

Kurs2: Pinball Wizard

Das Ziel des Kurses

Wir hatten uns ein ehrgeiziges Ziel gesteckt: Die Erstellung eines Programmes, das die Flugbahnen und Zusammenstöße von Kugeln in einem Flipperautomaten mit Hilfe von professionellen Programmierwerkzeugen naturgetreu nachrechnet und als graphische Animation in Echtzeit, also während die Berechnung läuft, auf dem Bildschirm ausgibt.

Vollkommen erreicht haben wir dieses Ziel leider nicht. Und so ist unser „Flipperautomat“ recht vollständig bis auf das Manko, keine Flipperriegel zu besitzen, mit denen ein virtueller Spielhallenbesucher interaktiv die Flugbahnen der Kugeln beeinflussen könnte.

Nichtsdestotrotz sind wir auf dem spannenden Weg zu unserer kleinen Simulation vielen mathematischen, physikalischen und informatischen Konzepten begegnet und haben uns mit ihnen vertraut gemacht. Wir haben dabei auch alle wesentlichen Kenntnisse erworben, mit denen jeder Teilnehmer nun für sich selbst das Ausgangsziel bezwingen kann.



Momsen Reincke ist 15 Jahre alt und kommt aus Freiburg. Seine wahre Größe zeigt er beim Skat, niemand hat gegen ihn eine Chance. Seine weitere Freizeit verbringt er oft am Computer. „Ich bin der Momsen, klingt komisch ist aber so! “

Der in Herrenberg wohnende **Meru Alagalingam** ist 12 Jahre alt und vertreibt sich seine Zeit gerne mit Modellbau. Die Mathematik versteht er besser als das Skatspielen.

Natascha Schiel aus Gernsbach-Lautenbach ist 14 Jahre alt und ihre Hobbys sind Leichtathletik, Klavierspielen und Lesen. Den Kurs heiterte sie durch ihr Gekicher auf. In Adelsheim hat sie das Skatspielen für sich neu entdeckt.

Jascha Fahr kommt aus Offenburg, wo er seine meiste Zeit im Schwimmbad verbringt. Die anderen

finden den 14-jährigen Jungen sehr sympathisch. In seiner restlichen Freizeit spielt er gerne Basketball und Tischtennis.

Manuel Lieb spielt Billard im Verein, er ist 15 Jahre alt und kommt aus Ehingen. In seinem Heimatort betreibt er eines seiner Hobbys, Leichtathletik.

Rainer Mühlhoff, einer unserer Leiter, lernte Skat zu lieben und hat schon einen festen Platz in der abendlichen Skatrunde gefunden. Zur Zeit studiert er in Heidelberg Mathe und Physik. Auf alle Fragen über Mathematik findet unser Rainer eine Antwort. „Ist's wahr? - Ja es ist wahr.“

Andreas Potschka, unser Sunnyboy, weiß nicht nur in Mathe bestens Bescheid, sondern singt auch wie ein echter Profi. In der internen Academyband, konnte er sich mit Rainer zusammen so richtig entfalten.

Die geregelte Planung unseres Kurses hatte er immer gut im Griff.

Der kleingewachsene Schwabe **Karl Christ** aus Schorndorf ist immer gut drauf und erlaubt sich gerne einen Spaß mit Rainer. Er ist 14 Jahre alt.

Unser leidenschaftlicher Basketballer **Simon Jacobi** ist 15 Jahre alt und kommt aus Kirchentellinsfurt. Er ist stets ein sehr netter und hilfsbereiter Junge. Den Sieg unseres Kurses beim Basketballturnier haben wir größtenteils ihm zu verdanken.

Die Leidenschaften der ruhigen 15-jährigen **Tina Schmidt** sind Schwimmen, Lesen und Klavierspielen. Sie wohnt in Kornwestheim.

Fabian Ruf hat Rainer besonders lieb gewonnen. Er ist 14 Jahre alt und kam den weiten Weg aus Remchingen. Die Pinball-Simulation hatte es ihm besonders angetan.

Barbara Schimmele ist 14 Jahre alt und kommt aus der Goldstadt Pforzheim. Eines ihrer Hobbys ist Hockeyspielen. In Adelsheim nahm sie eifrig an der nächtlichen Skatrunde teil.

Matthias Rapp, 16 Jahre alt, aus Eichstegen kennt sich beim Programmieren bestens aus und ist beim Wasserball immer voll dabei.

Ferdinand Macher aus Überlingen, intern genannt Nando, war so fleißig im Dokumentieren, dass er nach kurzer Zeit nichts mehr zu tun hatte und seine Zeit mit Uga-Agga Spielen vertrieb. Im Kurs verhielt er sich stets ruhig und kippelte unaufhaltsam. Beim Konzert begeisterte er uns alle mit seinem Klaviervorspiel.

Atmosphäre im Kurs

Ein Außenstehender würde von einem Kurs voller Mathematikern, Physikern und Programmierern wohl eher eine trockene Atmosphäre erwarten. Und obwohl „Pinball Wizard“ doch zu einem nicht unbedeutenden Teil von ellenlangen Formeln, physikalischen Prinzipien und anderen für

Normalsterbliche unverständlichen Dingen geprägt war, gab es keinen Teilnehmer, der nicht mit Spaß und Eifer dabei war. Wir kamen alle gut miteinander aus, was nicht zuletzt auch an unserem gemeinsamen Interessengebiet lag. Auch die Beziehung zwischen den Leitern und uns kann man nur als freundschaftlich beschreiben.

Falls sich doch einmal nach stundenlanger Arbeit Müdigkeit, Erschöpfung und folglich auch steigende Lustlosigkeit unter den Teilnehmern breit zu machen schien, waren Andreas und Rainer keiner Lösung schuldig: Mit (zumindest bis zur zweiten oder dritten Wiederholung) mehr oder weniger ansprechenden und aufmunternden Klatschspielen, sowohl gemeinsam, als auch in bis zu vierstimmigem Canon, konnten sie die Truppe wieder auf Trab bringen.

Zeitweise wurden auch andere Kurse durch unsere Abwesenheit beim gemeinsamen Essen in Verwunderung versetzt: Wir hatten einfach keine Lust aufzuhören, und bis zur Problemlösung ca. 1 1/2 Stunden später konnte man uns außerhalb unseres etwas abgelegenen Kursraum nicht erblicken.

Zusammenfassend kann man nur sagen: An die Lernatmosphäre in der Akademie werden wir uns in so mancher langweiligen Schulstunde wehmütig erinnern.

Was ist Simulation?

Simulation leitet sich von dem lateinischen Wort „simulare“ ab, was soviel bedeutet wie „vortäuschen“. Der Begriff „Vortäuschung“ ist in der Tat auch für die Programme treffend, die im Rahmen unseres Kurses von den Teilnehmern entwickelt worden sind. Jedes dieser Programme versucht, einen Teilausschnitt der Realität (die Bewegung der Kugeln eines Flipperautomaten) nachzubilden. Eine Simulation leitet sich also aus der echten Realität ab und erzeugt eine sogenannte *virtuelle* Realität.

Die Entwicklung einer solchen Simulation ist kein einmaliges Unterfangen, sondern ein *iterativer* Prozess. Das bedeutet, dass das Simulationsprogramm nicht nach einer langen Programmierphase einmal getestet wird und dann funktionsfertig ist, sondern dass immer wieder kleine Zwischentests die Programmierphase unterbrechen, in denen der Entwickler die erzeugte virtuelle Realität mit der echten Realität vergleicht und aus diesem Vergleich Konsequenzen für die Weiterentwicklung des Programmes zieht. Diese Konsequenzen können Nachbesserungen in den mathematisch-physikalischen Formeln oder in der Art sein, wie die Formeln auf dem Computer ausgewertet werden. Im Allgemeinen nennt man die Zusammenfassung des zu simulierenden Teilgebietes der Realität in einen Satz von Formeln die *Modellierung* und die Berechnung dieser Formeln auf dem Computer die *Diskretisierung* einer Simulation.

Der Schritt der Diskretisierung rührt daher, dass der Modellierungsprozess in beinahe allen Fällen Formelsysteme entstehen lässt, die sich als unendlichdimensionale Probleme herausstellen, wenn man sie lösen will. Da Computer von ihrer Natur her sehr begrenzt und endlich sind (endlicher Speicher, endliche Rechengeschwindigkeit), müssen diese Modelle erst in das enge Kostüm der Endlichkeit gepresst werden, in welchem sie dann ein Rechner lösen kann. Die Fehler, die man bei der Modellierung durch Vernachlässigung gewisser Parameter (z. B. die Anziehungskraft des Planeten Jupiter auf eine Flipperkugel) und bei der Diskretisierung durch den Abstieg ins Endliche macht, sind die Gründe für den Unterschied der virtuellen Realität von der Wirklichkeit.

Der Kreislauf des Testens und Verfeinerns von Modellierung und Diskretisierung wird gebrochen, sobald sich der Entwickler der Simulation mit seiner erschaffenen virtuellen Wirklichkeit zufrieden gibt.

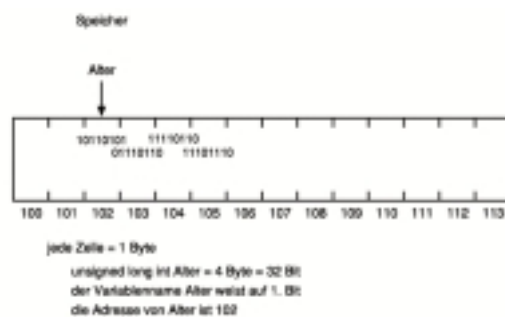
Diesen Kreislauf haben wir selbst in Adelsheim mitgemacht, wobei wir hauptsächlich an der Modellierung Verbesserungen vorgenommen haben, um unsere Simulationen zu erweitern. Zuerst ließen wir nur eine Kugel auf einer ebenen Platte rollen, im Anschluss simulierten wir eine schiefe Ebene, dann ließen wir die Kugeln von beliebigen Hindernissen abprallen und schließlich bewirkten wir, dass bei den Simulationen mit mehreren Kugeln keine durch eine andere hindurch rollen konnte. So konnten wir ein wenig

authentische Luft der Simulationsentwicklung schnuppern.

Modellierung und Diskretisierung in unserem Simulationsprogramm.

Die Modellierung bestand in unserem Fall aus den Formeln für gleichförmige und gleichmäßig beschleunigte Bewegungen, den drei Newton'schen Axiomen, den Sätzen der Impuls- und Energieerhaltung, sowie den Formeln für Kollisionserkennung und -behandlung. Diese haben wir in der Sprache der Euklidischen Ebene formuliert.

Bei der Diskretisierung stießen wir auf die unendlichdimensionale Problematik der Bewegungsvorgänge. Wir mussten, um diese auf den Computer zu übertragen, die ablaufenden Vorgänge auf endliche Art und Weise beschreiben.



Dies geschah in Form einer schrittweisen Berechnung. Wie bei einem Comicfilm, wo normalerweise 24 Bilder in der Sekunde abgespielt

werden, berechnet der Computer fortlaufend neue Bilder: bei jedem Schritt werden die neuen Positionen der Kugeln errechnet und auf dem Bildschirm grafisch visualisiert. Die dadurch entstehenden Fehler müssen wohl oder übel in Kauf genommen werden.

Programmierung

Unser Kursziel ist es eine Pinball-Simulation zu schreiben. Wir benötigen eine Programmiersprache um den Computer zu sagen, was er tun soll.

Wir wählen die Programmiersprache C, aus folgenden Gründen:

Effizienz:

Mit C können kleinere, schnellere Programme erstellt werden als mit den meisten anderen Programmiersprachen.

Graphische Programmierung:

Für C gibt es sehr starke Funktionen und Möglichkeiten zu graphischer Programmierung (OpenGL Bibliotheken), die C auch für Spieleprogrammierer interessant machen.

In der Spanne vom Vorbereitungswochenende bis zur Science-Academy im Sommer lernten wir über sogenannte C-Missionen, wie man in C programmiert. Bei den C-Missionen handelte es sich um Lern- und Aufgabenblätter von unseren Kursleitern. Diese C-Missionen wurden uns alle zwei bis vier Wochen (je nach Bearbeitungszeit der vorherigen Mission) zugeschickt. Dann bekamen wir

bis zu einem vorher festgelegten Termin Zeit für die Lösung dieser Aufgaben. Außerdem mussten wir eine Dokumentation über unsere Lösungen verfassen und diese mit der Lösung zusammen über einer Mailingliste an alle anderen Kursteilnehmer per Email schicken und diskutieren. In den ersten C-Missionen ging es im Wesentlichen um die Grundlagen der Programmierung (Variablen, Algorithmen, Bibliotheken, etc.).

Die einzelnen Missionen:

1. Ein Programm zur Berechnung des Alters in Tagen.
2. Ein Programm zur Berechnung von Pi.
3. Ein Programm, das ein Labyrinth und den dazugehörigen Lösungsweg graphisch darstellt.
4. Ein Programm, das einen Billardtisch darstellt, auf dem mehrere Kugeln sich mit konstanter Geschwindigkeit und ohne Reibung auf dem Billardtisch bewegen und an den Wänden abprallen.

