

# **Darstellung von Kurven und Flächen**

Proseminar Computergraphik, 10. Juni 2008  
Christoph Dähne

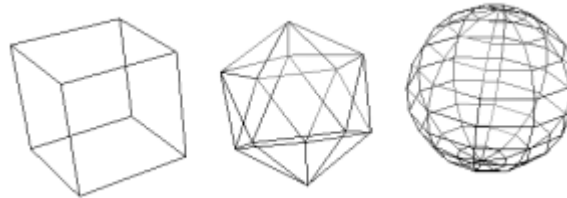
# Inhalt

<b>Polygonnetze</b>	<b>3</b>
Knotenliste	3
Kantenliste	3
<b>Parametrisierte kubische Kurven</b>	<b>4</b>
Definition	4
Stetigkeit	5
Berechnung	6
<b>Hermite-Kurven</b>	<b>7</b>
<b>Bézier-Kurven</b>	<b>9</b>
<b>Parametrisierte kubische Flächen</b>	<b>11</b>
<b>Fraktale Modelle</b>	<b>12</b>
Gebirge	13
Mandelbrot-Menge	14
<b>Quellen</b>	<b>15</b>

# Polygonnetze

Für die Darstellung von dreidimensionalen Objekten im Speicher sind Polygonnetze eine verbreitete und sinnvolle Lösung.

Die Oberfläche eines Objektes wird durch beliebig kleine Teilflächen (Polygone) angenähert, die durch ihre Eckpunkte (Knoten) eindeutig beschrieben werden.



## Knotenliste:

Folglich ist es naheliegend, die Knoten in einer Liste zu speichern. Nun kann man, wenn man weiß, welche Knoten zu welchem Polygon gehören, durch Zeichnen aller Polygone das Objekt darstellen. Dazu ist eine zusätzliche List von Polygonen mit Referenzen auf die Knotenliste nötig.

Diese Knotenliste hat die folgenden Nachteile:

- Die Kanten werden mitunter doppelt gezeichnet.
- Die Frage, welche Polygone an eine bestimmte Kante angrenzen, lässt sich nur mit viel (Rechen-) Aufwand beantworten.
- Das selbe gilt für die Suche nach Flächen an bekannten Knoten.

## Kantenliste:

Der erste Nachteil lässt sich umgehen, indem in der Polygonliste Referenzen auf eine Kantenliste ablegt werden. In der Kantenliste werden dann die Knoten referenziert.

Um das Objekt zu zeichnen, ist es nun nicht mehr nötig Kanten doppelt zu zeichnen.

Selbstverständlich gibt es Möglichkeiten, dreidimensionale Objekte suchfreundlicher im Speicher darzustellen. Wenn man die Suche durch zusätzliche Listen beschleunigt, heißt das aber auch, dass man zunehmend redundante Datensätze anlegt, deren Verwaltung zusätzliche Zeit in Anspruch nimmt, und die deshalb zum Beispiel nicht immer für die Echtzeitbearbeitung geeignet sind, sobald der Arbeitsspeicher nicht mehr ausreicht.

Ferner stellt sich nun die Frage, wie man die fehlenden geometrischen Informationen entlang der Kanten interpoliert. Die Polygone so klein zu wählen, dass man trotz einer linearen Interpolation, einer Verbindung der Knoten durch Geraden, eine runde Darstellung erreicht, ist möglich. Allerdings müssen die Kanten nicht zwangsläufig durch Geraden interpoliert werden.

# Parametrisierte kubische Kurven

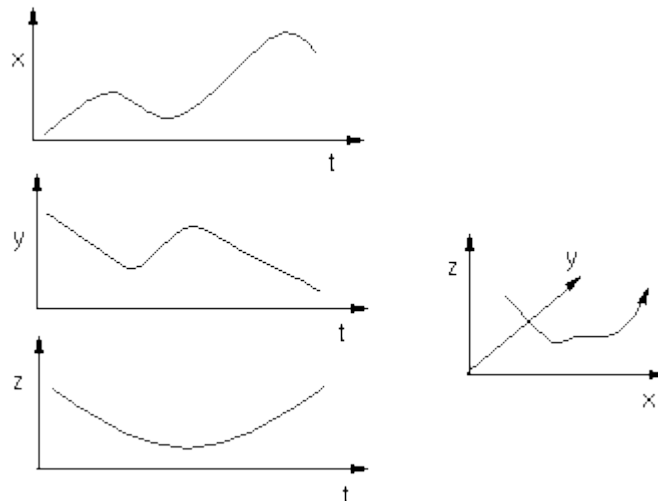
Bevor man konkrete Lösungen entwickeln kann, ist es nötig die allgemeineren mathematischen Grundlagen von parametrisierten kubischen Kurven zu verstehen.

