

Inhaltsverzeichnis

1	Einleitung	3
2	Erstellen einer virtuellen Umgebung	6
2.1	Szeneneditor	6
2.2	Animation	8
2.3	Darstellung der Objekte	10
2.4	Stereobildanzeige	11
3	Simulationsmodell	13
3.1	Lage und Orientierung eines Körpers im Raum	13
3.2	Mehrkörpersysteme	15
3.3	Bewegungsgleichungen	16
3.4	Verbindungen	17
3.5	Benutzerspezifische Kräfte	18
3.6	Stöße	19
3.7	Berührungsbehandlung	23
3.8	Reibung	24
3.9	Gravitation und Luftwiderstand	25
3.10	Kollisionserkennung	26
4	Berechnungsmethode	27
4.1	Einbindung der Bewegungsgleichungen in das Integrationsverfahren	27
4.2	Das Integrationsverfahren	28
4.3	Allgemeine Funktionsweise	29
4.4	Simulationsdurchführung	30
5	Beispiele	33
5.1	Pendel	33
5.2	Dosenwerfen	34
5.3	Fahrzeugsimulation	34
6	Ausblick	36
	Literaturverzeichnis	37

1 Einleitung

In den letzten Jahren haben durch Computer generierte Filme, sog. Computeranimationen, zunehmend an Bedeutung gewonnen. Besonders interessant sind hierbei Animationen, die aus Simulationen hervorgehen.

Im Bereich der Mechanik wird die Realität aus Gründen der rechnerischen Überschaubarkeit oft auf vergleichsweise simple Probleme reduziert, wie beispielsweise Geschwindigkeit und Position eines Massepunktes zu einem bestimmten Zeitpunkt. Entfernt man sich jedoch von der Abstraktion eines Körpers als Massepunkt und wirkliche Objekte mit Ausdehnungen, dann wird das Problem zunehmend schwieriger: jetzt gibt es Drehgeschwindigkeiten und -beschleunigungen, mehrere Körper können unter unterschiedlichen Winkeln zusammenstoßen, sich berührende Körper üben nicht mehr nur punktförmige, sondern auch flächige Kräfte aufeinander aus, und vieles mehr.

Diese aufwendigen Systeme lassen sich oft nicht mehr mathematisch geschlossen erfassen, sondern nur noch schrittweise simulieren. Wie verhalten sich z.B. einige Kugeln, die aus unterschiedlichen Höhen fallengelassen werden, beim Aufprall auf den unebenen Boden? Wann und wie stoßen die Kugeln in der Folge zusammen? (siehe Abb. 1)

Ein Simulations- und Animationssystem zur Visualisierung solcher Aufgabestellungen ist *AERO*. *AERO* steht für „Animations-Editor für Realistische Objektbewegungen“ und wurde in Form von drei Studienarbeiten am Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR) in der Abteilung Bildverstehen unter der Betreuung von Tho-

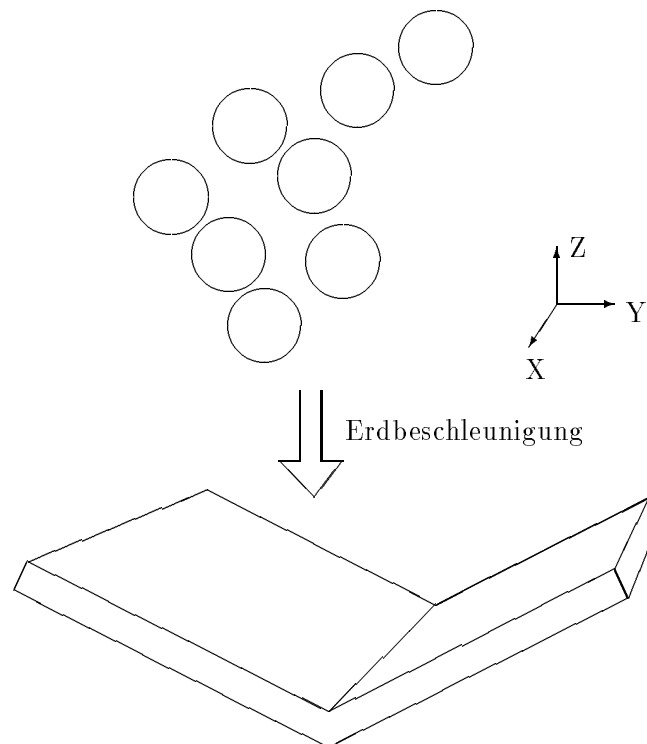


Abbildung 1: Beispiel für eine Simulation

mas Bräunl durchgeführt. Implementiert wurde *AERO* von Andreas Ziegler (3D-Szenen-Editor), Harmut Keller (3D-Grafik, Stereo-Darstellung, Szenengenerierung für Raytracing) und Horst Stolz (Simulationsberechnungen). Realisiert wurde das Projekt unter X-Windows auf UNIX-Workstations.

In *AERO* wird eine virtuelle Welt simuliert, in der sich alle darin definierbaren Körper entsprechend den geltenden physikalischen Gesetzmäßigkeiten bewegen. Auf dem Bildschirm wird dies in animierter Form dargestellt. Während bei anderen Animationsprogrammen die Bewegung überwiegend durch den Benutzer vorgegeben wird, soll in *AERO* ein realitätsnaher Ablauf allein durch Anwendung von Berechnungsvorschriften und Gesetzmäßigkeiten der Mechanik erreicht werden.

Zur Simulation wird mit Hilfe eines im Rahmen des Projekts entwickelten Editors für 3D-Daten eine Szenerie bestehend aus Objekten und angreifenden Kräften erstellt. Anschließend wird die Simulation gestartet und es kann die Bewegung und gegenseitige Beeinflussung der Objekte beobachtet werden. Besonders interessant ist die Möglichkeit, zu jedem Zeitpunkt Änderungen an der Szenerie vorzunehmen und mit den veränderten Parametern die Simulation fortzusetzen oder neu zu starten. Das wünschenswerte Ziel, eine fotorealistische Animation in Echtzeit zu erzeugen, in die der Benutzer interaktiv eingreifen kann, läßt sich auch mit den heutzutage üblichen Rechnern (Workstations) noch nicht erreichen. Darum wurde bei *AERO* ein Weg gewählt, der die Darstellung in drei Stufen erlaubt:

1. **Interaktive Stufe:** Die Szene wird in einer interaktiven Benutzeroberfläche erstellt, berechnet und dargestellt. Wie bei einem Videoschnittpult kann durch Eingreifen des Benutzers der zeitliche Ablauf angehalten, vor- und zurück„gespult“ und auch verändert werden. Durch dieses interaktive Eingreifen wird es möglich, auf das Geschehen der Animation zu reagieren. Wenn z.B. genau zum Zeitpunkt eines Zusammenstoßes von Objekten eine neue externe Kraft angesetzt werden soll, kann dies i.a. erst dann erfolgen, wenn der Zeitpunkt des Zusammenstoßes bekannt ist. In *AERO* wird die Sequenz gestartet und simuliert, bis der Zusammenstoß eintritt. An dieser Stelle wird der Ablauf angehalten und die neue Kraft angesetzt.
2. **Vorausberechnete Sequenz:** Besonders bei sehr aufwendigen Szenen mit vielen Objekten und komplizierten Zusammenhängen ist es möglich, daß die Berechnung der einzelnen Sequenzbilder relativ lange dauert. Darum ist es möglich, eine Sequenz „offline“ (d.h. im Stapelbetrieb (batch)) zu berechnen und anschließend nur noch anzuzeigen.
3. **Vollberechnung (Rendering):** Aus Zeitgründen ist bei den bisherigen beiden Stufen nur eine sehr vereinfachte grafische Darstellung als Drahtgitter möglich. Um fotorealistische Ergebnisse zu erzielen ist ein enormer Rechen- und damit auch Zeitaufwand nötig. In dieser Stufe kann eine Sequenz offline mit allen Details (Licht, Schatten, Oberflächentexturen, usw.) berechnet werden. Dazu kann *AERO* Quelltext für einen externen Raytracer ausgeben. Das Ergebnis, also eine Reihe von ge„rendert“en Bildern, kann dann mit speziellen Anzeige-Programmen als Computeranimation auf dem Bildschirm dargestellt oder auch auf Bildplatte bzw. Videoband festgehalten werden.

AERO wird von den Autoren als Public-Domain-Software kostenlos zur Verfügung gestellt.
Über das Internet kann *AERO* mittels „anonymous ftp“ kopiert werden:
Rechner: `ftp.informatik.uni-stuttgart.de` (zur Zeit 129.69.211.2)
Verzeichnis: `pub/AERO`

2 Erstellen einer virtuellen Umgebung

2.1 Szeneneditor

Nach Start des Programms befindet sich der Benutzer im Szeneneditor (siehe Abb. 2). Der Editor ermöglicht die dreidimensionale Eingabe von Objekten, Kräften und Verbindungen. Da aber nur zweidimensionale Medien zur Eingabe (Tastatur bzw. Maus) und zur Ausgabe (Bildschirm) zur Verfügung stehen, erfolgt die Eingabe über 4 Ansichten: eine Ansicht von vorne (XY-Ansicht), eine von rechts (YZ-Ansicht), eine von oben (XZ-Ansicht) und eine Ansicht von schräg vorne oben rechts (3D-Ansicht).

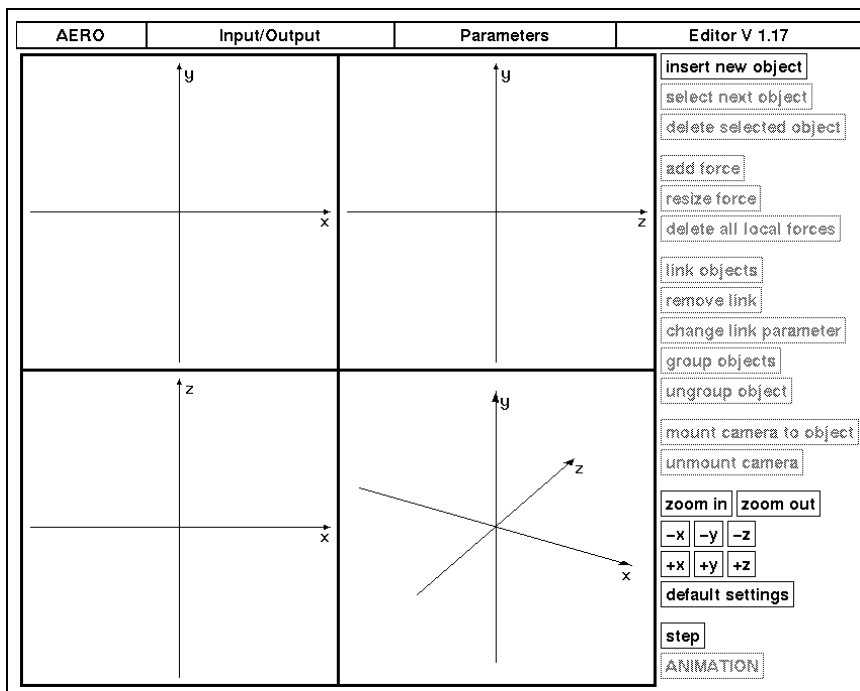


Abbildung 2: Das Editorfenster

Über die angebotenen Menüs können nun Objekte ausgewählt und in den Ansichten platziert werden. Zur Auswahl stehen unterschiedliche Grundkörper: Kugeln, Quader, Zylinder, Ebenen und Fixpunkte. Dabei können die Objektparameter wie beispielsweise Größe und Farbe über ein spezielles Objektfenster verändert werden (siehe Abb. 3). Durch die Auswahl unterschiedlicher Materialien können auch die physikalischen Eigenschaften wie Reibungskoeffizienten und Elastizität beeinflusst werden.

Objekte können miteinander verbunden werden. Dazu werden für jede Verbindung zwei Objekte ausgewählt und die Art der Verbindung angegeben. Es gibt starre Verbindungen, Stangen, Federn, Dämpfer und Gelenke zur Auswahl, deren Parameter ebenfalls eingestellt werden können (z.B. Ruhelänge und Federkonstante bei der Verbindungsart Feder). Auch mehrere Verbindungen zwischen zwei Körpern sind möglich, z.B. ein auf mehreren Federn auf einer Ebene ruhender Quader, oder ein gemeinsam durch Feder und Dämpfer realisierter Stoßdämpfer. Auf diese Weise lassen sich komplexe Objekte zusammensetzen.

