugeloberfläche einen gemeinsamen Punkt zu haben<sup>24</sup>. Die drei möglichen Lagebeziehungen sind in Abbildung 4.7 veranschaulicht.

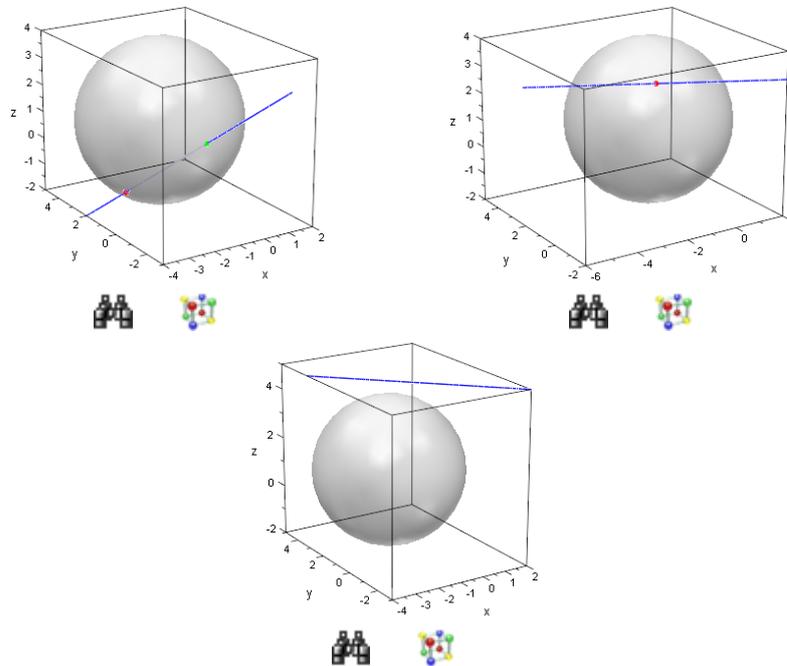


Abbildung 4.7: Lagebeziehungen von Kugeln und Geraden

In Abschnitt 4.3 hatten wir die Lagebeziehung von Kreisen und Geraden im  $\mathbb{R}^2$  diskutiert. Wir werden im Fall von Kugeln und Geraden im  $\mathbb{R}^3$  ganz analog vorgehen, jedoch müssen wir hier nicht unterscheiden, ob die Gerade in Koordinatenform oder in Parameterform vorliegt, da es für Geraden im  $\mathbb{R}^3$  keine Koordinatenform gibt.

Erinnern wir uns daran, wie wir die Lage einer Geraden zu einem Kreis im  $\mathbb{R}^2$  bestimmt hatten: Wir hatten einfach versucht, gemeinsame Punkte von Gerade und Kugel auszurechnen, indem wir die Komponenten der Parameterform der Geraden in die Kreisgleichung eingesetzt hatten. Dies lieferte uns eine quadratische Gleichung in dem Parameter der Parameterform der Geraden. Eine solche quadratische Gleichung kann immer genau zwei, genau eine oder keine Lösung haben, je nach dem ob es sich bei der Geraden um eine Sekante, eine Tangente oder eine Passante handelte. Für den Fall, dass die quadratische Gleichung uns Lösungen lieferte, konnten wir diese in die Parameterform der Geraden einsetzen und erhielten so die Ortsvektoren der Schnittpunkte bzw. den Ortsvektor des Berührungspunktes. Im Fall einer Geraden und einer Kugel im Raum gehen wir ganz analog vor.

Es sei  $K$  die Kugel mit Mittelpunkt  $M(-1|2|1)$  und Radius  $r = 3$ . Ferner seien die

<sup>24</sup>Eine solche Gerade nennt man eine *Passante*.

folgenden drei Geraden gegeben:

$$g1: \vec{x} = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \quad g2: \vec{x} = \begin{pmatrix} -2 \\ 0 \\ 3 \end{pmatrix} + k \cdot \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix},$$

$$g3: \vec{x} = \begin{pmatrix} -1 \\ 0 \\ 5 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}.$$

Wir wollen die Lagebeziehungen dieser Geraden zur Kugel  $K$  untersuchen. Zuerst definieren wir die Gleichung der Kugel sowie die Parameterformen der drei Geraden in MuPAD<sup>25</sup>:

- `assume(k, Type::Real):`  
`K:= (x + 1)^2 + (y - 2)^2 + (z - 1)^2 = 3^2:`  
`ParameterformG1:= matrix([-1,-1,1]) + k*matrix([1,-1,1]):`  
`ParameterformG2:= matrix([-2, 0,3]) + k*matrix([2,-1,0]):`  
`ParameterformG3:= matrix([-1, 0,5]) + k*matrix([1,-1,0]):`

Zuerst untersuchen wir die Lage der Geraden  $g1$  zu der Kugel  $K$ . Wir setzen die Komponenten der Parameterform der Geraden mit Hilfe von `subs` in die Kugelgleichung ein und erhalten auf diese Weise eine quadratische Gleichung in  $k$ , die wir mit `G1` bezeichnen:

- `G1:= subs(K, x = ParameterformG1[1],`  
`y = ParameterformG1[2],`  
`z = ParameterformG1[3])`

$$(k + 3)^2 + 2k^2 = 9$$

Die Gleichung lösen wir mit Hilfe von `solve` nach  $k$  auf nennen die Lösungsmenge `L`:

- `L:= solve(G1, k)`  
`{-2, 0}`

Es gibt zwei Lösungen, d.h. die Gerade  $g1$  schneidet die Kugel  $K$  in zwei Punkten  $S1$  und  $S2$ , deren Ortsvektoren wir erhalten, wenn wir die berechneten Werte für  $k$  in die Parameterform der Geraden  $g1$  einsetzen:

- `OrtsvektorS1:= subs(ParameterformG1, k = L[1]):`  
`OrtsvektorS2:= subs(ParameterformG1, k = L[2])`

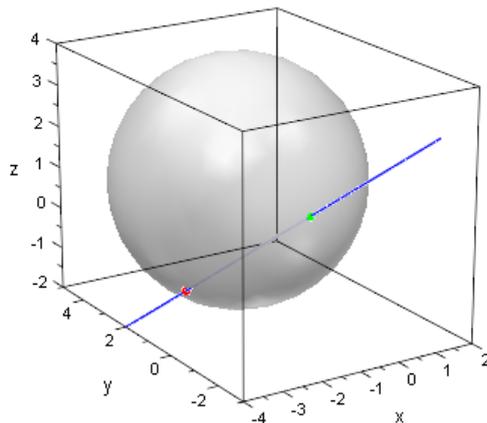
<sup>25</sup>Wir nehmen den Parameter  $k$  aus den Parameterformen der Geraden mit Hilfe von `assume` und `Type::Real` wieder explizit als reell an.

$$\begin{pmatrix} -3 \\ 1 \\ -1 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}$$

Wir können jetzt die Kugel  $K$  sowie die Sekante  $g_1$  und die beiden Schnittpunkte  $S_1$  und  $S_2$  in ein gemeinsames Koordinatensystem zeichnen<sup>26</sup>:

- `Kugel := plot::Sphere(3, matrix([-1,2,1]),  
Color = RGB::Grey.[0.9]):`
- `g1 := plot::Curve3d(ParameterformG1, k = -3..2):`
- `S1 := plot::Point3d(OrtsvektorS1, Color = RGB::Red):`
- `S2 := plot::Point3d(OrtsvektorS2, Color = RGB::Green):`
- `plot(Kugel, g1, S1, S2)`



Nun zu der Geraden  $g_2$ . Wir setzen die Komponenten der Parameterform wieder in die Kugelgleichung der Kugel  $K$  ein

- `G1 := subs(K, x = ParameterformG2[1],  
y = ParameterformG2[2],  
z = ParameterformG2[3])`

$$(2k + 2)^2 + (3k - 1)^2 + 4 = 9$$

und lösen die erhaltene Gleichung nach  $k$  auf:

- `L := solve(G1, k)`

<sup>26</sup>Mit Hilfe von `Color = RGB::Grey.[0.9]` zeichnen wir die Kugel in Grau und lassen sie mit einer Durchsichtigkeit von 90% darstellen, um die Schnitte besser zu veranschaulichen.

{0}

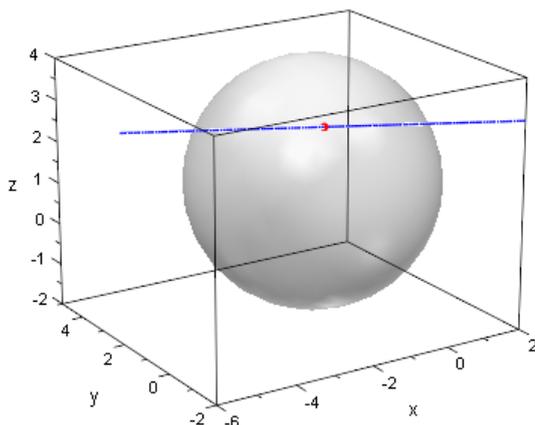
Es gibt nur eine Lösung, d.h.  $g_2$  ist eine Tangente an die Oberfläche der Kugel  $K$ . Den Ortsvektor des Berührungspunktes mit der Kugel erhalten wir wieder durch Einsetzen der berechneten Lösung in die Parameterform der Geraden:

- `OrtsvektorB := subs(ParameterformG2, k = L[1])`

$$\begin{pmatrix} -2 \\ 0 \\ 3 \end{pmatrix}$$

Jetzt zeichnen wir die Kugel<sup>27</sup>, die Tangente sowie den Berührungspunkt in ein gemeinsames Koordinatensystem:

- `g2 := plot::Curve3d(ParameterformG2, k = -2..2):`  
`B := plot::Point3d(OrtsvektorB, Color = RGB::Red):`  
`plot(Kugel, g2, B)`



Die Gerade  $g_3$  stellt sich als Passante heraus, denn sie hat mit der Kugeloberfläche der Kugel keine gemeinsamen Punkte:

- `G1 := subs(K, x = ParameterformG3[1],`  
`y = ParameterformG3[2],`  
`z = ParameterformG3[3])`

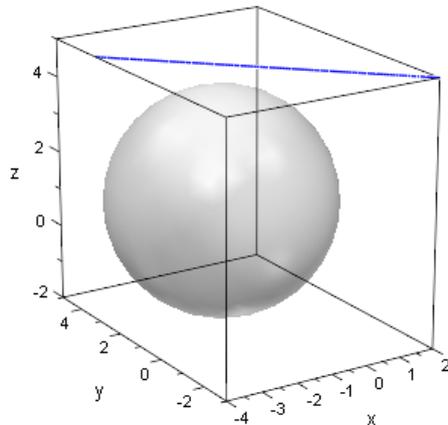
$$(k + 2)^2 + k^2 + 16 = 9$$

- `solve(G1, k)`  
`{}`

Auch für diesen Fall veranschaulichen wir die Situation mit MuPAD:

<sup>27</sup>Die Kugel hatten wir oben bereits als grafisches Objekt erzeugt und in der Variablen `Kugel` gespeichert. Wir definieren es hier nicht erneut, sondern übernehmen die Definition von oben.

- `g3 := plot::Curve3d(ParameterformG3, k = -3..3):`  
`plot(Kugel, g3)`



### Übungen

- 60.) Gegeben sei die Kugel  $K$  mit Mittelpunkt  $M(-2|3|5)$  und Radius  $r = 6$  sowie drei Geraden

$$g1 : \vec{x} = \begin{pmatrix} -2 \\ 2 \\ 4 \end{pmatrix} + k \cdot \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix},$$

$$g2 : \vec{x} = \begin{pmatrix} 3 \\ 6 \\ \sqrt{2} + 5 \end{pmatrix} + k \cdot \begin{pmatrix} -3 \\ 5 \\ 0 \end{pmatrix},$$

$$g3 : \vec{x} = \begin{pmatrix} 3 \\ 6 \\ 7 \end{pmatrix} + k \cdot \begin{pmatrix} -3 \\ 5 \\ 0 \end{pmatrix}.$$

Bestimmen Sie die Lagebeziehung der drei Geraden zu der Kugel  $K$ . Zeichnen Sie jede der drei Geraden zusammen mit der Kugel  $K$  und etwaigen Schnitt- bzw. Berührungspunkten in ein gemeinsames Koordinatensystem. Färben Sie die Kugel dabei in Grau und verwenden Sie eine leichte Transparenz.

- 61.) Berechnen Sie für jede der in der vorhergehenden Aufgabe angegebenen Geraden den Abstand der Gerade zum Mittelpunkt der Kugel. Wie lässt sich daraus auf die Lage der Geraden zur Kugel schließen?

TIPP: In Abschnitt 3.10.1 haben wir diskutiert, wie man den Abstand eines Punktes zu einer Geraden im Raum berechnet.

## 4.7 Lagebeziehung von Ebenen und Kugeln

Eine Ebene kann eine Kugel schneiden, sie kann diese in einem Punkt berühren<sup>28</sup> und sie kann natürlich auch keinen gemeinsamen Punkt mit der Kugeloberfläche gemeinsam haben. Die drei möglichen Lagebeziehungen sind in den Abbildungen 4.8, 4.9 und 4.10 dargestellt (im Fall, dass die Ebene die Kugel schneidet und im Fall, dass die Ebene die Kugel berührt, haben wir die Szene jeweils aus zwei unterschiedlichen Blickwinkeln dargestellt).

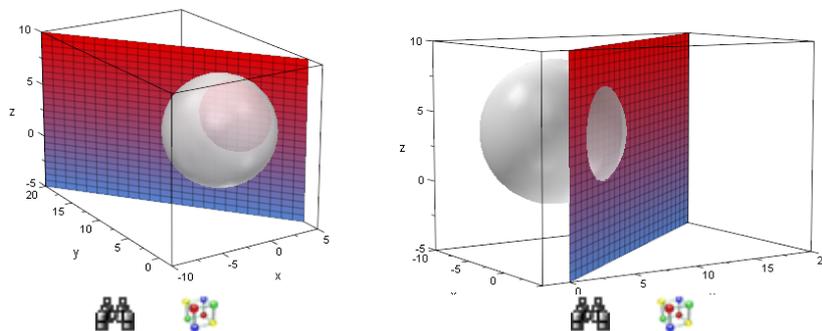


Abbildung 4.8: Eine Ebene schneidet eine Kugel

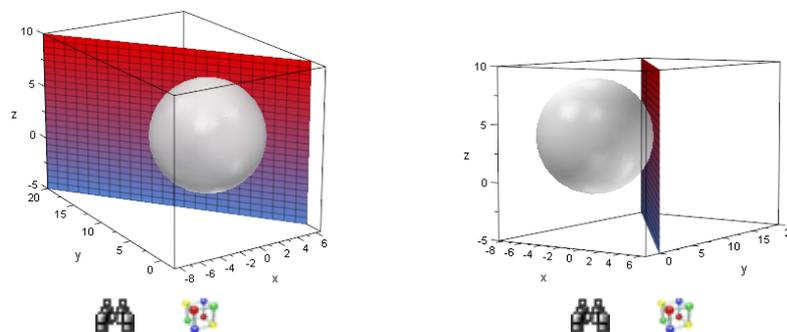


Abbildung 4.9: Eine Ebene berührt eine Kugel

Wir müssten im folgenden im Prinzip die beiden Fälle unterscheiden, ob die gegebene Ebene in Parameterform oder in Koordinatenform vorliegt, sofern wir versuchen, die gegenseitige Lage der Ebene und der Kugel über die Berechnung gemeinsamer Punkte zu ermitteln. Abgesehen von der etwas lästigen Fallunterscheidung “Ebene in Parameterform”–“Ebene in Koordinatenform” ist der Versuch der Berechnung gemeinsamer Punkte von Kreis und Ebene kein so glücklicher Ansatz. Grund dafür ist die Tatsa-

<sup>28</sup>Eine solche Ebene nennt man *Tangentialebene* und den Punkt, in dem sie die Kugel berührt, den Berührungspunkt. Wir kommen in Abschnitt 4.7.1 noch einmal gesondert auf Tangentialebenen sowie ihre Bestimmung für den Fall, dass der Berührungspunkt auf der Kugel vorgegeben ist, zu sprechen.