Hier werden wir wieder allein auf den dreidimensionalen Fall eingegangen. Die Herangehensweise eignet sich natürlich für beliebige Dimensionen.

## Definition:

Sei  $Q(t) = (x(t) \ y(t) \ z(t))^T$  ein Kurvensegment, dann gilt:

$$\begin{aligned}x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x, \\y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y, \\z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z, \\0 &\leq t \leq 1\end{aligned}$$



Der zusätzliche Parameter  $t$  hat den Vorteil, dass die Koordinaten des Kurvensegments in alle drei Dimensionen  $x(t)$ ,  $y(t)$ ,  $z(t)$  unabhängig voneinander gesteuert werden können. Das ermöglicht zum Beispiel die Darstellung einer Schleife mit nur zwei zusammengesetzten Kurvensegmenten.

Wegen der Begrenzung des Kurvensegments durch das Intervall, reicht es außerdem aus, wenn das Kurvensegment den gewünschten Verlauf in eben diesem Intervall zeigt.

Ein übersichtlichere Darstellung erreicht man durch die Verwendung von Matrizen:

$$Q(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = C \cdot T$$

mit

$$C = \begin{pmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{pmatrix} \text{ und } T = \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}$$

### Stetigkeit:

Es ist weiterhin wichtig sich über die Stetigkeit der Übergänge zwischen den Kurvensegmenten Gedanken zu machen, bevor wir uns wieder den Kurvensegmenten selbst zuwenden können. Immerhin nützt das rundeste Kurvensegment nichts, wenn der Übergang zum nächsten nicht auch rund ist.

Es gibt folgende Arten der Stetigkeit:

- $G_0$ -Stetigkeit
- $G_1$ -Stetigkeit
- (...)
- $G_n$ -Stetigkeit
- $C_n$ -Stetigkeit

Eine Kurve aus zwei Kurvensegmenten ist zum Beispiel  $G_0$ -stetig, falls der letzte Punkt des ersten Kurvensegmentes gleich dem ersten Punkt des zweiten Kurvensegmentes ist. Anders ausgedrückt heißt das, dass die beiden Kurvensegmente miteinander verbunden sind.

Im Allgemeinen sind Kurven  $G_n$ -stetig, falls die Kurvensegmente an den Übergängen in der  $n$ -ten Ableitung bis auf einen Skalierungsfaktor  $k$  übereinstimmen:

$$Q_0^{(n)}(1) = k \cdot Q_1^{(n)}(0)$$

Ist darüber hinaus der Skalierungsfaktor  $k=1$ , so spricht man von einer  $C_n$ -Stetigkeit. Daraus folgt trivialerweise, dass  $C_n$ -Stetigkeit  $G_n$ -Stetigkeit nach sich zieht.  $C_1$ -Stetigkeit kann man sich veranschaulichen, indem man sich vorstellt, auf den Kurvensegmenten entlang zu fahren, indem man  $t$  mit konstanter Geschwindigkeit erhöht.

Wenn die Geschwindigkeit der Fahrt an den Übergängen konstant ist, so ist die Kurve  $C_1$ -stetig.

Im Allgemeinen können deckungsgleiche Kurvensegmente verschiedene Tangentenvektoren, also Anstiege haben. Dieser Zusammenhang lässt sich über eine Anpassung des Intervalls leicht zeigen. Das bedeutet aber, dass die Tangentenvektoren nicht graphisch ermittelt werden können. Das ist aber nicht besonders überraschend, da der Parameter  $t$  bei einer dreidimensionalen Darstellung nicht als Koordinate abgebildet wird.

Berechnung der Kurvensegmente:

Um kubische Polynome eindeutig lösen zu können, braucht man vier Kontrollpunkte ( $P_1$ - $P_4$ ). Sei

$$G = (P_1 \ P_2 \ P_3 \ P_4) \quad \text{und} \quad C = G \cdot M,$$

Dann ergibt sich aus

$$Q(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = C \cdot T$$

$$Q(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$$

$$= (P_1 \ P_2 \ P_3 \ P_4) \cdot \begin{pmatrix} m_{11} & m_{21} & m_{31} & m_{41} \\ m_{12} & m_{22} & m_{32} & m_{42} \\ m_{13} & m_{23} & m_{33} & m_{43} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{pmatrix} \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}$$

Wobei  $G$  „Geometriematrix“ und  $M$  „Basismatrix“ genannte werden. Die Matrix  $M$  enthält hierbei nur reelle Werte. Das Produkt aus  $M$  und  $T$  wird auch „Basisfunktion“ genannt.