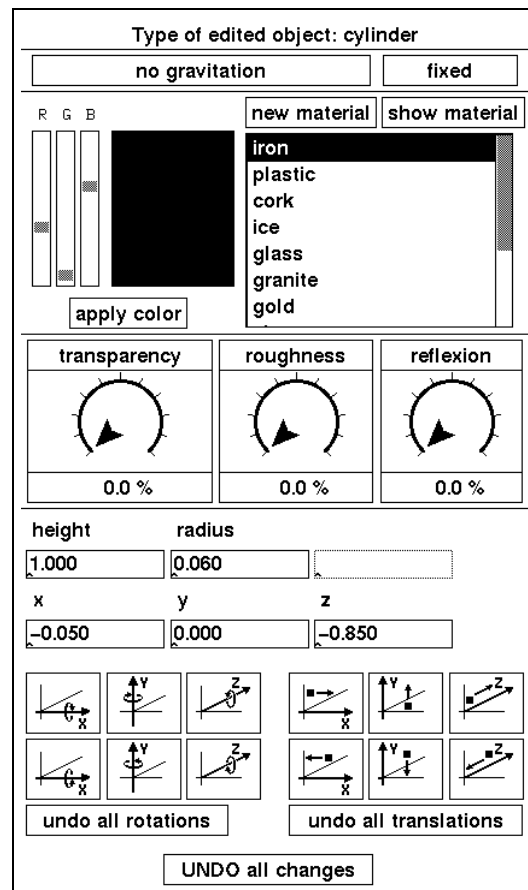


Abbildung 3: Einstellungsmöglichkeiten bei Objekten

Zusätzlich können Kräfte angegeben werden. Damit lassen sich Körper gezielt in eine Richtung beschleunigen oder in Drehung versetzen. Kräfte können auf drei verschiedene Arten angegeben werden:

- lokal zu einem einzelnen Körper
(z.B. eine immer tangential an einer Kugel wirkende Kraft zur Erzeugung eines Drehmoments)
- relativ zu einem anderen Körper
(eine Kraft die immer in Richtung des anderen, sich evtl. bewegenden Körpers zeigt; dies ermöglicht „Verfolgungen“)
- absolut im Raum
(z.B. zur geradlinigen Beschleunigung eines Körpers)

Bei Kräften kann zusätzlich noch das Zeitintervall angegeben werden, in dem die Kraft wirken soll. In Abb. 4 ist das Editorfenster mit einer eingegebenen Szenerie dargestellt. Zu sehen sind alle Körper und Verbindungen.

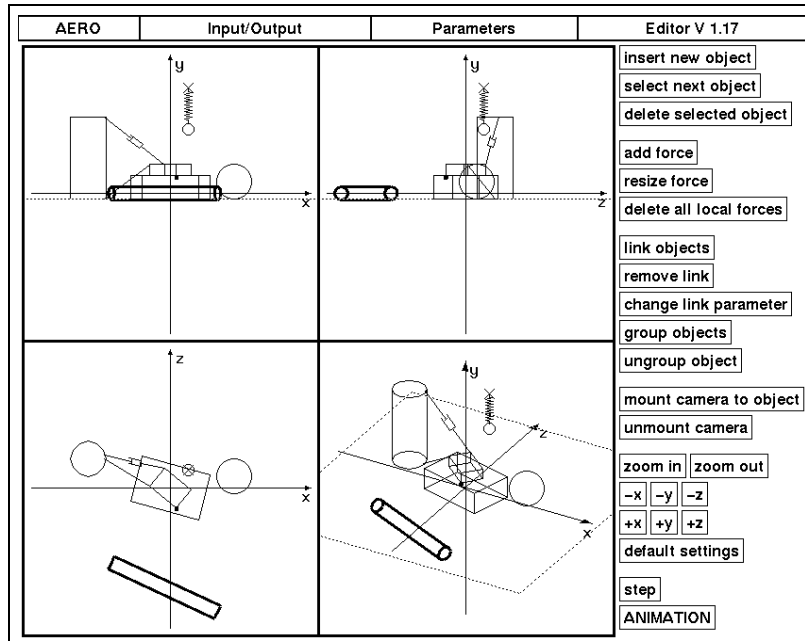


Abbildung 4: Editorfenster mit Objekten

2.2 Animation

Mit einer auf diese Weise eingegebenen Szenerie kann nun die Animation gestartet werden. Dadurch wird automatisch ein neues Fenster geöffnet, in dem die eigentliche Animation abläuft (siehe Abb. 5). Leider kann die Faszination der bewegten Bilder hier nicht vermittelt werden. Um aber einen gewissen Eindruck zu bekommen, sind in Abb. 6 einige Bilder einer Animationssequenz abgebildet.

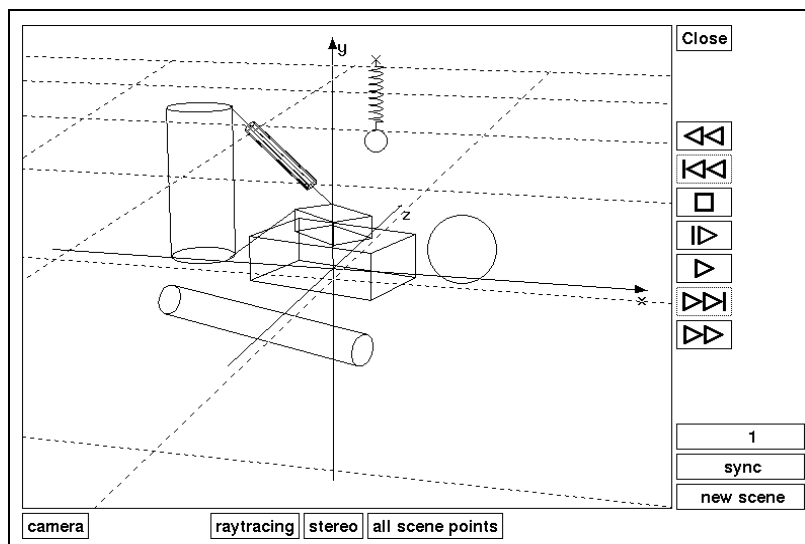


Abbildung 5: Das Animationsfenster

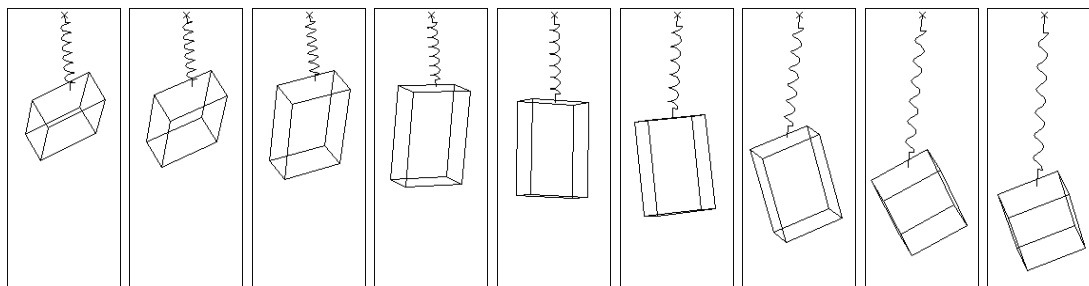


Abbildung 6: Einige Bilder aus einer Animationssequenz

Am rechten Rand des Animationsfensters befinden sich „Tasten“, die an die Laufwerksfunktionen eines Videorekorders erinnern und in *AERO* auch ähnliche Aufgaben haben (siehe Abb. 7). Hier kann im einfachsten Fall mit der Start-Taste die Animation gestartet und angeschaut werden. Entsprechend kann mit den anderen „Tasten“ der Ablauf der Animation beeinflusst werden.

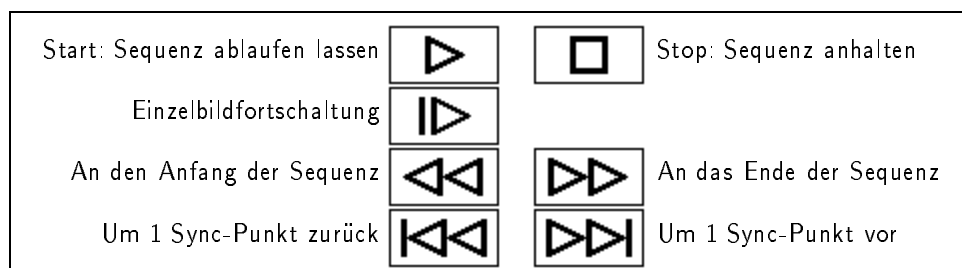


Abbildung 7: Die Lauffunktionen der Animation

Um die Szene aus unterschiedlichen Richtungen betrachten zu können, kann im Animationsfenster der Standort der (virtuellen) Kamera bestimmt werden. Diese Kameraposition kann auch jederzeit während des Animationsablaufs verändert werden.

Etwas außergewöhnlich ist das Konzept der Synchronisations-Punkte (Sync-Punkte). Von einer ablaufenden Animation wird im Programm nur der Anfangszustand und der aktuelle Zustand gespeichert. Alle Ablauffunktionen benutzen die Folgezustandsberechnung, die den aktuellen Zustand in einen neuen aktuellen Zustand überführt. Leider läßt sich die Berechnung nicht ohne weiteres umkehren, um entsprechende Rückwärts-Lauffunktionen zu erhalten. Trotzdem sollte man die Sequenz auch in kleineren Schritten zurück setzen können. Hier setzt die Idee der Sync-Punkte an. Ein Zustand kann jederzeit zu einem Sync-Punkt erklärt werden. Das bedeutet, daß die Ablaufsteuerung zusätzlich diese Stelle speichert und man jederzeit wieder dorthin springen kann. Da sich beliebig viele solcher Sync-Punkte setzen lassen, kann man die Sequenz entsprechend fein unterteilen. Will man also um ein kleines Stück zurücksetzen, springt man einfach rückwärts zum davorliegenden Sync-Punkt und rückt dann mit Einzelbildfortschaltung wieder vorwärts bis zur gewünschten Stelle.

An jeder beliebigen Stelle kann auch eine neue Szene angesetzt werden. Diese ähnelt einem Sync-Punkt, jedoch wird beim Übergang auf einen Szenenanfang *keine* Folgezustandsberechnung aufgerufen, sondern die Information aus diesem „Szenen-Sync-Punkt“

übernommen. Kehrt man also an einer solchen Stelle durch Schließen des Animationsfensters in den Editor zurück, kann man die Szenerie an dieser Stelle ändern und auf diese Weise sozusagen einen „Filmschnitt“ erzeugen.

Während dem Ablauf einer Animation kann die Ausgabe für den Raytracer zugeschaltet werden. Hierbei wird für jedes Bild ein zusätzliches File ausgegeben, das den aktuellen Zustand in Form der Szenenbeschreibungssprache des verwendeten Raytracers enthält. Hier wird der Persistence Of Vision Raytracer, kurz POV-Ray verwendet, der als Public-Domain-Software verfügbar ist. POV-Ray kann z.B. über das Internet mittels „anonymous ftp“ von `alfred.ccs.carleton.ca` (134.117.1.1) bezogen werden.

2.3 Darstellung der Objekte

In den einzelnen Stufen werden die Objekte auf unterschiedliche Art dargestellt (siehe Abb. 8). Im Editor ist es nötig, Entfernungen zwischen Objekten genau bestimmen zu können. Eine perspektivische Verzerrung wäre hier also fehl am Platz und darum wird zur Darstellung eine Parallelprojektion verwendet. Außerdem werden der Übersichtlichkeit wegen Verbindungen nur als Symbole angezeigt.

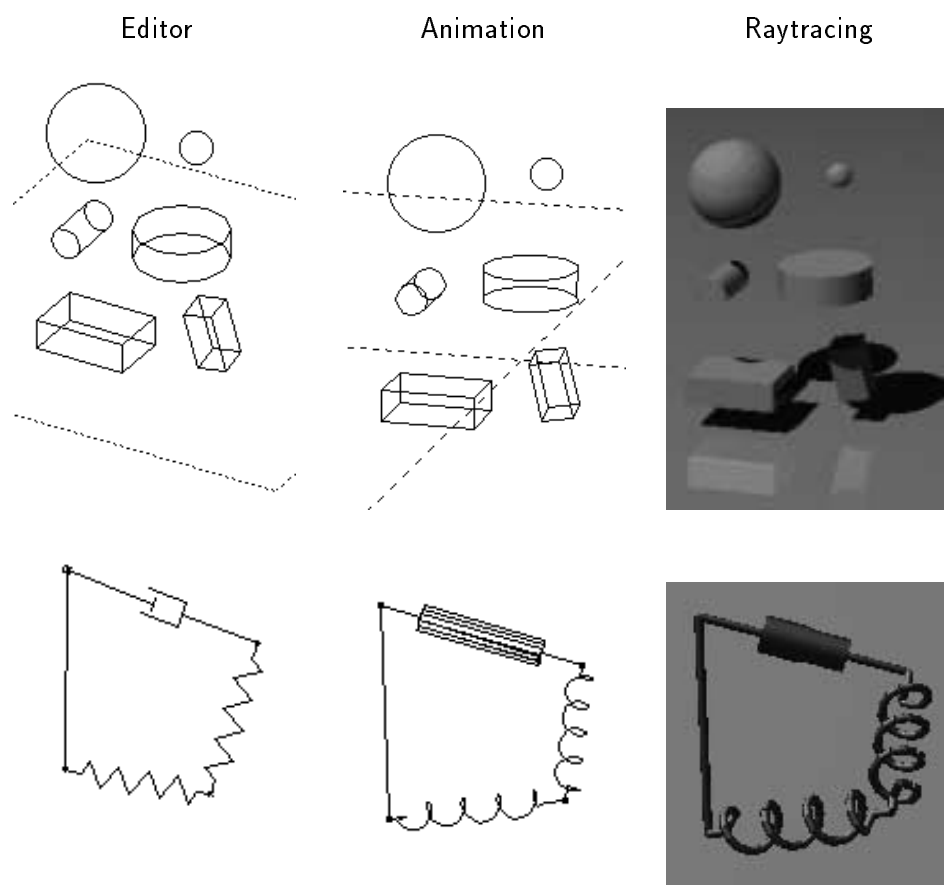


Abbildung 8: Darstellung der verschiedenen Objekte

Im Animationsfenster von *AERO* soll die Anzeige möglichst realistisch erfolgen. Darum wurde hierfür eine perspektivische Projektion (Zentralperspektive) gewählt. Auch die Verbindungen haben hier ihr korrektes Aussehen. So wird eine Federverbindung als Spiralfeder dargestellt, ein Dämpfer ist zylinderförmig.