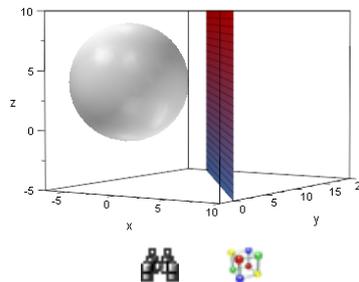


Abbildung 4.10: Eine Ebene passiert eine Kugel

che, dass die Berechnung gemeinsamer Punkte von Ebene und Kugel i.a. auf eine quadratische Gleichung in zwei Variablen führt<sup>29</sup>. Wir wählen daher einen anderen Ansatz zur Bestimmung der Lage einer Ebene und einer Kugel im, der im Prinzip der gleiche ist, der in der letzten Aufgabe des vorhergehenden Abschnitts für Geraden durchzuführen war: Wir berechnen schlicht den Abstand der gegebenen Ebene von dem Mittelpunkt gegebenen Kugel.

Ist dieser Abstand echt kleiner als der Radius der Kugel, so schneidet die Ebene die Kugel. Das Schnittgebilde ist dann ein Kreis (siehe auch Abbildung 4.8). Im Prinzip kennen wir die Gleichung eines Kreises, die zu Anfang dieses Kapitels behandelt wurde, jedoch kann der Schnittkreis völlig "schräg" im Raum liegen. Was wir immer bestimmen können, ist der Mittelpunkt dieses Schnittkreises sowie sein Radius.

Ist der Abstand der Ebene zu dem Mittelpunkt der Kugel gleich dem Radius der Kugel, so berührt die Kugel die Ebene in einem Punkt. Wir werden sehen, dass wir diesen Berührungspunkt leicht bestimmen können.

Ist der Abstand der Ebene zu dem Mittelpunkt der Kugel echt größer als der Radius, so haben Kugel und Ebene keine gemeinsamen Punkt.

Wir müssen zur Untersuchung der Lagebeziehung einer Ebene und einer Kugel also im Prinzip gar nichts Neues lernen, denn wie man den Abstand eines Punktes (in diesem Fall der Mittelpunkt der Kugel) von einer Ebene berechnet, haben wir in Abschnitt 3.10.4 bereits gesehen. Dort wurde auch die Fallunterscheidung getroffen, wie man vorzugehen hat, wenn die Ebenen in Parameterform oder aber in Koordinatenform vorliegt.

Betrachten wir einige Beispiele: Gegeben sei die Kugel  $K$  mit Mittelpunkt  $M(-2|3|4)$

<sup>29</sup>Als wir die gemeinsamen Punkte einer Geraden und einer Kugel berechnet hatten, führte dies auf eine quadratische Gleichung in dem Parameter, der in der Parameterform der Geraden vorkommt. Bei Ebenen tauchen zwei Parameter in ihrer Parameterdarstellung auf. Auch der Ansatz über Verwendung der Koordinatenform einer Ebene führt in der Regel auf eine quadratische Gleichung in zwei der Variablen  $x$ ,  $y$  und  $z$ .

und Radius  $r = 5$  sowie die drei Ebenen

$$F1: \vec{x} = \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix} + r \cdot \begin{pmatrix} -3 \\ 4 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$F2: \vec{x} = \begin{pmatrix} 13/2 \\ 0 \\ 0 \end{pmatrix} + r \cdot \begin{pmatrix} -3 \\ 4 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$F3: \vec{x} = \begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix} + r \cdot \begin{pmatrix} -3 \\ 4 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Ihre Lage zu der Kugel  $K$  wollen wir jetzt untersuchen. Wir beginnen mit der Ebene  $F1$

- `ParameterformF1:= matrix([5,0,0] + r*matrix([-3,4,0])  
+ s*matrix([ 0,0,1]))):`

und berechnen den Abstand von  $F1$  zum Punkt  $M$ . Wie in Abschnitt 3.10.4 bestimmen wir mit Hilfe von `linalg::crossProduct` einen Normalenvektor der Ebene

- `n:= linalg::crossProduct(matrix([-3,4,0]),  
matrix([ 0,0,1]))):`

$$\begin{pmatrix} 4 \\ 3 \\ 0 \end{pmatrix}$$

mit dessen Hilfe wir jetzt die Gleichung der Lotgeraden  $l$  durch  $M$  mit Richtungsvektor  $\vec{n}$  bestimmen können:

- `OrtsvektorM:= matrix([-2,3,4]):  
l:= OrtsvektorM + m*n`

$$\begin{pmatrix} 4m - 2 \\ 3m + 3 \\ 4 \end{pmatrix}$$

Nun gilt es den Schnittpunkt  $L$  von  $l$  mit  $F1$  zu bestimmen:

- `G1:= {ParameterformF1[1] = l[1],  
ParameterformF1[2] = l[2],  
ParameterformF1[3] = l[3]}:  
solve(G1, {m, r, s})`

$$\left\{ \left[ m = \frac{19}{25}, r = \frac{33}{25}, s = 4 \right] \right\}$$

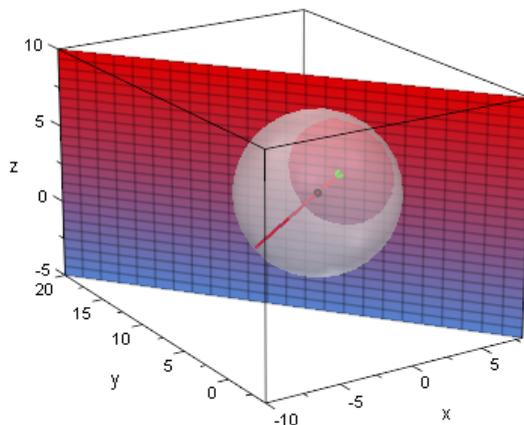
Damit ist der Ortsvektor des Lotfußpunktes  $L$  gegeben durch:

- `OrtsvektorL:= subs(1, m = 19/25)`

$$\begin{pmatrix} \frac{26}{25} \\ \frac{132}{25} \\ 4 \end{pmatrix}$$

Zur Veranschaulichung dessen, was wir getan haben, zeichnen wir die Ebene  $F1$ , die Kugel  $K$ , den Mittelpunkt  $M$ , den Lotfußpunkt  $L$  sowie die Lotgerade  $l$  in ein gemeinsames Koordinatensystem:

- `F1:= plot::Surface(ParameterformF1, r = -1..5,  
s = -5..10):`
- `Kugel:= plot::Sphere(5, OrtsvektorM,  
Color = RGB::Grey.[0.5]):`
- `M:= plot::Point3d(OrtsvektorM, Color = RGB::Black):`
- `L:= plot::Point3d(OrtsvektorL, Color = RGB::Green):`
- `Lotgerade:= plot::Curve3d(1, m = -2..2,  
Color = RGB::Red):`
- `plot(F1, Kugel, M, L, Lotgerade)`



In MuPAD selbst lässt sich diese Grafik nun wie üblich mit einem Doppelklick aktivieren, man kann die Szene drehen und an interessante Stellen heran zoomen. Zwei Sichtweisen auf die gegebene Situation sind in Abbildung 4.11 dargestellt.

Die Grafiken zeigen uns, dass die Ebene die Kugel schneidet. Wollen wir dies mathematisch nachweisen, so müssen wir zeigen, dass der Abstand der Ebene vom Mittelpunkt der Kugel kleiner als der Radius  $r = 5$  der Kugel ist. Der Abstand der Ebene  $F1$  von der Kugel  $K$  ist gegeben durch die Länge des Vektors  $\overrightarrow{ML}$ . Mit Hilfe von `norm` berechnen wir die Länge des Vektors:

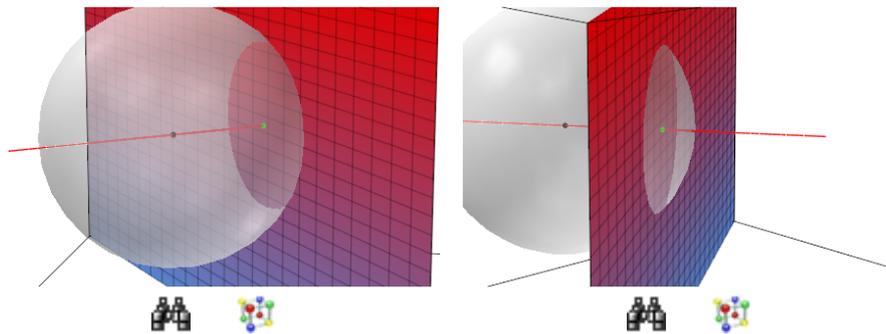


Abbildung 4.11: Der Abstand der Ebene zum Mittelpunkt der Kugel ist kleiner als der Radius der Kugel

- `d:= norm(OrtsvektorL - OrtsvektorM, 2);`  
`float(d)`  
 $\frac{19}{5}$   
 3.8

Der Abstand der Ebene  $F1$  von dem Mittelpunkt der Kugel beträgt also 3.8 Längeneinheiten und ist damit in der Tat echt kleiner als der Radius der Kugel.

Nun zu der Ebene  $F2$ :

- `ParameterformF2:= matrix([13/2,0,0] + r*matrix([-3,4,0])`  
`+ s*matrix([ 0,0,1]));`

Da die Spannvektoren der Ebene mit denen der Ebene  $F1$  übereinstimmen, ist der oben berechnete Normalenvektor  $\vec{n}$  der Ebene  $F1$  auch ein Normalenvektor der Ebene  $F2$ . Wir brauchen ihn also nicht erneut berechnen. Da die Lotgerade  $l$  von oben als Stützvektor den Ortsvektor des Mittelpunktes der Kugel und als Richtungsvektor den Normalenvektor  $\vec{n}$  hat, ändert sich auch die Lotgerade nicht, d.h. wir können die Definition von  $l$  von oben übernehmen und direkt den Schnittpunkt mit der Ebene  $F2$  berechnen:

- `G1:= {ParameterformF2[1] = l[1],`  
`ParameterformF2[2] = l[2],`  
`ParameterformF2[3] = l[3]}:`  
`solve(G1, {m, r, s})`  
 $\left\{ \left[ m = 1, r = \frac{3}{2}, s = 4 \right] \right\}$

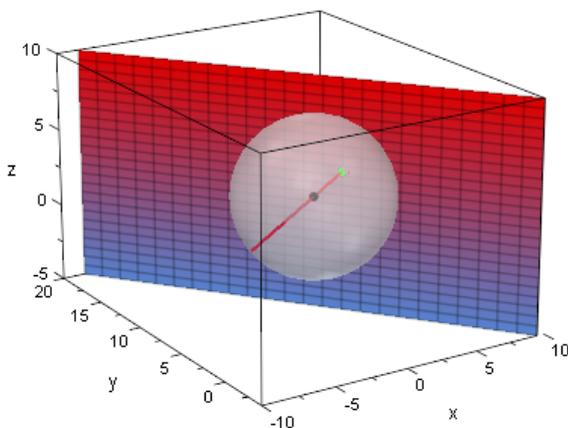
Damit ist der Ortsvektor des Lotfußpunktes  $L$  gegeben durch:

- `OrtsvektorL:= subs(l, m = 1)`

$$\begin{pmatrix} 2 \\ 6 \\ 4 \end{pmatrix}$$

Zur Veranschaulichung dessen, was wir getan haben, zeichnen wir wieder die Ebene  $F2$ , die Kugel  $K$ , den Mittelpunkt  $M$ , den Lotfußpunkt  $L$  sowie die Lotgerade  $l$  in ein gemeinsames Koordinatensystem<sup>30</sup>:

- `F2:= plot::Surface(ParameterformF2, r = -1..5, s = -5..10):`  
`L:= plot::Point3d(OrtsvektorL, Color = RGB::Green):`  
`plot(F2, Kugel, M, L, Lotgerade)`



Die Grafik deutet und an, dass die Ebene  $F2$  die Kugel  $K$  in genau einem Punkt berührt. Zwei andere Sichtweisen auf die gegebene Situation sind in Abbildung 4.12 dargestellt.

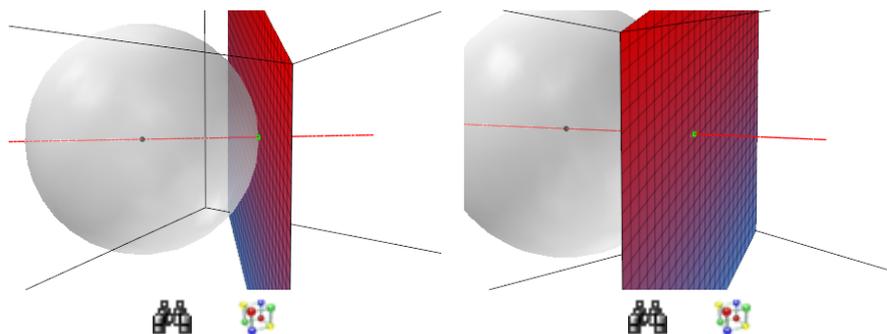


Abbildung 4.12: Der Abstand der Ebene zum Mittelpunkt der Kugel ist gleich dem Radius der Kugel

Der Abstand der Ebene  $F2$  von der Kugel  $K$  ist wieder gegeben durch die Länge des

<sup>30</sup>Die Kugel, ihren Mittelpunkt sowie die Lotgerade hatten wir oben bei der Betrachtung von  $F1$  bereits definiert. Wir müssen sie also hier nicht erneut in MuPAD definieren.

Vektors  $\overrightarrow{ML}$ :

- `d:= norm(OrtsvektorL - OrtsvektorM, 2);`  
5

Der Abstand der Ebene  $F2$  von dem Mittelpunkt der Kugel beträgt also 5 Längeneinheiten und stimmt damit mit dem Radius der Kugel überein: Die Ebene  $F2$  berührt die Kugel im Punkt  $L(2|4|6)$ .

Schließlich untersuchen wir noch die Lage der Ebene  $F3$  zu der Kugel  $K$ :

- `ParameterformF3:= matrix([10,0,0] + r*matrix([-3,4,0])  
+ s*matrix([ 0,0,1]));`

Da die Spannvektoren der Ebene mit denen der Ebenen  $F1$  und  $F2$  übereinstimmen, können wir die Lotgerade  $l$  wieder von oben übernehmen und müssen wieder nur den Lotfußpunkt  $L$  auf  $F3$  als Schnittpunkt der Lotgeraden mit der Ebene  $F3$  neu berechnen:

- `G1:= {ParameterformF3[1] = l[1],  
ParameterformF3[2] = l[2],  
ParameterformF3[3] = l[3]}:`  
`solve(G1, {m, r, s})`  
 $\left\{ \left[ m = \frac{39}{25}, r = \frac{48}{25}, s = 4 \right] \right\}$

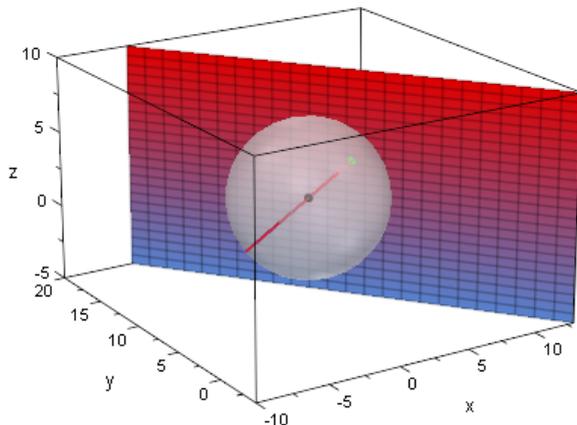
Damit ist der Ortsvektor des Lotfußpunktes  $L$  gegeben durch:

- `OrtsvektorL:= subs(l, m = 39/25)`  
 $\begin{pmatrix} \frac{106}{25} \\ \frac{192}{25} \\ 4 \end{pmatrix}$

Zur Veranschaulichung dessen, was wir getan haben, zeichnen wir auch hier wieder die Ebene  $F3$ , die Kugel  $K$ , den Mittelpunkt  $M$ , den Lotfußpunkt  $L$  sowie die Lotgerade  $l$  in ein gemeinsames Koordinatensystem<sup>31</sup>:

- `F3:= plot::Surface(ParameterformF3, r = -1..5,  
s = -5..10):`  
`L:= plot::Point3d(OrtsvektorL, Color = RGB::Green):`  
`plot(F3, Kugel, M, L, Lotgerade)`

<sup>31</sup>Die Kugel, ihren Mittelpunkt sowie die Lotgerade hatten wir oben bei der Betrachtung von  $F1$  bereits definiert. Wir müssen sie also hier nicht erneut in MuPAD definieren.