Nun haben wir eine allgemeine Darstellung eines Kurvensegments gefunden, in dem die Kontrollpunkte direkt enthalten sind, und man lediglich die Basismatrix  $M$  bestimmen muss, um die vollständige Darstellung eines bestimmten Kurvensegment zu haben. Es zeigt sich, dass je nach Wahl der Kontrollpunkte allgemeingültige Lösungen für  $M$  bei bestimmten Wahlkriterien der Kontrollpunkte gefunden werden können.

Als Beispiel werden hier Hermite- und Bézierkurven betrachtet.

# Hermite-Kurven

Hermitekurven sind parametrisierte kubische Kurven, bei denen man als Kontrollpunkte den Start- und Endpunkt der Kurve sowie die Anstiege an diesen Punkten auswählt.

Seien  $P_1$  und  $P_2$  die Randpunkte und  $R_1$  und  $R_2$  die Anstiege, so ergibt sich:

$$G_H = (P_1 \quad P_2 \quad R_1 \quad R_2)$$

$$Q(t) = G_H \cdot M_H \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix} \qquad Q'(t) = G_H \cdot M_H \cdot \begin{pmatrix} 3t^2 \\ 2t \\ 1 \\ 0 \end{pmatrix}$$

Die Hermite-Basismatrix wird mit  $M_H$  bezeichnet.

Um die Basismatrix zu bestimmen, müssen wir nun die Kontrollpunkte einsetzen, anschließend zusammenfassen und nach  $M_H$  auflösen:

$$Q(0) = P_1 = (P_1 \quad P_2 \quad R_1 \quad R_2) \cdot M_H \cdot (0 \quad 0 \quad 0 \quad 1)^T,$$

$$Q'(0) = R_1 = (P_1 \quad P_2 \quad R_1 \quad R_2) \cdot M_H \cdot (0 \quad 0 \quad 1 \quad 0)^T$$

$$Q(1) = P_2 = G_H \cdot M_H \cdot (1 \quad 1 \quad 1 \quad 1)^T,$$

$$Q'(1) = R_2 = G_H \cdot M_H \cdot (3 \quad 2 \quad 1 \quad 0)^T$$

$$(P_1 \quad P_2 \quad R_1 \quad R_2) = G_H = G_H \cdot M_H \cdot \begin{pmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$\Rightarrow M_H = \begin{pmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}$$

Es ergibt sich:

$$Q(t) = (P_1 \quad P_2 \quad R_1 \quad R_2) \cdot \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}$$

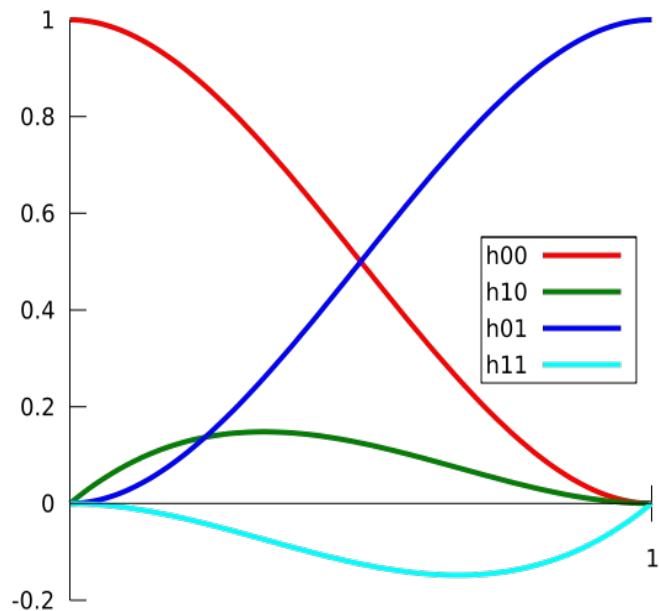
$$Q(t) = G_H \cdot B_H, \qquad B_H = M_H \cdot T \text{ heißt Basisfunktion}$$

Wie bereits erwähnt ist die Basisfunktion für alle Kontrollpunkte gleich. Es ist also nicht nötig, die aufwendige Berechnung der inversen Matrix für jedes Kurvensegment neu anzustellen, sondern es ist zweckmäßiger Weise so, dass die Matrix fest implementiert ist.