Für die Programmierung der Simulation mussten wir uns noch während des Kurses in Adelsheim ein wichtiges Werkzeug von C aneignen: Zeiger. Zeiger sind Variablen, die die Adresse anderer Variablen aufnehmen können. Wenn wir von unserer Struktur Vektor eine Kopie **A** und eine Kopie **B** erzeugen und die Werte der Kopie **A** ändern, wirkt es sich nicht auf die Kopie **B** aus. Bei Benutzung von Zeigern wirkt sich die Änderung auch auf alle anderen Kopien der Struktur aus (in unserem Beispiel die Kopie **B**). Um die Funktionsweise von Zeigern zu verstehen, muss

man sich erst einmal den Speicher in Computern anschauen. Der Speicher (RAM) ist aus Sicht eines Programmierers eine lineare Ansammlung von Bits: Ein Bit ist die kleinste Informationseinheit. Ein Bit kann man darstellen als Ja/Nein, 1/0, Spannung/keine Spannung. Acht Bit werden zu einem Byte zusammengefaßt. Die Bytes im Speicher werden durchnummeriert. Die „Nummer“ einer Speicherzelle (1 Byte) heißt Speicheradresse. Hier nun ein kleines Programmbeispiel für Zeiger:

```
int a = 47;
int *ipa = &a;
/* es wurde nun eine Variable a
definiert und ein Zeiger auf a
definiert und initialisiert*/

*ipa = 100;
/* Da die Variable ipa auf die Adresse
von der Variable a zeigt, wird durch
eine Neuzuweisung der Wert der
Variablen a und der Zeigervariable ipa
beide auf den Wert in unserem Beispiel
auf 100 gesetzt*/
```

Aufbau unseres Pinball-Programmes:

Unsere Simulation beinhaltet im wesentlichen folgende Funktionen:

Bewegung:

Berechnung eines neuen Bildes, anhand der Koordinaten und Geschwindigkeiten der Kugeln.

Kollisionserkennung:

Die Kollisionserkennung besteht aus `if` (wenn) Anweisungen in denen überprüft wird, ob eine Kugel - Kugel oder eine Kugel - Linien Kollision stattfinden wird.

Kollisionsbehandlung:

Wenn die Kollisionserkennung eine Kollision erkannt hat, ruft sie die Funktion der Kollisionsbehandlung auf. In dieser Funktion werden die neuen Kugelgeschwindigkeiten berechnet und zurückgegeben.

Zeichnen:

Ausgabe des neuen Bildes auf dem Bildschirm. Dies macht der Computer indem er das alte Bild löscht und durch das aktuelle ersetzt.

Leider wurden wir in dem Kurs nicht ganz fertig mit der Simulation, was heißen soll, das man noch nicht über Eingaben mit der Maus oder Tastatur die Kugeln irgendwie beeinflussen kann.

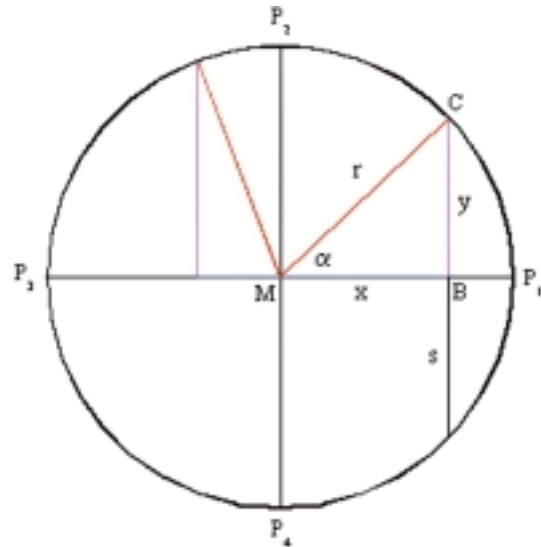


Abb. 1

Winkelfunktionen

MBC : rechtwinkliges Dreieck

s : Sehne des Kreises

x : Ankathete von MBC

y : Gegenkathete von MBC

r : Radius = Hypotenuse von MBC

α : Winkel am Mittelpunkt

Das rechtwinklige Dreieck im Kreis

Man betrachte das rechtwinklige Dreieck MBC in der Abbildung.

Man stelle sich nun vor, wie der Punkt C des Dreiecks MBC entlang des Kreisbogens vom Punkt P_1 zum Punkt P_2 „wandert“. Dabei werden alle möglichen Dreiecke durchlaufen. x verändert sich dabei vom maximalen Wert ($= r$) bis 0 und y von 0 zum maximalen Wert ($= r$)(siehe Abb. 1).

Auf ähnliche Weise kann man sich überlegen, wie sich x und y ändern, wenn C von P_2 nach P_3 , von P_3 nach P_4 oder von P_4 nach P_1 wandert.

Das Verhältnis der beiden Katheten bei gegebenen Winkel α ist also vollständig bestimmt, man spricht deshalb von funktionaler Abhängigkeit von dem Winkel α oder kurz von einer Funktion. Diese Funktionen heißen Winkelfunktionen, es gibt jedoch drei von ihnen, denn prinzipiell gibt es drei Verhältnisse im Dreieck:

$$\frac{y}{r}, \frac{x}{r}, \frac{x}{y}$$

Bezogen auf den Winkel α heißen diese Verhältnisse:

$$\sin \alpha = \frac{y}{r} \text{ (Sinus)}$$

$$\cos \alpha = \frac{x}{r} \text{ (Cosinus)}$$

$$\tan \alpha = \frac{y}{x} \text{ (Tangens)}$$

In dem rechtwinkligen Dreieck MBC ist r die Hypotenuse, x die Ankathete und y die Gegenkathete bezüglich des Winkels α (siehe Abb. 2).

Deswegen können wir für rechtwinklige Dreiecke allgemein auch schreiben:

$$\sin \alpha = \frac{\text{Gegenkathete}}{\text{Hypotenuse}}$$

$$\cos \alpha = \frac{\text{Ankathete}}{\text{Hypotenuse}}$$

$$\tan \alpha = \frac{\text{Gegenkathete}}{\text{Ankathete}}$$

Es macht auch keinen Sinn eine Winkelfunktion zu finden, mit der man alleine aus einem Winkel, unter Anwendung einer Funktion, eine Seitenlänge erhält. Das gilt schon bei nur ähnlichen Dreiecken: Der Funktionswert der Winkelfunktion wird in beiden ähnlichen Dreiecken gleich sein. Die Seitenlänge mit Sicherheit nicht. Das Verhältnis zweier Seiten ist jedoch in ähnlichen Dreiecken in ähnlichen Dreiecken gleich. Das folgt aus den Strahlensätzen.

Am Einheitskreis hat der Radius und damit auch die Hypotenuse des Dreiecks MBC die Länge 1 und deswegen gilt hier:

$$\sin \alpha = \frac{\text{Gegenkathete}}{1}$$

und

$$\cos \alpha = \frac{\text{Ankathete}}{1}$$

Ankathete und Gegenkathete sind dabei die Koordinaten des Punktes C .

Der Graph der Sinus-Funktion sieht so aus:

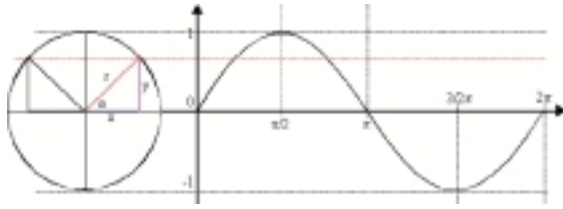


Abb. 2

Die euklidische Ebene als reeller Vektorraum

Motivation

Für das Kursziel, die Programmierung eines Pinball-Programms benötigen wir ein geeignetes mathematisches Modell, um eine Grundlage für eine physikalische Theorie zu haben. Hierfür ziehen wir die Vektorrechnung heran und führen zunächst die grundlegende mathematische Struktur des reellen Vektorraums ein, die auch in anderen Gebieten der Mathematik ihre Anwendung findet.

Der reelle Vektorraum

Naiv gesprochen ist eine reelle Zahl eine Dezimalzahl, die in ihrer Darstellung entweder abbricht, periodisch ist oder weder abbricht noch eine Periode hat (dies ist dann eine sogenannte irrationale Zahl). Die Menge all dieser Zahlen wird die Menge \mathfrak{R} genannt.

Ein reeller Vektorraum ist eine Menge V mit einer inneren Verknüpfung $+: V \times V \rightarrow V$ (genannt Addition) und einer äußeren Verknüpfung

$\cdot: \mathfrak{R} \times V \rightarrow V$ (genannt skalare Multiplikation). Im Fall des reellen Vektorraums nennen wir auch \mathfrak{R} die Menge der Skalare. Für eine Menge K bedeutet $K \times K \rightarrow K$, dass zwei Elemente aus K verknüpft werden und das Ergebnis wieder ein Element der Menge K ist. Für die Addition und die Multiplikation soll Folgendes gelten:

1. $\vec{v} + \vec{w} = \vec{w} + \vec{v}$ für alle $\vec{v}, \vec{w} \in V$ (Kommutativität der Addition)
2. Es gibt ein $\vec{0} \in V$ mit $\vec{0} + \vec{v} = \vec{v}$ für alle $\vec{v} \in V$ (Existenz des Nullelements bezüglich der Vektoraddition)
3. $\vec{v} + (\vec{w} + \vec{u}) = (\vec{v} + \vec{w}) + \vec{u}$ für alle $\vec{v}, \vec{w}, \vec{u} \in V$ (Assoziativität der Addition)
4. Für jedes $\vec{v} \in V$ gibt es ein $-\vec{v} \in V$ mit $\vec{v} + (-\vec{v}) = \vec{0}$ (Existenz des Inversen)
5. a) $(a+b) \cdot \vec{v} = a \cdot \vec{v} + b \cdot \vec{v}$ für alle $a, b \in \mathfrak{R}$ und $\vec{v} \in V$ (Distributivität)
b) $a \cdot (\vec{v} + \vec{w}) = a \cdot \vec{v} + a \cdot \vec{w}$ für alle $a \in \mathfrak{R}$ und alle $\vec{v}, \vec{w} \in V$ (Distributivität)
6. $(ab) \cdot \vec{v} = a \cdot (b \cdot \vec{v})$ für alle $a, b \in \mathfrak{R}$ und alle $\vec{v} \in V$
7. $1 \cdot \vec{v} = \vec{v}$ für alle $\vec{v} \in V$

Elemente eines Vektorraums nennen wir Vektoren.

Die Definition eines Vektorraums ist im ersten Moment sehr abstrakt, man wird jedoch in den nächsten Abschnitten sehen, dass sie sehr hilfreich und sinnvoll ist.

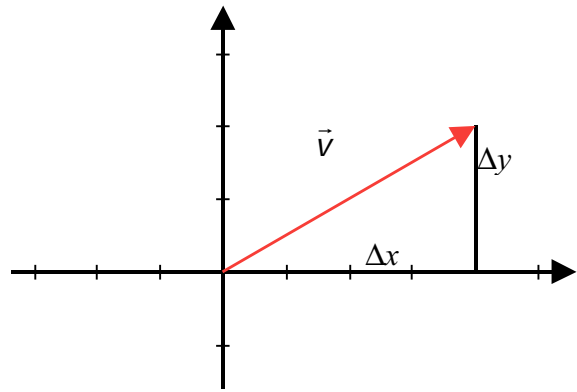
Die Menge \mathfrak{R}^2

Nun bilden wir das kartesische Produkt von \mathfrak{R} mit sich selber, $\mathfrak{R} \times \mathfrak{R} = \mathfrak{R}^2$. Unter dem kartesischem Produkt zweier Mengen K und L versteht man die Menge $K \times L = \{(x, y) \mid x \in K \wedge y \in L\}$

der geordneten Paare von Elementen aus K und L , d.h. man bildet eine neue Menge, deren Elemente jeweils aus einem Element der Menge K und einem Element der Menge L bestehen.

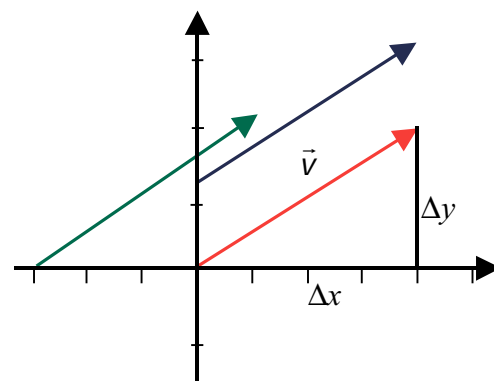
Koordinatensysteme, Pfeile und Vektoren

Es wird nun das aus der Schule bekannte kartesische Koordinatensystem verwendet. Jeder Punkt in dieser Ebene lässt sich als Paar zweier reellen Zahlen auffassen, seiner x- und seiner y-Koordinate. Deshalb kann man \mathfrak{R}^2 als die Menge all dieser Punkte in dem Koordinatensystem ansehen. Diese nennt man die Euklidische Ebene. Es wird nun in dieses Koordinatensystem ein Pfeil eingezeichnet:

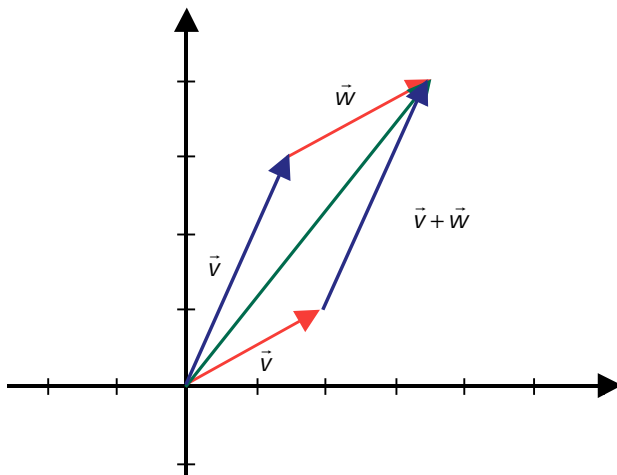


Wir überlegen uns nun, was diesen Pfeil charakterisiert: Er hat eine Länge und eine Richtung. Diese lässt sich präzise durch Δx und Δy ausdrücken. Dies nennen wir jetzt die x- und die y-Komponenten des Pfeiles. Wir schreiben jetzt also:

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \text{ mit } v_1 = \Delta x \text{ und } v_2 = \Delta y.$$



Leider ist dieser Pfeil jedoch noch nicht eindeutig beschrieben: Sowohl der blaue als auch der rote Pfeil lassen sich mit den gleichen Komponenten beschreiben. Doch daran wollen wir nichts ändern, man wird später sogar z.B. bei der Vektoraddition sehen, dass dies recht nützlich sein kann. Ab jetzt wollen wir aus der Euklidischen Ebene einen Vektorraum machen. Als Vektor der Euklidischen Ebene nehmen wir die Menge aller Pfeile mit denselben Komponenten.