Die Perspektive, aus der wir hier auf das Geschehen blicken, scheint etwas ungünstig zu sein. Wir können zwar erkennen, dass die Kugel die Ebene sicherlich nicht in einem Kreis schneidet, aber wir können schlecht erkennen, ob die Ebene  $F_3$  die Kugel etwa in einem Punkt berührt oder gar keinen Punkt mit der Kugel gemeinsam hat. Zwei andere Sichtweisen auf die gegebene Situation sind in Abbildung 4.13 dargestellt und geben mehr Aufschluss über die Lage der Ebene zur Kugel.

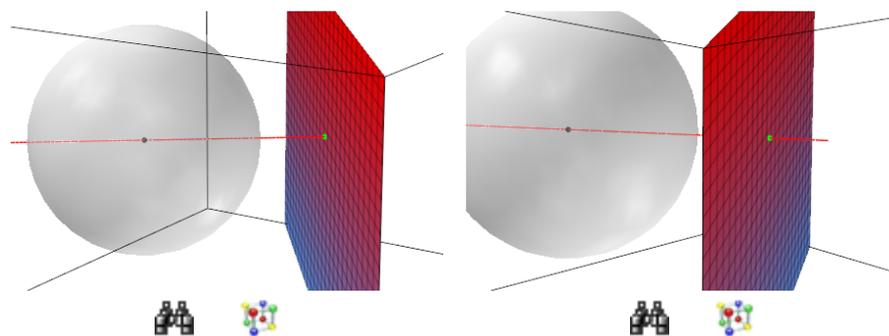


Abbildung 4.13: Der Abstand der Ebene zum Mittelpunkt der Kugel ist größer als der Radius der Kugel

Anhand der Abbildung 4.13 vermuten wir, dass  $K$  und  $f_3$  keine gemeinsamen Punkte haben. Der Abstand der Ebene  $F_3$  von der Kugel  $K$  beträgt:

- `d:= norm(OrtsvektorL - OrtsvektorM, 2);`  
`float(d)`  
 $\frac{39}{5}$   
 7.8

Damit ist der Abstand von  $F_3$  echt größer als der Radius der Kugel  $K$ , d.h.  $F_3$  hat mit  $K$  in der Tat keine gemeinsamen Punkte.

**Übungen**

62.) Berechnen Sie den Abstand der Ebenen

$$F1 := \vec{x} = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

$$F2 := \vec{x} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

$$F3 := \vec{x} = \begin{pmatrix} 0 \\ 0 \\ 7 \end{pmatrix} + r \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

von der Kugel  $K$  mit Mittelpunkt  $M(1|2|1)$  und Radius  $r = 2$ . Zeichnen Sie für jede Ebene den Mittelpunkt der Kugel, die Kugel selbst, die Ebene, die Lotgerade und den Lotfußpunkt in ein gemeinsames Koordinatensystem. Färben Sie die Kugel in Grau und wählen Sie eine Durchsichtigkeit von 50% für die Kugeloberfläche.

63.) Berechnen Sie den Abstand der Ebenen

$$F1 : 2 \cdot z - 6 = 0, \quad F2 : z - 7 = 0, \quad F3 : 2 \cdot z - 5 = 0$$

von der Kugel  $K$  mit Mittelpunkt  $M(1|2|1)$  und Radius  $r = 2$ . Zeichnen Sie für jede Ebene den Mittelpunkt der Kugel, die Kugel selbst, die Ebene, die Lotgerade und den Lotfußpunkt in ein gemeinsames Koordinatensystem. Färben Sie die Kugel in Grau und wählen Sie eine Durchsichtigkeit von 50% für die Kugeloberfläche.

**4.7.1 Tangentialebenen**

Im vorhergehenden Abschnitt, wo wir uns mit den möglichen Lagebeziehungen von Ebenen und Kugeln beschäftigt haben, haben wir gesehen, dass eine Ebene eine Kugel in genau einem Punkt berühren kann. Eine solche Ebene nennt man dann eine *Tangentialebene* und den Punkt, den die Kugel und die Ebene gemeinsam haben, den *Berührungspunkt*. Wir wollen uns jetzt die folgende Frage stellen: Gegeben eine Kugel  $K$  mit Mittelpunkt  $M$  und Radius  $r$  und ein Punkt  $B$  auf der Kugeloberfläche, wie findet man die Tangentialebene an die Kugel in diesem Punkt?

Die zentrale Beobachtung ist, dass der Vektor vom Kugelmittelpunkt  $M$  hin zum Punkt  $B$  und jeder beliebige Vektor zwischen einem Punkt  $X$  in der Ebene und dem Berührungspunkt  $B$  orthogonal zueinander sind, d.h. es gilt für die Ortsvektoren der Punkte  $M$ ,  $B$  und  $X$  die Gleichung

$$(\vec{b} - \vec{m}) * (\vec{x} - \vec{m}) = 0.$$

Die Situation ist in Abbildung 4.14 dargestellt, wo zwei Vektoren in der Ebene eingezeichnet sind, die beide Senkrecht auf dem Vektor vom Kugelmittelpunkt hin zum Berührungspunkt  $B$  stehen.

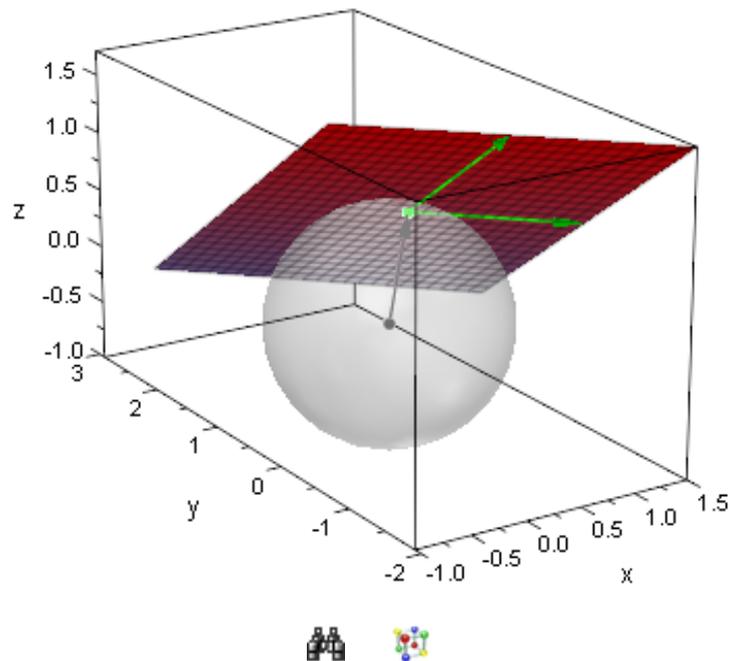


Abbildung 4.14: Beschreibung einer Tangentialebene bei gegebenem Berührungspunkt auf der Kugel

Sind uns also der Berührungspunkt  $B$  und der Mittelpunkt der Kugel  $M$  gegeben, so ist die Gleichung  $(\vec{b} - \vec{m}) * (\vec{x} - \vec{m}) = 0$  bereits eine Koordinatenform der Tangentialebene an die Kugel  $K$ . Die Gleichung einer Tangentialebene zu bestimmen ist also sehr einfach. Wir schauen uns ein Beispiel an: Gegeben sei die Kugel  $K$  mit Mittelpunkt  $M(-2|1|2)$  und Radius  $r = 3$ . Die Punkte  $B1(-1|2|2 + \sqrt{7})$  und  $B2(-1|2|2 - \sqrt{7})$  liegen auf der Kugeloberfläche. Dies können wir leicht nachweisen, indem wir ihre Koordinaten für  $x$ ,  $y$  und  $z$  in die Kugelgleichung einsetzen<sup>32</sup>:

- $K := (x + 2)^2 + (y - 1)^2 + (z - 2)^2 = 3^2$ :  
`subs(K, x = -1, y = 2, z = 2 + sqrt(7));`  
`subs(K, x = -1, y = 2, z = 2 - sqrt(7))`  
 $9 = 9$   
 $9 = 9$

Wir wollen jetzt die Tangentialebene  $T1$  bzw.  $T2$  im Punkt  $B1$  bzw.  $B2$  an die Kugel berechnen. Zuerst definieren wir die Ortsvektoren der Punkte  $B1$  und  $B2$  sowie des Mittelpunkts  $M$  der Kugel  $K$  in MuPAD:

- `OrtsvektorB1 := matrix([-1, 2, 2+sqrt(7)]):`

<sup>32</sup>Mit Hilfe von `subs` setzen wir die Koordinaten der Punkte in die Kugelgleichung ein. Die Wurzel aus 7 definieren wir wie gewohnt mit Hilfe des Befehls `sqrt`. Alternativ kann natürlich auch die Potenzschreibweise  $7^{(1/2)}$  verwendet werden.

```
OrtsvektorB2:= matrix([-1,2,2-sqrt(7)]):
OrtsvektorM := matrix([-2,1,2]):
```

Jetzt müssen wir nur noch die entsprechenden Ortsvektoren in die Formel zur Bestimmung der Koordinatenform einer Tangentialebene einsetzen<sup>33</sup>. Zur Berechnung des Skalarprodukts greifen wir wie üblich auf `linalg::scalarProduct` zurück:

- `assume({x,y,z}, Type::Real):`  
`OrtsvektorX:= matrix([x,y,z]):`  
`KFormF1:= linalg::scalarProduct(`  
`OrtsvektorB1 - OrtsvektorM,`  
`OrtsvektorX - OrtsvektorM) = 0`

$$x + y + \sqrt{7}z - 2\sqrt{7} + 1 = 0$$

Ganz analog ergibt sich auch die Gleichung der Tangentialebene  $F2$  im Berührungspunkt  $B2$  und die Kugel  $K$ :

- `KFormF2:= linalg::scalarProduct(`  
`OrtsvektorB2 - OrtsvektorM,`  
`OrtsvektorX - OrtsvektorM) = 0`

$$x + y - \sqrt{7}z + 2\sqrt{7} + 1 = 0$$

## Übungen

- 64.) Gegeben sei die Kugel  $K$  mit Mittelpunkt  $M(1|4|-2)$  und Radius  $r = 5$ . Zeigen Sie, dass die Punkte  $B1(-3|4|1)$  und  $B2(-3|4|-5)$  auf der Kugeloberfläche liegen. Bestimmen Sie anschließend Koordinatenformen der Tangentialebenen  $T1$  bzw.  $T2$  an die Kugeloberfläche im Berührungspunkt  $B1$  bzw.  $B2$ .

<sup>33</sup>Mit Hilfe von `assume` und `MexType::Real` zeichnen wir die Variablen  $x$ ,  $y$  und  $z$  explizit als reell.

## 5 Aufgaben aus Abiturprüfungen

Die in diesem Kapitel aufgeführten Abituraufgaben zur Analytischen Geometrie sind dem Buch *Analytische Geometrie* von *Rudolf Eckart, Franz Jehle* und *Wilhelm Vogel*, erschienen im *Bayerischen Schulbuchverlag – München*, entnommen. Diese Aufgaben sind typische Beispiele für Abiturprüfungen, wie sie Ende der Siebziger und in den Achtziger Jahren gestellt wurden. Zu dieser Zeit hatten die Computeralgebra-Systeme noch keinen Einzug in den Mathematikunterricht der Sekundarstufe II gehalten, so dass die Aufgaben nicht explizit für eine Bearbeitung mit einem Computeralgebra-System ausgelegt sind. Hier soll es daher auch in erster Linie darum gehen, zu demonstrieren, wie sich die auch heute für das Abitur im Fach Mathematik noch durchaus aktuellen Fragestellungen mit Hilfe von MuPAD bearbeiten lassen.

Viele, in den Aufgaben auftretende Sachverhalte, müssen laut Aufgabenstellung nicht explizit visualisiert werden<sup>1</sup>. Wir wollen daher in so fern eine Erweiterung der “Standardaufgaben” vornehmen, als dass wir die jeweilige Situation, in der wir uns zu einem zentralen Zeitpunkt der Aufgabenbearbeitung befinden, stets mit Hilfe von MuPAD unaufgefordert visualisieren.

Der erste Block von Aufgaben wurde im Rahmen der Abiturprüfung von Grundkursen gestellt. Der zweite anschließend aufgeführte Block von Aufgaben entstammt den Abiturprüfungen von Leistungskursen.

### 5.1 Grundkursniveau

Die folgenden Aufgaben wurden im Jahr 1983 im Rahmen der Abiturprüfung für Grundkurse in Bayern gestellt<sup>2</sup>:

**Aufgaben zur Analytischen Geometrie:** In einem kartesischen Koordinatensystem mit dem Ursprung  $O$  ist die Ebene  $E$  durch die Punkte  $A(3|-3|3)$ ,  $B(5|1|-1)$  und  $C(1|5|1)$  festgelegt.

1. a) Stellen Sie für die Ebene  $E$  eine Gleichung in Parameterform und eine Gleichung in Normalenform auf.  
(Mögliches Teilergebnis:  $2 \cdot x_1 + x_2 + 2 \cdot x_3 - 9 = 0$ )

---

<sup>1</sup>Dies ist auch kein Wunder, denn z.B. die Darstellung einiger Ebenen einer Ebenenschar ist ohne ein entsprechendes Werkzeug wie etwa MuPAD schwerlich im Rahmen einer Prüfung einforderbar.

<sup>2</sup>Die Aufgabenstellung ist wörtlich aus [2] übernommen: Siehe [2], Anhang A. 5. Seite 198

- b) Zeigen Sie, dass die Vektoren  $\overrightarrow{AB}$  und  $\overrightarrow{BC}$  aufeinander senkrecht stehen und den gleichen Betrag haben.
- c) Berechnen Sie die Koordinaten eines in der Ebene  $E$  gelegenen Punktes  $D$ , für den das Viereck  $ABCD$  ein Quadrat ist. [Ergebnis:  $D(-1|1|5)$ ]
2. Die Punkte  $A, B, C, D$  und der Ursprung  $O$  legen eine quadratische Pyramide mit Spitze in  $O$  fest. Dabei ist  $D$  der in Teilaufgabe 1 c) bestimmte Punkt.
- a) Zeigen Sie, dass das von  $O$  aus auf der Ebene  $E$  gefällte Lot durch den Mittelpunkt des Quadrats  $ABCD$  geht.
- b) Ermitteln Sie den spitzen Winkel (auf eine Dezimale gerundet), den die Seitenkante  $OA$  mit der von  $A$  ausgehenden Diagonalen der Grundfläche  $ABCD$  einschließt.
- c) Berechnen Sie die Höhe und das Volumen der Pyramide.
- d) Eine Ebene  $F$  ist parallel zur Grundfläche  $ABCD$  und schneidet die Pyramide so, dass die Höhe halbiert wird. Bestimmen Sie eine Gleichung der Ebene  $F$ .