Aus der Basisfunktion ergibt sich folgende Gewichtung der Kontrollpunkte für Hermitekurven,

wobei die Ordinate das Gewicht des jeweiligen Kontrollpunkts darstellt und auf der Abszisse der Parameter  $t$  abgetragen ist.

- Rot –  $P_1$
- Blau –  $P_2$
- Grün –  $R_1$
- Hellblau –  $R_2$



Hermitekurven haben die Vorteil, dass alle Kontrollpunkte interpoliert werden. Die Anstiege  $R_1$  und  $R_2$  sind hierbei gesondert zu betrachten, und gelten nicht als „klassische“ Kontrollpunkte. Ferner haben Hermitekurven immer den Stetigkeitsgrad  $C_0$ , da die Kurvensegmente selbstverständlich zusammenhängend gewählt werden. Der Stetigkeitsgrad  $C_1$  kann sehr leicht erreicht werden, indem man darauf achtet, dass an allen Übergängen  $R_{2a}$  gleich  $R_{1b}$  ist.



## Bézier-Kurven

Bézierkurven sind nach Pierre Bézier benannt worden, der dieses Verfahren während seiner Anstellung bei Renault entwickelt hat. Die Kontrollpunkte werden wie folgt gewählt:

$$Q(0) = P_1$$

$$Q(1) = P_4$$

$$Q'(0) = R_1 = 3(P_2 - P_1)$$

$$Q'(1) = R_2 = 3(P_4 - P_3)$$

Zur Bestimmung der Basisfunktion können wir die Basisfunktion der Hermitekurven verwenden. Dazu müssen wir die neuen Kontrollpunkte folgendermaßen anpassen:

$$\begin{aligned} G_H &= (P_1 \quad P_2 \quad R_1 \quad R_2) \\ &= (P_1 \quad P_2 \quad P_3 \quad P_4) \cdot \begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{pmatrix} \end{aligned}$$

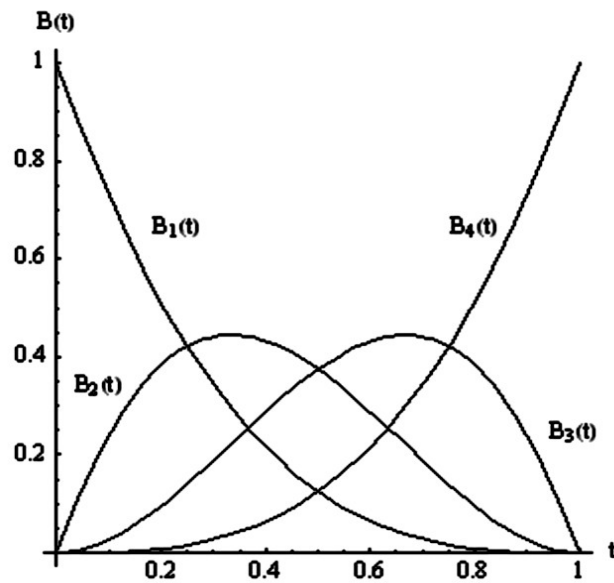
Jetzt können wir die Hermitekurve mit der Hermitegeometriematrix die Kurve ermitteln, indem wir die Hermitebasisfunktion  $B_H = M_H \circ T$  verwenden:

$$\begin{aligned} Q(t) &= G_H \cdot M_H \cdot T \\ &= G_B \cdot \begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{pmatrix} \cdot M_H \cdot T \\ &= G_B \cdot M_B \cdot T \end{aligned}$$

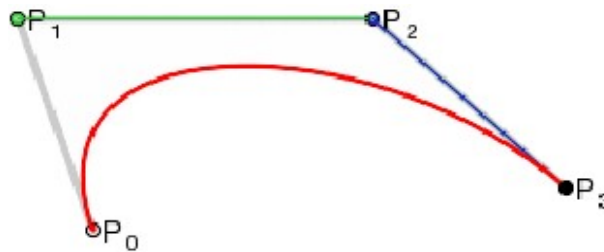
Nun haben wir die Bézierbasismatrix  $M_B$ :

$$\Rightarrow M_B = \begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{pmatrix} \cdot M_H = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Analog zu der Hermitebasisfunktion ergibt sich folgende Gewichtung der Kontrollpunkte in Abhängigkeit von dem Parameter  $t$ :