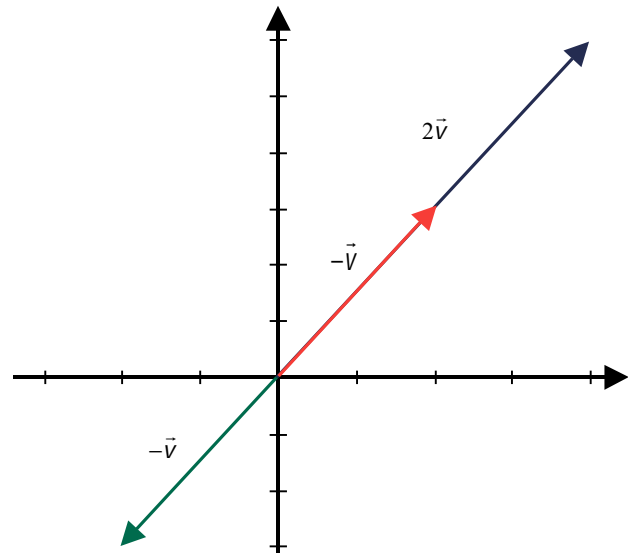


Addition und skalare Multiplikation

Jetzt legen wir eine Addition fest: Zwei Vektoren werden addiert, indem man die sie repräsentierenden Pfeile in der Ebene aneinander hängt.

In der Komponentendarstellung wird addiert, indem man einfach komponentenweise addiert. Die geometrische und die rechnerische Addition sind dieselbe Addition, wie man sich leicht in der Ebene

veranschaulichen kann. Der Nullvektor (Axiom 2) ist der Vektor $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Durch die Zurückführung dieser Addition von Vektoren auf die Addition der reellen Zahlen sind Axiome 1, 2, 3 und 4 leicht zu beweisen.



Die skalare Multiplikation eines Vektors $\vec{v} \in \mathfrak{R}^2$ mit einem Skalar $a \in \mathfrak{R}$ wird in der Ebene als Strecken bzw. Stauchen des Vektors um den Faktor a definiert (zentrische Streckung). Die skalare Multiplikation wird berechnet, indem man die Komponenten mit dem Skalar multipliziert.

Rechnerisch sieht das so aus:

$$a \cdot \vec{v} = \begin{pmatrix} av_1 \\ av_2 \end{pmatrix}$$

Bei der skalaren Multiplikation ändert sich im Gegensatz zur Vektoraddition die Richtung *nicht*, nur die Länge (diese wird später nicht nur anschaulich sondern auch rechnerisch mittels der euklidischen Norm definiert). Nun lässt sich leicht zeigen, dass die Axiome 5, 6 und 7 gelten, da wir auch die skalare Multiplikation auf eine Multiplikation reeller Zahlen zurückgeführt haben.

Linearkombinationen von Vektoren

In einem reellen Vektorraum V nennen wir einen Vektor $\vec{v} \in V$ eine Linearkombination der Vektoren $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n \in V$ wenn es Skalare $a_1, a_2, \dots, a_n \in \mathbb{R}$ so gibt, dass

$$\vec{v} = a_1 \cdot \vec{v}_1 + a_2 \cdot \vec{v}_2 + \dots + a_n \cdot \vec{v}_n .$$

Wir nennen Vektoren $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n \in V$ linear unabhängig, wenn aus

$$a_1 \cdot \vec{v}_1 + a_2 \cdot \vec{v}_2 + \dots + a_n \cdot \vec{v}_n = \vec{0} \text{ folgt:}$$

$$\forall a_i \ 1 \leq i \leq n \ a_i = 0$$

Basis und Dimension eines Vektorraums

Wir nennen eine Menge B von Vektoren aus V eine Basis eines Vektorraums V , wenn sich jeder Vektor $\vec{v} \in V$ als eine Linearkombination von Vektoren b

aus B darstellen lässt. Außerdem müssen die Vektoren aus B jeweils linear unabhängig sein. Man kann beweisen, dass jeder Vektorraum solch eine Basis besitzt. Die Kardinalität der Basis nennen wir die Dimension des Vektorraums. Man kann zeigen, dass diese eindeutig ist. Es gibt interessanterweise sogar unendlich-dimensionale Vektorräume.

Die Standardbasis des \mathbb{R}^2

Jeder Vektor $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in \mathbb{R}^2$ lässt sich als

Linearkombination $\vec{v} = v_1 \cdot e_1 + v_2 \cdot e_2$ mit

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ und } \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ darstellen.}$$

Also ist $\{e_1, e_2\}$ eine Basis des \mathbb{R}^2 , wir nennen sie die „Standardbasis des \mathbb{R}^2 “.

Ergebnis:

Wir sehen, dass die Menge \mathbb{R}^2 der Punkte in unserem Koordinatensystem zusammen mit der Vektoraddition (dem „Aneinanderhängen“) und der skalaren Multiplikation („dem Stauchen und Strecken“) einen Vektorraum bildet, d.h. die Axiome eines Vektorraums erfüllen. Diesen nennen wir den Vektorraum der Euklidischen Ebene.

Die euklidische Norm

Wenn man eine Strecke und einen Vektor vergleichen will, dann hat man das Problem, dass der Vektor und die Strecke zu verschiedenen Mengen gehören. Der Vektor ist ein Element des Vektorraumes \mathfrak{R}^2 und die Strecke ist eine bestimmte Untermenge der Menge \mathfrak{R}^2 .

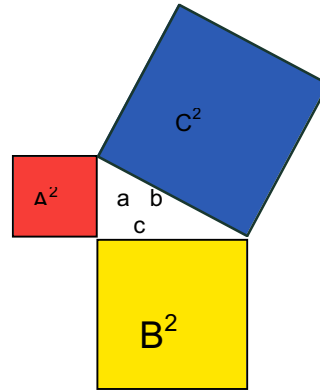
Um dieses Problem zu lösen gibt es die euklidische Norm. Die Norm ist eine Abbildung, die den Vektor auf eine reelle Zahl, das heißt seine Länge, abbildet.

Jeder Vektor in $\vec{v} \in \mathfrak{R}^2$ lässt sich in zwei Komponenten zerlegen. Diese Komponenten lassen sich als senkrecht aufeinander stehende Strecken im Koordinatensystem darstellen.

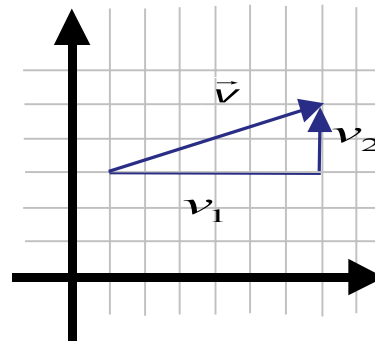
Da diese beiden Strecken senkrecht aufeinander stehen, kann man mit Hilfe des Satzes von Pythagoras die Länge des Vektors ausrechnen:

Einschub: Der Satz des Pythagoras

Der Satz des Pythagoras besagt, dass in einem rechtwinkligen Dreieck die Summe der Quadrate über den Katheten dem Quadrat über der Hypotenuse entspricht.



$$a^2 + b^2 = c^2$$



In diesem Dreieck sind a und b die Katheten, c die Hypotenuse, deshalb gilt: **Vektorzerlegung:**

Zerlegung von \vec{v} in v_1 und v_2

Der Satz des Pythagoras geht in die euklidische Norm ein.

Definition: Euklidische Norm

Sei $\vec{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in \mathfrak{R}^2$, dann heißt die Abbildung

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \mapsto \sqrt{a_1^2 + a_2^2}$$

die euklidische Norm.

Die formale Definition der allgemeinen Norm ist:

Definition: Norm

Sei V ein reeler Vektorraum.

Eine Norm ist eine Abbildung

$$\|\cdot\|: V \rightarrow \mathfrak{R},$$

die den folgenden Forderungen genügt:

$$\|\vec{v}\| = 0 \text{ genau dann, wenn } \vec{v} = \vec{0} \text{ für alle } \vec{v} \in V$$

$$\|\lambda \vec{v}\| = |\lambda| \|\vec{v}\| \text{ für alle } \lambda \in \mathfrak{R}, \vec{v} \in V$$

$$\|\vec{v} + \vec{w}\| \leq \|\vec{v}\| + \|\vec{w}\| \text{ für alle } \vec{v}, \vec{w} \in V$$

Im Folgenden soll gezeigt werden, dass die euklidische Norm den Anforderungen der allgemeinen Norm genügt:

1. Anforderung:

Da $\vec{v} = \vec{0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ gelten soll, sind $a_1^2 = 0$ und

$a_2^2 = 0$, deshalb gilt auch:

$$\left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\| = \sqrt{0^2 + 0^2} = \sqrt{0} = 0$$

Nun muss noch gezeigt werden, dass wenn $\vec{v} \neq \vec{0}$ auch $\|\vec{v}\| \neq 0$ ist.

Es sei $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, also $v_1 \neq 0$ oder

$v_2 \neq 0$, dann gilt $v_1^2 > 0$ oder bzw. und $v_2^2 > 0$

$$\text{und somit } \|\vec{v}\| = \left\| \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \right\| = \sqrt{v_1^2 + v_2^2} > 0$$

Bemerkung:

Da Quadrate nie kleiner 0 werden, ist der Radikand nie negativ.

2. Anforderung:

Eine kurze Rechnung liefert die 2. Anforderung:

$$\begin{aligned}\|\lambda \vec{v}\| &= \left\| \lambda \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \right\| \stackrel{(1)}{=} \left\| \begin{pmatrix} \lambda v_1 \\ \lambda v_2 \end{pmatrix} \right\| \\ &\stackrel{(2)}{=} \sqrt{(\lambda v_1)^2 + (\lambda v_2)^2} \\ &\stackrel{(3)}{=} \sqrt{\lambda^2 (v_1^2 + v_2^2)} \\ &\stackrel{(4)}{=} |\lambda| \cdot \sqrt{v_1^2 + v_2^2} \stackrel{(5)}{=} |\lambda| \cdot \|\vec{v}\|\end{aligned}$$

Erklärungen:

1. Multiplikation des Vektors mit λ
2. Definition der euklidischen Norm
3. Distributivgesetz (Ausklammern)
4. Teilweises Radizieren
5. Definition der euklidischen Norm

Somit hat man durch Umformungen und Einsetzen die Aussage bewiesen, da man die linke Seite der Gleichung in die rechte umgeformt hat.

3. Anforderung:

Die 3. Anforderung wird auch Dreiecksungleichung genannt. Diese besagt, dass die Summe zweier Dreiecksseiten stets kleiner bzw. gleich der längsten Dreiecksseite ist. Diese geometrische Erkenntnis dient uns als Beweis für die 3. Forderung.

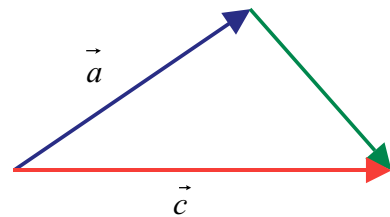
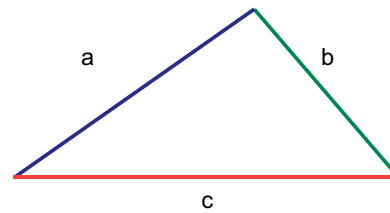
$$\|\vec{a}\| = a, \|\vec{b}\| = b, \|\vec{c}\| = c$$

$$a + b \geq c \Rightarrow \|\vec{a}\| + \|\vec{b}\| \geq \|\vec{c}\| \quad (1)$$

$$\vec{a} + \vec{b} = \vec{c} \Rightarrow \|\vec{a} + \vec{b}\| = \|\vec{c}\| \quad (2)$$

Aus (1) und (2) folgt:

$$\|\vec{a} + \vec{b}\| \geq \|\vec{a}\| + \|\vec{b}\|$$



Wenn wir nun für die Seiten v und w Vektoren einsetzen, dann können wir die beiden Vektoren addieren und erhalten dadurch die dritte Seite.

Einheitsvektoren:

Einen Vektor der Länge 1 nennt man Einheitsvektor. Multipliziert man einen nicht verschwindenden

Vektor \vec{v} skalar mit dem Kehrwert seiner Norm, erhält man einen Einheitsvektor mit gleicher Richtung. Dies fassen wir in der nächsten Definition zusammen.

Definition: Einheitsvektor von \vec{v}

Es sei V ein reeller Vektorraum und $\vec{v} \in V$ mit

$\vec{v} \neq \vec{0}$. Dann heißt $\vec{v}^0 = \frac{1}{\|\vec{v}\|} \cdot \vec{v}$ der zugehörige

Einheitsvektor (für $\vec{v} = \vec{0}$ ist der zugehörige Einheitsvektor undefiniert).

Beispiel: Gesucht wird der Einheitsvektor von

$$\vec{v} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

$$\vec{v}^0 = \frac{1}{\|\vec{v}\|} \cdot \vec{v} = \frac{1}{\sqrt{v_1^2 + v_2^2}} \cdot \vec{v}$$

$$= \frac{1}{\sqrt{3^2 + 4^2}} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \frac{1}{5} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

$$= \begin{pmatrix} 0,6 \\ 0,8 \end{pmatrix}$$

Wenn nun alles stimmt, dann muss die Norm dieses Vektors 1 sein.

$$\|\vec{v}^0\| = \sqrt{v_1^2 + v_2^2} = \sqrt{0,6^2 + 0,8^2}$$

$$= \sqrt{0,36 + 0,64} = 1$$

Die Norm der Standardbasisvektoren $\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

und $\vec{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ist 1.

$$\|\vec{e}_1\| = \|\vec{e}_2\| = 1$$

Polardarstellung

Die Polardarstellung ist die Darstellung eines Vektors aus dem Produkt seiner Norm (Länge) und seinem Einheitsvektor (Richtung):

$$\vec{v} = \|\vec{v}\| \cdot \vec{v}^0$$

Diese Darstellung hat den Vorteil, dass man sofort Länge und Richtung eines Vektors ablesen kann und deshalb zwei Vektoren nach Länge und Richtung schneller vergleichen kann.

