

# Grundlegende Verfahren und Techniken

## Transformationen und Koordinatensysteme

1. Es gilt

$$\sum_{i=1}^n \lambda_i X_i = \left( \frac{\sum_{i=1}^n \mathbf{x}_i}{\sum_{i=1}^n \lambda_i} \right).$$

falls die Punkte gegeben sind durch  $X_i = (\mathbf{x}_i, 1)^T$ . Für  $\sum \lambda_i = 1$  erhält man einen Punkt für die affine Kombination. Ist die Summe  $\sum \lambda_i = 0$ , dann ist das Ergebnis ein Vektor. Hat die Summe einen Wert  $\sum \lambda_i \neq 0$ , dann kann durch diesen Wert dividiert werden; das Ergebnis ist ein Punkt.

2. Die Polygonzüge A und B und die gesuchten Lösungen:

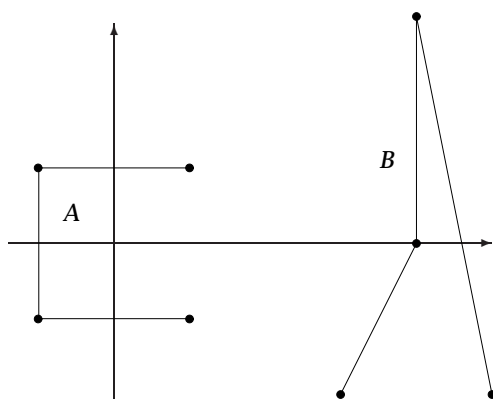


Abbildung 1: Die Polygonzüge A und B

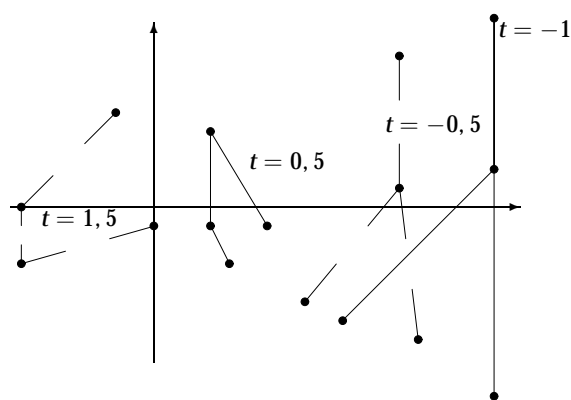


Abbildung 2: Die Polygonzüge  $tA + (1-t)B$

3. Durch eine Koordinatentransformation wird die Gerade durch den Ursprung auf die  $x$ -Achse abgebildet; dann die Reflexion durchgeführt und anschließend wieder die Koordinatentransformation rückgängig gemacht. Die Koordinatentransformation beschreiben wir durch die beiden Winkel  $\varphi$  und  $\theta$ .  $\varphi$  beschreibt den Winkel zwischen der  $x$ -Achse und der Achse in der  $xy$ -Ebene,  $\theta$  den Winkel zwischen  $x$ -Achse und der Achse in der  $xz$ -Ebene. Dann ist die allgemeine Reflexion an einer Achse gegeben durch  $R_y(-\theta)R_z(\varphi)Ref_xR_z(-\varphi)R_y(-\theta)$ :

$$Ref = \begin{pmatrix} \cos^2 \theta \cos 2\varphi - \sin^2 \theta & \cos \theta \sin 2\varphi & -\sin 2\theta \sin \varphi^2 \\ \cos \theta \sin 2\varphi & -\cos 2\varphi & \sin \theta \sin 2\varphi \\ -\sin 2\theta \sin \varphi^2 & \sin \theta \sin 2\varphi & \cos^2 \theta + \sin^2 \theta \cos 2\varphi \end{pmatrix}$$

Setzen Sie  $\theta = \varphi = 0$ , dann erhalten Sie  $Ref_x$  zurück. Interessant ist der Fall  $\theta = 0$ , einer

Reflexion an einer Achse in der  $xz$ -Ebene:

$$Ref_{\theta=0} = \begin{pmatrix} \cos 2\varphi & \sin 2\varphi & 0 \\ \sin 2\varphi & -\cos 2\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4. Analog zur letzten Aufgaben wird wieder die Drehachse durch die beiden Winkel  $\varphi, \theta$  beschrieben. Dann ist die allgemeine Rotation mit Winkel  $\beta$  gegeben durch

$$R_y(-\theta)R_z(\varphi)R_x(\beta)R_z(-\varphi)R_y(-\theta).$$

Im Buch auf Seite 20 müssen Sie noch die Koordinaten der Drehachsen durch den Vektor  $(\cos \theta \cos \varphi, \sin \varphi, \cos \varphi \sin \theta)^T$  ersetzen. Haben Sie sich dazu entschieden, die Drehachse auf die  $y$  oder  $z$ -Achse zu drehen, ergibt sich ein anderer Ausdruck, aber das gleiche Endergebnis.

5. Das Auto unterscheidet sich vom Fahrrad auf Seite 24 grob durch die Anzahl der Räder. Beim Propellerflugzeug kommen zu den beiden Vorderrädern ein Hinterrad und der Propeller hinzu.

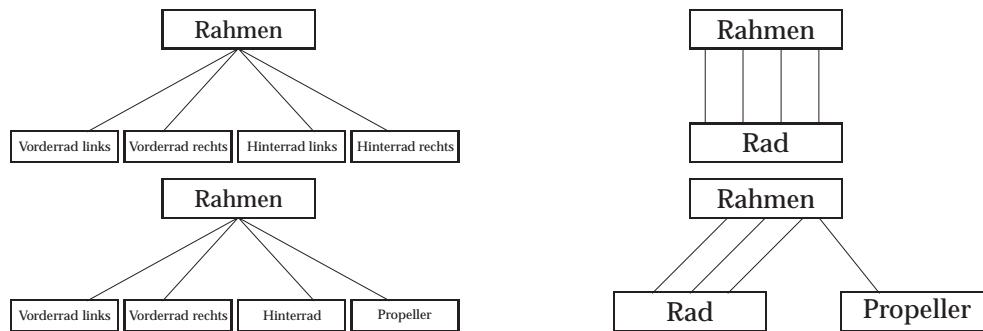


Abbildung 3: Ein Auto (oben) und ein Propellerflugzeug (unten) als Baum und als gerichteter azyklischer Graph

## Projektionen und Kameramodelle

- .
- Das Ergebnis stimmt mit der Abbildung 2.25 auf Seite 29 überein.
- Die Kavaliers- und Kabinettprojektion mit  $\alpha = 60^\circ$  finden Sie in Abbildung 4.



Abbildung 4: Die Kabinett- (links) und Kavalierprojektion mit  $\alpha = 60^\circ$

4. Das grafische Ergebnis sehen sie in Abbildung 5. Die Matrizen sind

$$P_{parallel} = \begin{pmatrix} -0,7071 & 0 & 0,7071 & 0 \\ -0,3015 & 0,9045 & -0,3015 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und

$$P_{persp} = \begin{pmatrix} 3,3166 & 0 & -3,3166 & 0 \\ -1,4142 & 4,2426 & -1,4142 & -4,2426 \\ 0 & 0 & 0 & 0 \\ -0,6396 & -0,4264 & -0,6396 & 5,11682 \end{pmatrix}$$



**Abbildung 5:** Die Parallel- (links) und Zentralprojektion für Aufgabe 4

5. Die Projektion der Geraden ist gegeben durch

$$X_u^*(\lambda) = d \frac{A_u + \lambda c_u}{-A_n - \lambda c_n}, X_v^*(\lambda) = d \frac{A_v + \lambda c_v}{-A_n - \lambda c_n}.$$

Für den Fall  $c_n = 0$  ergibt dies die parametrische Form einer Geraden in der Bildebene mit Steigung  $\frac{c_v}{c_u}$ :

$$\frac{d}{-A_n} (A_u + \lambda c_u, A_v + \lambda c_v).$$

Dieses Bild ist unabhängig von der Position der Geraden, es kommt immer die gleiche Steigung heraus.

Für den Fall  $c_n \neq 0$  müssen wir den Grenzwert  $\lambda \rightarrow \infty$  betrachten. Es ist

$$\lim_{\lambda \rightarrow \infty} = d \left( -\frac{c_u}{c_n}, -\frac{c_v}{c_n} \right).$$

Der Fluchtpunkt hänge nur vom Richtungsvektor ab! Also haben parallele Geraden alle den gleichen Fluchtpunkt.

## Clipping und Rasterung

1. .

## Sichtbarkeit

1. .

## Fallstudien

1. .