**Lösung:** Zuerst definieren wir – wie üblich – die Ortsvektoren der vier Punkte  $A, B, C$  und  $O$  in MuPAD:

- OrtsvektorA:= matrix([3,-3, 3]):
- OrtsvektorB:= matrix([5, 1,-1]):
- OrtsvektorC:= matrix([1, 5, 1]):
- OrtsvektorO:= matrix([0, 0, 0]):

**1.a)** Zuerst müssen wir eine Parameterform der Ebene  $E$  angeben<sup>3</sup>. Als Stützvektor für die Ebene  $E$  wählen wir den Ortsvektor des Punktes  $A$  und als Spannvektoren die Differenzvektoren der Ortsvektoren von  $B$  und  $A$  bzw. von  $C$  und  $A$ :

- Stuetzvektor:= OrtsvektorA:
- Spannvektor1:= OrtsvektorB - OrtsvektorA:
- Spannvektor2:= OrtsvektorC - OrtsvektorA:
- ParameterformE:= Stuetzvektor + k \* Spannvektor1  
+ l \* Spannvektor2

$$\begin{pmatrix} 2k - 2l + 3 \\ 4k + 8l - 3 \\ 3 - 2l - 4k \end{pmatrix}$$

<sup>3</sup>Die Ebene  $E$  sollte nicht unter dem Variablennamen  $E$  in MuPAD definiert werden, denn  $E$  ist in MuPAD eine Kurzschreibweise für die *Eulersche Zahl*.

Zur Bestimmung einer Koordinatenform der Ebene  $E$  berechnen wir zuerst einen Normalenvektor der Ebene. Diesen erhalten wir wieder zum Beispiel als Kreuzprodukt der beiden Spannvektoren `Spannvektor1` und `Spannvektor2` der Ebene  $E$ . Das Kreuzprodukt können wir wie gewohnt mit Hilfe der Funktion `linalg::crossProduct` bestimmen:

- `Normalenvektor:= linalg::crossProduct(  
Spannvektor1, Spannvektor2)`

$$\begin{pmatrix} 24 \\ 12 \\ 24 \end{pmatrix}$$

Die Koordinatenform der Ebene  $E$  ergibt sich also mit Hilfe des Skalarprodukt<sup>4</sup>:

- `assume({x1,x2,x3}, Type::Real):  
KFormE:= linalg::scalarProduct(  
Normalenvektor,  
matrix([x1,x2,x3]) - Stuetzvektor) = 0`  
 $24x_1 + 12x_2 + 24x_3 - 108 = 0$

Als Zwischenergebnis wurde in der Aufgabenstellung 1.a) die Koordinatenform  $2 \cdot x_1 + x_2 + 2 \cdot x_3 - 9 = 0$  angegeben. Haben wir uns verrechnet? Ein scharfer Vergleich zwischen der als Zwischenergebnis angegebenen Koordinatenform und der von uns berechneten Koordinatenform zeigt, dass wir unsere Koordinatenform nur durch 12 dividieren müssen, um die als Zwischenergebnis angegebene Koordinatenform zu erhalten:

- `KFormE:= KFormE/12`  
 $2x_1 + x_2 + 2x_3 - 9 = 0$

Grafisch haben wir damit die folgende Situation, auf der auch unsere Überlegungen in den folgenden Aufgabenteilen basieren werden<sup>5</sup>:

- `PunktA:= plot::Point3d(OrtsvektorA,  
Color = RGB::Green):  
PunktB:= plot::Point3d(OrtsvektorB,  
Color = RGB::Black):  
PunktC:= plot::Point3d(OrtsvektorC,  
Color = RGB::Blue):  
PunktO:= plot::Point3d(OrtsvektorO,  
Color = RGB::Red):  
EbeneE:= plot::Surface(ParameterformE,`

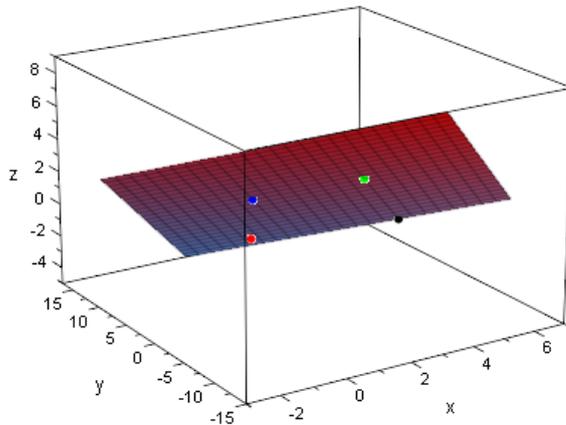
<sup>4</sup>Hier kommt wieder die Funktion `linalg::scalarProduct` zum Einsatz. Mit Hilfe von `assume` und `Type::Real` zeichnen wir  $x_1$ ,  $x_2$  und  $x_3$  explizit als reelle Variablen aus (siehe auch Seite 31).

<sup>5</sup>Wir zeichnen die vier Punkte  $A$ ,  $B$ ,  $C$  und den Ursprung  $O$  sowie die Ebene  $E$  in ein gemeinsames Koordinatensystem.

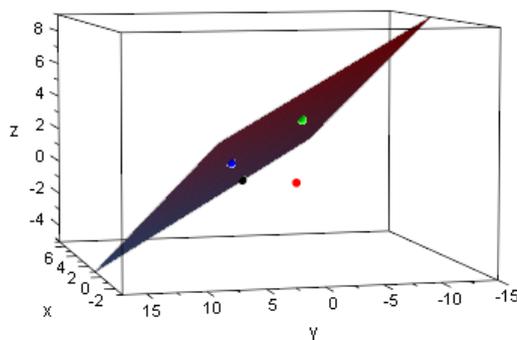
```

k = -1..1, l = -1..1):
plot(PunktA, PunktB, PunktC, PunktO, EbeneE)

```



Wir haben den Punkt  $A$  in grüner Farbe (`RGB::Green`), den Punkt  $B$  in schwarzer Farbe (`RGB::Black`), den Punkt  $C$  in blauer Farbe (`RGB::Blue`) und den Ursprung  $O$  in roter Farbe (`RGB::Red`) dargestellt. Erzeugt man die obige Grafik in MuPAD, so lässt diese sich wie üblich mit einem Doppelklick aktivieren und die grafische Szene kann in dem Grafikprogramm VCam (in MuPAD integriert) gedreht werden. Die folgende Ansicht auf unsere Szene zeigt besser, dass z.B. der Ursprung  $O$  nicht in der Ebene  $E$  enthalten ist:



**1.b)** Wir müssen zeigen, dass die Vektoren  $\vec{AB}$  und  $\vec{BC}$  senkrecht aufeinander stehen und den gleichen Betrag haben. Um zu zeigen, dass die beiden Vektoren senkrecht aufeinander stehen, genügt es, nach zurechnen, dass ihr Skalarprodukt Null ist:

- `VektorAB:= OrtsvektorB - OrtsvektorA:`  
`VektorBC:= OrtsvektorC - OrtsvektorB:`  
`linalg::scalarProduct(VektorAB,`  
`VektorBC)`

0

Den Betrag der Vektoren  $\vec{AB}$  und  $\vec{BC}$  berechnen wir wieder mit Hilfe des Befehls `norm`:

- `norm(VektorAB, 2), norm(VektorBC, 2)`  
6, 6

Beide Vektoren haben den Betrag 6 und damit die gleiche Länge.

**1.c)** Wir müssen den Punkt  $D$  bestimmen, für den  $ABCD$  ein Quadrat in der Ebene  $E$  bildet. Da  $A$ ,  $B$  und  $C$  in der Ebene  $E$  liegen, erhalten wir den Ortsvektor des Punktes  $D$  analog zu der Vorgehensweise in Abschnitt 2.7. Es muss gelten  $\overrightarrow{AB} = \overrightarrow{DC}$ <sup>6</sup>, d.h.:

- `OrtsvektorD := OrtsvektorA - OrtsvektorB + OrtsvektorC`

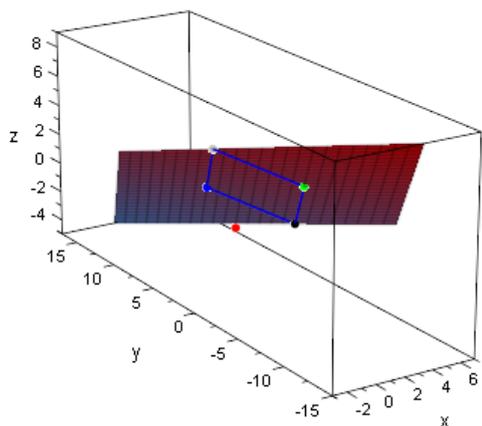
$$\begin{pmatrix} -1 \\ 1 \\ 5 \end{pmatrix}$$

Das Ergebnis deckt sich mit dem Zwischenergebnis, dass für die Koordinaten des Punktes  $D$  am Ende von Aufgabenteil 1.c) angegeben ist. Wir zeichnen zum Abschluss dieser Teilaufgabe die fünf Punkte  $A$ ,  $B$ ,  $C$ ,  $O$ , und  $D$  und das von  $A$ ,  $B$ ,  $C$  und  $D$  aufgespannte Quadrat sowie die Ebene  $E$  in ein gemeinsames Koordinatensystem<sup>7</sup>:

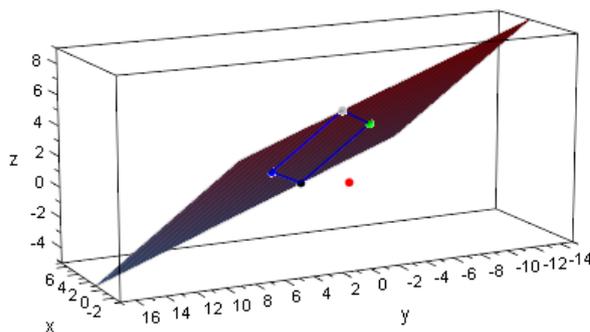
- `PunktD := plot::Point3d(OrtsvektorD,  
                          Color = RGB::Grey):  
StreckeAB := plot::Line3d(OrtsvektorA,  
                          OrtsvektorB):  
StreckeBC := plot::Line3d(OrtsvektorB,  
                          OrtsvektorC):  
StreckeCD := plot::Line3d(OrtsvektorC,  
                          OrtsvektorD):  
StreckeDA := plot::Line3d(OrtsvektorD,  
                          OrtsvektorA):  
Quadrat := StreckeAB, StreckeBC,  
          StreckeCD, StreckeDA:  
plot(PunktA, PunktB, PunktC, PunktD,  
      PunktO, EbeneE, Quadrat,  
      Scaling = Constrained)`

<sup>6</sup>Die Ecken eines Vierecks werden gemeinhin im mathematisch positiven Sinn, also gegen den Uhrzeigersinn, mit  $A$ ,  $B$ ,  $C$  und  $D$  beschriftet. Die Kenntnis dieser Konvention wird hier schlicht vorausgesetzt.

<sup>7</sup>Die Punkte  $A$ ,  $B$ ,  $C$ ,  $O$  und die Ebene  $E$  hatten wir oben bereits definiert, so dass wir hier auf die Variablen `PunktA`, `PunktB`, `PunktC`, `PunktO` und `EbeneE` einfach so zugreifen können. Wir verwenden darüber hinaus wieder die Option `Scaling = Constrained`, damit die Achsen des Koordinatensystems gleich skaliert werden und das von den Punkten aufgespannte Quadrat auch als solches erscheint. Mit Hilfe von `plot::Line3d` verbinden wir je zwei der vier Punkte, so dass sich das Quadrat  $ABCD$  in der Ebene  $E$  abzeichnet.



Auch hier kann die Grafik in MuPAD mit einem Doppelklick aktiviert und interaktiv erforscht werden: Die folgende Grafik zeigt – im Vorgriff auf Aufgabenteil 2 – bereits deutlich, dass die vier Punkte  $A$ ,  $B$ ,  $C$  und  $D$  in der Tat die Grundfläche für eine Pyramide mit Spitze im Ursprung  $O$  festlegen:



2. Die in der Aufgabenstellung beschriebene Pyramide lässt sich mit Hilfe von `plot::Polygon3d` visualisieren. Der MuPAD-Funktion `plot::Polygon3d` können wir eine Folge von Ortsvektoren von Punkten im Raum übergeben, die in einer Liste – also in eckigen Klammern – eingeschlossen sind<sup>8</sup>. Sie zeichnet dann einen geschlossenen Streckenzug durch diese Punkte (im Fall von drei Punkten also ein Dreieck, im Fall von vier Punkten ein Viereck). Die Pyramide besteht aus den vier Dreiecksflächen  $ABO$ ,  $BCO$ ,  $CDO$  und  $DAO$  sowie der Grundfläche  $ABCD$ <sup>9</sup>:

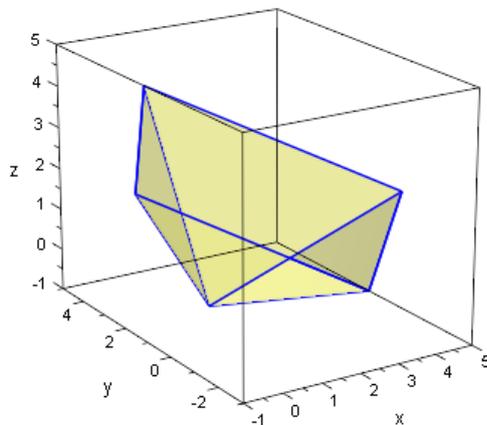
<sup>8</sup>Was Listen als Datenstrukturen für Eigenschaften in MuPAD haben, muss der Schüler an dieser Stelle nicht wissen; das Zusammenfassen der Vektoren zu einer Liste kann hier schlicht als technisches Hilfsmittel bei der Kommunikation mit MuPAD aufgefasst werden.

<sup>9</sup>Wir verwenden beim Zeichnen mit Hilfe des Befehls `plot` die Füllfarbe `LightYellow` (also ein zartes Gelb) mit einer Durchsichtigkeit von 80%. Mit Hilfe von `Filled = TRUE` sorgen wir dafür, dass die Grundfläche und die Seitenflächen der Pyramide vollständig ausgemalt werden. Andernfalls liefert `plot::Polygon3d` nur einen Streckenzug durch die angegebenen Punkt, so dass wir nur die Umrisslinien der Pyramide sehen würden.

```

• Pyramide:=
  plot::Polygon3d([OrtsvektorA,
                  OrtsvektorB,
                  OrtsvektorO]),
  plot::Polygon3d([OrtsvektorB,
                  OrtsvektorC,
                  OrtsvektorO]),
  plot::Polygon3d([OrtsvektorC,
                  OrtsvektorD,
                  OrtsvektorO]),
  plot::Polygon3d([OrtsvektorD,
                  OrtsvektorA,
                  OrtsvektorO]),
  plot::Polygon3d([OrtsvektorA,
                  OrtsvektorB,
                  OrtsvektorC,
                  OrtsvektorD]):
plot(Pyramide, Filled = TRUE,
     FillColor = RGB::LightYellow.[0.8],
     Scaling = Constrained)

```



**2.a)** Wir müssen zeigen, dass das von  $O$  aus gefällte Lot auf die Ebene  $E$  durch den Mittelpunkt des Quadrates  $ABCD$  verläuft. Dazu berechnen wir zuerst den Mittelpunkt des Quadrates. Dieser ist der Schnittpunkt der Diagonalen, d.h. sein Ortsvektor  $\vec{m}$  ist z.B. gegeben durch

$$\vec{m} = \vec{a} + \frac{1}{2} \cdot \overrightarrow{AC} = \vec{a} + \frac{1}{2} \cdot (\vec{c} - \vec{a}) = \frac{1}{2} \cdot \vec{a} + \frac{1}{2} \cdot \vec{c}.$$

Mit MuPAD erhalten wir:

```

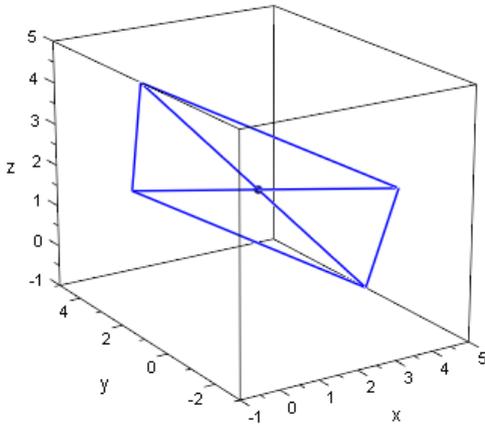
• OrtsvektorM:= 1/2*OrtsvektorA + 1/2*OrtsvektorC

```

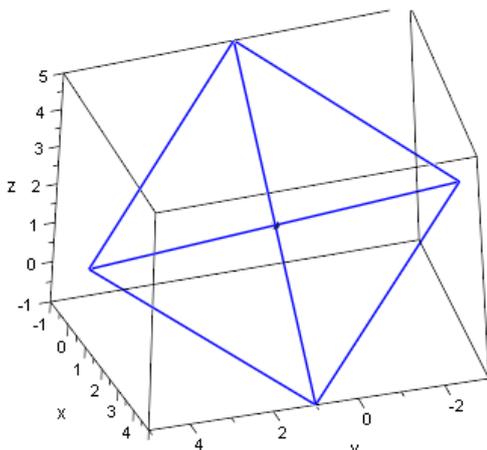
$$\begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$

Zur Kontrolle zeichnen wir das Quadrat  $ABCD$ , die Diagonalen  $AC$  und  $BD$  sowie den Punkt  $M$  in ein gemeinsames Koordinatensystem<sup>10</sup>:

- `DiagonaleAC:= plot::Line3d(OrtsvektorA, OrtsvektorC):`  
`DiagonaleBD:= plot::Line3d(OrtsvektorB, OrtsvektorD):`  
`PunktM:= plot::Point3d(OrtsvektorM):`  
`plot(Quadrat, DiagonaleAC, DiagonaleBD,`  
`PunktM, Scaling = Constrained)`



Die Sicht auf das Quadrat ist ein kleines ungünstig. Am besten aktiviert man die Grafik wieder mit einem Doppelklick und dreht die gesamte Szene. Auf diese Weise erhält man z.B. die folgende Ansicht, in der sich leicht erkennen lässt, dass  $M$  in der Tat der Schnittpunkt der Diagonalen und damit der Mittelpunkt des Quadrates ist.