Beispiel:

Gesucht wird die Polardarstellung von $\vec{v} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$:

$$\vec{v} = \begin{pmatrix} 0,6 \\ 0,8 \end{pmatrix}$$

$$\left\| \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right\| = \sqrt{3^2 + 4^2} = 5$$

$$\vec{v} = 5 \cdot \begin{pmatrix} 0,6 \\ 0,8 \end{pmatrix}$$

Das euklidische Skalarprodukt

Definition: Das Skalarprodukt

Sei V ein reeller Vektorraum.

Die Abbildung $\circ : V \times V \rightarrow \mathfrak{R}$

heißt Skalarprodukt, wenn folgende Eigenschaften erfüllt sind:

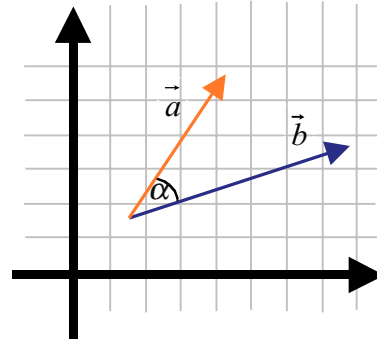
1. $\vec{v} \circ \vec{w} = \vec{w} \circ \vec{v}$ für alle $\vec{v}, \vec{w} \in V$
2. $(\vec{u} + \vec{v}) \circ \vec{w} = \vec{u} \circ \vec{w} + \vec{v} \circ \vec{w}$ für alle $\vec{u}, \vec{v}, \vec{w} \in V$
3. $r(\vec{v} \circ \vec{w}) = (r\vec{v}) \circ \vec{w} = \vec{v} \circ (r\vec{w})$
für alle $r \in \mathfrak{R}, \vec{v}, \vec{w} \in V$
4. $\vec{v} \circ \vec{v} > 0$ genau dann, wenn $\vec{v} \neq \vec{0}$

Für unsere physikalischen Phänomene benötigen wir ein spezielles Skalarprodukt, welches Euklidisches Skalarprodukt genannt wird.

Definition: Euklidisches Skalarprodukt

Sei $\vec{a}, \vec{b} \in \mathfrak{R}^2$

α bezeichne den nicht überstumpfen Winkel, den \vec{a} und \vec{b} einschließen.



Diese Abbildung zeigt die zwei Vektoren \vec{a}, \vec{b} und den eingeschlossenen Winkel α .

Dann ist $\vec{a} \circ \vec{b} = \|\vec{a}\| \|\vec{b}\| \cdot \cos \alpha$ das Euklidische

Skalarprodukt der Vektoren \vec{a} und \vec{b} .

Stehen die Vektoren \vec{a} und \vec{b} senkrecht

aufeinander, so beträgt $\alpha = 90^\circ$ und

$\cos \alpha = \cos 90^\circ = 0$. **Damit ist $\vec{a} \circ \vec{b} = 0$.**

Wenn nun die beiden Vektoren übereinander liegen, ist $\cos \alpha = 1$ somit $\vec{a} \circ \vec{b} = \|\vec{a}\| \|\vec{b}\|$.

Euklidische Skalarprodukt in Koordinatendarstellung

Nun stellt sich uns die Frage: „Was müssen wir tun wenn wir die Koordinaten der Vektoren \vec{a} und \vec{b} haben und das Skalarprodukt $\vec{a} \circ \vec{b}$ ausrechnen wollen?“

Wir stellen dieses Problem in der Koordinatenschreibweise dar.

Diese sieht wie folgt aus:

$$\vec{a} \circ \vec{b} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = ?$$

Um nun diese Formel zu lösen benötigen wir die Standardbasisvektoren \vec{e}_1 und \vec{e}_2 .

Wir benötigen außerdem das 2. und 3. Axiom des Skalarproduktes.

Wir stellen nun \vec{a} und \vec{b} als Linearkombination der Standardbasisvektoren dar:

$$\vec{a} = a_1 \vec{e}_1 + a_2 \vec{e}_2$$

$$\vec{b} = b_1 \vec{e}_1 + b_2 \vec{e}_2$$

Wenn wir dies nun in das Skalarprodukt einsetzen, ergibt sich diese Formel:

$$\vec{a} \circ \vec{b} = (a_1 \vec{e}_1 + a_2 \vec{e}_2) \circ (b_1 \vec{e}_1 + b_2 \vec{e}_2)$$

Dies rechnen wir nun mit dem 2. Axiom durch und bekommen folgendes heraus:

$$= a_1 \vec{e}_1 \circ b_1 \vec{e}_1 + a_1 \vec{e}_1 \circ b_2 \vec{e}_2 + a_2 \vec{e}_2 \circ b_1 \vec{e}_1 + a_2 \vec{e}_2 \circ b_2 \vec{e}_2$$

Dies müssen auch noch mit dem 3. Axiom durchrechnen:

$$= (\vec{e}_1 \circ \vec{e}_1) \cdot a_1 b_1 + (\vec{e}_1 \circ \vec{e}_2) \cdot a_1 b_2 + (\vec{e}_2 \circ \vec{e}_1) \cdot a_2 b_1 + (\vec{e}_2 \circ \vec{e}_2) \cdot a_2 b_2$$

Nun wenden wir unsere Erkenntnis über das Skalarprodukt an und überprüfen die Standardbasisvektoren. Die Skalarprodukte $\vec{e}_1 \circ \vec{e}_1$ und $\vec{e}_2 \circ \vec{e}_2$ ergeben nach unseren Erkenntnissen 1, da die Vektoren aufeinanderliegen und die Länge 1 haben. Die Skalarprodukte $\vec{e}_1 \circ \vec{e}_2$ und $\vec{e}_2 \circ \vec{e}_1$ ergeben 0 da sie senkrecht aufeinander stehen.

Wenn wir die Formel nun auflösen erhalten wir das Endergebnis:

$$a_1 b_1 + a_2 b_2$$

Das Ergebnis beweist die Formel:

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

Berechnung von Winkeln zwischen Vektoren

Sei $\vec{a}, \vec{b} \in \mathbb{R}^2, \vec{a}, \vec{b} \neq 0$

Dann gilt:

$$a_1 b_1 + a_2 b_2 = \vec{a} \circ \vec{b} = \|\vec{a}\| \|\vec{b}\| \cdot \cos \alpha$$

Wenn man diese Formel nun nach $\cos \alpha$ auflöst

so erhält man:
$$\cos \alpha = \frac{a_1 b_1 + a_2 b_2}{\|\vec{a}\| \|\vec{b}\|}$$

Diese Berechnungsweise ist sehr einfach, da wir allein durch die Kenntnis der Komponenten von zwei Vektoren ihren Zwischenwinkels berechnen können.

Das erste Newton'sche Axiom

Erklärung wichtiger Grundbegriffe:

Zeit und Raum:

Wir können Zeit und Raum (Längen) messen.

Die Grundeinheit der Zeit t ist die Sekunde:

$$[t] = 1s$$

Die Grundeinheit der Länge l ist das Meter:

$$[l] = 1m$$

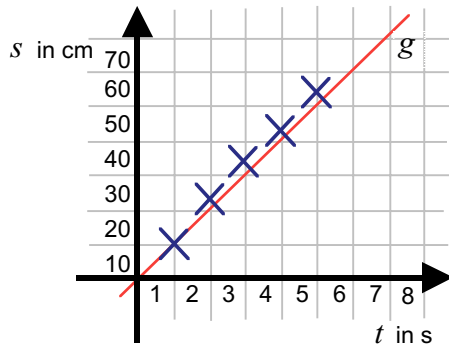
Die gleichförmige Bewegung

Versuch:

Wir nehmen eine Modelleisenbahn und lassen sie in einer waagrechteten Ebene auf einem Blatt Papier fahren. Jede Sekunde machen wir eine Makierung, wo sich die Eisenbahn befindet. Dadurch würde man dann eine Tabelle dieser Art erhalten:

abgelaufene Zeit t in s	1	2	3	4	5
zurückgelegter Weg s in cm	10	21	32	42	53

Wenn man diese Werte in ein Koordinatensystem einzeichnet, erhält man ein Orts-Zeit-Diagramm.



→ Bild 1: Messwerte der Tabelle + optimale Gerade

Die Punkte liegen ziemlich genau auf einer Geraden. Da man den Ort nicht immer ganz genau markieren konnte, sind Messfehler entstanden. Eine optimale Tabelle sähe z.B. so aus:

Abgelaufene Zeit t in s	1	2	3	4	5
Zurückgelegter Weg s in cm	10	20	30	40	50

In gleich langen Zeitabschnitten wurde immer die gleiche Strecke zurückgelegt. Für die doppelte Strecke wurde die doppelte Zeit benötigt, für die dreifache Strecke die dreifache Zeit, usw. Man sagt, die Werte für die Zeit und die Strecke sind zueinander proportional (das heißt, der Quotient aus Strecke und Zeit ist konstant). Der Quotient aus Strecke und Zeit ist die Geschwindigkeit v . v kommt von *velocitas*, dem lateinischen Wort für Schnelligkeit.

Definition: Geschwindigkeit v :

Die Ortsänderung pro Zeiteinheit eines Körpers

$$v = \frac{\Delta x}{\Delta t}$$

heißt die Geschwindigkeit des Körpers.

Mit Δx ist die Ortsänderung während des Zeitintervalls Δt gemeint. Das Δ ("Delta") steht für eine Änderung.

Beispiel:

Ein Sprinter legt eine Strecke von 100 Metern in 10 Sekunden zurück.

→ Δx (zurückgelegte Strecke) = 100m;

Δt (benötigte Zeit) = 10s

Dies setzt man nun in die Formel für die Geschwindigkeit ein:

$$v = \frac{\Delta x}{\Delta t} = \frac{100m}{10s} = 10 \frac{m}{s}$$

Man spricht von einer geradlinig gleichförmigen Bewegung, wenn die Fortbewegung (z.B. das Fahren der Eisenbahn) mit konstanter (gleichbleibender) Geschwindigkeit v stattfindet und geradlinig verläuft.

Wenn die Geschwindigkeit konstant ist, sind die Werte für die Strecke und die Zeit proportional. Wenn zwei Werte zueinander proportional sind, dann ist der Graph, in dem der eine Wert über den anderen aufgetragen ist, eine Ursprungsgerade.

Bei zwei verschiedenen schnellen Bewegungen unterscheiden sich die Quotienten aus zurückgelegter Strecke und vergangener Zeit. Bei der schnelleren Bewegung ist der Quotient größer als

bei der langsameren, da bei der schnellen Bewegung in der gleichen Zeit eine längere Strecke zurückgelegt wird.

Wenn wir die Geschwindigkeit $v_1 = \frac{1m}{1s} = 1 \frac{m}{s}$

verdoppeln, dann bewegen wir uns doppelt so weit pro Zeiteinheit fort und erhalten somit die

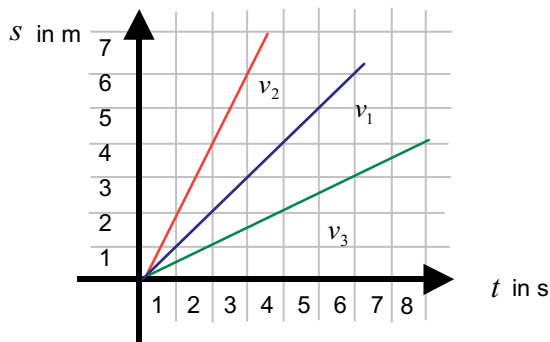
Geschwindigkeit $v_2 = \frac{2m}{1s} = 2 \frac{m}{s}$.

Wenn wir die Geschwindigkeit $v_1 = \frac{1m}{1s} = 1 \frac{m}{s}$

halbieren, bewegen wir uns nur noch halb so weit pro Zeiteinheit fort und erhalten somit die

Geschwindigkeit: $v_3 = \frac{0,5m}{1s} = 0,5 \frac{m}{s}$. In einem

Diagramm sieht dies folgendermaßen aus:



→ Bild 2: $x-t$ -Diagramm v_1 , v_2 und v_3

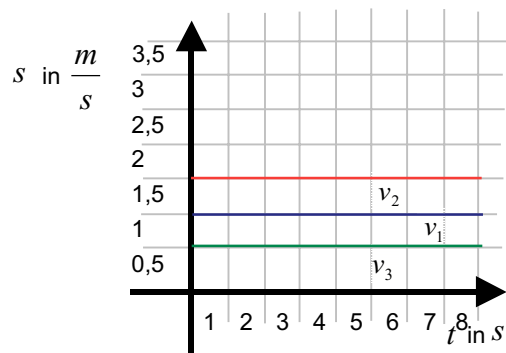
Man stellt fest, dass weder v_2 noch v_3 den Winkel zwischen v_1 und der x- bzw. y-Achse in 2 gleichgroße Winkel teilt. Die Verdoppelung oder Halbierung der Geschwindigkeit hat also nichts mit der Verdoppelung oder Halbierung dieser Winkel zu tun.

Alle drei Geschwindigkeiten lassen sich auch in einem Geschwindigkeits-Zeit-Diagramm darstellen: Man bestimmt von einem beliebigen Punkt auf der Geraden die Strecke s und die vergangene Zeit t und setzt dies in die Formel für die Geschwindigkeit

ein: $v_2 = \frac{\Delta x}{\Delta t} = \frac{2m}{1s} = 2 \frac{m}{s}$. Da wir eine

gleichförmige Bewegung vorliegen haben, ist die Geschwindigkeit konstant, das heißt sie ist zu jedem Zeitpunkt t dieselbe.