Der Vorteil der Hermitekurven, dass alle Kontrollpunkte interpoliert werden, geht hier verloren, da die Anstiege nun auch durch zusätzliche Kontrollpunkte definiert werden. Die Kontrollpunkte bilden die konvexe Hülle des Kurvensegments. Das bringt Vorteile bei bestimmten Berechnung, zum Beispiel bei der Frage, ob das Kurvensegment in einem sichtbaren Bereich liegt oder nicht. Bézierkurven sind immer  $C_0$ -stetig und  $C_1$ -stetig, falls an allen Übergängen  $R_{2a}$  gleich  $R_{1b}$  ist.

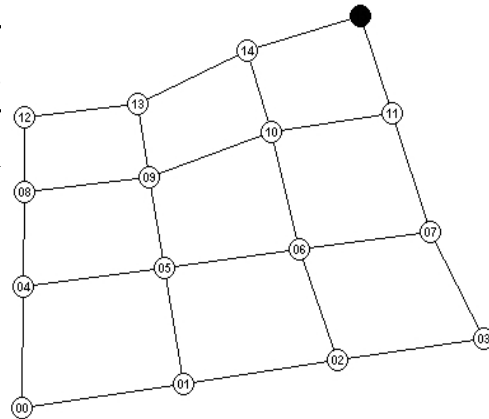


# Parametrisierte kubische Flächen

Es reicht in der Computergraphik nicht aus nur Kurven darzustellen. Der weit häufigere Anwendungsfall ist die Darstellung von ganzen Flächen.

Flächen bestehen andererseits nur aus einer unendlichen Menge von Kurven, die auf der Fläche liegen. Es reicht für die Darstellung aus, die Kurven zu betrachten, die parallel zueinander liegen, und muss diese entlang einer anderen Kurve zeichnen.

Folglich benötigt man 4x4 Kontrollpunkte.



$$Q(s, t) = (P_1(t) \ P_2(t) \ P_3(t) \ P_4(t)) \cdot M \cdot S$$

$$= T^T \cdot M^T \cdot \begin{pmatrix} P_{11} & P_{21} & P_{31} & P_{41} \\ P_{12} & P_{22} & P_{32} & P_{42} \\ P_{13} & P_{23} & P_{33} & P_{43} \\ P_{14} & P_{24} & P_{34} & P_{44} \end{pmatrix} \cdot M \cdot S$$

mit  $P_i(t) = T^T \cdot M^T \cdot (P_{i1} \ P_{i2} \ P_{i3} \ P_{i4})^T$

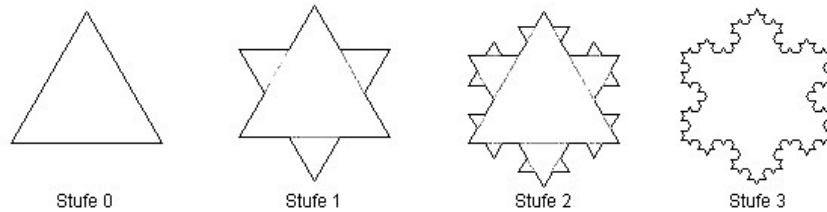
Das Flächensegment ist also wie folgt beschrieben:

$$Q(s, t) = T^T \cdot M^T \cdot G \cdot M \cdot S$$

Durch Einsetzen konkreter Basismatrizen in die Gleichung kann man wahlweise Hermite- oder Bézierflächen erstellen.

# Fraktale Modelle

Fraktale Modelle werden neben den Polygonnetzen als anderer Ansatz zur Darstellung verwendet. Der Begriff „Fraktal“ wurde von Benoît Mandelbrot geprägt. Fraktale sind selbstähnliche Muster.



Selbstähnlich bedeutet in diesem Zusammenhang, dass das dargestellte Objekt sich selbst enthält. In der Natur vorkommende Dinge sind auch selbstähnlich, wie Berge, Pflanzen, Küstenlinien und Flussverläufe, wobei die Selbstähnlichkeit weniger streng und weniger oft vorhanden ist.