Wird die fotorealistische Ausgabe über den Raytracer gewählt, erhalten die Objekte noch materialabhängige Oberflächentexturen. So wird ein Körper aus Holz mit einer gewissen Maserung berechnet, Glaskörper sind durchsichtig und es wird die Lichtbrechung beachtet.

2.4 Stereobildanzeige

Bilder sind zweidimensional, was daran liegt, daß sie auf einer Ebene abgebildet sind und ihnen die 3. Dimension, die Tiefe, fehlt. Der Mensch sieht jedoch dreidimensional. Dies kommt dadurch, daß er mit seinen beiden Augen jeweils Bilder aus einem leicht unterschiedlichen Blickwinkel wahrnimmt. Das Gehirn kann daraus Entfernungen bestimmen, oder allgemeiner gesagt, die Tiefe wahrnehmen. Um nun bei einem Film oder eben bei einer Computeranimation einen dreidimensionalen Effekt zu erzielen, muß man jedem der beiden Augen das jeweilige zweidimensionale Bild zur Verfügung stellen, das es auch in der Realität gesehen hätte, also Bilder aus leicht unterschiedlichem Blickwinkel. Bei 3D-Filmen wird dies über zwei Kameras erreicht, die im Augenabstand auf ein Stativ montiert sind. Bei Computeranimationen kann man das zweite Bild natürlich genauso berechnen wie das erste, nur mit einer um den Augenabstand (ca. 6–7 cm) seitlich versetzten und um einen geringen Winkel einwärts gedrehten virtuellen Kamera, um die Parallaxe des menschlichen Augenpaares auszugleichen.

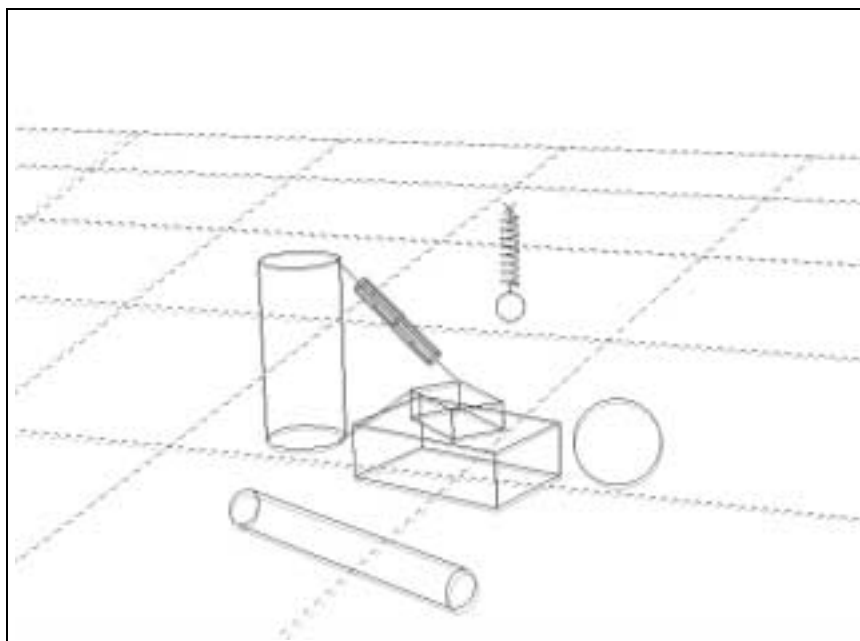


Abbildung 9: Stereoskopische Rot-Grün-Darstellung der Animation

Eine Möglichkeit, diese Bilder den beiden Augen getrennt vorzuführen, ist die sog. Rot-Grün-Stereobildanzeige (siehe Abb. 9). Dabei setzt sich der Betrachter eine Brille auf,

deren rechtes Glass grün und linkes Glass rot eingefärbt ist. Gleichzeitig wird das Bild für das rechte Auge ausschließlich in roten und das für das linke Auge in ausschließlich grünen Farbtönen dargestellt. Durch die Filterwirkung der bunten Brillengläser und die Tatsache, daß Rot und Grün Komplementärfarben sind, sieht der Betrachter mit jedem Auge nur dasjenige Bild, das für dieses Auge auch bestimmt ist. Allerdings ist damit nur eine Quasi-Schwarzweiße Darstellung möglich. Diese Stereo-Darstellung ist im Animationsfenster von *AERO* realisiert.

Eine andere Möglichkeit sind 3D-Bildschirme. Hier wird durch ein LC-Display vor der Bildröhre das von der Mattscheibe ausgehende Licht abwechselnd in eine von zwei Richtungen polarisiert. Synchron dazu wird immer das Bild für das eine und für das andere Auge angezeigt. Setzt sich der Betrachter nun eine spezielle Brille mit ebenfalls polarisierten Gläsern auf, so sieht immer nur ein Auge ein Bild. Zum anderen Auge kann aufgrund der passend gewählten Polarisationsrichtungen kein Licht durchdringen. Durch die wechselnde Polarisation des Bildschirms sieht der Betrachter abwechselnd mit dem linken Auge das eine und mit dem rechten Auge das andere Bild. Wird nun schnell genug umgeschaltet, nimmt man den Wechsel gar nicht mehr wahr und erhält so einen dreidimensionalen Eindruck. *AERO* ermöglicht die Ausgabe solcher Bilder bei der Verwendung des Raytracers.

Bei *AERO* kann sowohl bei der Animationsdarstellung als Drahtgitter als auch bei der Erzeugung von Raytracing Files mit Stereobildern gearbeitet werden. Dies bedeutet, daß pro Animations-Frame je ein Bild für das linke und ein Bild für das rechte Auge erzeugt wird. Werden nun den beiden Augen eines Betrachters diese Bilder getrennt gezeigt, dann wird ein hervorragender räumlicher Effekt erzielt.

3 Simulationsmodell

Der Zweck eines Modells ist eine berechenbare Nachbildung von Vorgängen der Mechanik, die unter den gewünschten Anforderungen der Realität hinreichend genau nahekommmt. Zum einen sollte aufgrund des Einsatzes in einem interaktiven System die Antwortzeit möglichst kurz sein, auf der anderen Seite müssen, um einen realistischen Bewegungsablauf sicherzustellen, möglichst viele physikalische Effekte und Gesetzmäßigkeiten berücksichtigt werden.

Um nun den zeitlichen Verlauf von Bewegungen zu erhalten sind sogenannte **Bewegungsgleichungen** nötig. Diese stellen eine mathematische Beziehung zwischen den Bewegungsvariablen, wie Position, Geschwindigkeit und Beschleunigung, und den angreifenden **Kräften** her.

Ein weiterer wichtiger Punkt ist die Modellierung von physikalischen Elementen. Die Begrenzung auf starre räumliche **Körper** liegt an dem Vorhandensein von fundiertem, erprobten Wissen auf das man zurückgreifen kann. Zwar existieren auch Modelle welche die Benutzung von elastischen, deformierbaren Körpern zulassen. Da deren Ursprung in den Finiten-Elementen-Methoden liegen ist eine zeitkritische Anwendung, wie in diesem Fall, jedoch unangebracht. So bleiben auch Flüssigkeiten und Gase unberücksichtigt, lediglich eine rudimentäre **Luftwiderstandskraft** wird eingeführt.

Soll eine Durchdringung von Körpern verhindert werden, so geschieht dies entweder durch den Impuls beim **Stoß**, oder durch Kräfte die bei der **Berührung** von Körpern entsteht. Vor dieser Kollisionsbehandlung steht aber erst die **Kollisionserkennung**, welche den Schnitt zwischen zwei räumlichen Objekten bestimmt und die für die Behandlung notwendigen Daten ermittelt.

Zwangsbedingungen¹ sollen in Form von **Verbindungen** die Kopplung zwischen Körpern definieren. Diese gehen wie auch die **Gravitation** und **Reibung** als Kräfte in die Bewegungsgleichungen ein. Schließlich können auch noch **benutzerspezifische Kräfte** angebracht werden um Körper in Bewegung zu versetzen.

3.1 Lage und Orientierung eines Körpers im Raum

Zunächst ist es nötig die Lage und Orientierung eines Körpers im 3-dimensionalen Raum zu definieren.

Um die Lage eines Körpers zu beschreiben genügt ein Vektor² \vec{r}_{OS} vom Ursprung O des raumfesten Koordinatensystems K zum Schwerpunkt S .

Um die Verdrehung zwischen dem körperfesten Koordinatensystem K' zum Inertialsystem K zu definieren werden Eulerparameter, auch Quaternions genannt, herangezogen. Wird K um die Drehachse \vec{u} im Uhrzeigersinn um den Winkel γ gedreht erhält man K' . Dabei

¹Das sind gewöhnlich Gleichungen in Abhängigkeit von den Körperpositionen mit der Form $\varphi(x_1, \dots, x_n) = 0$

²In den Bildern sind Vektor durch Fettdruck erkennbar; die Vektoren mit Apostroph sind vom Koordinatensystem K' aus zu sehen.

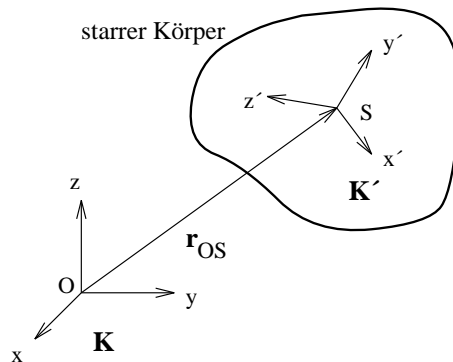


Abbildung 10: Lage eines Körpers im Raum

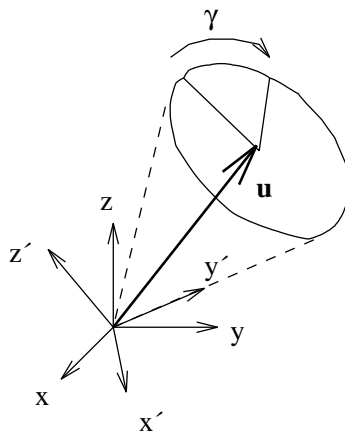


Abbildung 11: Verdrehung zwischen zwei Koordinatensystemen

berechnen sich die vier Eulerparameter $q = (q_0, q_1, q_2, q_3)$ zu $q_0 = \cos \frac{\gamma}{2}$ und $(q_1, q_2, q_3)^T = \vec{u} \cdot \sin \frac{\gamma}{2}$.

Eine Annehmlichkeit der Quaternionen ist die einfache Aufaddierung von Drehungen durch $q \times v = (q_0, \vec{q}) \times (v_0, \vec{v}) = (q_0 \cdot v_0 - \vec{q} \cdot \vec{v}, q_0 \cdot \vec{v} + v_0 \cdot \vec{q} + \vec{q} \times \vec{v})$.

Zur Umrechnung eines Vektors \vec{r}' aus dem Koordinatensystem K' in den Vektor \vec{r} des Koordinatensystems K dient die Matrix A :

$$A = \begin{pmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{pmatrix}$$

Hieraus ergeben sich die Beziehungen $\vec{r}' = A \cdot \vec{r}$ und $\vec{r} = A^{-1} \cdot \vec{r}'$. Die Umkehrmatrix A^{-1} erhält man einfach durch Umkehrung der Drehachse \vec{u} . Man setzt statt (q_0, q_1, q_2, q_3) lediglich $(q_0, -q_1, -q_2, -q_3)$ in obige Matrix-Gleichung ein. Eine detailliertere Darstellung von Euler-Parameter, Euler-Winkeln und Bryant-Winkeln findet sich unter anderem in [Wittenburg 77], die Beziehung zwischen zwei Drehungen stammt hingegen aus [Shoemake 85].