Im Geschwindigkeits-Zeit-Diagramm sieht dies folgendermaßen aus:

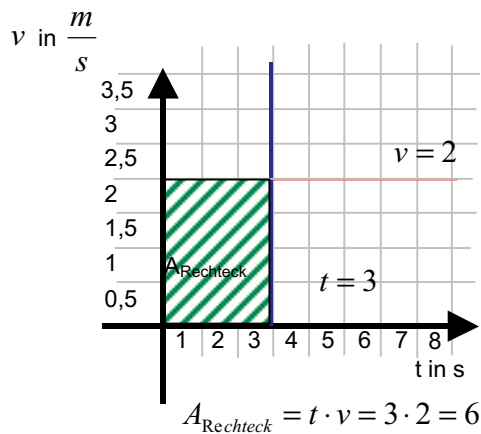


→ Bild 3: $v-t$ -Diagramm v_1 , v_2 und v_3

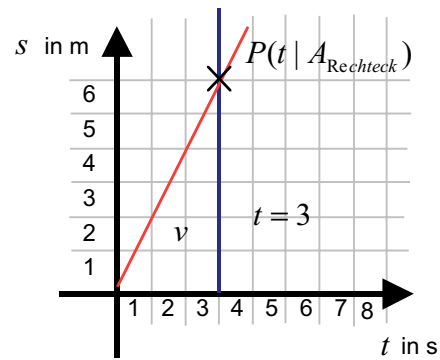
Geht dies auch rückwärts?

Ja, man bestimmt im Geschwindigkeits-Zeit-Diagramm einen beliebigen Zeitpunkt t (ungleich 0) auf der Zeit-Achse. Von diesem Punkt aus denkt man sich eine Gerade parallel zur Geschwindigkeits-Achse. Die Fläche des entstandenen Rechtecks ist: Nun zeichnet man im Orts-Zeit-Diagramm den selben Zeitpunkt t ein. Seine zugehörige y-Koordinate ist der Flächeninhalt des Rechtecks:

$$A_{\text{Rechteck}} = v \cdot t$$



→ Bild 4: $v - t$ -Diagramm mit Rechteck



→ Bild 5: zugehöriges $s - t$ -Diagramm

Erklärung: Die Formel $v = \frac{\Delta x}{\Delta t}$ lässt sich für die

gleichförmige Bewegung auch folgendermaßen vereinfachen, da die zurückgelegte Strecke pro

Zeiteinheit konstant ist: $v = \frac{x}{t}$

Mit s ist die zurückgelegte Strecke und mit t die seit dem Start vergangene Zeit gemeint.

Diese Formel kann man mit Hilfe von Äquivalenzumformungen nach s , der benötigten Größe für das Orts-Zeit-Diagramm, auflösen:

$$s = v \cdot t$$

$v \cdot t$ ist die Fläche des Rechteckes. Also ist

$$s = v \cdot t = A_{\text{Rechteck}}$$

1. Newtonsches Axiom (Trägheitsprinzip):

„Ein Körper verharrt im Zustand der Ruhe oder der gleichförmigen Bewegung, solange keine äußeren Kräfte auf ihn einwirken.“

Dieses Newtonsche Axiom wird auch Trägheitsprinzip genannt, weil jeder Körper den Bewegungszustand (das ist seine momentane Geschwindigkeit), in dem er sich befindet, beibehalten will.

Beispiel:

Im Weltraum befinden wir uns in einem Zustand der Schwerelosigkeit, d.h. keine Anziehungskräfte wirken auf uns ein. Reibungsphänomene treten auch nicht auf. Wenn wir nun einen Ball nehmen und ihn von uns weg werfen, dann wird er zuerst von uns beschleunigt (was dies ist soll später erklärt werden), sobald wir ihn aber los lassen, wirken keine äußeren Kräfte mehr auf ihn ein. Deshalb würde er solange bis irgendeine Kraft auf ihn einwirkt, mit konstanter (gleichbleibender) Geschwindigkeit geradeaus weiterfliegen.

Zu unserem Kursziel:

Anfangs hatten wir nur Geschwindigkeiten im Eindimensionalen betrachtet (Sprinter und Eisenbahn), das oben genannte Weltraumbeispiel ist allerdings schon dreidimensional. Der Flipper in unserer Simulation ist zweidimensional. Um die Bewegungen zweier Kugeln miteinander zu vergleichen interessiert uns nicht nur der Betrag der Geschwindigkeit sondern auch die Richtung, in die sich jede Kugel bewegt. Eine Richtung lässt sich durch einen Vektor beschreiben. Mit seiner Länge,

d.h. seiner Norm, kann man den Betrag der Geschwindigkeit angeben. Die Länge stellt also die Strecke x dar, die während des konstanten Zeitintervalls zurückgelegt wurde.

Das erste Newtonsche Axiom besagt, dass die Geschwindigkeit eines Körpers konstant bleibt, wenn keine Kräfte auf ihn einwirken. Auf die Vektoren bezogen heißt dies nun, dass der Geschwindigkeitsvektor konstant (d.h. gleich) bleiben muss. Das heißt weder die Geschwindigkeit noch die Richtung der Kugel würden sich ändern, wenn keine äußeren Kräfte (wie zum Beispiel in unserer Pinball-Simulation die Erdanziehungskraft oder die Reibungskraft) auf diese Kugel wirken würden.

Die Erfassung der Geschwindigkeit als vektorielle Größe hat auch bei der Programmierung der Pinball-Simulation entscheidende Vorteile. In der Computersimulation wurde der Standort der Kugel immer in Koordinaten angegeben. Deshalb ist es einfacher den neuen Standort der Kugel auszurechnen, wenn die x- und die y-Komponenten der Geschwindigkeit getrennt sind. „einfacher“ heißt in diesem Fall auch schneller, da der Computer für jede Rechnung Zeit benötigt und wir ein möglichst fortlaufendes Bild haben wollen.

Um den neuen Ort der Kugel auszurechnen, muss

man erstmal die Formel $v = \frac{\Delta x}{\Delta t}$ nach

Δx auflösen: $\Delta x = v \cdot \Delta t$.

Die Geschwindigkeit der Kugel wurde am Anfang per Zufall festgelegt. Mit Δt ist in diesem Fall die

Zeit, die zwischen Bild 1 und Bild 2 vergeht gemeint. Da diese Zeitspanne feststeht, kann man sie schon in die Geschwindigkeit mit einbeziehen. Somit ergibt sich bei unserer Simulation sofort die Strecke, die während einer festgelegten Zeiteinheit zurückgelegt wird.

Um den neuen Ort zu erhalten muss noch Δx zur ursprünglichen x Koordinate addiert werden. Im Quellcode sieht dies so aus:

```
for (i=0; i<=n; i++)
{
    kugel[ i ].x += kugel[ i ].geschw_x;
    kugel[ i ].y += kugel[ i ].geschw_y;
}
```

Diese Schleife durchläuft das Programm für jede Kugel.

Arbeit und Energie

Was haben Arbeit und Energie eigentlich mit unserem Kursziel zu tun? Die Begriffe Arbeit und Energie treten auf, wenn sich bei unserer Computersimulation die Kugeln bewegen.

Nachdem wir nun wissen wie wichtig Arbeit und Energie für unseren Kurs sind, beginnen wir diese Begriffe für uns genauer zu definieren. Jeder von Ihnen wird diese Begriffe schon mehrfach gehört haben und häufig im Alltag benutzen. Wird ein Gegenstand hoch gehoben, ein Wagen eine Strecke gezogen, ein Brett gehobelt oder ein Stück Metall gefeilt, so sagt man schon im täglichen Leben, dass dabei Arbeit verrichtet wird.

76

Aber um was handelt es sich bei der physikalischen Arbeit und Energie genau?

Definition Arbeit W:

Zuerst einmal muss das Zeichen für die Arbeit, nämlich das W geklärt werden. Warum ausgerechnet $W \text{ S}^2$ Das Zeichen W kommt vom englischen Wort „work“, das nichts anderes als Arbeit heißt. Nachdem geklärt ist warum Arbeit mit W abgekürzt wird, können wir uns mit der physikalischen Aufgabe der Arbeit beschäftigen. Wenn die Kraft F entlang einer Strecke s auf einen Körper wirkt, dann verrichtet die Kraft an diesem Körper die Arbeit W .

Formel:
$$W = \vec{F} \circ \vec{s}$$

In Worten: Die Arbeit kann man berechnen, indem man den Vektor \vec{F} der Kraft, mit dem Vektor \vec{s} der Strecke skalar multipliziert.

Nun ist uns der physikalische Begriff der Arbeit genau bekannt und wir kennen auch die Formel um mit ihr zu rechnen, doch was uns jetzt noch fehlt, ist die Einheit. Die Einheit der Arbeit heißt Joule.

Herleitung der Einheit der Arbeit

$$[W] = \left[\vec{F} \circ \vec{s} \right] = 1N \cdot 1m = 1Nm = 1J$$

Erklärung der Einheit

Die Einheit der Arbeit, Joule, wird mit dem Buchstaben J abgekürzt. Warum ausgerechnet Joule? Die Einheit Joule ist nach dem britischen Physiker James Prescott Joule (1818-1889) benannt. Dieser war ein Schüler des bekannten Naturforschers John Dalton (1766-1844) und interessierte sich bei seiner Forschung vor allem für Elektrizität und Thermodynamik.



James Prescott Joule

Rechenbeispiel:

Wenn ein Kleinkind (Gewicht: $30kg$) mit der Gewichtskraft $F = 300N$ die Strecke $s = 4m$ der Leiter einer Rutsche hinaufklettert, so verrichtet es damit eine mechanische Arbeit von $1200Nm$ oder $1200J$.

$$F = 300N$$

$$s = 4m$$

$$W = ?$$

$$W = F \cdot s = 300N \cdot 4m = 1200Nm = 1200J$$

Definition Energie

Jeder Mensch braucht Energie um sich zu bewegen und so ist das auch in der Physik, denn Energie ist das Vermögen eines Körpers Arbeit zu verrichten. Es gibt verschiedene Arten von Energie. Nein, ich rede nicht von Atom- oder Sonnenenergie. Bei der Art von Energie, die ich Ihnen vorstellen möchte handelt es sich um Lageenergie. Aber was können Sie sich unter Lageenergie vorstellen? Eine Keksdose, die auf dem Küchenschrank steht, hat eine bestimmte Energie, da sie auf den Boden fallen könnte.

Definition Lageenergie (=potenzielle Energie)

Da wir nun ein konkretes Beispiel für Lageenergie kennen, müssen wir unser Beispiel nur noch verallgemeinern. Ein Körper der Masse m mit einer Höhendifferenz h zum zuvor festgelegten Nullpunkt der Energieskala hat die Lageenergie:

$$E = m \cdot g \cdot h$$

Dies gilt im Schwerfeld der Erde. Bei g handelt es sich um die Ihnen schon bekannte Erdbeschleunigung.

Jetzt müssen wir nur noch das Problem klären, wohin wir den Nullpunkt der potenziellen Energie legen. Denn wenn die Keksdose aus dem Fenster fallen würde, hätten wir eine um einiges höhere kinetische Energie, als wenn sie nur auf den Boden

fällt. Die Auswahl des Nullpunktes ist wichtig, der Nullpunkt ist aber trotzdem frei wählbar.

Bei der Arbeit handelt es sich um eine Energieübertragung, da an einem Körper Arbeit verrichtet wird und sich dadurch auf ihn Energie überträgt.

Herleitung der Einheit der Energie

$$[E] = [W] = 1J$$

Bei der Einheit der Energie handelt es sich um nichts anderes als um die Einheit der Arbeit.

Erklärung kinetische Energie

Kinetische Energie hört sich erst einmal ziemlich kompliziert an, ist sie aber nicht, denn wir alle haben jeden Tag mit ihr zu tun. Das Wort kinetisch kommt aus dem Griechischen und heißt bewegt. Bei der kinetischen Energie handelt es sich also um Bewegungsenergie. Mit ihr wird gemessen, wieviel Arbeit man an einem Gegenstand verrichten musste, um ihn auf seine jetzige Geschwindigkeit zu beschleunigen. Die gleiche Arbeit kann der Gegenstand beim Abbremsen wieder verrichten. Die kinetische Energie hängt nicht nur davon ab, wie stark man einen Gegenstand beschleunigen will, sondern auch von dessen Masse.

Das wissen alle, die gerne mit Spielzeugautos gespielt haben und heute ein richtiges Auto fahren.

Wie einfach konnte man das Spielzeugauto beschleunigen aber wenn die Autobatterie streikt, kann man das Auto nur schwer anschieben.

$$\text{Formel: } W_{kin} = \frac{1}{2} \cdot m \cdot v^2$$

Die Arbeit W , die von der kinetischen Energie verrichtet werden kann, berechnet man, indem man $\frac{1}{2}$ mit der Masse m und der quadrierten Geschwindigkeit v eines Körpers multipliziert.

Rechenbeispiel:

Vergleichen Sie die kinetische Energie einer Straßenwalze von $6000kg$ bei einer

Geschwindigkeit von $3,6 \frac{km}{h}$ mit der eines

Geschosses von $20g$ bei der Geschwindigkeit von

$$900 \frac{m}{s}.$$

Straßenwalze:

$$m = 6000kg;$$

$$v = 3,6 \frac{km}{h} = 1 \frac{m}{s}$$

$$W_{kin} = \frac{1}{2} \cdot m \cdot v^2 = \frac{1}{2} \cdot 6000kg \cdot 1 \frac{m^2}{s^2} = 3000J$$

Geschoss:

$$m = 20g = 0,02kg$$

$$v = 900 \frac{m}{s}$$

$$W_{kin} = \frac{1}{2} \cdot m \cdot v^2 = \frac{1}{2} \cdot 0,02kg \cdot \left(900 \frac{m}{s}\right)^2$$

$$= 8100J$$

Dieses Ergebnis wird Sie vielleicht zuerst überraschen, da die kinetische Energie des Geschosses trotz seiner kleinen Masse, die kinetische Energie der Straßenwalze übertrifft. Aber eigentlich ist dies einfach zu erklären, denn der Einfluss der Geschwindigkeit überwiegt, da diese bei der Berechnung quadriert wird.