<sup>10</sup>Auch hier sollte einmal wieder darauf geachtet werden, die Option `Scaling = Constrained` zu verwenden, damit die Achsen gleich skaliert werden und das Quadrat nicht verzerrt erscheint.

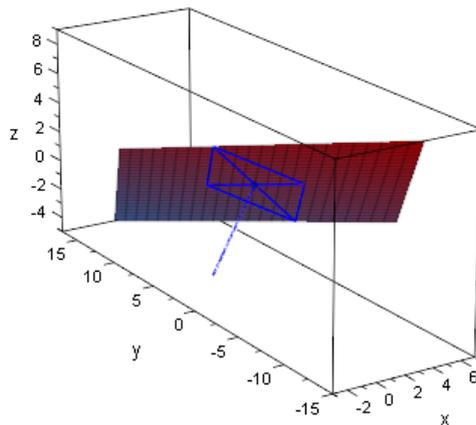
Nun müssen wir vom Ursprung  $O$  das Lot auf die Ebene  $E$  fällen. Dazu bestimmen wir eine Parameterform der Lotgeraden, indem wir den Koordinatenursprung  $O$  als Stützvektor und einen Normalenvektor der Ebene (den wir oben in Aufgabenteil 1.a) bereits berechnet und in der Variablen `Normalenvektor` abgespeichert hatten) als Richtungsvektor der Geraden verwenden<sup>11</sup>:

- `ParameterformL := OrtsvektorO + m*Normalenvektor`

$$\begin{pmatrix} 24m \\ 12m \\ 24m \end{pmatrix}$$

Jetzt müssen wir nur noch zeigen, dass der Schnittpunkt der Lotgeraden mit der Ebene  $E$  tatsächlich durch den Punkt  $M$  gegeben ist. Grafisch können wir uns diese Vermutung wieder ganz leicht plausibel machen:

- `Lotgerade := plot::Curve3d(ParameterformL, m = -0.1..0.1):`  
`plot(EbeneE, Quadrat, PunktM, DiagonaleAC, DiagonaleBD,`  
`Lotgerade, Scaling = Constrained)`



Zur Berechnung des Schnittpunktes der Lotgeraden mit der Ebene  $E$ , setzen wir die Komponenten der Parameterform der Geraden für  $x_1$ ,  $x_2$  und  $x_3$  in die Koordinatenform der Ebene  $E$  ein, die wir unter Teilaufgabe 1.a) berechnet haben und lösen anschließend die entstehende Gleichung nach  $m$  auf:

- `G1 := subs(KFormE, x1 = ParameterformL[1],`  
`x2 = ParameterformL[2],`  
`x3 = ParameterformL[3]):`  
`solve(G1, m)`

<sup>11</sup>Natürlich hätte man hier den Vektor `OrtsvektorO` auch einfach weglassen können, denn es handelt sich ja eh um den Nullvektor.

$$\left\{ \frac{1}{12} \right\}$$

Einsetzen des Werte  $m = \frac{1}{12}$  in die Parameterform der Lotgerade liefert den Ortsvektor des Lotfußpunktes auf der Ebene  $E$ :

- `OrtsvektorL:= subs(ParameterformL, m = 1/12)`

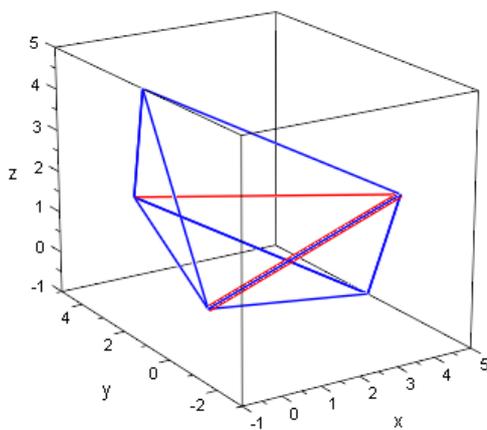
$$\begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$

Dies ist aber genau der Ortsvektor des Punktes  $M$ , den wir als Mittelpunkt des Quadrates oben berechnet hatten.

**2.b)** In dieser Teilaufgabe soll der spitze Winkel bestimmt werden, den die Seitenkante  $OA$  der Pyramide mit der von  $A$  ausgehenden Diagonalen des Quadrates  $ABCD$  einschließt. Wir veranschaulichen uns die Situation zuerst grafisch. Wir zeichnen die Seitenkante  $OA$  der Pyramide explizit als rote Linie mit Hilfe von `plot::Line3d` in die Pyramide ein. Die Farbe der Diagonalen `DiagonaleAC` setzen wir ebenfalls auf `RGB::Red`. Damit man die rot markierte Seitenkante darüber hinaus besser erkennen kann, verwenden wir das Attribut `LineWidth`, mit dessen Hilfe sich die Linienstärke erhöhen lässt:

- `SeitenkanteOA:= plot::Line3d(OrtsvektorO,  
OrtsvektorA,  
Color = RGB::Red,  
LineWidth = 1.2):`

```
DiagonaleAC::Color:= RGB::Red:
plot(Pyramide, SeitenkanteOA, DiagonaleAC,
      Scaling = Constrained)
```



Die roten Linien in der obigen Darstellung der Pyramide schließen den Winkel ein, den wir berechnen müssen. Wir müssen also den Winkel zwischen dem Vektor  $\overrightarrow{AO} = -\vec{a}$  und dem Vektor  $\overrightarrow{AC} = \vec{c} - \vec{a}$  bestimmen. Zur Berechnung von Winkeln zwischen Vektoren wird wie gewohnt die MuPAD-Funktion `linalg::angle` verwendet:

```
• a := linalg::angle(-OrtsvektorA,
                    OrtsvektorC - OrtsvektorA)
```

$$\arccos\left(\frac{\sqrt{72}\sqrt{27}}{54}\right)$$

In Grad ergibt das

```
• float((a * 180)/PI)
35.26438968
```

also beträgt der Winkel etwa  $35^\circ$ .

**2.c)** Die Höhe der Pyramide ist einfach die Länge des Vektors  $\overrightarrow{OM} = \vec{m}$ , wobei  $M$  der Mittelpunkt des Quadrates ist, den wir in Aufgabe 2.a) berechnet hatten. Damit ist die Höhe  $h$  gegeben durch

```
• h := norm(OrtsvektorM, 2)
3
```

Zur Berechnung des Volumens müssen wir diesen Wert nur mit der Masszahl der Fläche der Grundfläche der Pyramide multiplizieren. Das Quadrat  $ABCD$  hat den Flächeninhalt

```
• A := norm(OrtsvektorB - OrtsvektorA)^2
```

Damit beträgt das Volumen  $V$  der Pyramide

```
• V := h * A
48
```

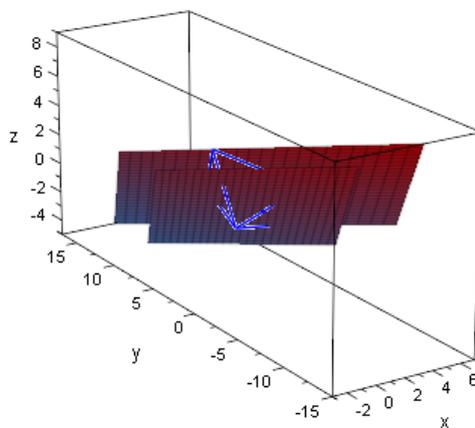
**2.d)** In der letzten Teilaufgabe sollen wir eine Ebene  $F$  bestimmen, die parallel zur Grundfläche der Pyramide verläuft und die Höhe der Pyramide halbiert. Wenn  $F$  parallel zur Grundfläche der Pyramide liegen soll, dann muss  $F$  auch parallel zu unserer ursprünglichen Ebene  $E$  sein, d.h. wir können für  $F$  die gleichen Spannvektoren wie zur Angabe der Parameterform von  $E$  aus Teilaufgabe 1.a) verwenden. Als Antragsvektor für die Ebene  $F$  wählen wir den Ortsvektor des Mittelpunktes  $M_2$  der Strecke  $OM$ , also  $\vec{m}_2 = \frac{1}{2} \cdot \vec{m}$ . Damit ist eine Parameterform von  $F$  gegeben durch:

```
• Antragsvektor := 1/2*OrtsvektorM:
  ParameterformF := Antragsvektor + r*Spannvektor1
                  + s*Spannvektor2
```

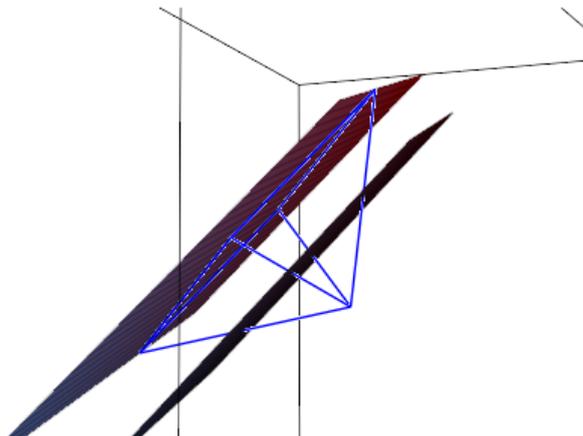
$$\begin{pmatrix} 2r - 2s + 1 \\ 4r + 8s + \frac{1}{2} \\ 1 - 2s - 4r \end{pmatrix}$$

Zeichnen wir nochmals die Pyramide sowie die beiden Ebenen  $E$  und  $F$  in ein gemeinsames Koordinatensystem, so ergibt sich anschaulich

- `EbeneF := plot::Surface(ParameterformF, r = -1..1, s = -1..1):`  
`plot(Pyramide, EbeneE, EbeneF, Scaling = Constrained)`



Dreht man die gesamte Szene und zoomt man ein wenig näher, so lässt sich ganz deutlich erkennen, dass die Ebene  $F$  parallel zur Ebene  $E$  verläuft und die Höhe der Pyramide halbiert.



Damit sind alle Aufgaben erfolgreich gelöst und wir haben das Abitur im Grundkurs Mathematik mit etwas Nachdenken und leichter Unterstützung von MuPAD bestanden ;-).

## 5.2 Leistungskursniveau

Die folgenden Aufgaben wurden im Jahr 1982 im Rahmen der Abiturprüfung für Leistungskurse in Bayern gestellt<sup>12</sup>:

**Aufgaben zur Analytischen Geometrie:** Gegeben ist eine Ebenenschar  $E_t$  in einem kartesischen Koordinatensystem

$$t \cdot x_1 + x_2 - x_3 - 2 \cdot t = 0, \quad t \in \mathbb{R}.$$

1. a.) Bestimmen Sie die Schnittgerade  $s$  von  $E_0$  und  $E_1$  in Parameterform und prüfen Sie, ob  $s$  auf allen Ebenen der Schar liegt. [Eine mögliche Lösung für  $s$  lautet:  $\vec{x} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \sigma \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ ,  $\sigma \in \mathbb{R}$ .]

$$\vec{x} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \sigma \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \sigma \in \mathbb{R}.$$

- b) Welchen Abstand hat der Ursprung von  $s$ ?

2. Die Scharebenen  $E_{t_1}$  und  $E_{t_2}$  sollen senkrecht aufeinander stehen. Welcher Bedingungsgleichung müssen  $t_1$  und  $t_2$  genügen? Zu welcher Ebene aus der Schar gibt es keine senkrechte Ebene aus dieser Schar?

3. Weiter ist eine Ebene  $F$  gegeben mit der Gleichung

$$\vec{x} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \mu \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \lambda, \mu \in \mathbb{R}.$$

Zeigen Sie, dass zwar  $s$  auch in  $F$  liegt,  $F$  aber keine Ebene der gegebenen Schar ist. Welchen Winkel bildet  $F$  mit  $E_1$  (auf Grad genau)?

4. Stellen Sie fest, ob es Parameterwerte  $t$  gibt, für die folgendes gilt: Der Punkt  $P(5|7|1)$  hat von  $E_t$  den Abstand  $\sqrt{3}$  und liegt bezüglich  $E_t$  in dem Halbraum, der den Ursprung nicht enthält. Führen Sie auch die Probe durch.