## **Affine Transformationen und Hierarchien in OpenGL**

Finden Sie bei den Quelltexten dieses Kapitels

## **Hierarchien in VRML und AliasWavefront MAYA**

Finden Sie bei den Quelltexten dieses Kapitels

## **Kameras in OpenGL**

Finden Sie bei den Quelltexten dieses Kapitels

# Geometrisches Modellieren mit Kurven und Flächen

## Parameterkurven

1. Alle Kurvenstücke sind linear; als Bild ergibt sich Abbildung 6.

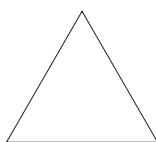


Abbildung 6: Die Lösung der Aufgabe 1

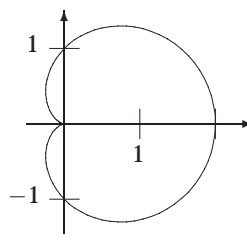


Abbildung 7: Die Kardioide, die Lösung der Aufgabe 3

2. Die Punkte  $(x(t), y(t))$  liegen alle auf einem Kreis; denn es ist  $x(t)^2 + y(t)^2 = a^2$ . Dann ist auch klar, dass der Ursprung der Mittelpunkt ist, der Radius ist  $a$ .
3. Die Lösung ist die *Kardioide* in Abbildung 7. Die Kardioide ist der geometrische Ort, der von einem Punkt eines Kreises von Radius  $2a$  beschrieben wird, der ohne zu gleiten auf einem anderen Kreis vom Radius  $2a$  rollt. In der Aufgabe war  $a = \frac{1}{2}$ . Als Bogenlänge ergibt das Integral den Wert 8.
4. Die Tangenten für eine Ellipse mit den Radien  $R_1, R_2$  sind gegeben durch

$$\mathbf{t}_0 = \begin{pmatrix} 0 \\ R_2 \end{pmatrix}, \mathbf{t}_{\frac{\pi}{4}} = \frac{1}{2}\sqrt{2} \begin{pmatrix} -R_1 \\ R_2 \end{pmatrix}, \mathbf{t}_{\frac{\pi}{2}} = \begin{pmatrix} -R_1 \\ 0 \end{pmatrix}, \mathbf{t}_{\pi} = \begin{pmatrix} 0 \\ -R_2 \end{pmatrix}.$$

Die Skizze sehen Sie in Abbildung 8.

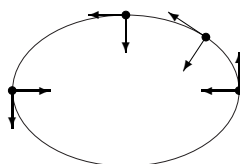
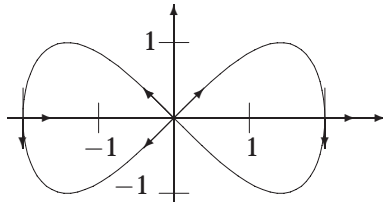


Abbildung 8: Die Ellipse, Tangenten und Normalen für Aufgabe 4

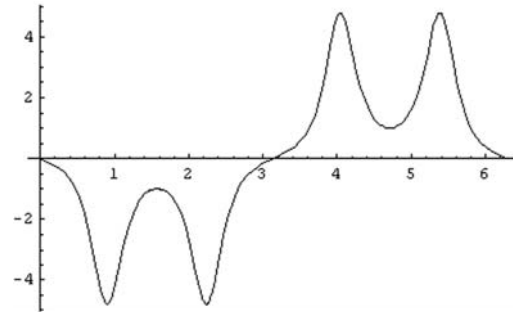
5. Die Lösung ist die Lemniskate von Gerono; die Skizze sehen Sie in Abbildung 9. Die Krümmung  $\kappa_2$  ist gegeben durch

$$\kappa_2(t) = \frac{-4 \cos^2 t \sin t + \sin t (\cos^2 t - \sin^2 t)}{(\cos^2 t + (\cos^2 t - \sin^2 t)^2)^{\frac{3}{2}}}$$

Abbildung 10 zeigt den Verlauf der Krümmung. Die Bogenlänge ist durch das Integral gegeben durch 6,09722.



**Abbildung 9:** Die Lemniskate von Gerono als Lösung der Aufgabe 5



**Abbildung 10:** Die Krümmung  $\kappa_2$  der Lemniskate von Gerono

Die Tangenten und Normalen finden Sie in der Abbildung 9 und in Tabelle 1.

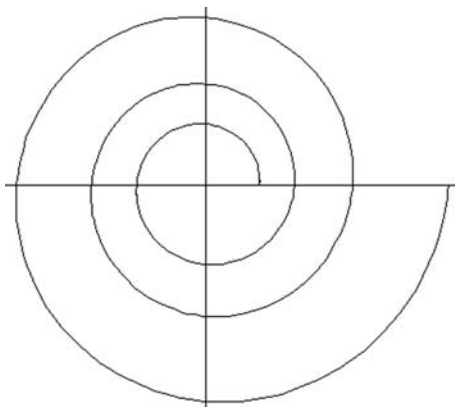
Parameterwerte	$t = 0$	$t = \frac{\pi}{2}$	$t = \pi$	$t = \frac{3\pi}{2}$	$t = 2\pi$
Tangenten	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
Normalen	$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$

**Tabelle 1:** Die unnormalisierten Tangenten und Normalen an die Lemniskate

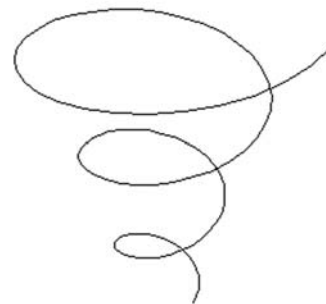
Die Ellipse in Abbildung 8 ist gegeben als der geometrische Ort der Punkte, für die die Summe der Abstände zu zwei gegebenen Brennpunkten  $F_1$  und  $F_2$  konstant ist. Die Lemniskate von Bernoulli ist gegeben geometrischer Ort aller Punkte, für die das Produkt der Abstände zu den Brennpunkten konstant ist. Allgemein ist eine Lemniskate von Bernoulli gegeben durch die Parameterdarstellung

$$K_a(t) = \left( \frac{a \cos t}{1 + \sin^2 t}, \frac{a \sin t \cos t}{1 + \sin^2 t} \right).$$

Dabei liegen die beiden Brennpunkte dann in  $(\pm a\sqrt{2}, 0)$ .



**Abbildung 11:** Die logarithmische Spirale im Parameterintervall  $[0; 6\pi]$



**Abbildung 12:** Die Schraubenlinie für die logarithmische Spirale im Parameterintervall  $[0; 6\pi]$

6. Eine grafische Darstellung der logarithmischen Spirale ( $ae^{bt} \cos t, ae^{bt} \sin t$ ) sehen Sie in Abbildung 11; die Schraubenlinie darüber ist durch  $(ae^{bt} \cos t, ae^{bt} \sin t, ct)$  gegeben; ihr Bild finden Sie in Abbildung 12. Für das Frenet'sche Bezugssystem ergibt sich:

$$\mathbf{v}(t) = \frac{1}{\sqrt{c^2 + a^2(1+b^2)}e^{2bt}} \begin{pmatrix} ae^{bt}(b \cos t - \sin t) \\ ae^{bt}(\cos t + b \sin t) \\ c \end{pmatrix},$$

$$\mathbf{b}(t) = \frac{1}{(1+b)\sqrt{c^2 + a^2}e^{2bt}} \begin{pmatrix} -c(2b \cos t + (b^2 - 1) \sin t) \\ c((b^2 - 1) \cos t - 2b \sin t) \\ a(1+b^2)e^{bt} \end{pmatrix},$$

$$\mathbf{n}(t) = \mathbf{b} \times \mathbf{v}.$$

## Polynomiale Kurven

1. Die Kurve ist eine Parabel, denn die Diskriminante ist Null. In Abbildung 13 sehen Sie ein Skizze.

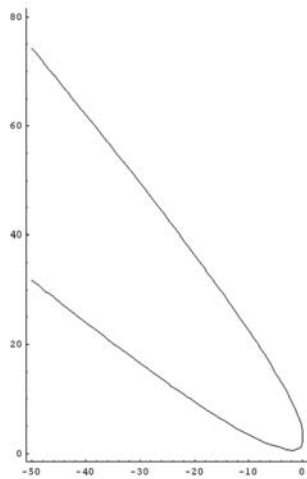


Abbildung 13: Der Kegelschnitt zu Aufgabe 1

2. Die Formel kann geschrieben werden als  $\sum_{i=1}^3 \lambda_i P_i$ , dabei sind die Koeffizienten gegeben durch

$$\lambda_1 = \frac{(1-t)^2}{(1-t)^2 + 2wt(1-t) + t^2}, \quad \lambda_2 = \frac{2wt(1-t)}{(1-t)^2 + 2wt(1-t) + t^2}, \quad \lambda_3 = \frac{t^2}{(1-t)^2 + 2wt(1-t) + t^2}.$$

Diese addieren sich offensichtlich zu 1 auf.

3. Das Gewicht ist für alle drei Kurvenstücke immer  $w = \cos 60^\circ = 0,5$ .

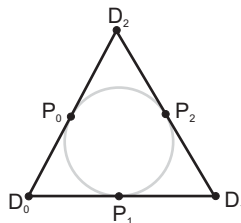


Abbildung 14: Der Kreis als rationale Kurve mit den Kontrollpunkten

Die Segmente ergeben sich durch die folgenden Kontrollpunkte:

- $P_0, D_2$  und  $P_2$ , erstreckt sich von  $P_0$  bis  $P_2$ ;
- $P_2, D_1$  und  $P_1$ , erstreckt sich von  $P_2$  bis  $P_1$ ;
- $P_1, D_0$  und  $P_0$ , erstreckt sich von  $P_1$  bis  $P_0$ .

Für das Dreieck ist jedes gleichwinklige Dreieck zu verwenden. Eine mögliche Wahl für das Dreieck ist  $D_0 = (0, 0)$ ,  $D_1 = (1, 0)$  und  $D_2 = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ . Wenn Sie an den Punkten  $P_i$  jeweils die Tangenten und Krümmungen bestimmen, dann erkennen Sie, dass der mit den rationalen Polynomen dargestellte Kreis nur  $C^1$ - bzw.  $G^1$ -stetig ist. Und er ist nicht nach seiner Bogenlänge parametrisiert!

4. .



# Polygonale Netze

## Polygone und Polyeder

1. Für den Tetraeder gilt  $4 - 6 + 4 = 2$ ; für die vierseitige Pyramide  $5 - 8 + 5 = 2$  und für das Prisma ganz rechts in Abbildung 4.4 auf Seite 193 gilt  $6 - 9 + 5 = 2$ .