Energieerhaltungssatz der Mechanik

Um was handelt es sich beim Energieerhaltungssatz der Mechanik? Dass es überhaupt einen Energieerhaltungssatz gibt, wird die meisten von ihnen verwirren. Denn wir alle meinen im ersten Moment, dass immer Energie verloren geht. Sonst bräuchten wir Menschen ja nichts essen und auch nicht schlafen und wären trotzdem immer top leistungsfähig. Dass dies leider nicht stimmt mussten wohl die meisten von ihnen schon am eigenen Leib erfahren. Jedoch geht Energie niemals verloren sondern sie wandelt sich immer in eine andere Energie um, zum Beispiel in Wärme durch Reibung. Auch in der Mechanik gilt der Energieerhaltungssatz. Allerdings nur in einem reibungsfrei ablaufenden System wandelt sich die kinetische Energie nicht in andere Energieformen um. Ein Beispiel für ein reibungsfrei ablaufendes System wäre zum Beispiel, wenn wir uns eine so glatte Fläche vorstellen, dass ein Spielzeugauto auf ihr ohne jeglichen Widerstand fahren kann. Dann

bräuchten wir das Auto nur ein einziges Mal anschieben und durch den Energieerhaltungssatz würde es für immer vor sich hin fahren.

Definition Energieerhaltungssatz der Mechanik

In einem abgeschlossenen System ist die Summe der mechanischen Energien, das heißt die Summe aus kinetischer und potentieller Energie konstant, solange die Vorgänge im System reibungsfrei laufen.

Formel: $E = E_{kin} + E_{pot} = \text{konstant}$

Die Gesamtenergie eines Körpers besteht aus der Summe seiner kinetischen und seiner potentiellen Energie, unter den oben genannten Bedingungen ist diese konstant.

Im Schwerfeld der Erde kann man die Potenzielle Energie eines Körpers wie zum Beispiel einer Keksdose berechnen, indem man die Gewichtskraft mit der Fallhöhe multipliziert. Da die Formel der Gewichtskraft $F = m \cdot g$ lautet, handelt es sich bei dieser Formel um nichts anderes als um die Ihnen schon bekannte Formel der Lageenergie.

Rechenbeispiel

Eine Keksdose mit der Gewichtskraft $0,2N$ steht auf einem Küchenschrank mit der Höhe $2,0m$.

Wie groß ist seine potenzielle Energie und wie hoch ist seine Aufprallgeschwindigkeit?

Berechnung der potenziellen Energie:

$$F = 0,2N$$

$$h = 2,0m$$

$$W_{pot} = F \cdot h = 0,2N \cdot 2,0m = 0,4Nm = 0,4J$$

Die potentielle Energie beträgt $0,4J$.

Berechnung der Aufprallgeschwindigkeit:

$$W_{kin} = 0,4J$$

$$m = 0,2kg$$

$$W_{kin} = \frac{1}{2} \cdot m \cdot v^2$$

$$0,4J = \frac{1}{2} \cdot 0,2kg \cdot v^2$$

$$0,4J = 0,1kg \cdot v^2$$

$$4 \frac{m^2}{s^2} = v^2$$

$$v = 2 \frac{m}{s}$$

Die Keksdose prallt mit der Geschwindigkeit $2 \frac{m}{s}$

auf dem Boden auf.

Das zweite Newton'sche Axiom

Das erste Newtonsche Axiom besagt, dass ein Körper im Zustand der Ruhe oder der geradlinig

gleichförmigen Bewegung verharrt, solange keine äußeren Kräfte auf ihn einwirken. Daraus folgt, dass ein Körper sich genau dann beschleunigt bewegt, wenn auf ihn eine Kraft einwirkt.

Nun stellt man sich die Frage, welche Kraft auf einen Körper einwirken muss, damit er sich mit einer bestimmten Beschleunigung bewegt. Genau diese Frage beantwortet uns das zweite Newtonsche Axiom.

Das zweite Newtonsche Axiom lautet wie folgt:

Um einen Körper der Masse m mit der Beschleunigung a zu beschleunigen, muss auf den Körper die Kraft

$$F = m \cdot a$$

einwirken.

Um die Gleichung des zweiten Newtonschen Axioms zu erklären, müssen die beiden Begriffe Masse m und Beschleunigung a eingeführt werden.

So wie Raum und Zeit ist auch die Masse eine wichtige physikalische Grundgrößen.

Jeder Körper hat eine Masse. Möchte man diese Masse bestimmen, so vergleicht man die

Gewichtskraft des Körpers mit der Gewichtskraft eines Normkörpers (Urkilogramm). Dies geschieht indem man diesen Körper beispielsweise mit einer Balkenwaage wiegt. Die **Masse** m besitzt die Einheit Kilogramm (kg).

Zur Berechnung der Kraft F benötigt man weiterhin die Beschleunigung a .

Jeder Körper dessen Geschwindigkeit sich ändert, führt eine beschleunigte Bewegung aus. Die **Beschleunigung** a eines Körpers definiert sich also einfach als Geschwindigkeitsänderung pro Zeiteinheit des Körpers.

$$a = \frac{\Delta v}{\Delta t}$$

Das v steht in dieser Gleichung für die Geschwindigkeit und t für die Zeit. Das Zeichen Δ steht jeweils für eine Änderung, so steht Δv für die Geschwindigkeitsänderung und Δt für die Zeitänderung.

Die Geschwindigkeit v hat die bekannte Einheit m/s (km/h), die Zeiteinheit ist bekanntlich die Sekunde s , daraus folgt nun die **Einheit der Beschleunigung** a

$$a = \frac{\frac{m}{s}}{s} = \frac{m}{s^2}$$

Es fehlt schließlich noch die **Einheit der Kraft**. Sie folgt aus der Gleichung $F = m \cdot a$ des zweiten Newtonschen Axioms. Wir haben die Einheit der Beschleunigung a (m/s^2) und der Masse m (kg). Daraus folgt als Einheit für die Kraft:

$$[F] = kg \cdot \frac{m}{s^2}$$

Zu Ehren des bahnbrechenden Physikers kürzt man diese Einheit mit Newton ab:

$$1kg \frac{m}{s^2} = 1N$$

Mithilfe des zweiten Newtonschen Axioms kann man nun berechnen welche Kraft man aufwenden muss, um einen Körper der Masse m mit der Beschleunigung a zu beschleunigen.

Ein Beispiel:

Welche Kraft wirkt auf einen Apfel, der vom Baum fällt ?

Diese Frage kann man mittels des zweiten Newtonschen Axioms beantworten. Dazu braucht man eine weitere Größe, die Erdbeschleunigung g . Die **Erdbeschleunigung** g beträgt in mittleren

Breiten $9,81 \text{ m/s}^2$. Sie verändert sich nicht bei Körpern mit verschiedener Masse.

Zusätzlich benötigt man die Masse m des Apfels, z.B. 100 g. Mithilfe des zweiten Newtonschen Axioms berechnet man die Schwerkraft, (oder auch Erdanziehungskraft genannt), folgendermaßen :

$$F = m \cdot a$$

$$F = 0,1 \text{ kg} \cdot 9,81 \text{ m/s}^2$$

$$F = 0,981 \text{ kg} \cdot \text{m/s}^2 = 0,981 \text{ N}$$

Genau genommen trifft dieses Ergebnis nur auf die mittleren Breiten zu.

Die Erdbeschleunigung g ist ortsabhängig,

deshalb nennt man sie auch **Ortsfaktor**.

Nachfolgend sind einige Werte angegeben:

am Äquator : $9,787 \text{ m/s}^2$

100 km über der Erde : $9,52 \text{ m/s}^2$

auf dem Mond : $1,62 \text{ m/s}^2$

auf dem Mars : $3,71 \text{ m/s}^2$

auf der Sonne : 274 m/s^2 .

Das dritte Newton'sche Axiom

Beim Abschuss einer Pistole erhält man einen kräftigen Stoß nach hinten. Dasselbe Verhalten wird bei Düsentriebwerken benutzt, indem man einen Stoß nach hinten abgibt, der das Flugzeug nach vorne befördert (Rückstoßprinzip).

Formulierung des dritten Newton'schen Axioms

Formal definiert besagt das dritte Newton'sche

Axiom, dass jede Kraft \vec{F} eine Reaktionskraft besitzt. Diese Reaktionskraft hat den gleichen Betrag, zeigt jedoch in die entgegengesetzte Richtung:

$$\vec{F} = -\vec{F}'$$

Das bedeutet, dass die beiden Vektoren dieselbe Länge besitzen.

$$\|F\| = \|F'\|$$

Diese Gleichung nennt man actio = reactio

Die Angriffspunkte der Kräfte müssen in zwei verschiedenen Körpern liegen.

Der Impuls

Wir kommen jetzt zu dem eigentlichen Problem unseres Kurses „Pinball Wizard“ zurück: Um eine Kollision zwischen zwei Kugeln exakt zu berechnen, benötigt man eine weitere Größe: den Impuls. Der Impuls ist das Produkt aus der Masse und der Geschwindigkeit.

Man bezeichnet ihn mit \vec{p} . Die Formel zur Berechnung lautet folglich:

$$\vec{p} = \text{Masse} \cdot \text{Geschwindigkeit} = m \cdot \vec{v}.$$

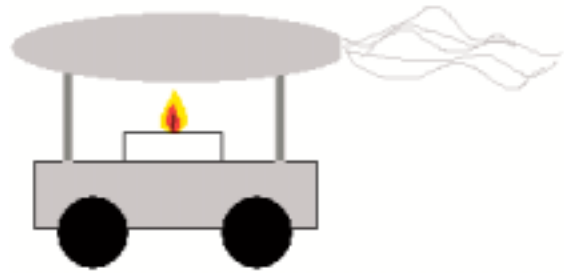
Auch seine Einheit lässt sich leicht erschließen:

$$1\text{kg} \cdot 1 \frac{\text{m}}{\text{s}} = 1 \frac{\text{kg} \cdot \text{m}}{\text{s}^2} \cdot \text{s} = 1\text{Ns}$$

Der Impulserhaltungssatz

Der Impulserhaltungssatz besagt, dass sich die Summe aller Impulse in einem abgeschlossenen System nie ändert. Einfacher ausgedrückt bedeutet das, dass bei unserer Pinball-Simulation die Summe der Impulse nach einer Kollision zwischen zwei Kugeln dieselbe ist wie vor dem Zusammenstoß. Wenn man nun ein Wägelchen hat, welches in Ruhe ist und nun von einem anderen Wägelchen mit gleicher Masse und einer konstanter Geschwindigkeit zentral getroffen wird, so bleibt dieser Wagen stehen und der erste bewegt sich mit der ursprünglichen Geschwindigkeit des zweiten fort. Der Impuls überträgt sich also auf den ersten Wagen. Somit ist die Summe der Impulse gleich groß geblieben. Um nun wieder etwas näher an die Praxis heran zu kommen nehmen wir uns ein neues Beispiel, an dem man gut die Anwendung des Impulses und des dritten Newton'schen Axioms erläutern kann.

Anwendungsbeispiel



Skizze des Wagens mit „Düsenantrieb“

Wir haben nun einen kleinen, fahrbaren Wagen mit einem „Düsenantrieb“ (vergleichbar mit einem Flugzeug mit Düsenantrieb). Dieser Wagen hat eine Halterung auf der ein Behälter mit einer kleinen Öffnung am hinteren Ende des Fahrzeugs angebracht ist. In diesen Behälter füllt man nun 20g Wasser. Die winzige Menge Luft, die sich noch im Behälter befindet, kann man bei unserer Berechnung vernachlässigen, da sie nur ein kleines Gewicht hat, welches unsere Berechnung nicht sehr stark beeinflusst. Wenn man nun unter diesen Behälter einige Kerzen anbringt, erwärmen sie das Wasser, sodass es nach einiger Zeit als Dampf nach hinten aus dem Loch entweicht und den Wagen aufgrund des dritten Newton'schen Axioms nach vorne befördert (Rückstoßprinzip). Wenn wir nun die Geschwindigkeit des austretenden

Wassers, sagen wir $10 \frac{\text{m}}{\text{s}} = 36 \frac{\text{km}}{\text{h}}$, kennen,

können wir den Impuls des Wassers berechnen

$$p = m \cdot v = 20g \cdot 10 \frac{m}{s} = 200 \frac{g \cdot m}{s} .$$

Wegen des Satzes der Impulserhaltung ist dieser Impuls betragsmäßig gleich groß wie der Impuls der Vorwärtsbewegung, da die Summe der Impulse im System vor dem Ereignis 0 war, müssen sich die Impulse aufheben. Nun löst man die Formel für den

Impuls nach der Geschwindigkeit v auf:
$$v = \frac{p}{m}$$

Wenn die Masse des Wagens $200g$ beträgt, folgt daraus für den Impuls der Vorwärtsbewegung:

$$v = \frac{p}{m} = \frac{200 \frac{g \cdot m}{s}}{100g} = 2 \frac{m}{s}$$

Das bedeutet, der Wagen hat eine Geschwindigkeit

von $2 \frac{m}{s} = 7,2 \frac{km}{h}$, was ungefähr der

Geschwindigkeit eines schnellen Fußgängers entspricht.

Vektorgeometrie der Geraden und Strecken

Motivation

Die Vektorraumstruktur der Euklidischen Ebene macht es möglich, Geraden und Strecken mit Vektoren zu definieren.

Für unseren Kurs brauchen wir das, um die Kollisionserkennung und -behandlung verstehen zu können.

Vektorgeometrie der Geraden

Eine Gerade ist eine Punktmenge, sie besteht aus unendlich vielen Punkten, die zusammen die Gerade ergeben.