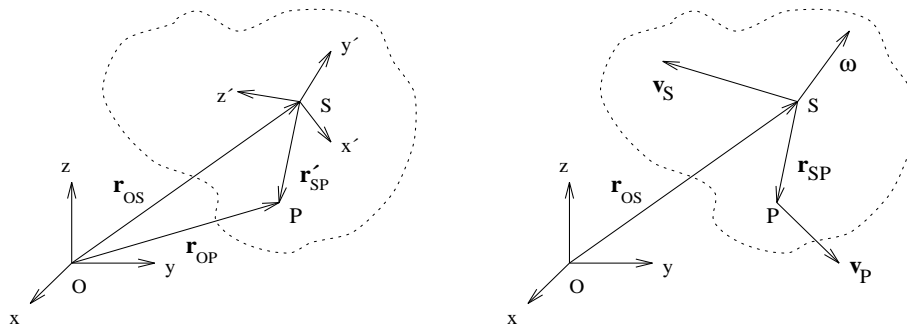


Abbildung 12: Position und Geschwindigkeit eines körperfesten Punktes

Somit ergeben sich für einen beliebigen körperfesten Punkt P folgende Formel:

$$\vec{r}_{OP} = \vec{r}_{OS} + \vec{r}_{SP} = \vec{r}_{OS} + A^{-1} \cdot \vec{r}'_{SP}$$

Für die Geschwindigkeit $\vec{v} = \dot{\vec{r}}$ des Punktes P gilt weiter:

$$\vec{v}_P = \vec{v}_S + \vec{\omega} \times \vec{r}_{SP}$$

Dabei ist $\vec{\omega}$ der Drehgeschwindigkeitsvektor, dessen Richtung die Drehachse um den Schwerpunkt angibt und $|\vec{\omega}|$ die Umdrehungsgeschwindigkeit um diese Achse.

3.2 Mehrkörpersysteme

Aus der Überlegung dem Benutzer einen Baukasten zu geben mit dem er experimentieren kann entstand ein Modell das aus 6 Grundkörpern besteht, welche eine geometrische Form haben und physikalische Eigenschaften entsprechend einem realen Material. Um den Rechenaufwand möglichst klein zu halten soll ein System aus starren, also nicht deformierbaren, Körpern verwendet werden. Man spricht dann in der Mechanik auch von einem Mehrkörpersystem. Einige einfache vielseitig einsetzbare räumliche Formen sollen den Zwang der Annäherung von komplexen Objekten durch Polygonflächen vermeiden und somit die Anforderungen bei der Kollisionsberechnung verringern.

Zu einem Körper gehören dabei unterschiedliche **Attribute**: Masse³, eine räumliche Ausdehnung und Materialeigenschaften. Jedes Attribut hat dabei Auswirkungen auf die Simulation. So ist erst durch das Vorhandensein einer Masse die Bewegung bei Krafteinwirkung möglich. Nur Körper mit einer räumlichen Ausdehnung können kollidieren. Schließlich beeinflussen die Materialeigenschaften den Stoß, die Berührung, und die Reibung zwischen Körpern.

Nicht jeder der Körper in Abbildung 13 besitzt alle Eigenschaften. So haben bis auf Nagel und Massepunkt alle Körper eine räumliche Ausdehnung und somit auch Materialeigenschaften. Allerdings hat die unendliche Ebene keine Masse, da sie nur ein 2-dimensionales

³Genauer gesagt Masse m und Trägheitstensor J , wobei letzterer in jedem technischen Tabellenbuch für die gebräuchlichsten räumlichen Formen nachgeschlagen werden kann.

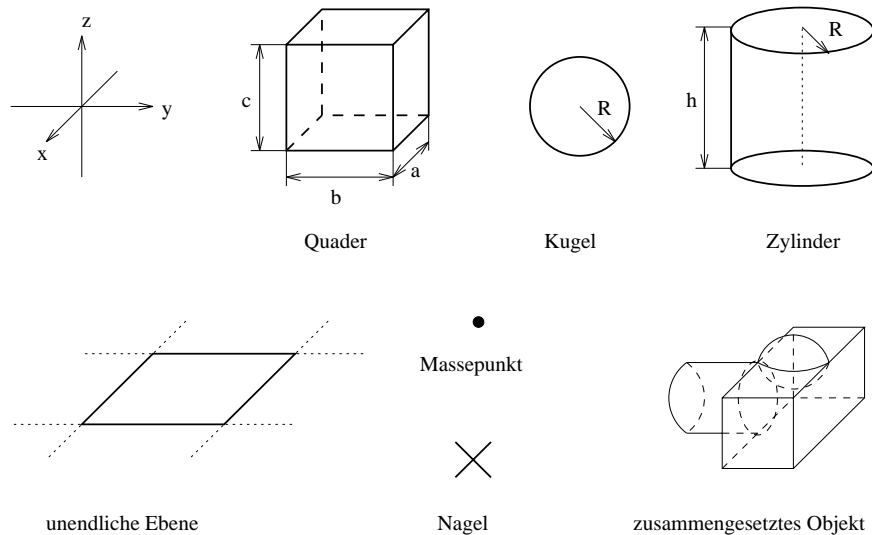


Abbildung 13: Verfügbare Körperarten

Objekt ist, was zu Folge hat, daß sie sich nicht bewegen kann. Auch der Nagel hat die Masse 0kg , somit hat man einen raumfesten Punkt an dem durch Verbindungen andere Körper festgemacht werden können. Der Massepunkt hingegen kann sich bewegen, läßt sich aber durch räumliche Hindernisse nicht aufhalten.

Der zusammengesetzte Körper wiederum kann aus Massepunkten, Quadern, Kugeln und Zylindern bestehen und ermöglicht das Erstellen eines komplexen starren Körpers. Die zueinander unbeweglichen Grundkörpern werden dabei mit ihren entsprechenden Materialeigenschaften berücksichtigt.

Neben den obigen 3 Attributen sind noch weitere wie Farbe, Textur usw. vorhanden um die Darstellung auf dem Bildschirm zu regeln. Diese Attribute sind für die Berechnung des Folgezustandes allerdings irrelevant. Siehe hierzu [Ziegler 93] und [Keller 93], die für die Auswahl und Darstellung dieser Attribute verantwortlich sind.

3.3 Bewegungsgleichungen

Durch die Wahl des Körperschwerpunktes als Bezugspunkt für eine starren Körper ergeben sich einfachste Formen für den Impuls- und Drallsatz, welche zusammengenommen die Bewegungsgleichungen eines Körpers darstellen.

$$m \cdot \ddot{\vec{r}}_{OS} = \sum_{i=1}^n \vec{F}_i \quad (\text{Impulssatz})$$

$$J_S \cdot \dot{\vec{\omega}}_S = \vec{M}_S + \sum_{i=1}^n (\vec{r}_i \times \vec{F}_i) \quad (\text{Drallsatz})$$

Dabei sind \vec{F}_i die an dem Körper mit der Masse m angreifenden Kräfte, $\ddot{\vec{r}}_{OS} = \dot{\vec{v}}_S = \vec{a}_S$ ist

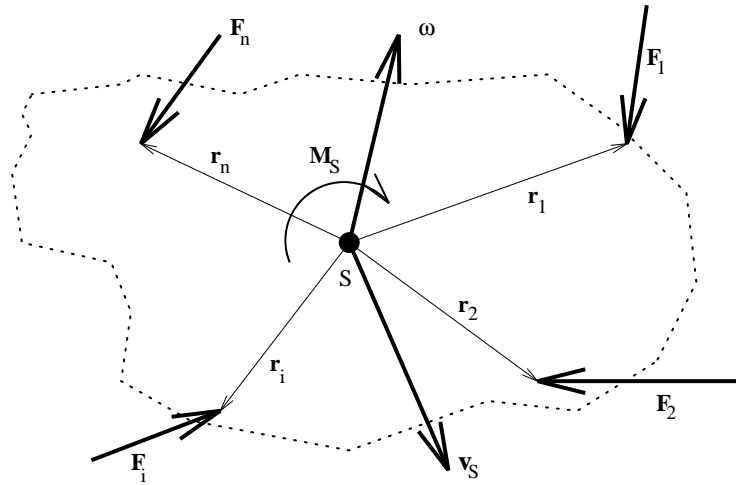


Abbildung 14: Kräfte am starren Körper

die Beschleunigung des Massenschwerpunktes und \vec{w}_S die Drehbeschleunigung. Die Matrix J_S ist der sogenannte Trägheitstensor, er repräsentiert die räumliche Massenverteilung um den Schwerpunkt, vergleichbar mit der Trägheitseigenschaft der Masse m für den Impulssatz. So ist zum Beispiel die Drehgeschwindigkeit einer Pirouette im Eiskunstlauf wegen dem größeren Trägheitsmoment bei ausgestreckten Armen kleiner als bei anliegenden.

Weil die Unbekannte \vec{r}_{OS} in den Bewegungsgleichungen mit ihrer zeitlichen Ableitung vertreten ist spricht man von einem Differentialgleichungssystem. Dieses kann mit einem **numerischen Integrationsverfahren** gelöst werden.

3.4 Verbindungen

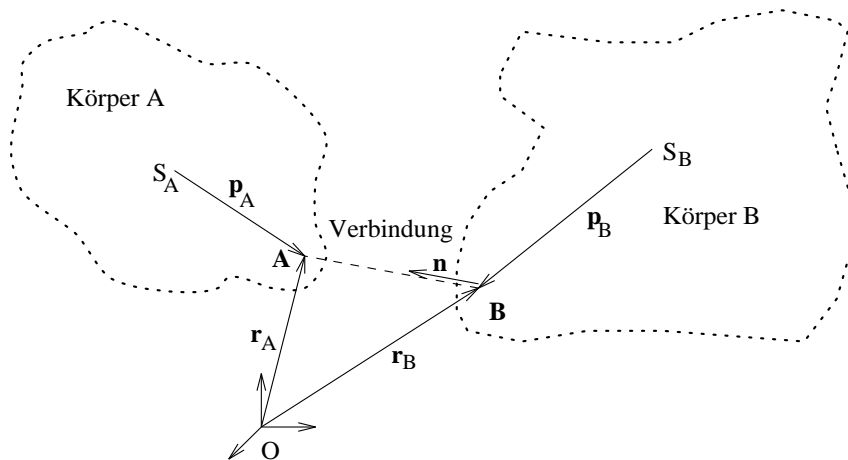


Abbildung 15: Verbindung zwischen zwei Körpern

Eine Verbindung soll einen Punkt A eines Körpers mit einem Punkt B eines anderen Körpers durch eine mathematische Beziehung koppeln. Dazu wird an den Verbindungs-

stellen eine der Verbindungsart entsprechende Kraft⁴ angebracht.

Wegen ihrer universellen Verwendbarkeit, der allgemein bekannten Funktionsweise und ihrer einfachen Implementierbarkeit wurde eine Feder, ein Dämpfer sowie ein Stange, die aus einem kombinierten Feder-Dämpfer-Glied besteht, als Verbindungsarten gewählt. Setzt man die Stangenlänge auf 0 erhält man angenähert ein Kugelgelenk.

Zu jedem Zeitpunkt werden die Kraftgleichungen ausgewertet und das Ergebnis in die Bewegungsgleichungen eingesetzt.

Es sollte an dieser Stelle darauf hingewiesen werden, daß eine aus Feder und Dämpfer zusammengesetzte Stange sich in ihrer Länge verändern muß um eine Kraft zu übertragen, es handelt sich hierbei um einen Art "Stoßdämpfer". Durch die Wahl der Parameter c und d läßt sich jedoch eine Stange hinreichend genau simulieren.

Ein allgemeineres Modell zu Kopplung verschiedener Körper sind durch Zwangsbedingungen erreichbar. So zeigen [Witkin 90], [Barzel et al. 88] und [Isaacs 87] analytische Methoden auf die allgemeinere Verbindungsarten zulassen, somit auch die Realisierung von starren Stangen und Gelenken. Diese Verfahren sind im Bereich der Simulation für technisch-wissenschaftliche Zwecke⁵ häufig im Einsatz und beruhen auf den Methoden der analytischen Mechanik.

Dieser Weg wurde aus zwei Gründen nicht gewählt:

- Sollen Berührungskräfte oder Reibungskräfte auch noch analytisch berücksichtigt werden erhält man ein nahezu beliebiges Gleichungssystem.
- Die starre Verbindung würde beim Stoß auch Impulse weitergeben, was die Stoßbehandlung noch schwieriger macht.

3.5 Benutzerspezifische Kräfte

Um Körper zusätzlich zu manipulieren, können konstante Kräfte, die an einem festen Punkt des Körpers angreifen und in einem vorgebbaren Zeitintervall wirken, angebracht werde. Die Richtung der Kraft ist entweder in Körperkoordinaten wie \vec{F}'_1 , in raumfesten Koordinaten wie \vec{F}_2 oder durch Angabe eines weiteren Punktes auf einem anderen Körper (\vec{r}_4, \vec{F}_3) möglich.

Mit diesen drei Typen lassen sich gleichmäßig beschleunigte Translations- und Rotationsbewegungen erstellen.