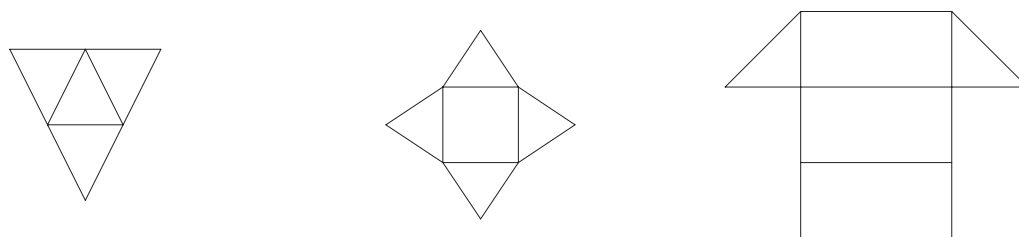


Abbildung 15: Die planaren Graphen der Aufgabe 1

2. Ja. Mögliche Skizzen sehen Sie in Abbildung 16.

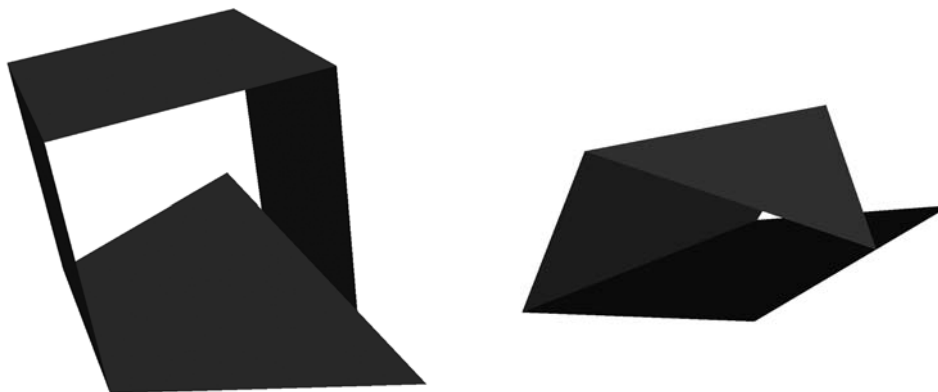


Abbildung 16: Skizzen der Objekte aus Aufgabe 2

3. Das Objekt aus Abbildung 4.10 auf Seite 195 besitzt eine Kante, die zu 3 Facetten gehört! Wie schon im Text behauptet, können Sie leicht nachrechnen, dass das Objekt genauso viele Ecken, Kanten und Facetten wie ein Würfel aufweist.

Das Objekt in Abbildung 4.9 hat 24 Ecken,  $12 + 32 = 44$  Kanten und  $16 + 6 = 22$  Facetten; dann gilt  $V - E + F = 24 - 44 + 22 = 2$ . Zu der rechten Seite verallgemeinerten Euler'schen Formel muss noch die Anzahl der „Löcher“ in Facetten dazugezählt werden, diese beträgt in diesem Fall 2.

4. Für das Objekt in Abbildung 4.8 gilt  $V = 24$ ,  $E = 46$ ,  $F = 15$ ,  $H = 3$  und  $G = 1$ ; damit ergibt sich  $24 - 36 + 15 - 3 = 0$ . Wie schon im Text angegeben hat das Objekt in Abbildung 4.7 48 Ecken, 96 Kanten und 48 Facetten.
5. Im linken Graphen in Abbildung 17 ist für die gekennzeichnete Ecke die Valenz 3 und der Ring gegeben durch die Facetten  $F_1$ ,  $F_2$  und  $F_3$ . Die Ecke  $V_1$  im rechten Graph in Abbildung 17 hat den Ring  $F_1$ ,  $F_2$ ,  $F_3$ ,  $F_4$  und Valenz 4.  $V_2$  hat Valenz 2 und den Ring  $F_3$ ,  $F_4$ .



Abbildung 17: Zwei Beispiel-Netze

## Speicherung von polygonalen Netzen

1. Die Eckenliste für Abbildung 4.6 ist  $V = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$  und

$$\begin{aligned} F_1 &= (1, 2, 3, 4), F_2 = (2, 1, 5, 6), F_3 = (2, 6, 10, 9), \\ F_4 &= (6, 5, 8, 7), F_5 = (7, 8, 4, 3), F_6 = (4, 8, 5, 1) \text{ und} \\ F_7 &= (7, 3, 2, 6). \end{aligned}$$

Dabei werden die Ecken wie im Buch nummeriert.

Das Netz in Abbildung 4.6 besitzt eine Kante, die zu drei Facetten gehört; eine „Kantenliste“ könnte dann so aussehen:

$$\begin{aligned} E_1 &= (1, 2; F_2, F_1), E_2 = (2, 3; F_7, F_1), E_3 = (3, 4; F_5, F_1), \\ E_4 &= (4, 1; F_6, F_1), E_5 = (1, 5; F_6, F_2), E_6 = (5, 6; F_4, F_2), \\ E_7 &= (6, 2; F_3, F_2, F_7), E_8 = (6, 10; F_3), E_9 = (10, 9; F_3), \\ E_{10} &= (9, 2; F_3), E_{11} = (5, 8; F_6, F_4), E_{12} = (8, 7; F_5, F_4), \\ E_{13} &= (7, 6; F_7, F_4), E_{14} = (8, 4; F_6, F_5), E_{15} = (3, 7; F_7, F_5). \end{aligned}$$

Die Facetten sind

$$\begin{aligned} F_1 &= (1, 2, 3, 4), F_2 = (1, 5, 6, 7), F_3 = (8, 9, 10, 7), \\ F_4 &= (6, 11, 12, 13), F_5 = (12, 14, 3, 15), F_6 = (14, 11, 5, 4), \text{ und} \\ F_7 &= (13, 15, 2, 7). \end{aligned}$$

In Abbildung 18 sehen Sie einen planaren Graphen für das Netz in Abbildung 4.10 im Buch, mit eingezeichneten Ecken und Facetten.

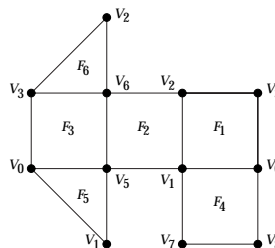


Abbildung 18: Ein planarer Graph für das Objekt in Abbildung 4.10

Eine Eckenliste ist dann gegeben durch

$$F_1 = (0, 1, 2, 3), F_2 = (2, 1, 5, 6), F_3 = (6, 5, 0, 3), \\ F_4 = (1, 0, 8, 7), F_5 = (0, 5, 1), F_6 = (6, 3, 2).$$

Auch hier gibt es wieder Kanten, die zu drei Facetten gehören. Eine „Kantenliste“ könnte so aussehen:

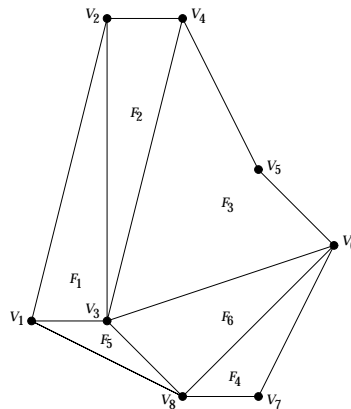
$$E_1 = (0, 1; 1, 4), E_2 = (1, 2; 1, 2), E_3 = (2, 3; 1, 6), E_4 = (3, 0; 1, 3), \\ E_5 = (2, 6; 6, 2), E_6 = (6, 5; 2, 3), E_7 = (5, 1; 5, 2), E_8 = (6, 3; 6, 3), \\ E_9 = (1, 7; 4), E_{10} = (7, 8; 4), E_{11} = (8, 0; 4), E_{12} = (0, 5; 5, 3).$$

und

$$F_1 = (1, 2, 3, 4), F_2 = (2, 5, 6, 7), F_3 = (6, 8, 4, 12), \\ F_4 = (1, 9, 10, 11), F_5 = (12, 1, 7), F_6 = (8, 3, 5).$$

Die doppelt-verkettete Kantenliste setzt die Konsistenzbedingungen voraus!

2. In Abbildung 19 sehen Sie das Ergebnis der Eckenliste mit markierten Ecken und Facetten.



**Abbildung 19:** Das Netz für Aufgabe 2 auf Seite 199

Dann ergibt sich die folgende Kantenliste:

$$E_1 = (1, 2; 1), E_2 = (2, 3; 1, 2), E_3 = (3, 1; 1, 5), E_4 = (2, 4; 2), E_5 = (4, 3; 2, 3), \\ E_6 = (4, 5; 3), E_7 = (5, 6; 3), E_8 = (6, 3; 3, 6), E_9 = (3, 8; 5, 6), \\ E_{10} = (8, 6; 4, 6), E_{11} = (6, 7; 4), E_{12} = (7, 8; 4), E_{13} = (8, 1; 5).$$

und

$$F_1 = (1, 2, 3), F_2 = (2, 4, 5), F_3 = (5, 6, 7, 8), \\ F_4 = (10, 11, 12), F_5 = (3, 9, 13), F_6 = (8, 9, 10).$$

Die doppelt-Verkettete Kantenliste finden Sie in Tabelle 2. Dabei wird nur NeF und VaF angegeben. + in der Facettentabelle bedeutet positive Orientierung.