Wir können eine Gerade g mit zwei voneinander verschiedenen Punkten A und B , die auf der Geraden liegen genau bestimmen (Abb. 1).

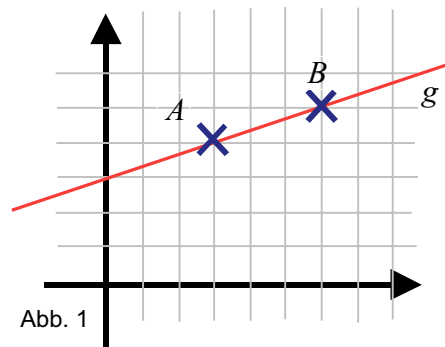


Abb. 1

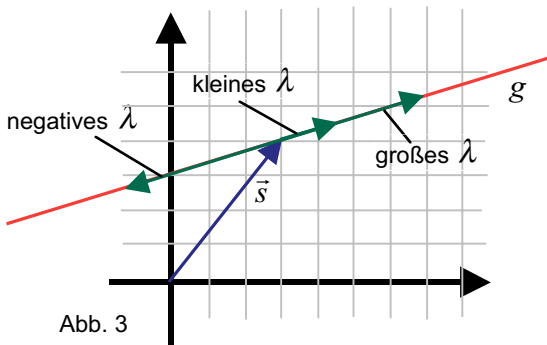
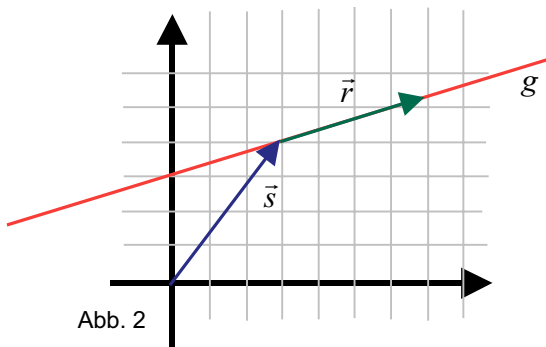
Mit der Vektorrechnung können wir diese Gerade nun sehr einfach als Formel hinschreiben:

$$g = \{ \vec{s} + \lambda \vec{r} \mid \lambda \in \mathfrak{R} \}$$

Dabei gilt:

- \vec{s} ist ein Vektor vom Ursprung bis zu einem beliebigen Punkt auf der Geraden. Hierzu eignet sich z.B. der Ortsvektor des Punktes A . Diesen nennen wir Stützvektor, weil er die Gerade sozusagen "stützt".

- \vec{r} ist der sogenannte Richtungsvektor, da er in die Richtung der Geraden zeigt (Abb. 2). Wenn \vec{a} und \vec{b} die Ortsvektoren der Punkte A und B sind, erhält man ihn z.B. durch $\vec{r} = \vec{b} - \vec{a}$.
- Das Lambda (λ) skaliert den Richtungsvektor. Da es jede beliebige reelle Zahl einmal annimmt, führt das dazu, dass mit dem Term $\vec{s} + \lambda\vec{r}$ alle Punkte auf der Geraden erreicht werden (Abb. 3).



Vektorgeometrie der Strecken

Den oben vorgestellten Ausdruck für eine Gerade müssen wir nur geringfügig modifizieren, um die Formel für eine Strecke zu erhalten.

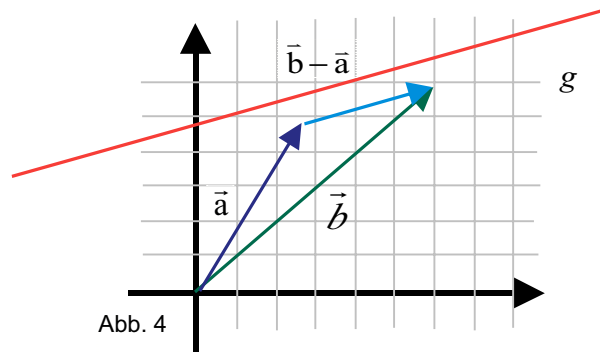
Denn wie die Gerade ist natürlich auch die Strecke eine Punktmenge, nur diesmal müssen wir λ einschränken, da die Strecke sonst keine Endpunkte hätte, sprich eine Gerade wäre.

Als Stützvektor wählen wir zunächst den Ortsvektor \vec{a} des Endpunktes A der Strecke.

Als Richtungsvektor nehmen wir wieder $\vec{b} - \vec{a}$, wobei \vec{b} der Ortsvektor des Endpunktes B der Strecke ist. λ muss zwischen 0 und 1 festgelegt werden, sodass genau alle Punkte der Strecke erreicht werden (Abb. 4).

Dann lautet die Formel für eine Strecke:

$$g = \{\vec{a} + \lambda(\vec{b} - \vec{a}) \mid \lambda \in \mathbb{R} \wedge 0 \leq \lambda \leq 1\}$$



Der Einheitsnormalenvektor

- Der Einheitsvektor eines Vektors ist der Vektor mit der gleichen Richtung nur der Norm 1;
- Ein Normalenvektor eines Vektors ist ein Vektor der zu dem Vektor orthogonal (senkrecht) ist;
- Ein Einheitsnormalenvektor ist demnach ein Vektor der orthogonal ist *und* die Norm 1 hat.

Zu einem Vektor $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ sind

$$\vec{n}_1 = \begin{pmatrix} -v_2 \\ v_1 \end{pmatrix} \text{ (nach rechts gedreht)}$$

$$\text{und } \vec{n}_2 = \begin{pmatrix} v_2 \\ -v_1 \end{pmatrix} \text{ (nach links gedreht)}$$

die beiden Normalenvektoren.

Einheitsnormalenvektor ist jetzt

$$\vec{n}^0 = \frac{\mathbf{I}}{\|\vec{n}\|} \cdot \vec{n} \quad (\text{Abb. 5}).$$

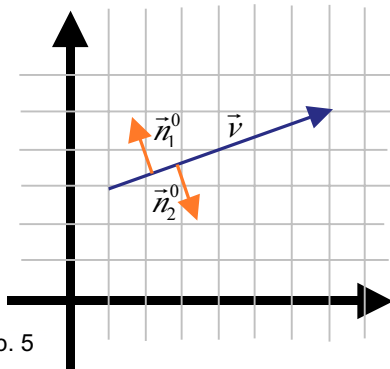


Abb. 5

Der orientierte Abstand zwischen Punkt und Gerade

Jetzt müssen wir wissen, wie wir den Abstand eines Punktes zu einer Geraden ausrechnen. Diesen brauchen wir für die Kollision einer Kugel mit einem Rand und mit einer Linie. Den orientierten Abstand kann man gut mit folgendem Einstiegsproblem angehen:

Herr Bauer sitzt nachmittags vor seinem Fernseher und schaut sich einen Film an. Dann hört er plötzlich Lärm und Geräusche auf der Straße: Oh nein, denkt er, es sind schon wieder diese lauten, störenden Kinder, die den ganzen Nachmittag Fußball spielen und immer den Ball über meine Mauer schießen, wobei sie ganz genau wissen, dass sie das nicht dürfen!

Herr Bauer überlegt, was er machen kann, denn er will unbedingt seinen Film zu Ende schauen aber auch nicht tatenlos zusehen, wie sein Garten ruiniert wird!

Da kommt ihm eine Idee: Ah, mein Sohn hat doch neulich mal auf der Science-Academy etwas über Vektoren und orientierten Abstand zwischen Punkt und Linie gelernt! Das kann man doch hier gebrauchen! Und Herr Bauer stellt seinen superschlauen Sohn vor das Haus und lässt ihn fortlaufend den orientierten Abstand vom Ball zur Mauer ausrechnen. Sobald dieser negativ wird, schreit der Sohn, und Herr Bauer weiß, dass der

Ball bei ihm im Garten ist. Dann kann er rausrennen und die Kinder schimpfen...

Und wieder einmal sieht man, zu was das, was man bei der Science-Academy lernt, alles gut sein kann!

Orientierter Abstand heißt nämlich, dass der Wert auch negativ sein kann, was dann bedeutet, dass der Punkt auf der anderen Seite der Geraden liegt. Das ist sehr praktisch weil man so immer weiß, auf welcher Seite der beliebig gewählte Punkt liegt.

Es wird also eine beliebige Gerade und ein beliebiger Punkt gewählt. Die Gerade soll in unserem Beispiel g und der Punkt P sein.

Die Vorgehensweise:

\vec{p} sei der Ortsvektor von P . Gegeben sind der Einheitsnormalenvektor \vec{n}^0 und der Stützvektor \vec{s} der Geraden.

Jetzt wenden wir die Projektion an (wir projizieren $\vec{p} - \vec{s}$ auf \vec{n}^0) und kommen dann auf folgende

Formel:

$$d = (\vec{p} - \vec{s}) \circ \vec{n}^0$$

Nach dieser Formel kann d wie gesagt auch negativ sein, das heißt eigentlich ist der Betrag von d der richtige Abstand von Punkt zur Geraden.

Wenn der Punkt auf der Seite liegt, in die auch der Normaleneinheitsvektor zeigt, ist der orientierte Abstand positiv; wenn er auf der anderen Seite liegt ist er negativ.

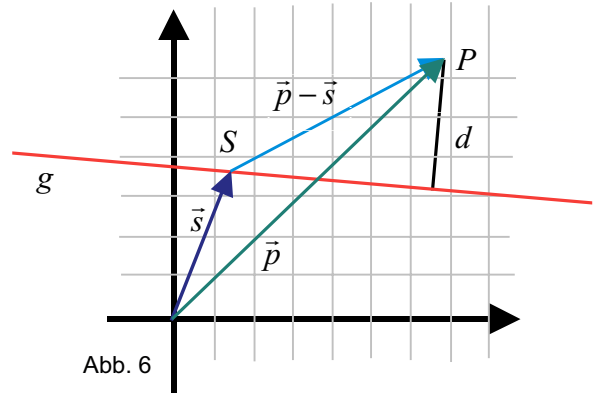


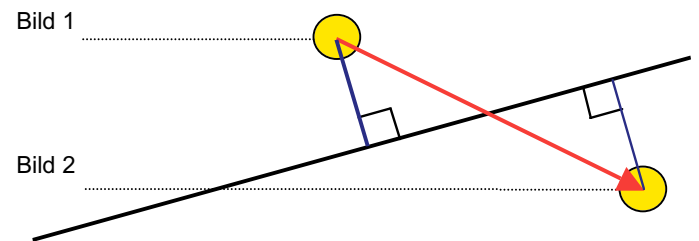
Abb. 6

Kollisionserkennung

Ein Problem bei der Realisierung unserer Flippersimulation war die Erkennung von Kollisionen. Dabei haben wir verschiedene Kollisionen unterschieden.

Kugel-Strecken Kollision

Der erste zu betrachtende Fall ist, wenn eine Kugel mit einer Strecke kollidiert. Dies zu erkennen ist für unseren Flipper sehr wichtig, da es hier viele



Hindernisse gibt, die linienförmig sind.

Zunächst betrachten wir den Fall einer Kollision mit einer Geraden.

Sei \vec{k} der Ortsvektor der Kugel \vec{e} der Stützvektor der Geraden und \vec{n} der Einheitsnormalvektor der Geraden. Dann ist $\vec{s} \circ \vec{n} = d$ mit $\vec{k} - \vec{e} = \vec{s}$ der orientierte Abstand der Kugel von der Geraden. Jetzt muss nur noch überprüft werden, ob der Betrag von d kleiner ist als der Radius der Kugel. Im Programm berechnet man d dann so:

$$d = s.x * n.x + s.y * n.y$$

wobei $s.x$ und $s.y$ die Komponenten von \vec{s} sind.

Für die Berechnung des Betrags gibt es den Befehl `abs(d)`. Im Programm sieht das dann so aus:

```
abs_d = abs(d);
```

Die Kollisionserkennung für eine Gerade sieht dann im Programm folgendermaßen aus:

```
d = s.x * n.x + s.y * n.y;

if (abs_d <= kugel[ i ].radius)
{
    ...
}
```

Wir haben hier aber einen Spezialfall noch nicht behandelt: Was ist wenn die Kugel sich so schnell bewegt, dass sie im nächsten Bild schon so weit ist, dass sie sich hinter der Geraden befindet?

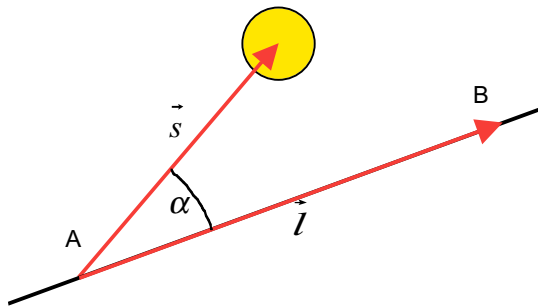
Hier kommt der vorhin angesprochene Vorteil des Abstandes als orientierte Größe ins Spiel. Sei d_1 der orientierte Abstand vor der Kollision und d_2 der orientierte Abstand nach der Kollision. Wir multiplizieren einfach d_1 mit d_2 und wenn dieses Produkt kleiner null ist (also d_1 und d_2 verschieden Vorzeichen haben), wird auch eine Kollision erkannt. Die vollständige Kollisionserkennung für eine Gerade sieht im Programm dann so aus:

```
d = s.x * n.x + s.y * n.y

if ( ( abs_d <= kugel[ i ].radius ) || (
d * d2 < 0 ) )
{
    ...
}
```

Doch eigentlich wollten wir eine Kollisionserkennung für eine Strecke und nicht für eine Gerade. Hierzu brauchen wir jetzt die Eigenschaft des Skalarprodukts, das negativ wird, wenn der eingeschlossene Winkel größer als 90° ist. Sei \vec{l} der Verbindungsvektor zwischen den Endpunktender Strecke A und B. Sei \vec{s} der Verbindungsvektor zwischen einem Endpunkt der Strecke und der Kugel und α der von \vec{s} und \vec{l} eingeschlossene Winkel. Wenn $\vec{s} \circ \vec{l}$ kleiner 0 ist

oder, wenn es kleiner \vec{l}^2 , so trifft der Ball die Strecke



Im Programm sieht das dann folgendermaßen aus:

```
if ( ( s.x * l.x + s.y * l.y > 0 ) &&
    ( l.x * l.x + l.y * l.y > s.x * l.x +
      s.y * l.y ) )
{
    ...
    if ( ( abs_d <= kugel[ i ].radius ) || (
        d * d2 < 0 ) )
    {
        ...
    }
}
```

Bei der Kugel-Strecken Kollision gilt es nun nur noch einen Sonderfall zu erkennen: Was ist, wenn die Kugel den Endpunkt der Strecke trifft? Dies ist recht einfach. Dazu berechnet man mit Hilfe des Satzes von Pythagoras den Abstand des

Kugelmittelpunktes zum Endpunkt der Strecke. Ist dieser dann kleiner als der Kugelradius, so wird eine

Kollision erkannt. Also $x^2 + y^2 \leq r^2$, mit $\begin{pmatrix} x \\ y \end{pmatrix}$

als Verbindungsvektor vom Endpunkt der Strecke zur Kugel.