**Lösung:** Wir definieren als erstes die Ebenenschar in MuPAD, wobei wir MuPAD mit Hilfe von `assume(x1, x2, x3, t, Type::Real)` explizit mitteilen, dass die Variablen `x1`, `x2`, `x3` und der Parameter `t` reell sind:

- `assume({x1,x2,x3,t}, Type::Real):`  
`KFormE_t:= t*x1 + x2 - x3 - 2*t = 0`

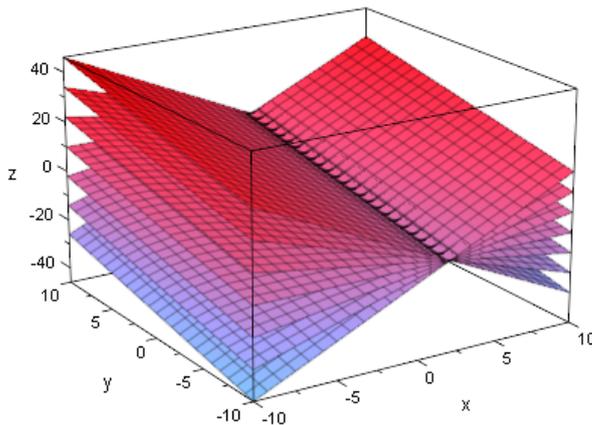
<sup>12</sup>Die Aufgabenstellung ist wörtlich aus [2] übernommen: Siehe [2], Anhang A. 5. Seiten 201/202

$$x_2 - 2t - x_3 + tx_1 = 0$$

Vorab können wir einige der Ebenen der Schar direkt mit Hilfe von `plot::Surface` darstellen<sup>13</sup>. Wir stellen die Ebenen  $E_t$  für  $t = -3, -2, -1, 0, 1, 2, 3$  dar<sup>14</sup>:

```
• E_t:= plot::Surface([x1, x2, t*x1 + x2 - 2*t],
                    x1 = -10..10, x2 = -10..10)
  $ t = -3..3:

plot(E_t)
```



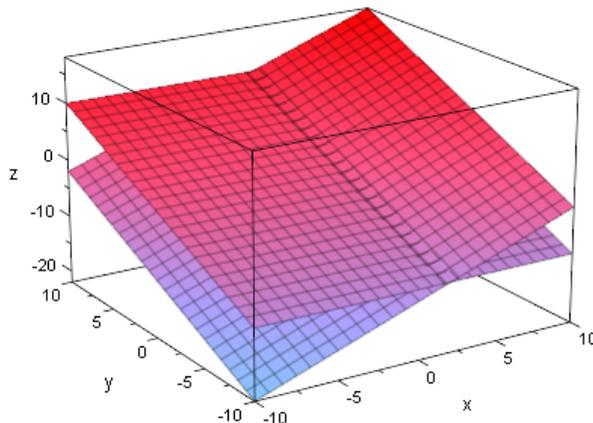
**1.a)** Gesucht ist zunächst die Schnittgerade der Ebenen  $E_0$  und  $E_1$ . Wir zeichnen die beiden Ebenen zunächst in ein gemeinsames Koordinatensystem:

```
• E_t:= plot::Surface([x1, x2, t*x1 + x2 - 2*t],
                    x1 = -10..10, x2 = -10..10)
  $ t in {0, 1}:

plot(E_t)
```

<sup>13</sup>`plot::Surface` akzeptiert im Prinzip nur Parameterformen von Ebene. Wir definieren hier aber aus Bequemlichkeitsgründen nicht extra eine Parameterform der Ebenenschar  $E_t$ , sondern stellen die Koordinatenform der Ebenenschar nach  $x_3$  um und übergeben an `plot::Surface` die Liste `[x1, x2, t*x1 + x2 - 2*t]`, so dass sich die Ebenenschar mit Hilfe von `plot::Surface` darstellen lässt. Alternativ kann auch die Funktion `plot::Implicit3d` verwendet werden, der man direkt die Koordinatenform einer Ebene übergeben kann. Da wir diese MuPAD-Funktion aber bisher in diesem Buch nicht verwendet haben, wollen wir auch an dieser Stelle darauf verzichten. Der interessierte Leser erhält auf der Hilfeseite zu `plot::Implicit3d` (zu erreichen über `?plot::Implicit3d`) alle nötigen Informationen sowie viele Anwendungsbeispiele zur Verwendung dieser Funktion.

<sup>14</sup>Wäre die Aufgabenstellung auf die Bearbeitung mit einem Computeralgebra-System wie MuPAD ausgelegt, so könnten einige der folgenden Teilaufgaben so formuliert werden, dass die Schülerinnen und Schüler die nachzuweisenden Hypothesen über die Lage der Ebenen und Geraden durchaus selbstständig aufstellen sollen. Die grafischen Visualisierungsmöglichkeiten von MuPAD eignen sich hier einmal mehr hervorragend zur eigenen Hypothesenbildung.



Die Grafik veranschaulicht, dass sich  $E_0$  und  $E_1$  in der Tat in einer Geraden schneiden. Wir definieren ihre Koordinatenform in MuPAD und weisen dies explizit nach. Dabei verfahren wir ganz analog zu der in Abschnitt 3.7.3 kennen gelernten Vorgehensweise:

```

• KFormE_0:= subs(KFormE_t, t = 0);
  KFormE_1:= subs(KFormE_t, t = 1);
  solve({KFormE_0, KFormE_1}, {x1, x2, x3})
  x2 - x3 = 0

```

$$x_1 + x_2 - x_3 - 2 = 0$$

$$\{[x_1 = 2, x_2 = x_3]\}$$

MuPAD sagt uns, dass  $x_2$  durch den Wert 2 festgelegt ist und gibt uns  $x_2$  in Abhängigkeit von  $x_3$  an. Unser entsprechendes Gleichungssystem ist also lösbar, d.h. die Ebenen  $E_0$  und  $E_1$  schneiden sich in einer Geraden. Wir müssen jetzt noch die Gleichung der Schnittgerade aus der Lösung ablesen. Dazu beobachten wir, dass wir  $x_3$  frei wählen können. Setzen wir also  $x_3 = k$ , so gilt

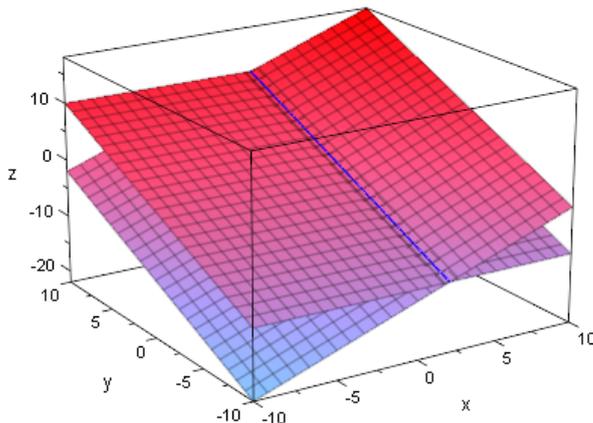
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ k \\ k \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

Damit haben wir eine Gleichung der Schnittgeraden bestimmt. Setzen wir für  $k$  gedanklich den Wert 1 ein, so erhalten wir den Ortsvektor des Punkt  $(2|1|1)$ , der als Stützvektor der als Lösung angegebenen Geraden in der Aufgabenstellung aufgeführt ist. Der Richtungsvektor der als Lösung angegebenen Geraden stimmt mit dem Richtungsvektor der von uns ermittelten Schnittgeraden überein. Damit haben wir korrekt gerechnet. Jetzt zeichnen wir die beiden Ebenen von oben sowie die Schnittgerade in ein gemeinsames Koordinatensystem:

```

• ParameterformS:= matrix([2,0,0]) + k*matrix([0,1,1]):
  GeradeS:= plot::Curve3d(ParameterformS, k = -10..10):
  plot(E_t, GeradeS):

```



Wir müssen noch die Frage beantworten, ob  $s$  auf allen Ebenen der Schar liegt. Dazu setzen wir die Komponenten der Parameterform von  $s$  in die allgemeine Koordinatenform der Ebenenschar  $E_t$  ein und lösen die entstehende Gleichung nach

- `G1 := subs(KFormE_t, x1 = ParameterformS[1],  
x2 = ParameterformS[2],  
x3 = ParameterformS[3])`

$$0 = 0$$

In der Tat liegt  $s$  auf allen Ebenen der Schar  $E_t$ .

**1.b)** Es gilt den Abstand der Geraden  $s$  vom Koordinatenursprung  $O(0|0|0)$  zu ermitteln. Wie wir prinzipiell vorzugehen haben, haben wir uns in Abschnitt 3.10.1 bereits überlegt (dort findet sich auch ein ausführliches Beispiel). Da wir das Lot vom Ursprung auf die Gerade  $s$  fällen müssen, von dem Lotfußpunkt  $L$  aber nur wissen, dass er ein Punkt der Geraden  $s$  sein muss, ist der Ortsvektor des Punktes  $L$  allgemein von der Form:

- `OrtsvektorL := ParameterformS`

$$\begin{pmatrix} 2 \\ k \\ k \end{pmatrix}$$

Das Skalarprodukt des Richtungsvektors der Geraden  $s$  mit diesem Vektor muss gleich Null sein:

- `linalg::scalarProduct(OrtsvektorL,  
matrix([0,1,1])) = 0`

$$2k = 0$$

Die Lösung der Gleichung ist  $k = 0$  (wofür wir hier MuPAD wirklich nicht mehr explizit bemühen wollen). Den Lotfußpunkt auf der Geraden  $s$  erhalten wir also, indem wir den Wert  $k = 0$  in die Parameterform der Gerade  $s$  einsetzen. Die Koordinaten des Lotfußpunktes sind also  $(2|0|0)$ :

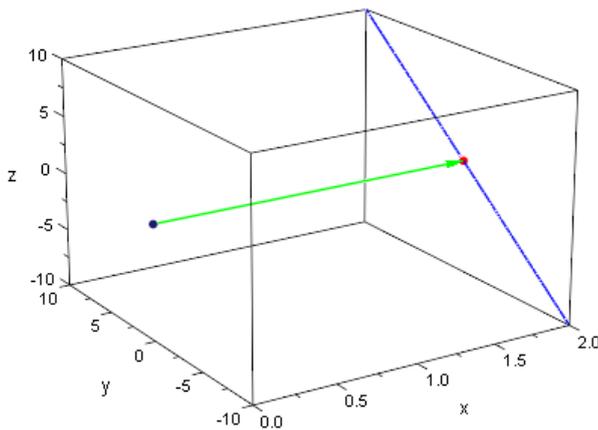
- `OrtsvektorL:= matrix([2,0,0]):`

Der Abstand  $d$  der Geraden  $g$  zum Ursprung  $O$  ist also gleich der Länge des Ortsvektors des Lotfußpunktes, d.h.

- `d:= norm(OrtsvektorL, 2)`  
2

Zur Veranschaulichung zeichnen wir den Ursprung, das Lot, den Lotfußpunkt sowie die Gerade  $s$  in ein gemeinsames Koordinatensystem<sup>15</sup>:

- `Ursprung:= plot::Point3d(matrix([0,0,0])):`  
`L:= plot::Point3d(OrtsvektorL,`  
`Color = RGB::Red):`  
`Lot:= plot::Arrow3d(OrtsvektorL,`  
`Color = RGB::Green):`  
`plot(Ursprung, L, Lot, Gerades):`



2. Die Ebenen  $E_{t_1}$  und  $E_{t_2}$  soll senkrecht aufeinander stehen. Dies ist genau dann der Fall, wenn ihre Normalenvektoren orthogonal zueinander sind. Die Koordinatenformen und  $E_{t_1}$  und  $E_{t_2}$  sind laut Aufgabenstellung ganz allgemein von der Form

$$t_1 \cdot x_1 + x_2 - x_3 - 2 \cdot t_1 = 0 \quad \text{und} \quad t_2 \cdot x_1 + x_2 - x_3 - 2 \cdot t_2 = 0.$$

Die Komponenten eines Normalenvektor der entsprechenden Ebene erhalten wir als Koeffizienten der jeweiligen Normalenform, d.h.

- `assume({t1,t2}, Type::Real):`  
`NormalenvektorE_t1:= matrix([t1,1,-1]):`  
`NormalenvektorE_t2:= matrix([t2,1,-1]):`

<sup>15</sup>Übergibt man der Funktion `plot::Arrow3d` nur einen Vektor, so wird automatisch der Pfeil vom Nullpunkt hin zu dem Punkt gezeichnet, dessen Ortsvektor angegeben wurde.

Sollen die beiden Vektoren orthogonal zueinander sein, dann muss ihr Skalarprodukt gleich Null sein, d.h. wir erhalten für  $t_1$  und  $t_2$  die Bedingung

- `G1:= linalg::scalarProduct(  
NormalenvektorE_t1,  
NormalenvektorE_t2) = 0`  
 $t_1 t_2 + 2 = 0$

Gilt also  $t_1 \cdot t_2 + 2 = 0$ , so sind  $E_{t_1}$  und  $E_{t_2}$  orthogonal zueinander. Damit ist auch klar, zu welcher Ebene der Schar es keine orthogonale Ebene geben kann: Für  $t_1 = 0$  oder  $t_2 = 0$  ist die Bedingung  $t_1 \cdot t_2 + 2 = 0$  nicht erfüllbar, d.h.  $E_0$  ist die einzige Ebene der Schar, zu der es keine senkrecht darauf stehende Ebene der Schar gibt.

3. Wir müssen zeigen, dass die Ebene  $F$  definiert durch

$$\vec{x} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \mu \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \lambda, \mu \in \mathbb{R}.$$

die Gerade  $s$  enthält, aber keine Ebene der Schar  $E_t$  ist. Da wir mit der Koordinatenform der Ebene durch direkten Vergleich der Koordinatenformen der Ebenen  $E_t$  besser feststellen können, dass  $F$  keine Ebene der Schar ist, definieren wir sofort auch die Koordinatenform der Ebene  $F$  in MuPAD. Ein Normalenvektor ist schnell gefunden, denn

offensichtlich ist der Vektor  $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$  orthogonal zu beiden Spannvektoren<sup>16</sup>:

- `ParameterformF:= matrix([2,0,0]) + k*matrix([0,1,0])  
+ l*matrix([0,0,1]):  
NormalenvektorF:= matrix([1,0,0]):  
StuetzvektorF:= matrix([2,0,0]):  
KFormF:= linalg::scalarProduct(  
NormalenvektorF,  
matrix([x1,x2,x3])-StuetzvektorF) = 0`  
 $x_1 - 2 = 0$

Dass  $F$  die Gerade  $s$  enthält, können wir jetzt leicht nachweisen, indem wir einfach die Komponenten der Parameterform der Geraden  $s$  in die Koordinatenform von  $F$  einsetzen. Da die Koordinatenform von  $F$  eh nur die Variable  $x_1$  enthält, brauchen wir auch nur die erste Komponente der Geraden  $s$  in die Koordinatenform einzusetzen:

- `subs(KFormF, x1 = ParameterformS[1])`

<sup>16</sup>Hier hätte man natürlich auch wieder die Funktion `linalg::crossProduct` bemühen können, um einen Normalenvektor der Ebene zu bestimmen, sofern man einen Kandidaten für denselben nicht direkt durch scharfes Ansehen der Gleichung ausmachen kann.

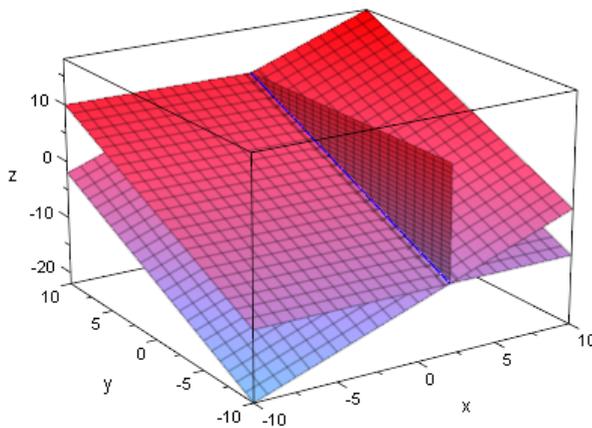
$$0 = 0$$

Die entstehende Gleichung hat unendlich viele Lösungen, d.h.  $s$  liegt in  $F$ . Dass  $F$  keine Ebene der Schar  $E_t$  sein kann, sehen wir unmittelbar durch Vergleich der Koordinatenformen  $F : x_1 - 2 = 0$  und  $E_t : t \cdot x_1 + x_2 - x_3 - 2 \cdot t = 0$ . Wäre  $F$  eine Ebene der Schar, so müsste  $t = 1$  gelten. Die beiden Koordinatenformen  $F$  und  $E_1$  stimmen aber weder überein, noch lassen sie sich durch Multiplikation mit einer entsprechenden Konstanten ineinander überführen. Also ist  $F$  keine Ebene der Schar. Zum Anschluss zeichnen wir noch einmal die Ebene  $F$ , die Gerade  $s$  sowie einige Ebenen der Schar  $E_t$  in ein gemeinsames Koordinatensystem:

```

• F:= plot::Surface(ParameterformF,
                    k = -10..10, l = -10..10):
  plot(E_t, GeradeS, F):

```



Diejenige Ebene, die in der obigen Darstellung vertikal durch die Schnittgerade  $s$  verläuft, ist genau die Ebene  $F$ . Bevor wir zum letzten Aufgabenteil kommen, müssen wir noch den spitzen Winkel berechnen, den  $F$  mit  $E_1$  bildet. Der Schnittwinkel zweier Ebenen ist durch den Winkel zwischen ihre Normalenvektoren gegeben (siehe auch Abschnitt 3.9.3):

```

• w:= linalg::angle(NormalenvektorF,
                  matrix([1,1,-1]));
  float((w * 180)/PI)

```

$$\arccos\left(\frac{\sqrt{3}}{3}\right)$$

54.73561032

Wir haben den Winkel einmal exakt im Bogenmaß und einmal auf 10 signifikante Stellen im Bogenmaß berechnet. In Grad beträgt er also etwa  $55^\circ$ .