3. Die Eckenliste ist gegeben durch

$$F_1 = (1, 2, 3, 4), F_2 = (5, 6, 2, 1), F_3 = (2, 6, 7, 3), \\ F_4 = (3, 7, 8, 4), F_5 = (8, 5, 1, 4), F_6 = (6, 5, 8, 7).$$

In Abbildung 20 sehen Sie eine Skizze des Quaders.

4. Für alle Facetten der doppelt-verketteten Kantenliste wird eine Eckenliste aufgebaut, beginnend mit der angegebenen Startkante. Je nach Orientierung werden Vor- bzw. Nachfolgerkanten der Eckenliste hinzugefügt, bis man wieder bei der Startkante angelangt ist.

Tabelle 2: Die doppelt-verkettete Kantenliste für Aufgabe 2

Kante	$V_a$	$V_e$	NeF	VaF
$E_1$	1	2	4	Null
$E_2$	2	3	3	5
$E_3$	3	1	1	13
$E_4$	2	4	5	Null
$E_5$	4	3	2	8
$E_6$	4	5	7	Null
$E_7$	5	6	8	Null
$E_8$	6	3	5	9
$E_9$	3	8	10	13
$E_{10}$	8	6	8	11
$E_{11}$	6	7	12	Null
$E_{12}$	7	8	10	Null
$E_{13}$	8	1	3	Null

Facette	Startkante	Orientierung
$F_1$	$E_1$	+
$F_2$	$E_4$	+
$F_3$	$E_5$	+
$F_4$	$E_{11}$	+
$F_5$	$E_{13}$	+
$F_6$	$E_8$	+

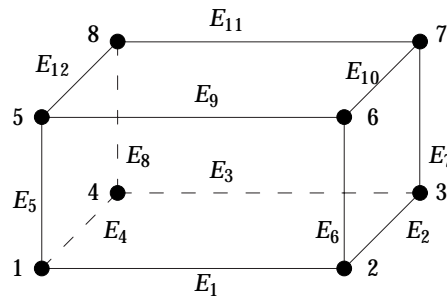


Abbildung 20: Eine Skizze des Quaders aus Aufgabe 3

## Normalenvektoren

- Beide Verfahren ergeben den Normalenvektor  $(-0, 182195, 0, 956544, 0, 227744)^T$ . Unnormalisiert ergibt das Vektorprodukt den Vektor  $(-4, 21, 5)^T$  und Newell ergibt unnormalisiert  $(-2, 10.5, 2.5)^T$ .
- Das Vektorprodukt ergibt das unnormalisierte Ergebnis  $(-2, 10, 4)^T$ , durch Normalisieren erhalten wir  $(-0.182574, 0.912871, 0.365148)^T$ . Newell ergibt als Ergebnis den Nullvektor.

Trotzdem liegen die vier Punkte in einer Ebene. Die Matrix  $\begin{pmatrix} 1 & 1 & 2 \\ 2 & 0 & 5 \\ 5 & 1 & 4 \\ 6 & 0 & 7 \end{pmatrix}$  hat den Rang 3. Es gilt  $\mathbf{v}_3 = -\mathbf{v}_0 + \mathbf{v}_1 + \mathbf{v}_2$ .

- Der Newell-Algorithmus mittelt die Normalen der Ebenen, in denen die Punkte liegen.

## Dreiecksnetze

- Als Startdreieck für den gierigen Algorithmus zur Konstruktion von Triangle Strips ergibt sich das Dreieck mit den Ecken 12, 13 und 6, da es nur einen Nachbarn im Adjazenzgraphen hat. Dann ergeben sich 5 Triangle Strips:

$$(12, 13, 6, 7, 1, 2) (2, 3, 7, 4, 8, 5, 9, 10) (5, 11, 10, 17) (13, 14, 7, 15) (15, 16, 9, 10).$$

Das Verhältnis zwischen übertragenen Punkten und Dreiecken ist  $\frac{13}{8}$ . Für den Generalized Triangle Strip im Buch ist das Verhältnis  $\frac{21}{16}$ .

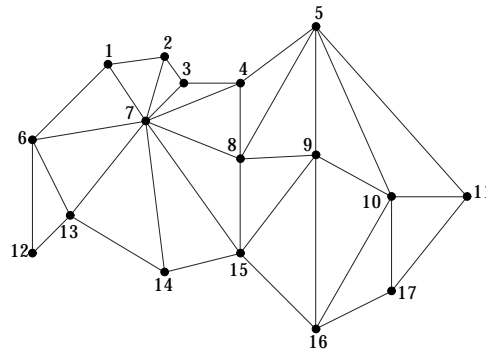
Für die Konstruktion des Generalized Triangle Meshes ist es wichtig, welche Randkanten

man verwendet zum Start. Verwendet man die Kanten mit den Ecken 12, 13, 14, 15, 16 und 17, das ist der untere Rand, dann ergibt sich das Generalized Triangle Mesh in Tabelle 3.

**Tabelle 3:** Ein Generalized Triangle Mesh zu Abbildung 21

12R	6ROp	13RO	7ROp	14RO	15RO	8RMp	9ROp
16RM	10RMp	17RO	11ROp	5RM	-3RM	-4RO	4RM
-5RO	3RO	2RO	1RM	-6RM			

Dafür erhält man das gleiche Verhältnis zwischen übertragenen Punkten und Dreiecken wie für die im Buch angegebene Variante, nämlich  $\frac{15}{16}$ .



**Abbildung 21:** Das Netz für Aufgabe 1

# Visualisierung

## Datenstrukturen

1. Die  $100^3$  Punkte werden als Volumen von  $100 \times 100 \times 100$  Punkten angenommen. Für `int` und `float` werden 4 Byte angesetzt.

**kartesisches Gitter:** Es werden 3 `float` Werte für den Ursprung und 3 `float` für die Schrittweiten in die Koordinatenrichtungen benötigt, plus 3 `int` Werte für die Anzahl der Punkte in jede Richtung. Dann werden für den Datensatz 36 Byte benötigt.

**rechtwinkliges Gitter:** Für den Ursprung 12 Byte, für die Anzahl der Punkte in jede Richtung 12 Byte und für jede Koordinatenrichtung 99 Schrittweiten, die wir auch als `float` annehmen. Insgesamt ergibt sich ein Speicherbedarf von rund 1.2 MB.

**strukturiertes Gitter:** Die Punktkoordinaten müssen explizit gespeichert werden plus 12 Byte für die Anzahl der Punkte in jeder Richtung. Dies ergibt einen Speicherbedarf von rund 4 MB.

**unstrukturiertes Gitter:** Wieder müssen die Punktkoordinaten explizit gespeichert werden, dafür werden 4 MB benötigt. Für jede Zelle (Hexaeder) werden 8 `int` Werte benötigt, um die Topologie abzuspeichern. Gehen wir von 1 000 Zellen aus, dann ergibt sich ein Speicherbedarf von rund 7, 2 MB. Allgemein ist der Speicherbedarf gegeben durch  $4 \text{ MB} + \text{NoZ} \cdot 32 \text{ Byte}$ ; wenn NoZ die Anzahl der Zellen bezeichnet.

2. Für `int` und `float` wird von einem Speicherbedarf von 4 Byte ausgegangen; bei `double` von 8 Byte. Für die  $100^3 = 1\,000\,000$  Punkte entstehen dann je nach Attributwert folgender Speicherbedarf für die Attribute:

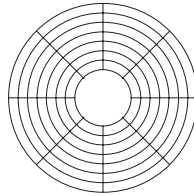
- `unsigned int`: 4 MB;
- `float`: 4 MB;
- `float[3]`: 12 MB;
- `double[3][3]`: 72 MB.

3. Für die Berechnung wird einmal davon ausgegangen, dass die Punkte im Datensatz als strukturiertes Gitter vorliegen. Dann benötigt man für den Datensatz 4 MB. Sind nur die Attribute zeitabhängig, dann entsteht pro Zeitschritt ein Speicherbedarf von 12 MB und insgesamt für die 100 Zeitschritte  $4 \text{ MB} + 100 \cdot 12 \text{ MB} \approx 1, 2 \text{ GB}$ .

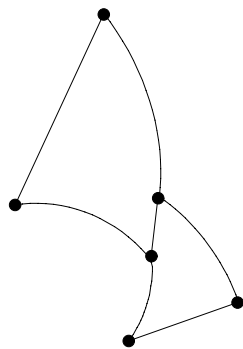
Ist auch die Geometrie zeitabhängig, dann entsteht ein Speicherbedarf von  $100 \cdot (4 + 12) \text{ MB} \approx 1, 6 \text{ GB}$ .

4. Ein kartesisches Gitter in Polarkoordinaten hat gleichmässige Schrittweiten im Radius und im Winkel. Dadurch entstehen konzentrische Kreise wie in Abbildung 22. Für ein rechtwinkliges Gitter wird ein Vektor mit Radiuswerten und ein Vektor mit Winkelwerten angegeben; mit gleichmässiger Topologie. Wenn wir immer von Vierecken ausgehen entsteht ein Bild wie in Abbildung 24. Für diese Abbildung werden die Radien (0.5, 1.0, 2.0, 2.5, 2.75) und die Winkel ( $0^\circ, 45^\circ, 60^\circ, 90^\circ, 135^\circ, 360^\circ$ ) angenommen.