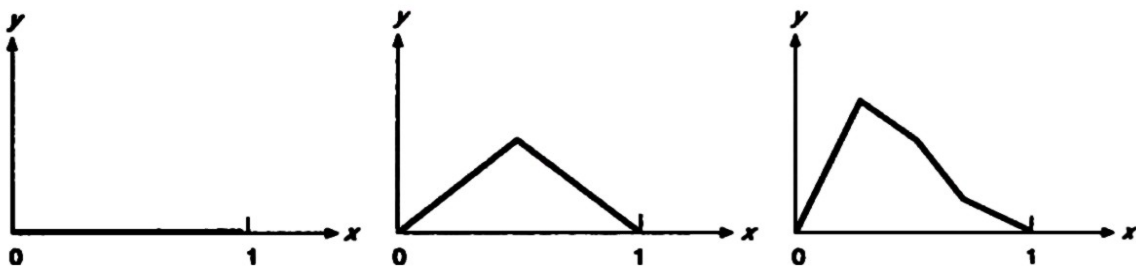
Der Gewinn aus fraktalen Modellen für die Computergraphik liegt in der Möglichkeit, komplexe dreidimensionale Formen mittels Fraktalen zu berechnen, anstatt die Formen als Polygonnetz oder anderweitig speichern zu müssen.



In automatisch generierten Landschaften erhält man durch Fraktale zusätzliche Abwechslung, wenn zum Beispiel mehrere gleichartige Berge dargestellt werden sollen. Für den Betrachter ist sehr schnell erkennbar, wenn es sich um identische Felsformationen handelt.

#### Fraktale Gebirge:

Das folgende Beispiel zum Generieren von Gebirgen ist für die bessere Verständlichkeit zunächst im zweidimensionalen Fall erläutert.



Zuerst wird die Grundlinie des Gebirges festgelegt. Nun wird der Mittelpunkt der Linie um einen durch den konkreten Algorithmus festgelegten Betrag erhöht, und dieser Schritt wiederholt. Durch mehrere Iterationen entsteht ein Gebirge. Implementiert man nun in dem Algorithmus nun etwas zufällige Höhenänderungen und Iterationslängen erhält man bei mehrmaligem Anwenden verschiedene Gebirge.





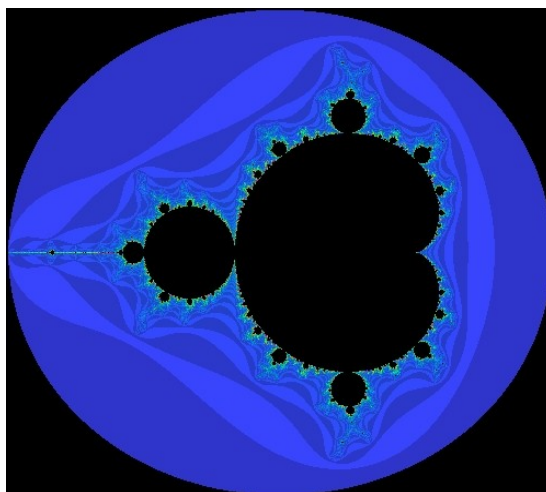
Die Mandelbrot-Menge (Apfelmännchen):

Neben den Fraktalen, mit denen natürliche Dinge abgebildet werden sollen, gibt es auch andere Fraktale mit einem ästhetischen Wert. Ein Beispiel hierfür ist die Mandelbrot-Menge, die manchmal auch Apfelmännchen genannt wird.

Aus der mathematischen Definition kann man erstaunlicher Weise nicht sofort erkennen, dass es sich um ein Fraktal handelt. Die Mandelbrot-Menge ist die Menge aller Punkte in der komplexen Zahlenebene, für deren Wert  $C$  die Reihe

$$X_{n+1} = X_n^2 + C, \text{ mit } X_0 = 0$$

konvergiert.



# Quellen

- James D. Foley, Andries van Dam, Steven K. Feiner,
- John F. Hughes, Richard L. Phillips:
- Grundlagen der Computergraphik, Addison-Wesley, 1994.
- Michael Bender, Manfred Brill:
- Computergrafik, Hanser, 2006.
- <http://www.cs.helsinki.fi/group/goa/mallinnus/curves/curves.html>
- <http://home.ph-freiburg.de/>
- <http://de.wikipedia.org/>
- <http://en.wikipedia.org/>

Apfelmännchen-Applets:

- <http://www.jjam.de/>

Fractal Landscape:

- <http://www.planetside.co.uk/terrigen/>