4. Es soll festgestellt werden, ob es Parameterwerte für  $t$  gibt, für die der Punkt  $P(5|7|1)$  von  $E_t$  den Abstand  $\sqrt{3}$  und bezüglich  $E_t$  in dem Halbraum liegt, der den Ursprung nicht enthält. Zur Beantwortung der Fragestellung können wir zunächst den Abstand des Punktes  $P$  von einer allgemeinen Ebene  $E_t$  der Schar berechnen. Für die Lotgerade  $l$  von  $P$  aus auf  $E_t$  können wir den Ortsvektor von  $P$  als Stützvektor und einen Normalenvektor der Ebene als Richtungsvektor verwenden (siehe auch Abschnitt 3.10.4):

- OrtsvektorP:= matrix([5,7,1]):  
ParameterformL:= OrtsvektorP + k\*matrix([t,1,-1])  
$$\begin{pmatrix} kt + 5 \\ k + 7 \\ 1 - k \end{pmatrix}$$

In Abhängigkeit von  $t$  können wir jetzt den Lotfußpunkt auf der Ebene  $E_t$  als Schnittpunkt von  $l$  und  $E_t$  bestimmen:

- Gl:= subs(KFormE\_t, x1 = ParameterformL[1],  
x2 = ParameterformL[2],  
x3 = ParameterformL[3]);

$$L := \text{solve}(Gl, k)$$

$$2k - 2t + t(kt + 5) + 6 = 0$$

$$\left\{ -\frac{3(t+2)}{t^2+2} \right\}$$

Den erhaltenen Wert für  $k$  setzen wir in die Parameterform der Lotgeraden  $l$  ein:

- OrtsvektorL:= subs(ParameterformL,  
k = L[1])

$$\begin{pmatrix} 5 - \frac{3t(t+2)}{t^2+2} \\ 7 - \frac{3(t+2)}{t^2+2} \\ \frac{3(t+2)}{t^2+2} + 1 \end{pmatrix}$$

Der Abstand der Ebene  $E_t$  von dem Punkt  $P$  entspricht jetzt also der Länge des Vektor vom Punkt  $P$  hin zum Lotfußpunkt:

- dt:= norm(OrtsvektorL - OrtsvektorP, 2)

$$3 \sqrt{\frac{2(t+2)^2}{(t^2+2)^2} + \frac{t^2(t+2)^2}{(t^2+2)^2}}$$

Wir suchen nach entsprechenden Werten für  $t$ , so dass dieser Abstand den Wert  $\sqrt{3}$  annimmt, d.h. wir müssen die Gleichung  $dt = \sqrt{3}$  nach  $t$  auflösen:

- solve(dt = sqrt(3), t)

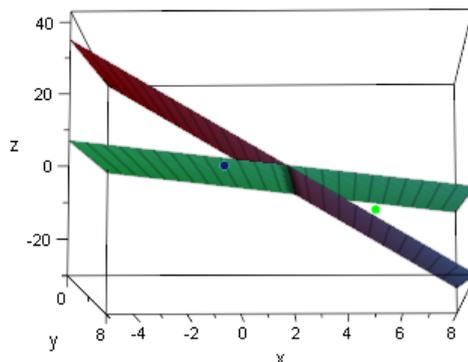
$$\{-5, -1\}$$

Potentielle Werte für  $t$  sind also  $-5$  und  $-1$ . Für beide Werte hat die entsprechende Ebene  $E_t$  den Abstand  $\sqrt{3}$  von dem Punkt  $P$ . Wir müssen jetzt noch feststellen, ob der Punkt  $P$  jeweils im gleichen Halbraum mit dem Koordinatenursprung liegt oder nicht. Bevor wir aber weiter rechnen, veranschaulichen wir uns die gegenwärtige Situation geometrisch. Wir zeichnen dazu die beiden Ebenen  $E_{-5}$  und  $E_{-1}$ , den Punkt  $P$  sowie den Koordinatenursprung in ein gemeinsames Koordinatensystem<sup>17</sup>:

```

• Eminus5:= plot::Surface([x1, x2, -5*x1 + x2 + 10],
                        x1 = -5..8, x2 = 0..8,
                        FillColor = RGB::Red):
Eminus1:= plot::Surface([x1, x2, -x1 + x2 + 2],
                        x1 = -5..8, x2 = 0..8,
                        FillColor = RGB::Green):
P:= plot::Point3d(OrtsvektorP,
                  Color = RGB::Green):
plot(Eminus5, Eminus1, Ursprung, P)

```



Die Grafik zeigt uns, dass der Punkt  $P$  (in Grün dargestellt) und der Koordinatenursprung (in Blau dargestellt) bezüglich der Ebene  $E_{-5}$  (in Rot dargestellt) in verschiedenen Halbräumen liegen. Bezüglich der Ebene  $E_{-1}$  (in Grün dargestellt) liegen beide Punkte im gleichen Halbraum. Diese Vermutungen müssen wir jetzt nur noch rechnerisch nachweisen.

<sup>17</sup>Die unten stehende Grafik wurde leicht gedreht, damit sich die Lage der Punkte  $P$  (in grüner Farbe dargestellt) und der Ursprung  $O$  (in blauer Farbe dargestellt) zu den Ebenen  $E_{-5}$  (in roter Farbe dargestellt) bzw.  $E_{-1}$  (in grüner Farbe dargestellt) besser erkennen lässt. Ferner haben wir den Koordinatenursprung in der Variablen **Ursprung** bereits bei der Bearbeitung von Teilaufgabe 1.b) in MuPAD definiert.

Eine Möglichkeit, dies zu tun, ist die folgende: Wir definieren die Gerade  $h$  mit

$$h : \vec{x} = m \cdot \begin{pmatrix} 5 \\ 7 \\ 1 \end{pmatrix}$$

die durch den Koordinatenursprung und den Punkt  $P$  verläuft und berechnen jeweils den Schnittpunkt mit der Ebene  $E_{-5}$  und der Ebene  $E_{-1}$ . In beiden Fällen erhalten wir, wenn wir die Komponenten der Parameterform von  $h$  in die betreffende Ebene einsetzen, einen Wert für den Parameter  $m$  aus der Geradengleichung. Gilt für den für  $m$  erhaltenen Wert  $0 < |m| < 1$ , so gibt es einen Schnittpunkt der Geraden  $h$  durch  $O$  und  $P$  mit der jeweiligen Ebene, der zwischen  $O$  und  $P$  liegt. Geometrisch bedeutet dies nichts anderes, als dass die jeweilige Ebene zwischen den beiden Punkten  $O$  und  $P$  verläuft. Ist der für  $m$  erhaltene Wert betragsmäßig größer als 1, also  $|m| > 1$ , so liegen beide Punkte bezüglich der betreffenden Ebene im gleichen Halbraum (der Fall  $m = 1$  ist uninteressant, denn setzen wir in die Parameterform der Geraden  $h$  für  $m$  den Wert 1 ein, so erhalten wir den Punkt  $P$  selbst). Wir rechnen dies mit MuPAD nach: Zuerst definieren wir die Parameterform der Geraden  $h$

- `ParameterformH:= m*OrtsvektorP`

$$\begin{pmatrix} 5m \\ 7m \\ m \end{pmatrix}$$

und setzen die Komponenten der Parameterform zunächst in die Koordinatenform der Ebene  $E_{-5}$  ein. Die erhaltene Gleichung lösen wir dann auch direkt nach  $m$  auf und erhalten somit:

- `KFormEminus5:= subs(KFormE_t, t = -5);`  
`G1:= subs(KFormEminus5, x1 = ParameterformH[1],`  
`x2 = ParameterformH[2],`  
`x3 = ParameterformH[3]);`

`solve(G1, m)`

$$x2 - 5x1 - x3 + 10 = 0$$

$$10 - 19m = 0$$

$$\left\{ \frac{10}{19} \right\}$$

Der für  $m$  erhaltenen Wert liegt betragsmäßig zwischen 0 und 1, d.h.  $P$  und der Ursprung liegen in der Tat in verschiedenen Halbräumen. Die gleiche Vorgehensweise für die Ebene  $E_{-1}$  liefert:

```

• KFormEminus1:= subs(KFormE_t, t = -1);
  G1:= subs(KFormEminus1, x1 = ParameterformH[1],
           x2 = ParameterformH[2],
           x3 = ParameterformH[3]);

solve(G1, m)
x2 - x1 - x3 + 2 = 0

m + 2 = 0

{-2}

```

Es gilt  $|m| = |-2| = 2 > 1$ , also liegen  $O$  und  $P$  bezüglich der Ebene  $E_{-1}$  im gleichen Halbraum. Damit ist die Aufgabe gelöst. Die folgenden vertiefenden Aufgaben stammen aus [2], Anhang A.5. Aufgaben aus Abiturprüfungen, Seite 199.

### Übungen

65.) Gegeben sind die Gerade  $g : \vec{x} = \begin{pmatrix} -2 \\ 4 \\ 7 \end{pmatrix} + k \cdot \begin{pmatrix} -3 \\ 2 \\ 6 \end{pmatrix}$  sowie die Punkte  $A(1|2|1)$ ,  $B(0|-2|10)$  und  $C(3|-4|4)$ .

- Zeigen Sie, dass der Punkt  $A$  auf der Geraden  $g$  liegt.
- Die Punkte  $B$  und  $C$  bestimmen die Gerade  $h$ . Zeigen Sie, dass  $g$  und  $h$  zwei verschiedene parallele Geraden sind.
- Bestimmen Sie eine Gleichung der von  $g$  und  $h$  aufgespannten Ebene  $F_1$  in Parameter- und in Normalenform. [Mögliches Teilergebnis:  $F_1 : 6 \cdot x_1 + 3 \cdot x_2 + 2 \cdot x_3 - 14 = 0$ ]
- Weiter ist die Ebene  $F_2 : \vec{x} = \begin{pmatrix} -2 \\ -1 \\ -10 \end{pmatrix} + l \cdot \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix} + m \cdot \begin{pmatrix} 0 \\ 2 \\ -3 \end{pmatrix}$  gegeben. Berechnen Sie den Abstand eines allgemeinen Punktes  $P(p_1|p_2|p_3)$  der Ebene  $F_2$  von der Ebene  $F_1$ . Welche Folgerung lässt sich aus dem Ergebnis hinsichtlich der gegenseitigen Lage von  $F_1$  und  $F_2$  ziehen?
- Bestimmen Sie eine Gleichung der Geraden durch  $A$ , die auf der Ebenen  $F_1$  senkrecht steht. In welchem Punkt  $L$  schneidet diese Lotgerade die Ebene  $F_2$ ? [Teilergebnis:  $L(-5|-1|-1)$ ]
- Welchen spitzen Winkel schließen die Geraden  $LA$  und  $LC$  ein?
- Die Gerade  $h$  schneidet die  $x$ - $z$ -Ebene im Punkt  $T$ . Bestimmen Sie die Koordinaten von  $T$ . Welcher der drei Punkte  $B$ ,  $C$  und  $T$  liegt zwischen den beiden anderen? In welchem Verhältnis teilt der mittlere Punkt die Strecke zwischen den beiden äußeren Punkten?

## 6 Bemerkungen zur Kompatibilität mit verschiedenen MuPAD-Versionen

Dieses Buch wurde für MuPAD Pro 3.1 geschrieben, eine Version von MuPAD, die zum Erscheinungszeitpunkt des Buches noch nicht auf dem Markt verfügbar war. Grund dafür ist die Tatsache, dass ab MuPAD Pro 3.1 alle wesentlichen Funktionen der Grafikbibliothek `plot` auch Matrizen bzw. Vektoren als Eingaben akzeptieren. Das Darstellen mathematischer Objekte – wie etwa die Visualisierung von Punkten, Geraden, Ebenen, Kreisen, Kugeln etc. aus dem Kontext der analytischen Geometrie – ist somit durch direkte Übergabe ihrer mathematischen Repräsentation (im Fall von Geraden und Ebenen z.B. deren Parameterform) an die entsprechenden Routinen der `plot`-Bibliothek möglich.

Mit der zum Erscheinungszeitpunkt des Buches aktuellen Version MuPAD Pro 3.0 können alle Visualisierungen, die in diesem Buch vorgestellt wurden, ebenfalls erstellt werden. Der einzige Unterschied ist, dass Funktionen wie etwa `plot::Surface` in der Version 3.0 noch keine Eingabe von Matrizen bzw. Vektoren erlauben. Stattdessen wird die Eingabe in Form einer Liste erwartet.

In MuPAD Pro 3.0 führt daher die folgende, für dieses Buch typische, Eingabe zu einer Fehlermeldung:

```
• ParameterformF:= matrix([1,0,1]) + k*matrix([1,0,0])
                    + l*matrix([0,1,0]):
F:= plot::Surface(ParameterformF,
                  k = -1..1, l = -1..1):
plot(F)
```

```
Error: First argument: expecting a list of 3 function
expressions/procedures or a procedure returning such
a list [plot::Surface::new]
```

MuPAD Pro 3.0 teilt uns explizit mit, dass `plot::Surface` eine Liste mit drei Einträgen erwartet. Diese Inkompatibilität zwischen den Grafik-Routinen und der mathematischen Repräsentation (hier der Parameterform einer Ebene  $F$ ) wurde für die Version 3.1 vollständig beseitigt, so dass die Visualisierung der mathematischen Objekte direkt über ihre mathematische Darstellung vorgenommen werden kann, ohne diese erst in ein anderes "Format" für die jeweilige Grafik-Routine konvertieren zu müssen. In diesem Buch wurden die ab MuPAD Pro 3.1 gültigen Eingaben bereits verwendet, da sich viele

Befehle dadurch reibungsloser und eleganter in MuPAD lesen bzw. schreiben lassen.

Da MuPAD Pro 3.1 ein so genanntes Service-Update der Firma SciFace Software GmbH & Co. KG ist, kann MuPAD Pro 3.1 kostenfrei von jedem Nutzer verwendet werden, der über eine lizenzierte Version von MuPAD Pro 3.0 verfügt. Die Version MuPAD Pro 3.1 wird nach offiziellen Erscheinen unter der Adresse

<http://www.mupad.de/schule/download>

zum freien Download bereitstehen.

Wer dennoch mit der Version 3.0 arbeiten möchte, sollte wie folgt vorgehen: Wann immer man in MuPAD Pro 3.0 beim Versuch der Visualisierung mathematischer Objekte aus diesem Buch eine Fehlermeldung ähnlich der oben aufgeführten erhält (wo also eine Liste statt einer Matrix-Eingabe gefordert wird), sollte man den jeweiligen Vektor bzw. die jeweilige Matrix in eine Liste konvertieren. Dies kann auf zwei verschiedene Arten geschehen.

Die erste Möglichkeit besteht darin, jede Komponente des Vektors explizit zu extrahieren und diese hintereinander in eine Liste zu schreiben. Dies führt zu einer gültigen Eingabe in MuPAD Pro 3.0. In dem Beispiel von oben liest sich dies dann wie folgt:

```
• F:= plot::Surface([ParameterformF[1],  
                    ParameterformF[2],  
                    ParameterformF[3]],  
                    k = -1..1, l = -1..1):
```

Anschließend kann die Ebene  $F$  problemlos mit Hilfe des Befehls `plot(F)` gezeichnet werden.

Wem diese Schreibweise zu lang ist, der kann auch auf die Funktion `op` zurückgreifen. Wendet man `op` auf einen Vektor (in unserem Fall auf den Vektor `ParameterformF`) an, so sorgt dieser dafür, dass die Komponenten des Vektors automatisch hintereinander geschrieben werden:

```
• ParameterformF;  
  op(ParameterformF)
```

$$\begin{pmatrix} k + 1 \\ l \\ 1 \end{pmatrix}$$

$k + 1, l, 1$

---

Anschließend muss man dann um den Befehl `op(ParameterformF)` nur noch eckige Klammern schreiben, um so die Komponenten des Vektors in eine Liste zu schreiben. Damit führt auch die folgende Definition der Ebene  $F$  in MuPAD Pro 3.0 zum Ziel:

- `F:= plot::Surface([op(ParameterformF)],  
k = -1..1, l = -1..1):`

Auch hier kann  $F$  mittels `plot(F)` anschließend problemlos visualisiert werden.