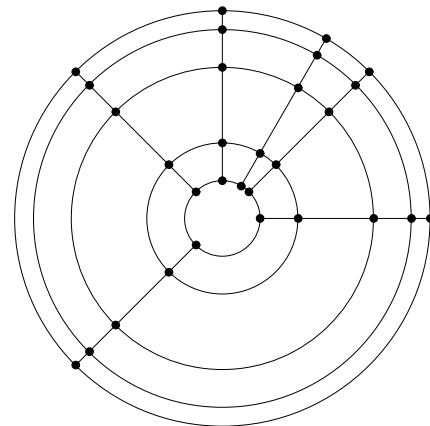
In Abbildung 23 sehen Sie ein kleines Beispiel eines strukturierten Gitters in Polarkoordinaten. Dabei sollen die Punkte  $V_1 = (0.5, 0^\circ)$ ,  $V_2 = (0.7, 30^\circ)$ ,  $V_3 = (0.4, 90^\circ)$ ,  $V_4 = (1.0, 10^\circ)$ ,  $V_5 = (0.9, 45^\circ)$  und  $V_6 = (1.5, 75^\circ)$  enthalten sein. Als Zellen sind die „Vierecke“ mit den Ecken 1, 4, 5, 2 und 2, 5, 6, 3 enthalten.



**Abbildung 22:** Ein kartesisches Gitter in Polarkoordinaten



**Abbildung 23:** Ein strukturiertes Gitter in Polarkoordinaten



**Abbildung 24:** Ein „rechtwinkliges“ Gitter in Polarkoordinaten

## Algorithmen für skalare Attribute

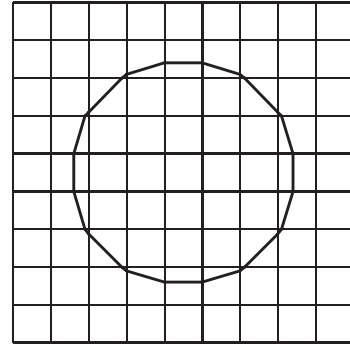
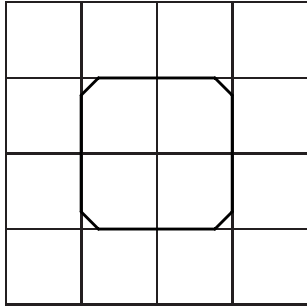
1. Schwarz, Weiß, Grün, Rot, Gelb, Cyan. Die minimale Differenz in der Luminanz ist dann 0.11.
2. Für die beiden Auflösungen finden sie in Tabelle 4 die entsprechenden Skalare; in Abbildung 25 die grafische Darstellung.
3. Man kommt durch Symmetrie und Rotation auf 4 Fälle:
  - (a) 0, 15;
  - (b) 1, 2, 4, 7, 8, 11, 13, 14;
  - (c) 3, 6, 9, 12;
  - (d) 5, 10.

## Direkte Volumen-Visualisierung

1. Den Filter nach Mitchell und Netravali sehen Sie in Abbildung 26. An der Stelle  $|x| = 2$  gilt immer  $H_t(B, C) = 0$ ; für  $|x| = 1$  ist der Funktionswert  $H_t(B, c) = B$  und an der Stelle  $x = 0$

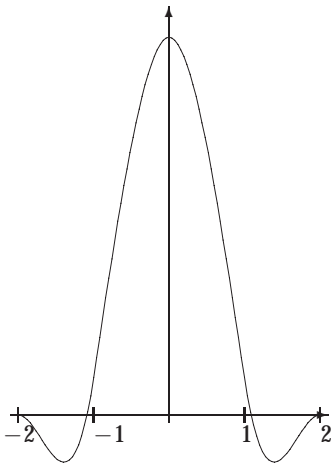
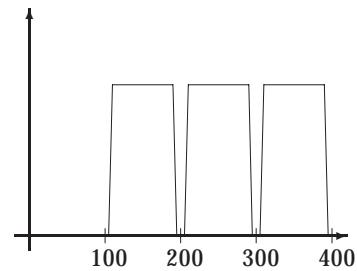
**Tabelle 4:** Die Daten für Aufgabe 2

4.5	2.8125	2.25	2.8125	4.5	4.5	3.61	2.94	2.5	2.27	2.27	2.5	2.94	3.61	4.5
2.8125	1.125	0.5625	1.125	2.8125	3.61	2.72	2.055	1.61	1.388	1.388	1.61	2.055	2.72	3.61
2.25	0.5625	0	0.5625	2.25	2.94	2.055	1.388	0.944	0.722	0.722	0.944	1.388	2.055	2.94
2.8125	1.125	0.5625	1.125	2.8125	2.5	1.61	0.944	0.5	0.277	0.277	0.5	0.944	1.61	2.5
4.5	2.8125	2.25	2.8125	4.5	2.27	1.388	0.722	0.277	0.055	0.055	0.277	0.722	1.388	2.27
					2.27	1.388	0.722	0.277	0.055	0.055	0.277	0.722	1.388	2.27
					2.5	1.61	0.944	0.5	0.277	0.277	0.5	0.944	1.61	2.5
					2.94	2.055	1.388	0.944	0.722	0.722	0.944	1.388	2.055	2.94
					3.61	2.72	2.055	1.61	1.388	1.388	1.61	2.055	2.72	3.61
					4.5	3.61	2.94	2.5	2.27	2.27	2.5	2.94	3.61	4.5

**Abbildung 25:** Die Berechnung von Konturlinien durch lineare Interpolation für Aufgabe 2; links Auflösung  $5 \times 5$ , rechts  $10 \times 10$ 

gilt  $H_t(B, C) = 6 - 2B$ .

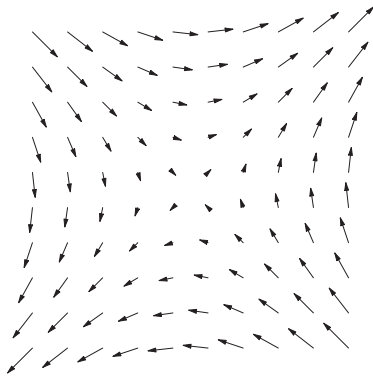
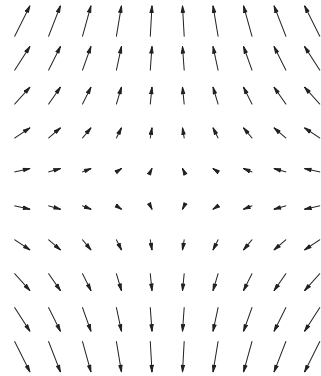
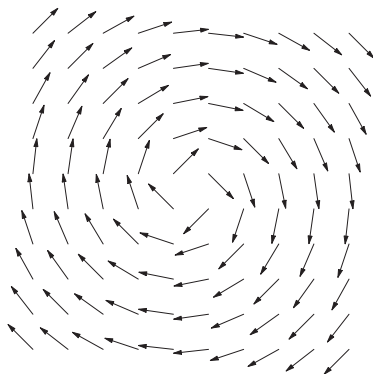
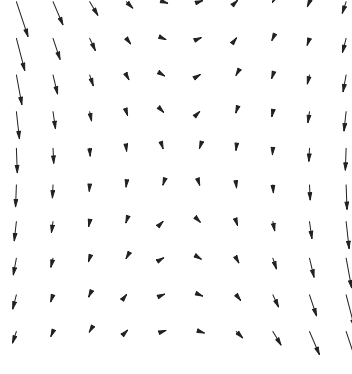
2. Eine scharfe Trennung ist nicht möglich bei der Lage; eine Möglichkeit wäre die Kurve in Abbildung 27. Möglich wäre auch eine Realisierung mit einer überlappenden, „fuzzy“ Transferfunktion.

**Abbildung 26:** Der Filter nach Mitchell und Nevanlina für  $B = 0.5$ ,  $C = 0.85$ **Abbildung 27:** Ein Vorschlag einer Transferfunktion für Aufgabe 2

## Visualisierung von Vektorfeldern

1. Die Skizzen finden Sie in den Abbildungen 28 bis 31.
2. Die Gradientenvektorfelder finden Sie in den Abbildungen 32 und 33.



Abbildung 28:  $f(x, y) = (y, x)$ Abbildung 29:  $f(x, y) = (-x, 2y)$ Abbildung 30:  $f(x, y) = \frac{1}{\sqrt{x^2+y^2}}(y, -x)$ Abbildung 31:  $f(x, y) = (y^2 - 2xy, 3xy - 6x^2)$ 

3. Durch Ableiten des Potentials bestimmt sich das Vektorfeld als

$$f(x, y, z) = v_0 \left[ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \frac{R^3}{2\sqrt{x^2 + y^2 + z^2}^5} \begin{pmatrix} -3xz \\ -3yz \\ x^2 + y^2 - 2z^2 \end{pmatrix} \right]$$

Auf den Koordinatenachsen erhält man damit

$$\begin{aligned} f(0, 0, \pm R) &= v_0 \left[ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \frac{R^3}{2R^5} \begin{pmatrix} 0 \\ 0 \\ -2R^2 \end{pmatrix} \right] = \mathbf{0}, \\ f(\pm R, 0, 0) &= v_0 \left[ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \frac{R^3}{2R^5} \begin{pmatrix} 0 \\ 0 \\ R^2 \end{pmatrix} \right] = \frac{3}{2} v_0 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \\ f(0, \pm R, 0) &= v_0 \left[ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \frac{R^3}{2R^5} \begin{pmatrix} 0 \\ 0 \\ R^2 \end{pmatrix} \right] = \frac{3}{2} v_0 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \end{aligned}$$

Für den Punkt  $P = (x, y, z)$  und  $R = \|P\|$  gilt

$$\langle f(x, y, z), P \rangle = v_0 \left( z + \frac{1}{2R^2} (-2z) R^2 \right) = 0.$$

Das Vektorfeld steht also immer senkrecht auf dem Punkt; also tangential zur umströmten Kugel! Die Abbildungen 6.43, 6.45 und 6.48 zeigen dieses Vektorfeld.