Zum Beispiel eine reine Rotationsbewegung läßt sich durch symmetrisches Anbringen zweier gleichgroßer Kräfte in Körperkoordinaten wie in Abb. 17 erreichen. Die translatorische Beschleunigung der beiden Kräfte hebt sich hier gegenseitig auf, man erhält das Moment $\vec{M} = 2 \cdot \vec{r}' \times \vec{F}'$.

⁴ $\vec{n} = \vec{r}_A + \vec{r}_B$ ist dabei der Normalenvektor zwischen den Punkten A und B mit $|\vec{n}| = 1$, und $x = |\vec{r}_A - \vec{r}_B|$ der Abstand zwischen den beiden Punkten. Die Geschwindigkeit ergibt sich aus $\vec{v} = \vec{v}_S + \vec{\omega} \times \vec{p}$.

⁵Weil sie auch die exakte Berechnung von Zwangskräften erlauben, weiter wird auf die Arbeiten in [Haug 84] verwiesen.

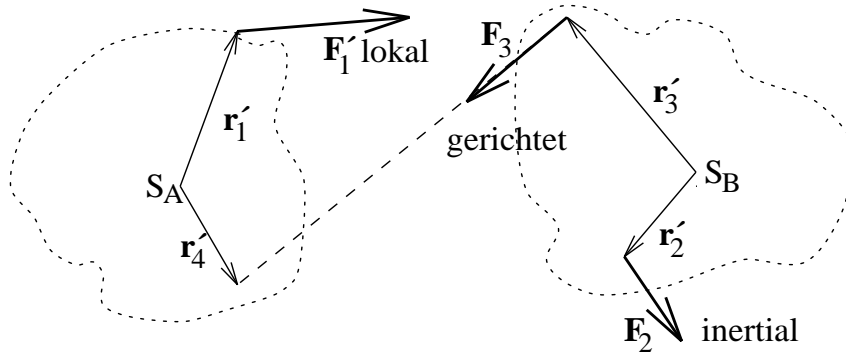


Abbildung 16: Mögliche Kraftarten

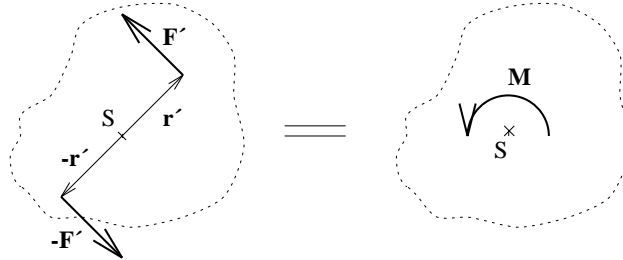


Abbildung 17: Erzeugung eines Drehmoments durch zwei Kräfte

3.6 Stöße

In Bild 18 sieht man die allgemeine Darstellung einer Kollision zwischen zwei Körpern.

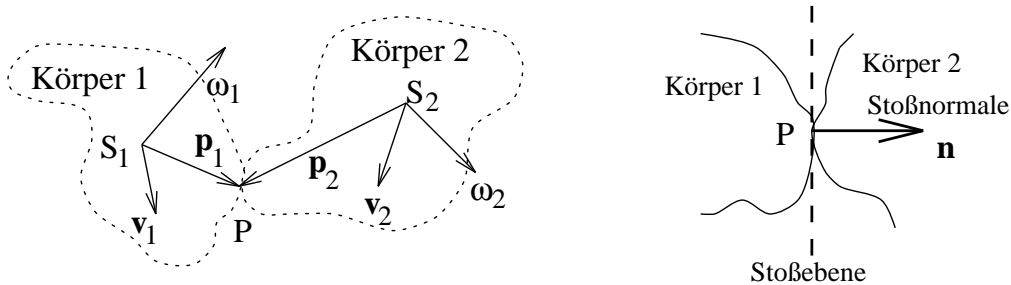


Abbildung 18: Stoß zweier Körper, Normalenrichtung

Neben der Normalenrichtung \vec{n} des Stoßes und der Durchdringungstiefe γ ist die Normallengeschwindigkeit im Kollisionspunkt P

$$v_N = \vec{n} \cdot (\vec{v}_1 + \vec{\omega}_1 \times \vec{p}_1 - \vec{v}_2 - \vec{\omega}_2 \times \vec{p}_2)$$

der beiden Körper von Bedeutung. Wenn $v_N = 0$, ist P relativ in Ruhe. Man spricht dann auch von einer **Berührung**. Diese wird genauer im nächsten Abschnitt behandelt. Der Fall $v_N < 0$ wird als **Loslösung** bezeichnet, den die Körper bewegen sie sich voneinander weg. Ein **Stoß** tritt erst für $v_N > 0$ auf.

Die Stoßzahl ϵ modelliert nun das Materialverhalten der involvierten Körper:

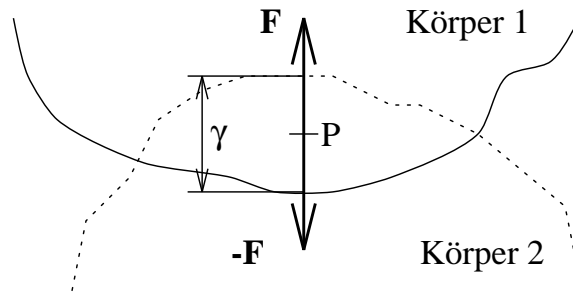


Abbildung 19: Kollisionsbehandlung durch eine Feder

$\epsilon = 1$: **Elastischer Stoß** — ein an die Wand geworfener Ball kommt mit derselben Geschwindigkeit zurück (v_{N1} und v_{N2} drehen sich um).

$0 < \epsilon < 1$: **Teilelastischer Stoß** — Normalengeschwindigkeiten werden entsprechend $(v_{N1} - v_{N2}) = \epsilon(v'_{N1} - v'_{N2})$ verteilt.

$\epsilon = 0$: **Plastischer Stoß** — Normalgeschwindigkeiten sind nach dem Stoß bei beiden Körpern gleich.

Zur Behandlung von Stößen, stehen zwei Verfahren zur Auswahl:

- **Kollisionsbehandlung durch Einsetzen von Federn**

Bei Auftreten einer Kollision wird eine steife Feder in den Kollisionspunkt eingesetzt. Die Federkraft ist dabei proportional zur Durchdringungstiefe. Die Federkonstante c ist dabei abhängig von der Materialpaarung. Auch wird die Stoß-Elastizität der Materialien bei der Loslösung der Körper voneinander berücksichtigt.

$$\vec{F} = \begin{cases} \gamma c \vec{n} & \text{für } v_N \geq 0 & (\text{Eindringen}) \\ \epsilon \cdot \gamma c \vec{n} & \text{für } v_N < 0 & (\text{Loslösung}) \end{cases}$$

Nachteilig ist dabei der große Berechnungsaufwand, der durch die Wahl von hohen Federkonstanten c entsteht, weshalb die Zeitschrittweite des Integrationsverfahrens erhöht werden muß um genau genug zu rechnen. Wählt man c zu niedrig, durchdringen sich die Körper je nach Masse und Geschwindigkeit sehr stark und bewegen sich unter Umständen sogar durcheinander durch.

- **Analytische Stoßbehandlung**

Die Analytische Methode bestimmt aus den momentanen Translations- und Rotationsgeschwindigkeiten $(\vec{v}_1, \vec{\omega}_1, \vec{v}_2, \vec{\omega}_2)$ die neuen Geschwindigkeiten $(\vec{v}'_1, \vec{\omega}'_1, \vec{v}'_2, \vec{\omega}'_2)$ nach dem Stoß unter Verwendung der Impuls- und Drallerhaltung (es geht somit keine Energie verloren).

$$\begin{aligned} m_1 \vec{v}'_1 &= m_1 \vec{v}_1 + \vec{R} \\ m_2 \vec{v}'_2 &= m_2 \vec{v}_2 - \vec{R} \end{aligned}$$

$$J_1 \vec{\omega}'_1 = J_1 \vec{\omega}_1 + \vec{p}_1 \times \vec{R}$$

$$J_2 \vec{\omega}'_2 = J_2 \vec{\omega}_2 - \vec{p}_2 \times \vec{R}$$

Um sicherzustellen, daß der Stoßimpuls \vec{R} , der zwischen den Körpern im Stoßzeitpunkt ausgetauscht wird, in Normalenrichtung verläuft, benutzt man noch:

$$\vec{R} \cdot \vec{e}_1 = 0$$

$$\vec{R} \cdot \vec{e}_2 = 0$$

$$\vec{n} \cdot (\vec{v}'_2 + \vec{\omega}'_2 \times \vec{p}_2 - \vec{v}'_1 - \vec{\omega}'_1 \times \vec{p}_1) = \epsilon \cdot \vec{n} \cdot (\vec{v}'_1 + \vec{\omega}'_1 \times \vec{p}_1 - \vec{v}'_2 - \vec{\omega}'_2 \times \vec{p}_2)$$

Dieses lineare Gleichungssystem mit den 15 Unbekannten⁶ $\vec{v}'_1, \vec{v}'_2, \vec{\omega}'_1, \vec{\omega}'_2$ und \vec{R} kann durch Standard-Verfahren wie den Gauß-Algorithmus gelöst werden. Insgesamt ist die analytische Behandlung schneller als das Einsetzen einer Feder, da sie nur für den Zeitpunkt des Stoßes, nicht aber über einen längeren Zeitraum angewandt werden muß.

Der Positionsverlauf einer Kugel die aus 0,05m Höhe auf den Boden prallt ist in Abb. 20 zu sehen.

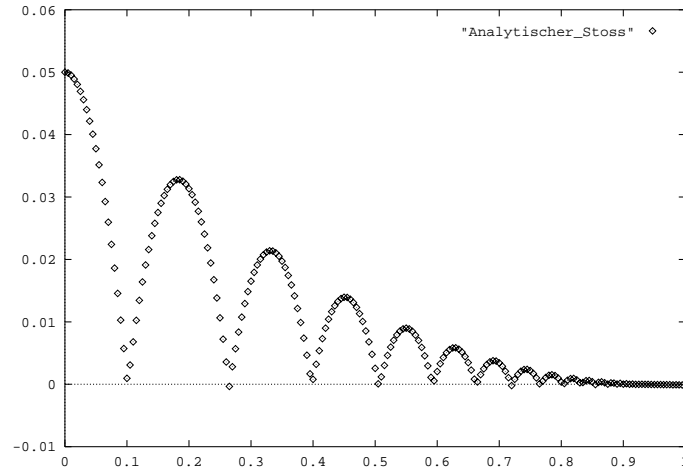


Abbildung 20: Positionsverlauf einer zu Boden fallenden Kugel mit $\epsilon = 0.81$

Daß man auch Reibungskräfte, z.B. für 2-dimensionale Anwendungen in der Arbeit von [Wang et al. 92] beschrieben, und zusätzlich auch die Stoßimpulsweitergabe von verbundenen Körperstrukturen mitberücksichtigen kann, steht in [Moore et al. 88], der sich dabei auf [MacMillan 60] beruft. Auch [Wittenburg 77] widmet in seinem Buch ein Kapitel der Stoßbehandlung von verbundenen Mehrkörpersystemen.

Doch das entscheidende Problem, die Behandlung von mehreren gleichzeitigen Stößen an einem Körper bleibt damit ungelöst.

⁶Dieses kann ohne weiteres auf 13 Unbekannte reduziert werden da \vec{R} parallel zu \vec{n} verläuft.

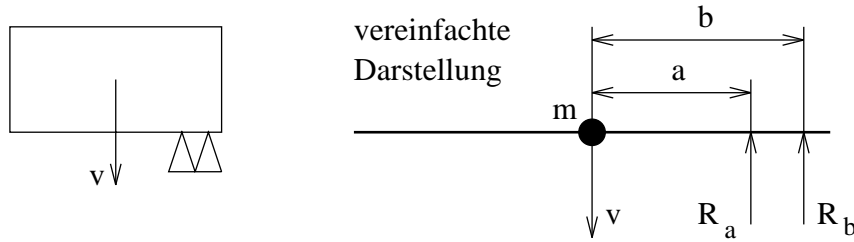


Abbildung 21: Quader mit zwei einseitigen Kollisionen

Betrachtet man einen Quader, der an zwei Stellen (wie in Bild 21) kollidiert, würde sich vereinfacht folgendes ergeben:

$$\begin{aligned}
 mv' &= mv + R_a + R_b \\
 Jw' &= Jw - R_a - R_b \\
 \left. \begin{aligned} v' + w'a &= 0 \\ v' + w'b &= 0 \end{aligned} \right\} \quad da \quad a \neq b \quad \Rightarrow \quad w' = 0
 \end{aligned}$$

Das heißt, der Quader würde sich nicht drehen, obwohl er einseitig aufkommt. Solange nun keine einfach anwendbare physikalische Grundlage vorhanden ist, bleibt man darauf beschränkt Stöße, die gleichzeitig auftreten, nacheinander zu behandeln.