Die Art und Weise die Ebene  $F$  auf die beiden oben beschriebenen Art und Weisen darzustellen wirkt recht technisch. Aus gutem Grund wurde hier also für die Version 3.1 Abhilfe geschaffen. Computeralgebra-Systeme allgemein und nicht nur MuPAD im speziellen werden ständig weiterentwickelt und an die Belange ihrer Benutzer angepasst. Kleine Inkompatibilitäten zwischen verschiedenen Versionen sind daher nie wirklich ausgeschlossen. In unserem Fall sind die Veränderungen, die den Inhalt dieses Buches betreffen und die sich von der Version 3.0 hin zur Version 3.1 ergeben, ein echter Schritt in die richtige Richtung zur Verbesserung der Benutzungsfreundlichkeit von MuPAD beim Einsatz in der analytischen Geometrie.

## 7 Interaktive Arbeitsblätter zu diesem Buch

Fast das gesamte vorliegende Buch ist in Form einzelner, interaktiver MuPAD-Arbeitsblätter verfügbar. Die folgenden Tabellen listen alle zu diesem Buch verfügbaren Arbeitsblätter auf. In der interaktiven Fassung dieses Buches können die Dateinamen mit der Maus angeklickt und die entsprechenden Arbeitsblätter direkt geöffnet werden. Abgesehen davon können alle hier aufgeführten Arbeitsblätter auf dem Web-Portal des MuPAD-Schule Teams der Universität Paderborn unter der Adresse

<http://schule.mupad.de/material>

kostenlos heruntergeladen werden.

### 7.1 Interaktive Arbeitsblätter zu thematischen Abschnitten

Nahezu alle thematischen Abschnitte dieses Buches sind in Form interaktiver Arbeitsblätter verfügbar (ausgenommen Anmerkungen in den Fußnoten und Animationen). Die folgende Tabelle listet alle Arbeitsblätter in der inhaltlichen Reihenfolge auf.

Name des Arbeitsblattes	Abschnitt
<a href="#">Vektoren-in-der-Ebene.mnb</a>	2.2
<a href="#">Vektoren-im-Raum.mnb</a>	2.3
<a href="#">Laenge-von-Vektoren.mnb</a>	2.4
<a href="#">Abstand-zweier-Punkte.mnb</a>	2.4.1
<a href="#">Standardskalarprodukt-und-Winkel-zwischen-Vektoren.mnb</a>	2.5
<a href="#">Kreuzprodukt-von-Vektoren.mnb</a>	2.6
<a href="#">Erste-elementare-Anwendungen-in-der-Geometrie.mnb</a>	2.7
<a href="#">Beschreibung-von-Geraden.mnb</a>	3.1
<a href="#">Lagebeziehung-von-Geraden-und-Punkten-im-R2-und-R3.mnb</a>	3.2
<a href="#">Lagebeziehung-zweier-Geraden.mnb</a>	3.3
<a href="#">Beschreibung-von-Ebenen.mnb</a>	3.4
<a href="#">Lagebeziehung-von-Ebenen-und-Punkten.mnb</a>	3.5
<a href="#">Lagebeziehung-von-Gerade-und-Ebene.mnb</a>	3.6
<a href="#">Lagebeziehung-zweier-Ebenen.mnb</a>	3.7
<a href="#">Schnittwinkelberechnung.mnb</a>	3.9
<a href="#">Abstand-eines-Punktes-von-einer-Ebene-im-R3.mnb</a>	3.10.1
<a href="#">Abstand-zweier-paralleler-Geraden.mnb</a>	3.10.2
<a href="#">Abstand-windschiefer-Geraden-im-R3.mnb</a>	3.10.3, 3.10.6

Name des Arbeitsblattes	Abschnitt
<a href="#">Abstand-eines-Punktes-von-einer-Ebene-im-R3.mnb</a>	3.10.4
<a href="#">Abstand-einer-Gerade-von-einer-Ebene-im-R3.mnb</a>	3.10.5
<a href="#">Beschreibung-von-Kreisen-im-R2.mnb</a>	4.1
<a href="#">Lagebeziehung-von-Punkten-und-Kreisen-im-R2.mnb</a>	4.4
<a href="#">Lagebeziehung-von-Geraden-und-Kreisen-im-R2.mnb</a>	4.3
<a href="#">Beschreibung-von-Kugeln.mnb</a>	4.4
<a href="#">Lagebeziehung-von-Punkten-und-Kugeln.mnb</a>	4.5
<a href="#">Lagebeziehung-von-Geraden-und-Kugeln.mnb</a>	3.2
<a href="#">Lagebeziehung-von-Ebenen-und-Kugeln.mnb</a>	4.7, 4.7.1
<a href="#">Aufgaben-aus-Abiturpruefungen-Grundkursniveau.mnb</a>	5.1
<a href="#">Aufgaben-aus-Abiturpruefungen-Leistungskursniveau.mnb</a>	5.2

## 7.2 Lösungen zu den Übungsaufgaben

Zu den Übungsaufgaben in diesem Buch existieren Lösungen in Form von interaktiven MuPAD-Arbeitsblättern. Die folgende Tabelle listet alle verfügbaren Arbeitsblätter sowie die zugehörige Aufgabennummer auf.

Name des Arbeitsblattes	Aufgabe
<a href="#">AnalyGeo-Uebung-01.mnb</a>	1
<a href="#">AnalyGeo-Uebung-02.mnb</a>	2
<a href="#">AnalyGeo-Uebung-03.mnb</a>	3
<a href="#">AnalyGeo-Uebung-04.mnb</a>	4
<a href="#">AnalyGeo-Uebung-05.mnb</a>	5
<a href="#">AnalyGeo-Uebung-06.mnb</a>	6
<a href="#">AnalyGeo-Uebung-07.mnb</a>	7
<a href="#">AnalyGeo-Uebung-08.mnb</a>	8
<a href="#">AnalyGeo-Uebung-09.mnb</a>	9
<a href="#">AnalyGeo-Uebung-10.mnb</a>	10
<a href="#">AnalyGeo-Uebung-11.mnb</a>	11
<a href="#">AnalyGeo-Uebung-12.mnb</a>	12
<a href="#">AnalyGeo-Uebung-13.mnb</a>	13
<a href="#">AnalyGeo-Uebung-14.mnb</a>	14
<a href="#">AnalyGeo-Uebung-15.mnb</a>	15
<a href="#">AnalyGeo-Uebung-16.mnb</a>	16
<a href="#">AnalyGeo-Uebung-17.mnb</a>	17
<a href="#">AnalyGeo-Uebung-18.mnb</a>	18

<b>Name des Arbeitsblattes</b>	<b>Aufgabe</b>
<a href="#">AnalyGeo-Uebung-19.mnb</a>	19
<a href="#">AnalyGeo-Uebung-20.mnb</a>	20
<a href="#">AnalyGeo-Uebung-21.mnb</a>	21
<a href="#">AnalyGeo-Uebung-22.mnb</a>	22
<a href="#">AnalyGeo-Uebung-23.mnb</a>	23
<a href="#">AnalyGeo-Uebung-24.mnb</a>	24
<a href="#">AnalyGeo-Uebung-25.mnb</a>	25
<a href="#">AnalyGeo-Uebung-26.mnb</a>	26
<a href="#">AnalyGeo-Uebung-27.mnb</a>	27
<a href="#">AnalyGeo-Uebung-28.mnb</a>	28
<a href="#">AnalyGeo-Uebung-29-30.mnb</a>	29, 30
<a href="#">AnalyGeo-Uebung-31-32.mnb</a>	31, 32
<a href="#">AnalyGeo-Uebung-33.mnb</a>	33
<a href="#">AnalyGeo-Uebung-34.mnb</a>	34
<a href="#">AnalyGeo-Uebung-35.mnb</a>	35
<a href="#">AnalyGeo-Uebung-36.mnb</a>	36
<a href="#">AnalyGeo-Uebung-37.mnb</a>	37
<a href="#">AnalyGeo-Uebung-38.mnb</a>	38
<a href="#">AnalyGeo-Uebung-39.mnb</a>	39
<a href="#">AnalyGeo-Uebung-40.mnb</a>	40
<a href="#">AnalyGeo-Uebung-41.mnb</a>	41
<a href="#">AnalyGeo-Uebung-42.mnb</a>	42
<a href="#">AnalyGeo-Uebung-43.mnb</a>	43
<a href="#">AnalyGeo-Uebung-44.mnb</a>	44
<a href="#">AnalyGeo-Uebung-45.mnb</a>	45
<a href="#">AnalyGeo-Uebung-46.mnb</a>	46
<a href="#">AnalyGeo-Uebung-47.mnb</a>	47
<a href="#">AnalyGeo-Uebung-48.mnb</a>	48
<a href="#">AnalyGeo-Uebung-49.mnb</a>	49
<a href="#">AnalyGeo-Uebung-50.mnb</a>	50
<a href="#">AnalyGeo-Uebung-51.mnb</a>	51
<a href="#">AnalyGeo-Uebung-52.mnb</a>	52
<a href="#">AnalyGeo-Uebung-53.mnb</a>	53
<a href="#">AnalyGeo-Uebung-54.mnb</a>	54
<a href="#">AnalyGeo-Uebung-55.mnb</a>	55
<a href="#">AnalyGeo-Uebung-56.mnb</a>	56
<a href="#">AnalyGeo-Uebung-57.mnb</a>	57
<a href="#">AnalyGeo-Uebung-58.mnb</a>	58
<a href="#">AnalyGeo-Uebung-59.mnb</a>	59

---

<b>Name des Arbeitsblattes</b>	<b>Aufgabe</b>
<a href="#">AnalyGeo-Uebung-60.mnb</a>	60
<a href="#">AnalyGeo-Uebung-61.mnb</a>	61
<a href="#">AnalyGeo-Uebung-62.mnb</a>	62
<a href="#">AnalyGeo-Uebung-63.mnb</a>	63
<a href="#">AnalyGeo-Uebung-64.mnb</a>	64
<a href="#">AnalyGeo-Uebung-65.mnb</a>	65

## Literaturverzeichnis

- [1] Creutzig, Gehrs, Oevel: *Das MuPAD Tutorium*, Springer, dritte Auflage, 2004
- [2] Eckart, Jehle, Vogel: *Analytische Geometrie*, BSV Mathematik, 1994
- [3] Gehrs, Postel: *MuPAD – eine Praktische Einführung, Mathematik 1 × anders – Materialien und Werkzeuge für computerunterstütztes Lernen*, Band 1, SciFace Software GmbH & Co. KG, dritte Auflage, 2004
- [4] Gehrs: *MuPAD Pro 3.0 - Installation und erste Schritte*, SciFace Software GmbH & Co. KG, 2004
- [5] Gehrs: *Geraden, Ebenen und Kugeln mithilfe des Programms MuPAD*, Unterrichtsmaterialien – Analytische Geometrie, Loseblattsammlung 707E-03, Beitrag H.53, STARK-Verlag, 2004
- [6] Gehrs: *Gerade und Kreis mithilfe des Programms MuPAD*, Unterrichtsmaterialien – Analytische Geometrie, Loseblattsammlung 707E-03, Beitrag O.29, STARK-Verlag, 2004
- [7] Grabinger: *Projekte und Aufgaben zur Analytischen Geometrie*, Schroedel, 1999
- [8] Heugl, Klinger, Lechner: *Mathematikunterricht mit Computeralgebra-Systemen*, Addison-Wesley, 1996.
- [9] Kroll, Reifert, Vaupel: *Analytische Geometrie / Lineare Algebra – Grund- und Leistungskurs*, Dümmler, 1997
- [10] Lindner: *Lineare Transformationen mit MuPAD, Mathematik 1 × anders – Materialien und Werkzeuge für computerunterstütztes Lernen*, Band 6, SciFace Software GmbH & Co. KG, zweite Auflage, 2004
- [11] Lindner: *Ausgleichsrechnung, überbestimmte LGS und Pseudoinverse mit MuPAD, Mathematik 1 × anders – Materialien und Werkzeuge für computerunterstütztes Lernen*, Band 7, SciFace Software GmbH & Co. KG, zweite Auflage, 2004

## Index

- \$, 44
- assume, 38, 109, 135, 155, 169, 172, 182
- bool, 19, 81
- Color, 18
- expand, 22, 134, 149
- FillColor, 152, 175
- Filled = TRUE, 175
- float, 41, 101, 103
- has, 134
- IgnoreSpecialCases, 45
- in, 44
- LegendText, 146
- LegendVisible, 146
- lhs, 134
- linalg, 38, 41
  - angle, 41, 49, 101, 103, 104, 180
  - crossProduct, 46, 47, 77, 104, 105, 120, 123, 161, 172, 187
  - scalarProduct, 38, 49, 56, 77, 109, 169, 172
- LineWidth, 57, 179
- matrix, 13
- norm, 31, 50, 110, 162, 173
- normal, 40, 46
- PI, 41
- plot, 12, 13, 175, 193
  - Arrow2d, 14
  - Arrow3d, 25, 109, 186
  - Circle2d, 130, 133, 145
  - Curve2d, 54, 55, 72
  - Curve3d, 53, 54, 72, 108
  - Function2d, 55, 57
  - Implicit2d, 136, 137
  - Implicit3d, 77, 149, 183
  - Line2d, 30
  - Line3d, 33, 48, 53, 174, 179
  - Parallelogram3d, 77
  - Point2d, 13, 54
  - Point3d, 24, 53, 74, 108
  - Polygon3d, 175
  - Sphere, 130, 149
  - Surface, 72, 77, 85, 183, 193
- PointSize, 84, 144
- RGB
  - Black, 109, 173
  - Blue, 24, 173
  - Green, 24, 109, 173
  - Grey, 152
  - LightYellow, 175
  - Red, 18, 173, 179
- rhs, 134
- Scaling = Constrained, 48, 109, 117, 121, 136, 150, 174, 177
- select, 134
- solve, 43, 60, 64, 80, 85, 87, 96, 109, 113, 116, 120, 135, 141, 155
- sqrt, 34, 152, 168
- subs, 44, 60, 67, 81, 109, 113, 138, 141
- Type::Real, 155, 169, 172, 182
- ViewingBox, 24



# Mathematik 1 x anders

## Band 10

*MuPAD* ist ein Computeralgebra-System, mit dem Problemstellungen der Mathematik sowie der Natur- und Ingenieurwissenschaften behandelt werden können. Es ist ein wertvolles Hilfsmittel für Schüler und Studenten, Lehrer und Wissenschaftler.

Dieses Buch richtet sich sowohl an Lehrkräfte als auch ausdrücklich an Schüler und Studierende. Es ist als Begleitbuch zu den üblichen Standardlehrbüchern gedacht, die im Unterricht der Sekundarstufe II zur Vermittlung der Inhalte der Analytischen Geometrie verwendet werden. Neben einer knappen Einführung in die Vektorrechnung werden in den folgenden Kapiteln die schulrelevanten Probleme, wie etwa die Bestimmung von Lagebeziehungen von Punkten, Geraden, Ebenen und Kugeln zueinander, Abstandsberechnungen sowie die Bestimmung von Schnittwinkeln mit MuPAD behandelt.

Besonderer Wert wird auf die Visualisierung dieser Objekte gelegt, so dass ein anschaulicher Zugang zur jeweiligen geometrischen Situation gegeben ist.

Mit der interaktiven PDF-Fassung dieses Buches sind darüber hinaus etwa 30 interaktive MuPAD Arbeitsblätter sowie über 150 Grafiken und Animationen per Mausklick verfügbar.

[www.sciface.com](http://www.sciface.com)

[schule.mupad.de](http://schule.mupad.de) / [schule@mupad.de](mailto:schule@mupad.de)

SciFace Software • Paderborn

**Sci**Face