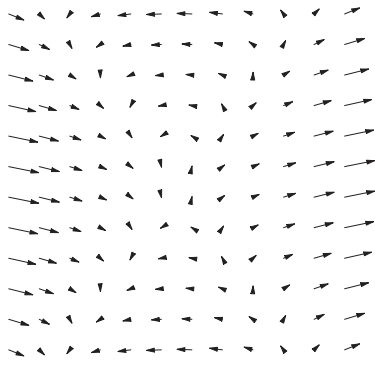


Abbildung 32:  $F(x, y) = x^2 - \frac{1}{2}y^2$

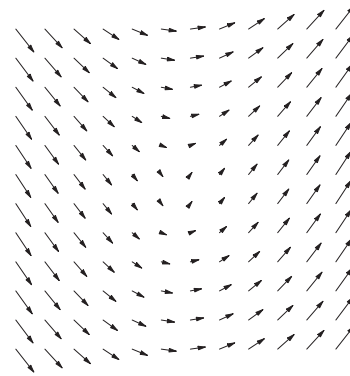


Abbildung 33:  $F(x, y) = \ln(\sqrt{x^2 + y^2})$

4. Die Feldlinien dieses Vektorfelds sind die Hyperbeln  $y = C/x$ . In Abbildung 34 sehen Sie die Trajektorien zum Zeitpunkt  $t = 0$ .

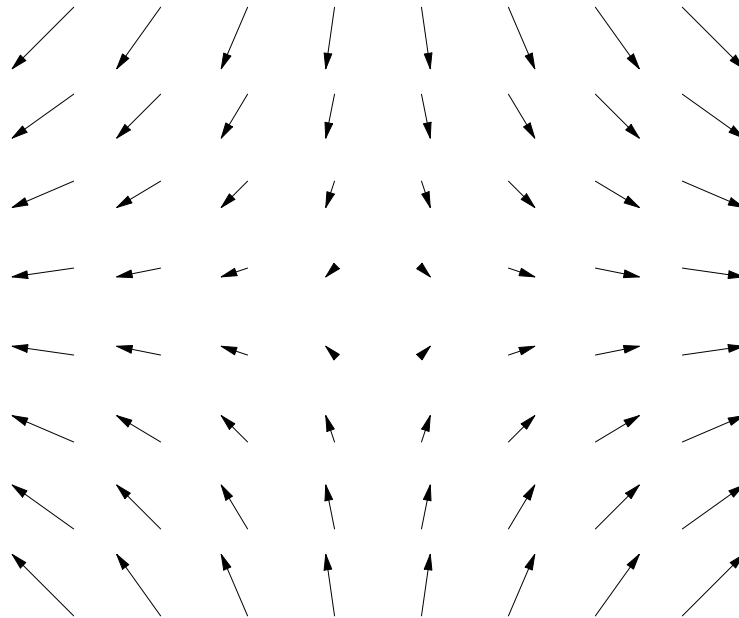


Abbildung 34: Die Trajektorien des Felds in Aufgabe 4 zum Zeitpunkt  $t = 0$

## Die Visualisierungs-Pipeline in VTK

- Finden Sie bei den Quelltexten dieses Kapitels
- Für alle drei Formate müssen auf jeden Fall die Attribute, also die Farben, gespeichert werden. Möglich wäre natürlich eine Zahl und eine Farbtabelle; oder direkt die RGB-Werte. Wenn wir 8 Bit für die Farbe pro Kanal ansetzen entsteht dadurch ein Bedarf von 30KB.

Für `vtkUnstructuredPoints` ergibt sich ein Speicherbedarf von 36 Byte; der Ursprung, die Anzahl der Punkte und das Spacing.

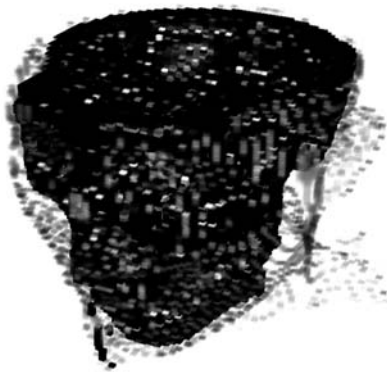
Für `vtkStructuredGrid` müssen die Punktkoordinaten explizit gespeichert werden; dadurch ergibt sich ein Bedarf von 40 KByte.

Für `vtkPolyData` ergibt sich ein Bedarf von 40 *KByte* für die Punktkoordinaten, plus 5 *int*-Werte pro Zelle für die Topologie; also insgesamt  $40 + 200 = 240$  *KByte*.

3. Finden Sie bei den Quelltexten dieses Kapitels

## Volumen-Visualisierung mit der VTK

1. In den Abbildungen bis finden Sie das Ergebnis eines Ray-Castings mit einer Pipeline wie für Bild 6.67 und Farbtafel 17; jeweils mit und ohne Schattierung und mit tri-linearer und mit nächster Nachbar- Interpolation. Die Bilder sind mit einem Datensatz mit Viertel-Auflösung („headsq/quarter“) berechnet; was man deutlich am Aliasing erkennt. Probieren Sie einmal den Datensatz mit voller Auflösung; Sie finden ihn unter den Quellen zu Kapitel 6; genauso wie die nötige Pipeline. Die Bilder sind alle mit einer Post-Shading Pipeline (in VTK `InterpolateFirst`) gerechnet; der Default in VTK.



**Abbildung 35:** Ray-Casting mit Nearest-Neighbour und Schattierung



**Abbildung 36:** Ray-Casting mit Nearest-Neighbour ohne Schattierung



**Abbildung 37:** Ray-Casting mit tri-linearer Interpolation und Schattierung



**Abbildung 38:** Ray-Casting mit tri-linearer Interpolation ohne Schattierung

2. Finden Sie bei den Quelltexten dieses Kapitels
3. In Abbildung 39 sehen Sie das Ergebnis der Pipeline für Abbildung 37 und Farbtafel 17; allerdings mit einer Pre-Shading-Pipeline. Beachten Sie unbedingt, dass Sie in VTK zwischen den

beiden Pipelines nur umschalten können, wenn Sie tri-lineare Interpolation verwenden!



**Abbildung 39:** Die Pipeline aus Abbildung 37 mit einer Pre-Shading Pipeline

4. Die Klasse `vtkLODProp3D` akzeptiert sowohl Ray-Casting-Mapper oder Texture-Mapping als auch von `vtkPolyDataMapper`. So ist es beispielsweise möglich, mit `vtkOutlineFilter` die Bounding-Box des Datensatzes zu extrahieren und als niedrigsten Level-Of-Detail einzustellen. In den Quellen zu diesem Kapitel finden Sie eine Pipeline. Für eine Instanz `volume` der Klasse `vtkLODProp3D` werden drei verschiedene Levels hinzugefügt. Der letzte Parameter bedeutet, dass VTK selbst versucht die benötigte Renderzeit zu schätzen.

```
// Bounding - Box mit eigenem Material
outline->SetInput(reader->GetOutput());
outlineMapper->SetInput(outline->GetOutput());
outlineProperty->SetColor(0.0, 0.0, 0.0);

// Texture2D Instanz als Level 2
lowresMapper->SetInput(reader->GetOutput());

// Ray-Casting als level 3
hiresMapper->SetInput(reader->GetOutput());
hiresMapper->SetVolumeRayCastFunction(composite);
hiresMapper->SetSampleDistance(distance);

int level1 = volume->AddLOD(outlineMapper, outlineProperty, 0.0);
int level2 = volume->AddLOD(lowresMapper, prop, 0.0);
int level3 = volume->AddLOD(hiresMapper, prop, 0.0)
```

## Visualisierung von Vektorfeldern mit der VTK

1. Finden Sie bei den Quelltexten dieses Kapitels
2. Finden Sie bei den Quelltexten dieses Kapitels

# Computer-Animation

## Computer-Animation

1. Sollten Sie die Filme nicht haben, dann bietet Pixar kleine Quicktime-Versionen (nicht nur) der Kinofilme auf <http://www.pixar.com>.
2. .
3. .

## Basistechnologien und Interpolation

1. Finden Sie bei den Quelltexten dieses Kapitels .
2. Finden Sie bei den Quelltexten dieses Kapitels .

Im Quelltext finden Sie drei verschiedene Inbetweens. In Abbildung 40 sehen Sie ein Screen-Capture des Programms. Einmal ganz oben linear, dann ease-in-ease-out, also zwei Keyframes und zu Beginn und am Ende die Ableitung Null.

Ganz unten sehen Sie ein Auto, das von einem Hermite-Spline gesteuert wird. Dabei wurden neben den beiden Anfangs- und Endframe noch nach einem Drittel und nach zwei Drittel der Wegstrecke Schlüsselszenen eingefügt; und dort die Ableitung, also die Geschwindigkeit des Autos, auf Null gesetzt.

Wollen Sie dieses Verhalten ändern, können Sie dies durch eine Änderung des Quelltexts in der Funktion `display tun`. Hier der Ausschnitt, in dem die Änderungen vorgenommen werden müssen.

```
// Hier die Schlüsselszenen.
keys[0] = minX;
keys[1] = (maxX - minX)/3.0;
keys[2] = 2.0 * keys[1];
keys[3] = maxX;

// Hier die Geschwindigkeitswerte
derivs[0] = 0.0;
derivs[1] = 0.0;
derivs[2] = 0.0;
derivs[3] = 0.0;

xPos0 = nonlinear(frame, numberOfFrames, keys, derivs);
xPos1 = easeineaseout(frame, numberOfFrames, minX, maxX);
xPos2 = linear(frame, numberOfFrames, minX, maxX);
```

3. Gegeben ist ein Quaternion  $q = s + ia + jb + kc$  mit  $|q| = 1$  mit  $s = \cos(\frac{\theta}{2})$  und  $a = \sin(\frac{\theta}{2})n_1$ ,  $b = \sin(\frac{\theta}{2})n_2$ ,  $c = \sin(\frac{\theta}{2})n_3$ , wenn die Drehachse gegeben ist durch  $(n_1, n_2, n_3)^T$ .