In dieser Implementierung wird zu einem Zeitpunkt die Stoßbehandlung von Punkten mit $v_N > 0$ so lange durchgeführt bis das sich alle Kollisionen voneinander lösen $v_N \leq 0$. Es werden aber maximal $2k$ Schritte durchgeführt, wenn k die Anzahl der aufgetretenen Kollisionen ist.

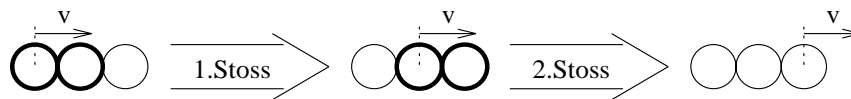


Abbildung 22: Stoßimpulsweitergabe von 3 Kugeln

Diese Weitergabe hat die nachteilige Auswirkung, daß z.B. ein Quader der auf seinen Eckpunkten wie in Abbildung 23 aufkommt, sich um eine Achse parallel zur Aufprallfläche zu drehen beginnt.

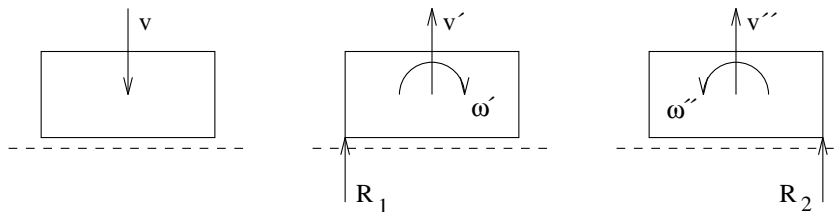


Abbildung 23: Quader fällt parallel auf eine Ebene

Deshalb hat der Benutzer die Möglichkeit zwischen der Federmethode, welche das obige Problem nicht kennt, und der analytischen Behandlung zu wählen.

3.7 Berührungsbehandlung

Man könnte theoretisch auch die Stoßbehandlung aus dem vorigen Kapitel für Berührungspunkte mit $v_N \approx 0$ anwenden, was auch tatsächlich geschieht wenn die Feder-Methode gewählt wird.

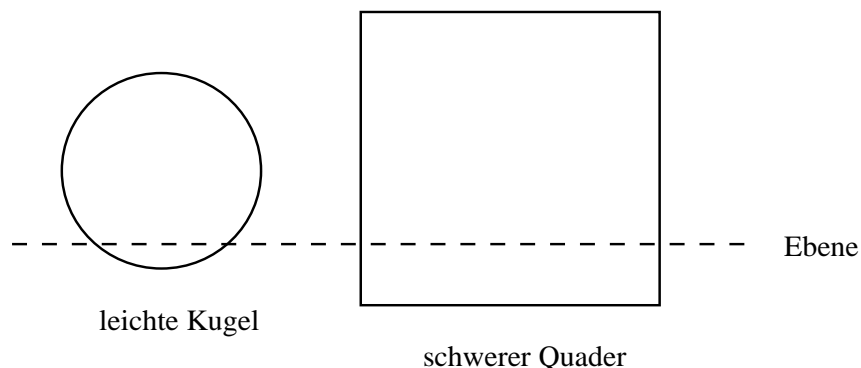


Abbildung 24: Berührungsbehandlung durch Penalty-Methode

Die analytische Stoßbehandlung ist dagegen zu aufwendig für die Benutzung in jedem Zeitschritt. Statt dessen wird eine "Penalty-Methode" mit besserem Einschwingverhalten als die Feder-Methode verwendet.

Dazu wird in jedem Punkt, dessen $|v_N|$ unterhalb einer vorgebbaren maximalen Berührungsgeschwindigkeit v_{Nmax} liegt, eine Kraft entgegen der Durchdringung angesetzt. Diese ist proportional zur Durchdringungstiefe γ und proportional gegen die Normalengeschwindigkeitsrichtung \vec{v}_N . Zur Parametrisierung steht für jedes Material der Dehnungsfaktor c und der Dämpfungsfaktor d zur Verfügung.

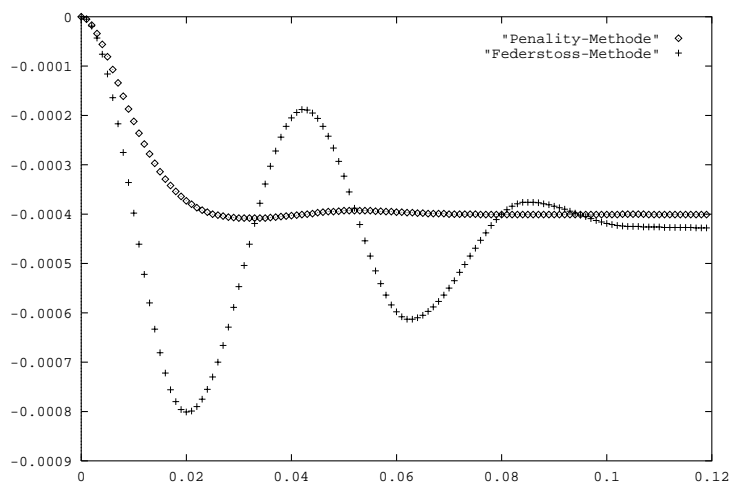


Abbildung 25: Vergleich des Einsinkens einer Kugel in eine Ebene mit beiden Berührungsbehandlungen

Durch eine geeignete Wahl von d kann die Einschwingzeit sowie ein starkes Überschwingen

beim Berührungsvorgang vermieden werden. Bild 25 zeigt das Einsinken einer Eisenkugel in eine Eisenebene mit beiden Methoden. Auf den vertikalen Achse ist dabei die Einsinktiefe in Metern aufgetragen, die horizontale Zeit-Achse hat als Einheit Sekunden.

Die analytische Behandlung von Berührungskräften, wie in [Baraff 89] beschrieben, wurde nicht implementiert. Sie stellt aber sicher ein lohnendes Ziel dar, da die unterschiedlichen Eindringtiefen von Körpern, wie in Bild 24 zu sehen, damit nicht mehr vorkommen. Außerdem erlaubt sie auch die Einbindung von Kräften die durch Zwangsbedingungen entstehen. Es sei jedoch angemerkt, daß bei der Beschäftigung Behandlungsverfahren eine Unteroutine zu Lösung von LCP-Problemen⁷ mittels Lemke's Complementary Pivot Algorithmus entstanden ist, und somit ein Fundament für zukünftige Erweiterungen in dieser Richtung zur Verfügung steht.

3.8 Reibung

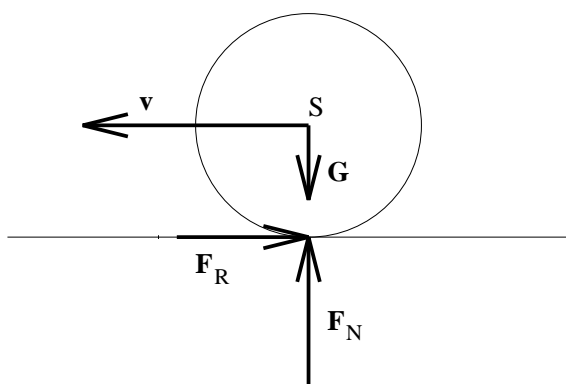


Abbildung 26: Gleitreibung bringt rutschende Kugel zum Rollen

Wenn sich berührende Körper relativ zueinander bewegen wirkt eine Reibungskraft entgegen die Bewegungsrichtung.

Man spricht dann auch von **Gleitreibung**. Die Gleitreibungskraft \vec{F}_{R0} berechnet einfach sich durch:

$$\vec{F}_{R0} = -\frac{\vec{v}_T}{|\vec{v}_T|} \cdot \mu_0 \cdot |\vec{F}_N|$$

F_N ist dabei die Berührungskraft welche im vorigem Kapitel berechnet wurde. Der Gleitreibungskoeffizient μ_0 ist dabei ein Materialpaar-abhängiger Wert aus dem Bereich $[0 \dots 1]$.

Wie wir gesehen haben ist die Implementierung des Gleitreibungsgesetzes kein Problem. Die **Haftreibung** muß jedoch angenähert werden. Die Haftreibungskraft \vec{F}_R wirkt normalerweise nur wenn die Tangentialgeschwindigkeit des Berührungspunktes $\vec{v}_T = \vec{0}$ ist und hält den Körper in Ruhe. Dazu liegt \vec{F}_R beliebig in der Berührungsebene und darf betragsmäßig nicht größer als die Normalenkraft \vec{F}_N sein, genauer gesagt $|\vec{F}_R| \leq \mu \cdot |\vec{F}_N|$.

⁷Linear Complementarity Problem: finde die Vektoren w und z für $w - Mz = q \wedge w \geq 0 \wedge z \geq 0 \wedge w_i z_i = 0 \quad \forall i$, wobei die Matrix M und der Vektor q vorgegeben werden, siehe dazu genauer [Murty 88].

Die Beschränkung $\vec{v}_T = \vec{0}$ wird aufgehoben und statt dessen werden alle Berührungen mit $|\vec{v}_T| \leq v_{Tmax}$ berücksichtigt. Oberhalb der maximalen Tangentialgeschwindigkeit v_{Tmax} wird dann mit der Gleitreibungsformel gerechnet. Statt der unbekanntem Wirkungsrichtung wird die Kraft entgegen der Bewegungsrichtung in der Berührungsebene angesetzt. Man erhält dann die Näherung:

$$\vec{F}_R = -\frac{\vec{v}_T}{v_{Tmax}} \cdot \mu \cdot |\vec{F}_N|$$

Es ist offensichtlich, daß diese Haftreibungskräfte keine Körperbewegung zum absoluten Stillstand bringen. Man kann nur alle Parameter möglichst klein wählen, um keinen sichtbaren Effekte in der Animation zu erhalten. Das Dilemma ist aber, daß dazu die Schrittweite des Integrationsverfahrens weit heruntergeschraubt werden müßte.

Eigentlich sollten Berührungskräfte und Reibungskräfte in ein großes Gleichungssystem gesetzt werden um eine analytische Lösung mit richtigen Wirkungsrichtungen zu erhalten. Wie im Artikel von [Baraff 91] gezeigt ist, sind solche gemischten Gleichungssysteme allerdings nur schwer mit heutigen mathematischen Verfahren lösbar.

3.9 Gravitation und Luftwiderstand

Damit die Körper sich nicht wie im schwerelosen Weltraum verhalten wirkt eine überall konstante Erdbeschleunigung g in negativer y-Richtung. Natürlich kann für g statt $9.81 \frac{m}{s^2}$ ein anderer Wert konfiguriert werden um die Simulation auf der Oberfläche eines anderen Planeten zu verlegen. Jeder Körper hat auch noch zusätzlich das Attribut Gravitation mit dem man die Erdbeschleunigung selektiv abschalten kann, wodurch das "Schieben" von Körpern durch eine Kraft vorhersagbarer wird.

Es ist optional möglich, die Gewichtskraft nicht konstant wirken zu lassen, sondern ihr einen gewissen Zufallsanteil in % mitzugeben. Dies hat den Vorteil, daß Sonderfälle wie das Stapeln von Kugeln nun mit dem erwarteten Zusammenbruch enden. Es soll aber nicht verschwiegen werden daß die Ersetzung der konstanten Gravitation durch eine Zufallsfunktion mit Mittelwert g unerwünschte Nebeneffekte besitzt. Bei einem frei beweglichem Körper führt dies nämlich zu einer unkontrollierten Energieänderung. Denn die Energie eines starren Körpers ergibt sich zu

$$E = E_{kin} + E_{pot} = \frac{m}{2} \vec{v}^2 + \frac{1}{2} J \vec{\omega}^2$$

Wobei ein zufälliges g , eben direkt $\vec{v} = \vec{a}$ beeinflusst.

So würde sich bei einem Pendel, bei dem kinetische Energie(Bewegungsenergie) in potentielle Energie(Lageenergie) umgewandelt wird, durch die Energieänderung, die Schwingungshöhe und -zeit vergrößern oder verkleinern. Oder Körper die in Ruhe am Boden liegen verschieben sich leicht.

Zusätzlich zur Gewichtskraft kann auch noch eine Luftwiderstandskraft, die entgegen der Bewegungsrichtung angesetzt ist, wirksam gemacht werden. Sie steigt proportional zum Geschwindigkeitsquadrat an und berücksichtigt einen Körperabhängigen

Luftwiderstandsbeiwert(c_w). Die Angriffsfläche A berechnet sich dabei konstant aus den Körpermaßen. Es ist offensichtlich, daß eine Analyse des Strömungsverhaltens selbst eines einfachen Körpers nicht durch

$$\vec{F}_{\text{Luftwiderstand}} = -c_w \cdot A \cdot |\vec{v}| \cdot \vec{v}$$

zu beschreiben ist. Der Zweck liegt auch eher in der Berücksichtigung einer energieverzehrenden Gegenkraft.