Im Programm Also:

```
if ( ( s.x * s.x + s.y * s.y <=
    kugel[ i ].radius * kugel[ i ].radius ) ||
    ( ( kugel[ i ].x - 80 ) * ( kugel[ i ].x
    - 80 ) + ( kugel[ i ].y - 80 ) * (
    kugel[ i ].y - 80 ) <= kugel[ i ].radius *
    kugel[ i ].radius ) )
```

Die komplette Kollisionserkennung im Programm:

```
// Erkennung, ob Kugel über Strecke
// fliegt

d = s.x * n.x + s.y * n.y;

if ( ( s.x * l.x + s.y * l.y > 0 ) &&
    ( l.x * l.x + l.y * l.y > s.x * l.x +
      s.y * l.y ) )
{

// Erkennung Kugel-Endpunkt Kollision

if ( ( s.x * s.x + s.y * s.y <=
    kugel[ i ].radius * kugel[ i ].radius ) ||
    ( ( kugel[ i ].x - 80 ) * ( kugel[ i ].x
    - 80 ) + ( kugel[ i ].y - 80 ) * (
```

```

kugel[ i ].y - 80 ) <= kugel[ i ].radius *
kugel[ i ].radius )
{
    ...
}
// Erkennung Kugel-Geraden Kollision
if ( ( abs_d <= kugel[ i ].radius ) || (
d * d2 < 0 ) )
{
    ...
}
}

```

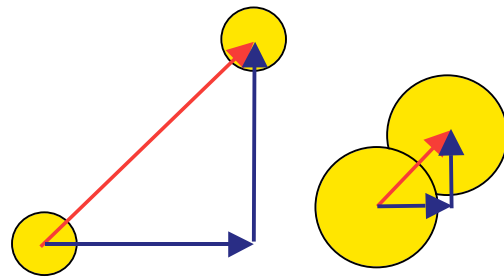
Kugel-Kugel Kollision

Als letztes muss unser Programm noch erkennen können, ob zwei Kugeln miteinander kollidieren. Dies kann man sehr einfach mit Hilfe des Satzes des Pythagoras herleiten.

Man berechnet mit Hilfe von Pythagoras den Abstand der Kugelmittelpunkte der dann kleiner oder gleich groß wie die Summe der Radien ist, wenn die Kugeln kollidieren. Also kollidieren die

Kugeln, wenn gilt: $x^2 + y^2 \leq 2r^2$, wenn $\begin{pmatrix} x \\ y \end{pmatrix}$ der

Verbindungsvektor der Kugeln ist.



Im Programm sieht das dann so aus:

```

if ( ( kugel[ i ].x - kugel[ z ].x ) * (
kugel[ i ].x - kugel[ z ].x ) +
(kugel[ i ].y - kugel[ z ].y ) * (
kugel[ i ].y - kugel[ z ].y ) <= (
kugel[ i ].radius + kugel[ z ].radius ) *
( kugel[ i ].radius + kugel[ z ].radius )
)
{
    ...
}

```

wobei Kugel z eine beliebige andere zu überprüfende Kugel ist.

Kollisionsbehandlung

Die Kollision zwischen zwei Kugeln

Nun kann man die verschiedenen Kollisionen im Programm erkennen. Es fehlt nur noch die Kollisionsbehandlung. Damit ist gemeint, dass Routinen eingebaut werden, die mittels bestimmter Formeln die Kollision wirklich auch *behandeln*, d.h. die Bewegungs- und Geschwindigkeitszustände der

Kugeln verändern. Zunächst wird die Kollision zwischen zwei Kugeln betrachtet:

Es wird gefordert, dass bei einer Kollision zwischen zwei Kugeln der Impulserhaltungssatz gilt. Dieser besagt, dass in einem abgeschlossenen System die Summe der Impulse \vec{P}_{gesamt} konstant bleibt, d.h. es soll gelten (Vektoren mit einem Strich meinen im Weiteren jeweils die Vektoren nach der Kollision):

$$\vec{P}_{\text{gesamt}} = \vec{P}'_{\text{gesamt}} \quad (1)$$

Nun seien \vec{p}_1 und \vec{p}_2 die Impulse der beiden Kugeln vor der Kollision. Es werden nun die durch den Stoß erfolgenden Impulsänderungen $\Delta\vec{p}_1$ und $\Delta\vec{p}_2$ betrachtet.

Nach (1) gilt

$$\begin{aligned} \vec{p}'_1 + \vec{p}'_2 &= \vec{p}_1 + \Delta\vec{p}_1 + \vec{p}_2 + \Delta\vec{p}_2 \\ &\Leftrightarrow \\ \Delta\vec{p}_1 &= -\Delta\vec{p}_2 \quad (2) \end{aligned}$$

Deswegen wird der Einfachheit halber gesetzt:

$$\Delta\vec{p} = \Delta\vec{p}_1 \quad (3)$$

Es seien nun \vec{x}_1 und \vec{x}_2 die Ortsvektoren der beiden Kugeln zum Zeitpunkt der Kollision. Man erwartet, dass die Impulsänderung die Richtung der Differenz von \vec{x}_1 und \vec{x}_2 hat.

Es wird nun gesetzt:

$$\Delta\vec{p} = \lambda \cdot (\vec{x}_1 - \vec{x}_2)^0$$

Für einen Körper in Bewegung gilt:

$$E_{\text{kin}} = \frac{m \cdot \vec{v} \circ \vec{v}}{2} = \frac{\vec{p} \circ \vec{p}}{m \cdot 2} \quad (4)$$

Jetzt wird λ mittels des Satzes der Energieerhaltung bestimmt. Dieser besagt, dass

$E_{\text{kin}} = E'_{\text{kin}}$, was wegen (4) äquivalent ist mit:

$$\frac{1}{2} \cdot \left(\frac{1}{m_1} \vec{p}_1^2 + \frac{1}{m_2} \vec{p}_2^2 \right) = \frac{1}{2} \cdot \left(\frac{1}{m_1} \vec{p}'_1^2 + \frac{1}{m_2} \vec{p}'_2^2 \right)$$

Jetzt wird (2) und (3) verwendet. Wenn die Gleichung vereinfacht wird, erhält man:

$$\frac{\vec{p}_1^2}{m_1} + \frac{\vec{p}_2^2}{m_2} = \frac{(\vec{p}_1 + \Delta\vec{p})^2}{m_1} + \frac{(\vec{p}_2 - \Delta\vec{p})^2}{m_2} \quad (5)$$

Nun wird die rechte Seite ausmultipliziert und umgeschrieben. Man erhält:

$$\Delta\vec{p}^2 \left(\frac{1}{m_1} + \frac{1}{m_2} \right) + 2 \cdot \Delta\vec{p} \circ (\vec{v}_1 - \vec{v}_2) = 0$$

Jetzt wird $\Delta\vec{p} = \lambda \cdot (\vec{x}_1 - \vec{x}_2)^0$ eingesetzt und die Gleichung nach λ aufgelöst. Man erhält:

$$\lambda = 2 \cdot \frac{\vec{v}_2 \circ \vec{n}^0 - \vec{v}_1 \circ \vec{n}^0}{\frac{1}{m_1} + \frac{1}{m_2}} \quad (6)$$

wobei $\vec{n}^0 = (\vec{x}_1 - \vec{x}_2)^0$ ist. Statt des Faktors 2 kann man auch $1 + \varepsilon$ mit $\varepsilon \in [0,1]$ setzen, dabei heißt ε „Restitutionskoeffizient“ und simuliert die Reibung.

Die Kugel-Geraden-Kollision

Eine weiteres Problem bei der Kollisionsbehandlung ist die Kollision einer Kugel mit einer Linie. Relativ bekannt in diesem Zusammenhang ist wohl jedem das Prinzip „Einfallswinkel = Ausfallswinkel“. Dieses mit Hilfe der Vektorrechnung zu formulieren, ist jedoch nicht ganz so einfach wie sich dieses Prinzip anhört, was daran liegt, dass wir dort überhaupt nicht mit Winkeln arbeiten (und dies auch nicht tun wollen).

Bekannt sind uns nur Radius, Orts- und Geschwindigkeitsvektor der Kugel, sowie der Richtungsvektor und ein Punkt der Linie. Ziel der Kollisionsbehandlung ist es, mit diesen Mitteln den Geschwindigkeitsvektor der Kugel so zu verändern, dass das oben genannte Prinzip zutrifft. Das heißt, wir wollen, wie in Abbildung 1.0 gezeigt, den Geschwindigkeitsvektor \vec{v} vor der Kollision (rot) in den neuen Geschwindigkeitsvektor \vec{v}' nach der Kollision (blau) umändern. Dazu zerlegen wir \vec{v} in zwei Komponenten: \vec{v}_1 und \vec{v}_2 , wobei \vec{v}_1 die Komponente parallel, und \vec{v}_2 die Komponente senkrecht zur Geraden ist.

Um \vec{v}_1 zu erhalten, führen wir eine Projektion von \vec{v} auf den normierten Richtungsvektor der Geraden durch. Für \vec{v}_2 gehen wir entsprechend vor: Der Geschwindigkeitsvektor \vec{v} wird auf den Einheitsnormalenvektor der Geraden (ein Vektor mit der Länge 1, der senkrecht zur Geraden steht; man kann ihn aus dem Richtungsvektor der Geraden errechnen) projiziert.

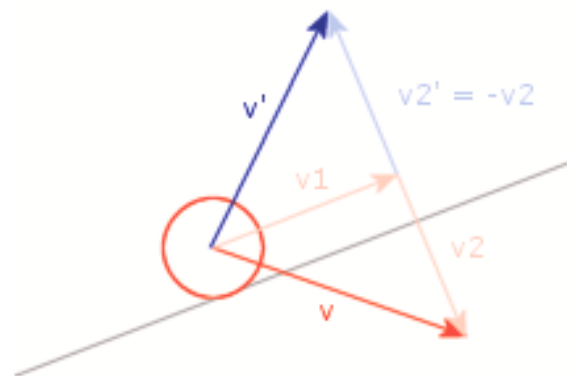


Abbildung 1.0 (Kollision Ball – Linie)

Nun kehren wir \vec{v}_2 um: $\vec{v}_2' = -\vec{v}_2$. Der neue Geschwindigkeitsvektor \vec{v}' ist die Summe aus \vec{v}_1 und \vec{v}_2' .

Dass die Bedingung „Einfallswinkel = Ausfallswinkel“ hier zutrifft ist insofern klar, als dass

bei dieser Vorgehensweise im Endeffekt nur der Geschwindigkeitsvektor an der Linie gespiegelt wird, und da eine Spiegelung eine Kongruenzabbildung ist bleiben alle Winkel erhalten.

Exkursion

Bei der Exkursion aller Kurse und Leiter nach Heidelberg ins DKFZ am 02.09.2003, verbrachte unser Kurs „Pinball Wizard“ die kursinterne Zeit im IWR, dem Interdisziplinären Zentrum für Wissenschaftliches Rechnen. Dort hörten wir einen Vortrag von Ulrich Brandt-Pollmann über Simulationen. Dabei erfuhren wir zuerst allgemein etwas über Simulationen und ihre Bedeutung. Dann bekamen wir erklärt, wo ihre Anwendungsgebiete liegen und aus welchen Gründen man Simulationen bevorzugt (niedrigere Kosten, weniger Zeitaufwand, ethische Gründe...). Nun wurden wir in ein konkretes Problem eingeführt. Hierbei ging es um eine chemische Reaktion. Dabei wurde klar, dass die Simulation gegenüber einer Methode von BASF aufgrund des niedrigeren Aufwands und den somit geringeren Kosten eindeutig im Vorteil ist. Wir bekamen auch noch eine Simulation eines Drachens bei verschiedenen Windverhältnissen gezeigt. Dieser Drache wurde von einem Programm immer auf eine bestimmte Flugbahn gelenkt. Insgesamt kann man sagen, dass es ein sehr interessanter und verständlicher Vortrag war.

Ausblick

Leider wurden wir mit unserem Programm nicht ganz fertig. Glücklicherweise besteht für uns Kursteilnehmer die Möglichkeit, über unsere Internet-Plattform <http://pinwiz.sourceforge.net> das Programm weiter zu entwickeln. Auf diese Art und Weise ist es uns möglich, die Simulation fertig zu schreiben. Wir werden auch versuchen den Kontakt über die Mailingliste aufrecht zu erhalten.

Zum Schluss noch ein Screenshot unserer Arbeit aus Adelsheim. In Aktion ist das Programm natürlich noch um einiges eindrucksvoller. Wer sich dafür interessiert, wie es mit unserem Programm weiter geht, für den besteht über unsere Homepage die Möglichkeit sich mit uns in Verbindung zu setzen.