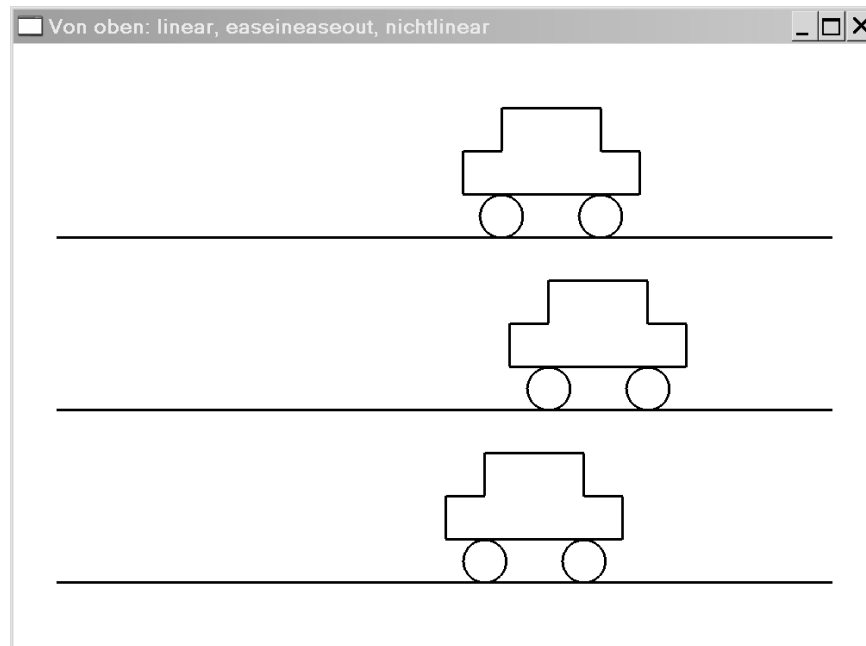


Abbildung 40: Screen-Capture zu Aufgabe 2

Eine solche Drehung kann dargestellt werden durch

$$R = \begin{pmatrix} tn_1^2 + c & tn_1n_2 - sn_3 & tn_1n_3 + sn_2 & 0 \\ tn_1n_2 + sn_3 & tn_2^2 + c & tn_2n_3 - sn_1 & 0 \\ tn_1n_3 - sn_2 & tn_2n_3 + sn_1 & tn_3^2 + c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

mit  $s = \sin \theta$ ,  $c = \cos \theta$  und  $t = 1 - \cos \theta$ ; diese Darstellung finden sie im Kapitel 2 auf Seite 20. Die euklidische Länge der Achse ist 1, damit kann die Matrix geschrieben werden als

$$R = \begin{pmatrix} 1 - t(n_2^2 + n_3^2) & tn_1n_2 - sn_3 & tn_1n_3 + sn_2 & 0 \\ tn_1n_2 + sn_3 & 1 - t(n_1^2 + n_3^2) & tn_2n_3 - sn_1 & 0 \\ tn_1n_3 - sn_2 & tn_2n_3 + sn_1 & 1 - t(n_1^2 + n_2^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Für die trigonometrischen Funktionen gilt:

$$2 \sin^2 \left( \frac{\theta}{2} \right) = 1 - \cos(\theta), \quad \sin(\theta) = 2 \sin \left( \frac{\theta}{2} \right) \cos \left( \frac{\theta}{2} \right).$$

Dann können wir die folgenden Gleichungen herleiten:

$$\begin{aligned} 2sa &= n_1 \sin(\theta), \quad 2sb = n_2 \sin(\theta), \quad 2sc = n_3 \sin(\theta), \\ 2a^2 &= n_1^2(1 - \cos(\theta)), \quad 2b^2 = n_2^2(1 - \cos(\theta)), \quad 2c^2 = n_3^2(1 - \cos(\theta)), \\ 2ab &= n_1n_2(1 - \cos(\theta)), \quad 2ac = n_1n_3(1 - \cos(\theta)), \quad 2bc = n_2n_3(1 - \cos(\theta)). \end{aligned}$$

Dann ist beispielsweise das Element  $r_{11}$  gegeben durch

$$r_{11} = 1 - (1 - \cos(\theta))(n_2^2 + n_3^2) = 1 - 2b^2 - 2c^2;$$

$r_{12}$  durch

$$r_{12} = -\sin(\theta)n_3 + (1 - \cos(\theta))n_1n_2 = -2sc + 2ab;$$

$r_{13}$  durch

$$r_{13} = \sin(\theta)n_2 + (1 - \cos(\theta))n_1n_3 = 2sb + 2ac.$$

Insgesamt ist die Rotationsmatrix  $R$  zum Quaternion  $q = s + ia + jb + kc$  mit  $|q| = 1$  gegeben durch

$$R = \begin{pmatrix} 1 - 2b^2 - 2c^2 & -2sc + 2ab & 2sb + 2ac \\ 2sc + 2ab & 1 - 2a^2 - 2b^2 & -2sa + 2bc \\ -2sb + 2ac & 2sa + 2bc & 1 - 2a^2 - 2b^2 \end{pmatrix}$$

4. .

## Animation hierarchischer Objekte

- Die Greiferposition für den ersten Satz von Parametern ist gegeben durch  $(4.22576, -2.1621)^T$ . Für den zweiten Satz von Parametern erhält man die Greiferposition  $(5.25653, 2.11126)^T$ .

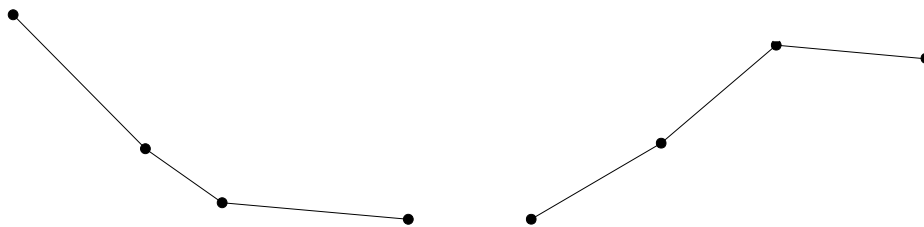


Abbildung 41: Die Positionen in Aufgabe 1; links der erste, rechts der zweite Parametersatz

- Man erhält eine ebene Fläche, da alle Gelenke des Roboters in einer Ebene liegen. In Abbildung 42 zeigt den Erreichbarkeitsbereich für die Winkel  $\theta_1, \theta_3 \in [-45^\circ, 45^\circ]$ . Eine Näherung erhält man durch Variierung beider Winkel, in der Abbildung mit  $1^\circ$  Schritten.

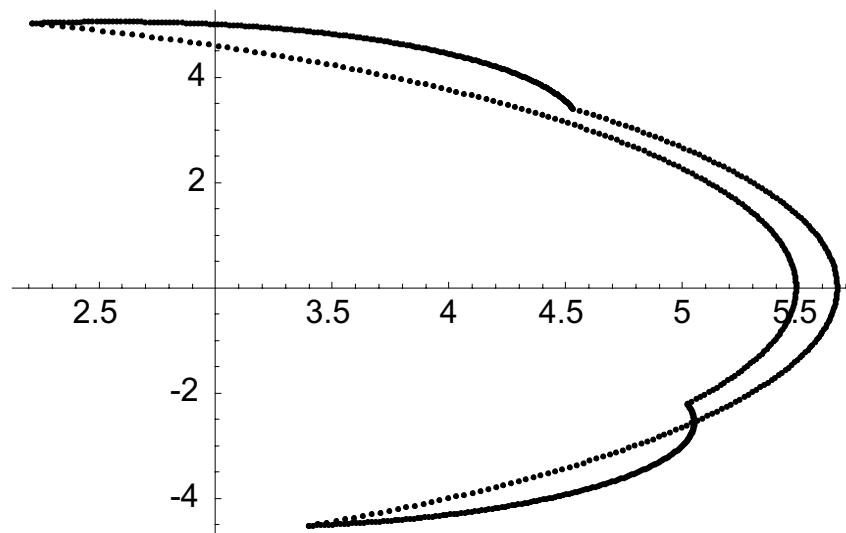


Abbildung 42: Der Erreichbarkeitsraum zu Aufgabe 2

- Wenn man diesmal die Kette von der Bodenplatte des Beins aufbaut, erhält man für das Bein das erste Gelenk mit einem Längenparameter  $a_1$  und dem Rotationswinkel  $\theta_1$ .

Als zweites Gelenk folgt jetzt der Kopf. Dabei ist in  $a_2$ , dem Achsabstand auch die Torsogröße zu berücksichtigen. Sie ist auf jeden Fall fest. Variabel ist der Kopfwinkel  $\theta_2$ . Jetzt folgen noch

zwei Drehgelenke für die beiden Schultern, und noch zwei Schiebegelenke für die beiden Daumen.

4. .

5. Die Lage des Endeffektors in der Ausgangslage war (vgl. Seite 463) gegeben durch  $3 + \sqrt{3}, 1 + \sqrt{3}) = (4.73205, 2.73205)$ .