3.10 Kollisionserkennung

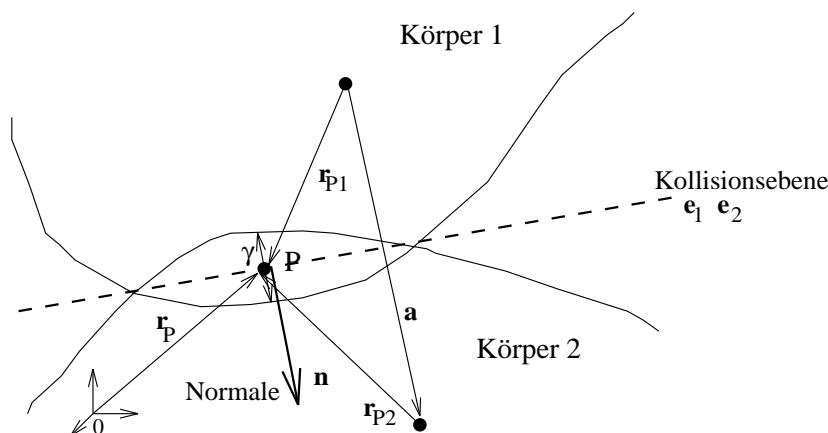


Abbildung 27: Kollisionserkennung

Damit eine Kollisionsbehandlung von Körpern erfolgen kann, muß die Kollisionserkennung die Durchdringung von Körpern feststellen und die relevanten Daten für eine Berührungs- oder Stoßbehandlung errechnen. Dazu wird jeder Körper mit jedem anderen überprüft und aufgetretene Kollisionen in einer Liste gesammelt.

Es wird davon ausgegangen, daß eine Berücksichtigung von einigen wenigen Kollisionspunkten ausreichend ist. So werden bei der Durchdringung von zwei Quadern die Eckpunkte des Schnittkörpers errechnet, um an ihnen die Berührungs- und Reibungskräfte angreifen zu lassen, bzw. um einen Punkt zur Berechnung des Stoßimpulses zu erhalten.

Relevante Parameter einer Kollision sind zuerst der Kollisionsort \vec{r}_P sowie dessen Pendant von den Körper aus \vec{r}_{P1} und \vec{r}_{P2} . Besondere Bedeutung haben die Normalenrichtung \vec{n} sowie die Durchdringungstiefe γ . Es ist wichtig, daß eine genügend große Gegenkraft die an einem Körper im Punkt P angelegt wird, die Durchdringungstiefe γ verkleinert. Genau dieses wird von den Berührungsmethoden vorausgesetzt.

Zwar wurden geometrisch einfache Körper ausgesucht, trotzdem sind bei $n = 4$ Körpern (Quader, Kugel, Zylinder und Ebene) immerhin $\frac{n}{2} \cdot (n+1) = 10$ Routinen nötig. Es hat sich leider jedoch herausgestellt daß selbst bei diesen einfachen Grundkörpern ein Kollisionstest schon sehr aufwendig ist. Deshalb konnte aus Zeitgründen der Zylinder bislang nicht

realisiert werden. Er wird daher durch einen Quader “angenähert” und durch die Quaderkollisionsroutinen behandelt. Eine ausführliche Beschreibung der einzelnen Routinen findet sich in [Stolz 93].

4 Berechnungsmethode

4.1 Einbindung der Bewegungsgleichungen in das Integrationsverfahren

Da die Bewegungsgleichungen ein Differentialgleichungssystem darstellen, ist eine Lösung mittels numerischer Integration angebracht. Dabei sind die meisten Integrationsverfahren nur für Systeme der Form

$$\dot{\vec{x}} = \vec{f}(t, \vec{x}) \quad \wedge \quad \vec{x}(t_0) = \vec{x}_0$$

anwendbar. Das Verfahren soll dabei aus $\vec{x}(t)$ den neuen Zustand an der Stelle $\vec{x}(t+h)$ berechnen.

Nun stellen aber die Bewegungsgleichungen ein Differentialgleichungssystem 2. Ordnung dar.

$$\begin{aligned} \ddot{\vec{r}} &= \frac{1}{m} \sum_i \vec{F}_i \\ \dot{\vec{\omega}} &= J^{-1} \cdot \left(\vec{M} + \sum_i (\vec{r}_i \times \vec{F}_i) \right) \end{aligned}$$

Um nun ein System 1. Ordnung zu erhalten werden noch folgende Gleichungen dazugenommen:

$$\begin{aligned} \dot{\vec{r}} &= \vec{v} \\ \dot{\vec{q}} &= g(\vec{q}, \vec{\omega}) \end{aligned}$$

Hier offenbart sich der Hauptgrund für die Benutzung der Eulerparameter, um die Orientierung eines Körpers zu beschreiben. Im Gegensatz zu Eulerwinkeln, die drei hintereinander ausgeführte Achsen-Drehungen benutzen, sind Eulerparameter ohne Probleme aufintegrierbar. Dazu steht die Beziehung

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = g(\vec{q}, \vec{\omega}) = \frac{1}{2} \begin{pmatrix} 0 & -w_1 & -w_2 & -w_3 \\ w_1 & 0 & w_3 & -w_2 \\ w_2 & -w_3 & 0 & w_1 \\ w_3 & w_2 & -w_1 & 0 \end{pmatrix} \cdot \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}$$

zur Verfügung.

Der Zustandsvektor \vec{x} enthält für jeden Körper die Position \vec{r} , die Orientierung \vec{q} , die Geschwindigkeit \vec{v} und die Drehgeschwindigkeit $\vec{\omega}$. Die rechten Seiten der obigen Gleichungen entsprechen dabei dem problemabhängigen Funktionenvektor $\vec{f}(t, \vec{x})$. Das Integrationsverfahren hat nun die Aufgabe, aus den Beschleunigungen $\ddot{\vec{r}}$ und $\dot{\vec{\omega}}$ die Geschwindigkeiten $\dot{\vec{r}}$ und $\dot{\vec{\omega}}$ und die Position \vec{r} und Orientierung \vec{q} aufzusummieren.

4.2 Das Integrationsverfahren

Angeregt durch den Artikel von [Heinzel 92] über Runge-Kutta-Verfahren fiel die Wahl auf ein eingebettetes Runge-Kutta-Verfahren⁸ mit Fehlersteuerung und Interpolation von Zwischenzuständen.

Das Verfahren der Ordnung $s = 7$ errechnet nun aus den Zwischenwerten

$$\begin{aligned} \vec{k}_0 &= \vec{f}(t, \vec{x}(t)) \\ \vec{k}_i &= \vec{f}(t + c_i \cdot h, \vec{x}(t) + h \cdot \sum_{j=0}^{i-1} a_{ij} \vec{k}_j) \quad i = 1, \dots, s-1 \end{aligned}$$

den neuen Zustand zu

$$\vec{x}(t+h) = \vec{x} + h \cdot \sum_{i=0}^{s-1} b_i \vec{k}_i$$

Durch die Fehlersteuerung wird nun die Schrittweite h entsprechend einer gewünschten Genauigkeit gewählt.

Indem man einen zweiten Zustandwert $\hat{\vec{x}}$ mit einer erniedrigten Ordnung durch

$$\hat{\vec{x}}(t+h) = \vec{x} + h \cdot \sum_{i=0}^{s-1} \hat{b}_i \vec{k}_i$$

bestimmt, erhält man eine absolute Abschätzung des Fehlers zwischen dem Schritt von t nach $t+h$ durch $\delta = \|\vec{x} - \hat{\vec{x}}\|$. Gibt man die Genauigkeit ϵ vor, so muß wenn selbige verletzt ist, also $\delta > \epsilon$ gilt, der Zustand $\vec{x}(t+h)$ verworfen werden und eine neue Schrittweite h_{neu} gewählt werden.

Eine geeignete Formel stellt dazu

$$h_{neu} = 0.9h \left(\frac{\epsilon}{\delta} \right)^{\frac{1}{p+1}}$$

dar. Darin ist $p = 4$ die Ordnung der Zustandsgleichung für $\hat{\vec{x}}$, welche um eins niedriger ist als die von \vec{x} , welche $q = p + 1 = 5$ ist. Man verwendet also das genauere Ergebnis weiter. Für die Praxis wünschenswert ist es, die Schrittweite nur in einem bestimmten Intervall variabel zu halten um unerträgliche lange Rechenzeiten zu vermeiden.

Weiter wird durch Interpolation, die sich auf die schon berechneten Zwischenwerte $\vec{k}_0, \dots, \vec{k}_{s-1}$ stützt, ein effizientes Abfragen von Zwischenzuständen $\vec{x}(t + \sigma h)$ im Intervall $t + \sigma h \in [t, t+h]$ ermöglicht. Dazu wird der Zustandsverlauf in diesem Intervall durch ein Polynom des Grades $d = 5$ angenähert. Dessen Polynomkoeffizienten ergeben sich zu

$$\begin{aligned} \vec{\alpha}_0 &= \vec{x}(t) \\ \vec{\alpha}_j &= h \sum_{i=0}^{s-1} \beta_{j-1,i} \vec{k}_i \quad j = 1, \dots, d \end{aligned}$$

⁸Dieses wird als RK5(4)7FM in den benutzten Artikeln [Dormand et al. 80] und [Heinzel 92] bezeichnet.

Das Ausrechnen eines Zwischenzustandes erfolgt nun ohne Funktionsauswertung von $\vec{f}(t, \vec{x})$ durch

$$\vec{x}(t + \sigma h) = \sum_{j=0}^d \alpha_j \sigma^j$$

Die Konstanten $a_{ij}, c_i, b_i, \hat{b}_i$ und $\beta_{i,j}$ für die obigen Formeln sind in [Heinzel 92] bzw. den Originaltexten [Dormand et al. 80], [Dormand et al. 81] und [Dormand et al. 86] zu finden.

4.3 Allgemeine Funktionsweise

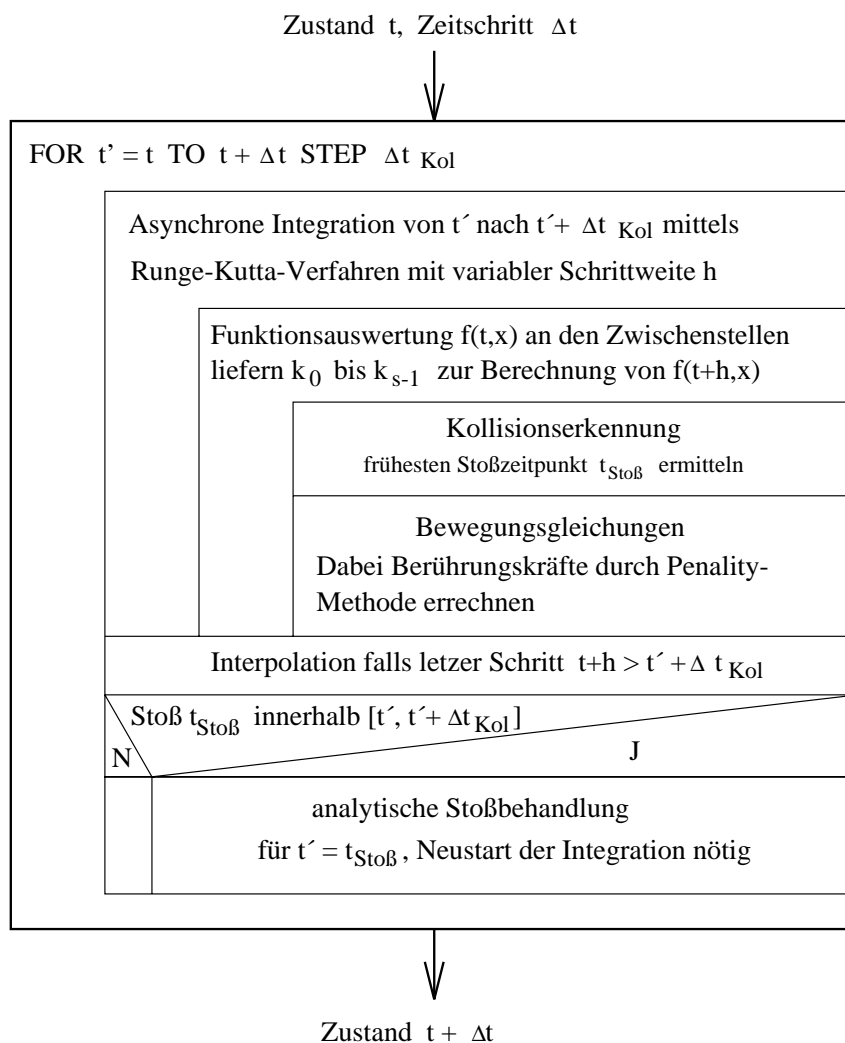


Abbildung 28: Funktionsschema der Folgezustandsberechnung

Insgesamt ergibt sich das Struktogramm in Bild 28 für den funktionalen Zusammenhang der einzelnen Komponenten zu der von *AERO* benutzten Folgezustandsberechnung.

Aus einem Systemzustand, der den Bewegungszustand aller Körper mitsamt Verbindungen und benutzerspezifischen Kräften repräsentiert, soll ein neuer Zustand für den Zeitpunkt $t + \Delta t$ errechnet werden.