Für den Zielpunkt  $(4.7, 2.6)$  erhält man den Korrektorvektor

$$(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3) = (-0.0010614, 0.000624155, 0.00101265)$$

und die Endeffektor-Position  $(4.69522, 2.59882)$ . Daraus ergibt sich ein Tracking-Fehler von  $5.55 \cdot 10^{-17}$ , das kann als Null angesehen werden, und als euklidischen Abstand zwischen Ziel und Endeffektor-Position von 0.153959.

Für den Zielpunkt  $(4.75, 2.7)$  erhält man den Korrektorvektor

$$(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3) = (-0.000113894, -0.000016647, 0.0000250411)$$

und die Endeffektor-Position  $(4.74989, 2.69994)$ . Daraus ergibt sich ein Tracking-Fehler von  $3.47 \cdot 10^{-18}$ , das kann als Null angesehen werden, und als euklidischen Abstand zwischen Ziel und Endeffektor-Position von  $3.8 \cdot 10^{-3}$ .

## Prozedurale Animationstechniken

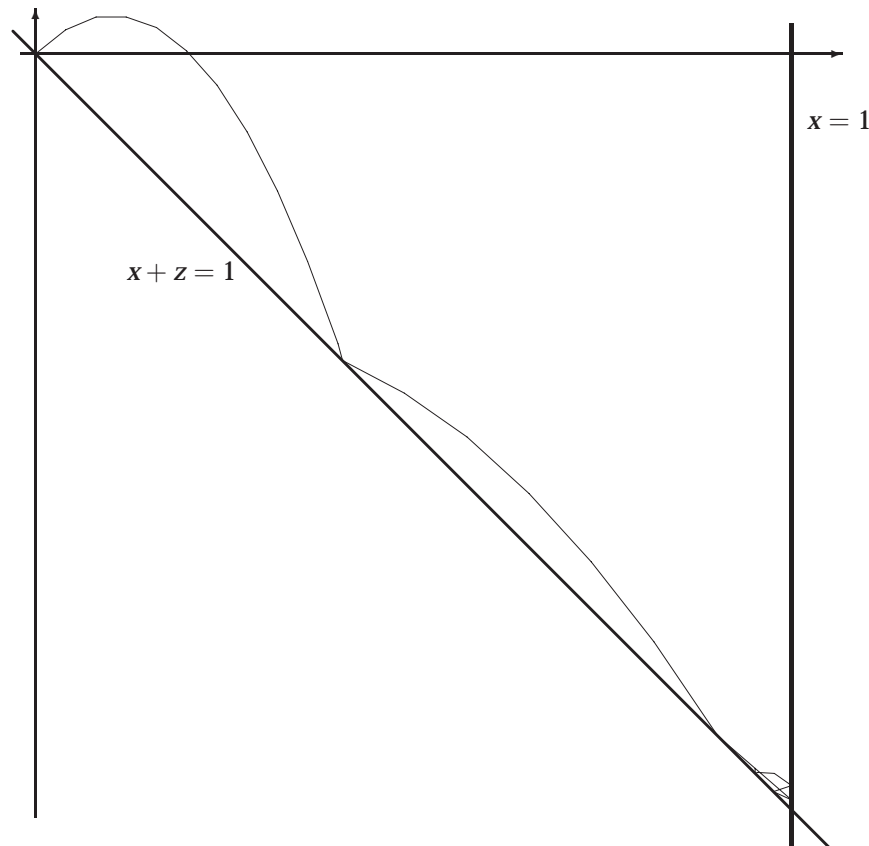


Abbildung 43: Der Pfad des Partikels für Aufgabe 1



1. In Abbildung 43 sehen Sie das Ergebnis der Berechnung mit einer Schrittweite von  $\Delta t = 0.4$  für  $t \in [0; 1]$ . Es treten nur Kollisionen mit den Ebenen  $x + z = 1$  und  $x = 1$  auf. Für alle Ebenen wurde  $\alpha = 0.5$  verwendet. Die Darstellung zeigt nur die  $xz$ -Ebene, denn die Kollisionen erzeugen nur Kräfte in dieser Ebene.
2. In Abbildung 44 sehen Sie die Ausgabe des Programms. Von links nach rechts und von oben nach unten wurden die Werte  $k_r = 0, 1, 5$  und  $k_r = 10$  verwendet. Die Quellen finden Sie unten den Quellcodes zu diesem Kapitel.

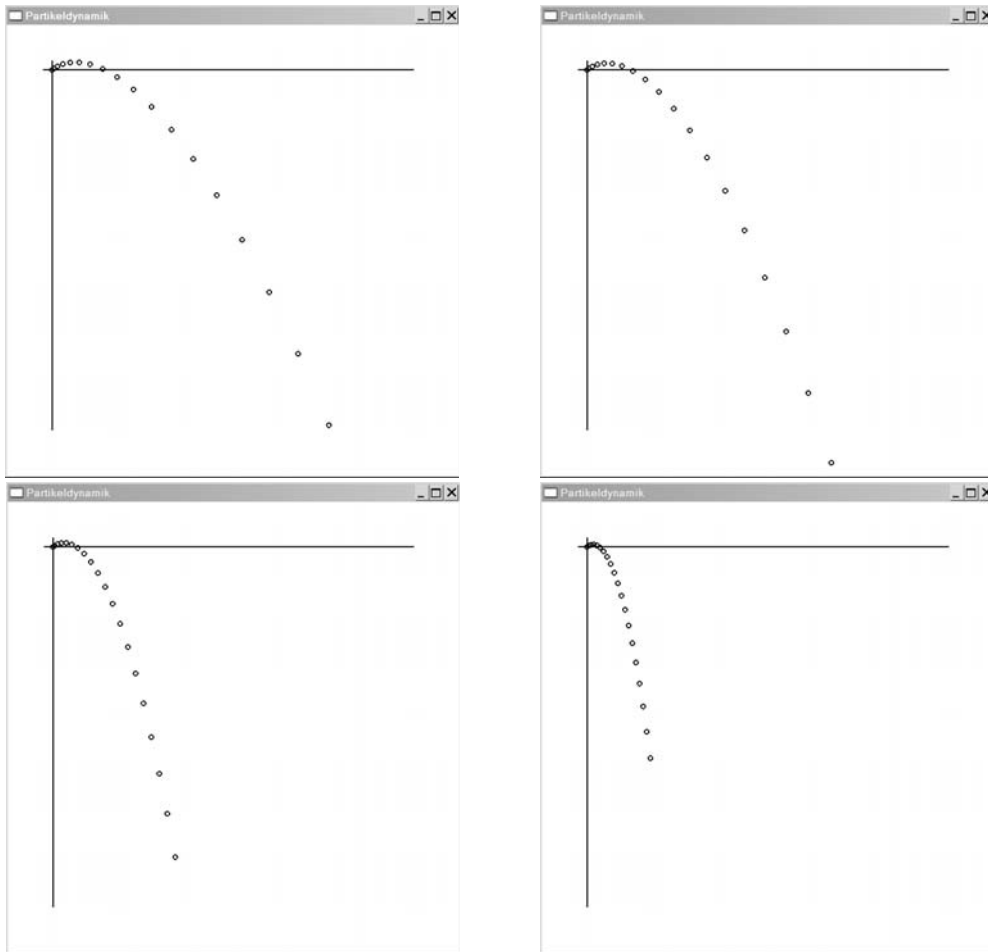


Abbildung 44: Die Abbildungen zu Aufgabe 2

Hier ein Auszug aus den Quellen; die Ausgabe wird auf dem Feld `points` abgelegt, das in `init` entsprechend der gewünschten Anzahl der Integrationsschritte dimensioniert wird.

```
void euler(double x0[2], double v0[2], double deltat, int steps, double kr,
           double **points)
{
    int i;
    double g = -9.8;
    double v[2], a[2]; // Die Geschwindigkeit und Beschleunigung

    points[0][0] = x0[0]; points[0][1] = x0[1];

    v[0] = v0[0];
    v[1] = v0[1];
```

```
for (i=1; i<= steps; i++) {
    a[0] = -kr*v[0];
    a[1] = g - kr*v[1];
    points[i][0] = points[i-1][0] + v[0]*deltat + 0.5*a[0]*deltat*deltat;
    points[i][1] = points[i-1][1] + v[1]*deltat + 0.5*a[1]*deltat*deltat;
    v[0] += deltat*a[0];
    v[1] += deltat*a[1];
}
```

3. .

## Key-Framing und Pfad-Animationen in Alias MAYA

1. Finden Sie bei den Quelltexten dieses Kapitels .
2. Finden Sie bei den Quelltexten dieses Kapitels .
3. Finden Sie bei den Quelltexten dieses Kapitels .
4. Finden Sie bei den Quelltexten dieses Kapitels .

## Expressions in Alias MAYA

1. Finden Sie bei den Quelltexten dieses Kapitels .
2. Finden Sie bei den Quelltexten dieses Kapitels .
3. Finden Sie bei den Quelltexten dieses Kapitels .
4. Finden Sie bei den Quelltexten dieses Kapitels .
5. Finden Sie bei den Quelltexten dieses Kapitels .

## Partikelsysteme und Partikeldynamik in Alias MAYA

1. Finden Sie bei den Quelltexten dieses Kapitels .
2. Finden Sie bei den Quelltexten dieses Kapitels .
3. Finden Sie bei den Quelltexten dieses Kapitels .
4. Finden Sie bei den Quelltexten dieses Kapitels .