Das Integrationsverfahren hat die Aufgabe, die Bewegungsgleichungen von t nach $t + \Delta t$ aufzusummieren. Es hat sich nun aber gezeigt daß es sinnvoll ist, das Intervall zuerst in kleinere Einheiten der Schrittweite Δt_{Kol} aufzuteilen, um eine stetige Kollisionserkennung in diesem Intervall zu ermöglichen, da die Integrationsrittweite h durchaus größer als Δt werden kann. Weiterhin muß bei der analytischen Stoßbehandlung die Integration neu gestartet werden, da die frisch errechneten Geschwindigkeiten als Anfangswerte des Differentialgleichungssystems anzusehen sind.

Die Bewegungsgleichungen sind nur von der Berechnungsreihenfolge in das Integrationsverfahren geschachtelt, Programmtechnisch gesehen ist das Verfahren völlig unabhängig vom Problem implementiert. Die Modellierung der Kräfte (Gravitationskräfte, Berührungskräfte, Reibungskräfte und Luftwiderstandskräfte) erlaubt eine direkte Errechnung aus den Zustandsvariablen \vec{x} . Da sie nur Position und Geschwindigkeit benutzen, werden sie direkt in den Impuls- und Drallsatz eingesetzt.

Um aber die Berührungskräfte berücksichtigen zu können, ist zuvor noch eine Kollisionserkennung nötig. Hierbei wird auch gleichzeitig der erste aufgetretene Stoß erfaßt und für eine spätere Verarbeitung gespeichert.

Dabei ist zu bemerken, daß aufgrund der variablen Schrittweite h das Integrationsverfahren asynchron zu den erfragten Zuständen $t' + t'_{Kol}$ läuft. Wenn es größere Schritte macht, werden die erfragten Zwischenzustände durch Interpolation angenähert, im anderen Fall müssen mehrere Zwischenschritte eingefügt werden.

4.4 Simulationsdurchführung

Um eine modellgetreue Simulation durchführen zu können sind 4 Voraussetzungen nötig. Wegen ihrer gegenseitigen Wechselwirkung dürfen sie nicht unabhängig voneinander gesehen werden.

Wenn ein Experiment simuliert wird, sollten die folgenden Voraussetzungen erfüllt sein. Schließlich sollte die daraus resultierende Animationssequenz auf unerklärliche Bewegungsänderungen oder große Körperschwingungen untersucht werden, um gegebenenfalls mit veränderten Simulationsparametern weiter zu arbeiten.

Numerische Stabilität

Eine Integration verläuft stabil, wenn das Aufsummieren einer abgeleiteten Funktion die Funktion selbst ausreichend annähert, also $f(x) \approx I(x) = \sum_{x'=x_0}^x f(x')$ gilt.

Um dies zu erreichen muß die Schrittweite h des Integrationsverfahrens zum Problem, der Funktion $f(x)$, passen. Die Fehlersteuerung wählt nun selbständig ein passendes h aus, wozu eine Zwischenschrittgenauigkeit ϵ herangezogen wird.

Da h sich nur in einem vorgegebenen Intervall bewegt, ist das Erreichen der unteren Intervall-Schranke, der minimalen Schrittweite also, ein Indikator, daß die Genauigkeit

nicht mehr eingehalten werden kann und somit die Summe $I(x)$ vom Sollwert $f(x)$ stark abweicht.

Dies führt im Extremfall zur völligen Divergenz, welche sich in der Animation durch plötzliche Positionsänderungen des betroffenen Körpers bemerkbar macht. Das kann so weit gehen, daß der Körper ganz verschwindet, weil seine Position sich ins “Unendliche” aufsummiert hat.

Ausreichende Kollisionserkennung

Ein rechtzeitiges Erkennung einer Kollision ist entscheidend für die Behandlung von Stößen und Berührungen. Die Kollisionsschrittweite Δt_{Kol} stellt die maximale Zeitdifferenz zwischen zwei Kollisionserkennungen dar. Sie sollte nach den räumlichen Maßen der simulierten Körper und der maximal auftretenden Geschwindigkeiten gewählt werden.

Ein zu großes Δt_{Kol} zieht sehr viele negative Effekte nach sich:

- Körper fliegen durcheinander durch, weil der Zeitraum in dem sie sich schneiden kleiner als Δt_{Kol} ist.
- Die Durchdringung von Körpern wird zu spät erkannt, so daß die Durchdringungstiefe γ sehr groß wird. Dies führt zu enormen Berührungskräften welche die Körper voneinander wegspringen lassen. Es ergibt sich somit auch kein zeitlich stetiger Verlauf der Berührungskräfte, was das Integrationsverfahren zu einer Verkleinerung der Schrittweite bis zur unteren Grenze veranlaßt.

Gemäßigte Stoßklassifizierung

Zur analytischen Stoßbehandlung muß eine Kollision erst als Stoß oder Berührung klassifiziert werden, was durch einen Vergleich mit der maximalen Berührungsgeschwindigkeit v_{Nmax} geschieht. Diese Geschwindigkeit darf nicht unabhängig von der Kollisionsschrittweite gesehen werden.

Ein Körper der auf einer unbeweglichen Unterlage steht und durch seine Gewichtskraft nach unten gezogen wird, soll nicht durch Stöße sondern durch Berührungskräfte auf seiner Position gehalten werden. Nun ist die Geschwindigkeitsdifferenz der Körper zwischen zwei Kollisionserkennungen etwa $g \cdot \Delta t_{Kol}$. Für die Praxis empfiehlt sich ein Wert für v_{Nmax} der um den Faktor 10 bis 100 höher ist um auch große Geschwindigkeitsdifferenzen aufzufangen. Diese entstehen beispielsweise wenn neben der Gewichtskraft noch weitere Kräfte wirken.

Ein zu kleines v_{Nmax} führt dazu, daß in der Einschwingphase zu einem ausgeglichenem Berührungskraftverhältnis Stöße auftreten. Dies bewirkt erstens eine Verlangsamung des Einschwingvorganges und zweitens erhöht es die Rechenzeit aufgrund der unnützen Stoßimpulsberechnung und des Integrationsneustarts.

Passende Körper- und Materialparameter

Die benutzte Modellierung der Berührungskräfte, die Federstoßmethode wie auch die Verbindungsarten Feder, Dämpfer und Stange, ergeben einen Regelkreis den es gilt stabil zu halten.

Als entscheidende Parameter des Regelkreises gelten dabei die Dehnungs- und Dämpfungsfaktoren der Materialien und Verbindungen, sowie die Gewichtskraft der Körper. Da sich die Masse aus den geometrischen Maßen ergibt, ist eine umsichtige Wahl des Materials entscheidend, dabei sollten allzu unterschiedliche Körpermassen vermieden werden. Noch besser ist die direkte Angabe der Körpermasse, da Vollkörper schnell zu große Massen aufweisen.

Dehnungs- und Dämpfungsfaktoren sollen den Einschwingvorgang von Berührungen oder das Dehnungsverhalten von Verbindungen sicherstellen, vergleichbar mit dem Stoßdämpfer eines Autos. Dabei sollte die Einschwingzeit möglichst kurz sein und das Überschwingen vermieden werden. Bild 25 aus dem Abschnitt über die Berührungsbehandlung zeigt das Einschwingverhalten des Einsinkens einer Kugel in eine Ebene mit der Penalty-Methode und der Federstoßmethode bei gleichem Dehnungsfaktor.

Da die Federstoßmethode die Schwingung lediglich beim Loslösen um die Stoßkennzahl ϵ abschwächt, entstehen langsam abklingende Schwingungen mit hohen Amplituden. Aus diesem Grund sollte bei der Simulation von Stoßproblemen mit hohen Geschwindigkeiten von dieser Methode Abstand genommen werden. Nur durch Wahl von größeren Materialfaktoren läßt sich dann die maximal erreichte Durchdringungstiefe herabsetzen. Allerdings gehen solche "steifen" Probleme zu Lasten der Rechenzeit. In der Numerik werden Probleme mit stark unterschiedlichen Zeitfaktoren in den Differentialgleichungen als steife Differentialgleichungen bezeichnet. Diese sind nur mit besonders kleinen Zeitschrittweiten lösbar.

Bei der Penalty-Methode hingegen wird durch das Berücksichtigen der Geschwindigkeitsdifferenz im Kollisionspunkt schneller und ohne großes Überschwingen die Ruheposition erreicht.

Die Grundeinstellungen der Dämpfungs- und Federfaktoren wurde dabei durch Versuche ermittelt, indem das Einsinkverhalten von Kugeln mit einer Masse zwischen 1 bis 600 kg untersucht wurde. Eine allgemeine Wahl der Faktoren, die für jeden erdenklichen Fall optimale Ergebnisse liefern, ist nicht möglich. Ein zu simulierendes Experiment sollte deshalb aus aufeinander abgestimmten Faktoren und Körpern aufgebaut sein.

Körper die sich in der Animation nicht bewegen sollen, können als masselos attribuiert werden. Dies spart Rechenzeit, weil diese Körper nicht in die Bewegungsgleichungen eingehen und keine Kollisionserkennung zwischen masselosen Körpern erforderlich ist.

5 Beispiele

5.1 Pendel

Wie man selbst mit einigen wenigen Objekten schon reizvolle Animationen erstellen kann zeigt das erste Beispiel.

Fünf Gummi-Kugeln werden über Stangen an einen raumfesten Punkt wie in Bild 29 zu sehen befestigt. Eine äußere Kugel wird in die waagrechte Position gebracht, so daß sie aufgrund der Schwerkraft auf einer Kreisbahn auf die ruhenden Kugeln fällt. Somit entsteht ein System aus fünf mathematischen Pendeln die ihre Energie durch Stoßimpulse weitergeben. Zur Verschönerung kann man noch ein Gestell dazu bauen. Dabei müssen die Zylinder als "feststehend" definiert werden. Um eine zu starke Verlängerung der Stangen zu vermeiden kann man die Dehnungs- und Dämpfungskonstanten um den Faktor 10 erhöhen.

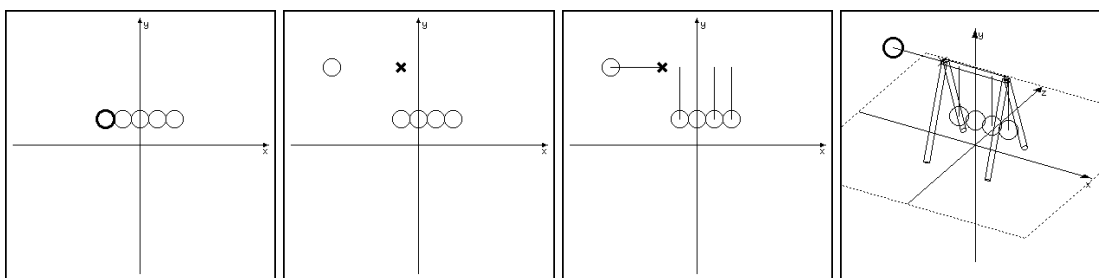


Abbildung 29: Schrittweise Erstellung eines 5-Kugel-Pendels

Nachdem alle Körper und Verbindungen eingegeben sind kann die Animation gestartet werden. Sinnvollerweise wählt man zuerst die passende Bildfrequenz aus, z.B. wurde die Animationssequenz aus Abbildung 30 mit 15 Bilder/Sekunde ausgegeben. Wenn man sich jedes Einzelbild als Szene mitprotokollieren läßt, kann durch einfaches Vor- und Zurückfahren die Sequenz ohne grossen Rechenaufwand mehrmals angesehen werden. So kann die optimale Kameraposition komfortabel gefunden werden. Läuft die Animation nach Wunsch und mit dem passenden Bildausschnitt ab, so läßt man sich für jedes Einzelbild den dazugehörigen Raytracer-Code erstellen und per Skript-File berechnen.

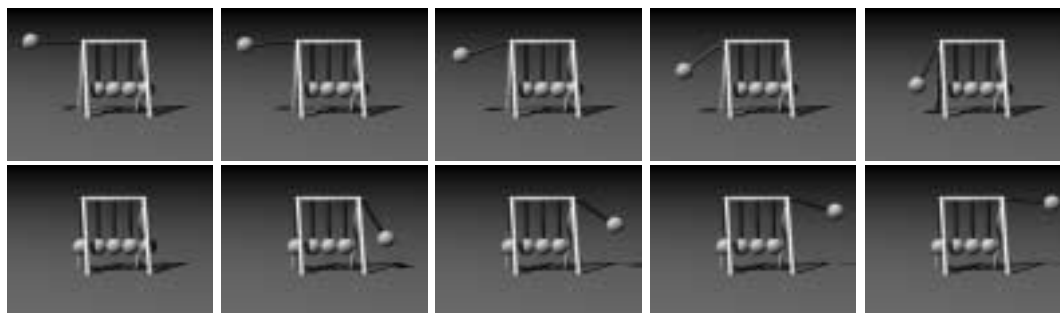


Abbildung 30: Animationssequenz des 5-Kugel-Pendels