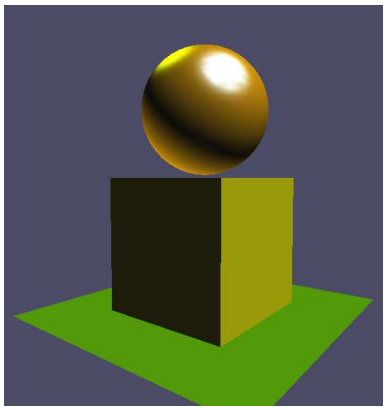
Praktische Mathematik für Medieninformatiker

PD Dr. Thorsten Hüls
Universität Bielefeld
Fakultät für Mathematik

Sommersemester 2015



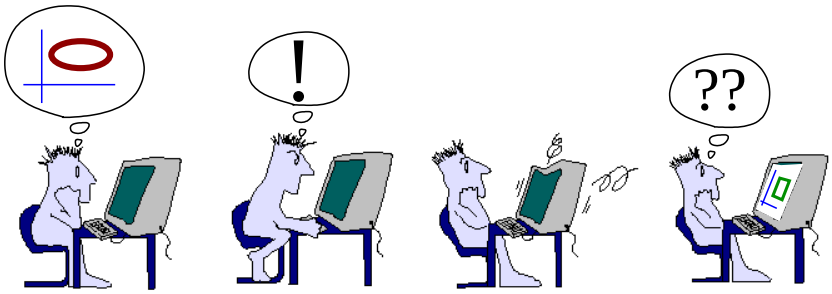
Inhaltsverzeichnis

Kapitel 1

Einleitung

Viele Anwender sind der Meinung, dass es zur sinnvollen Verwendung eines Computerprogramms ausreicht, die Benutzeroberfläche bedienen zu können. Die zugrundeliegenden Algorithmen werden als „Black Box“ angesehen und sind für den Benutzer nicht von Interesse.

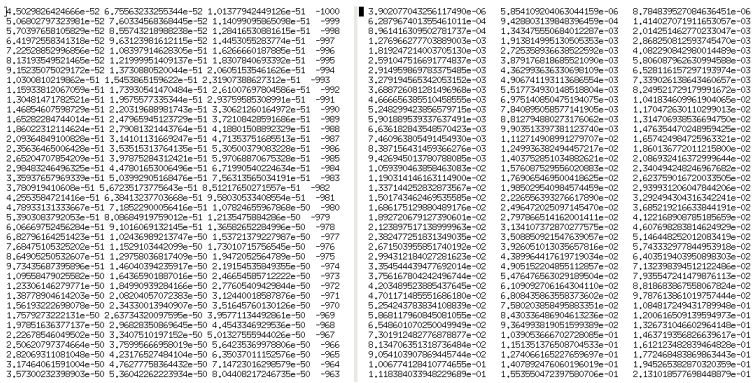
Dieses ist eine kurzsichtige Einstellung, denn beim Vorliegen spezieller Problemstellungen ist es sehr wohl hilfreich, die mathematischen Grundlagen der verschiedenen numerischen Verfahren zu kennen. Hierzu werden in dieser Vorlesung zunächst Kenntnisse aus dem Bereich der linearen Algebra und der Analysis vertieft. Das gewonnene Wissen erleichtert entscheidend die Auswahl eines geeigneten Software-Tools, bzw. ermöglicht erst die Entwicklung und Implementierung neuer Algorithmen.



Zur Illustration betrachten wir die Fragestellung: Es sollen Daten, die ein Anwender berechnet oder gemessen hat und die in Form zweier Listen vorliegen, in einer geeigneten Form visualisiert werden. Folgende Realisierungen sind denkbar:

Darstellung von Wertetabellen

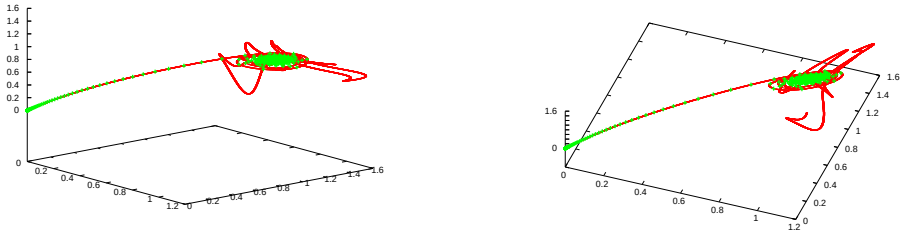
- Am einfachsten ist es, eine Wertetabelle auszugeben.



Aber man erkennt nicht, was berechnet wurde.

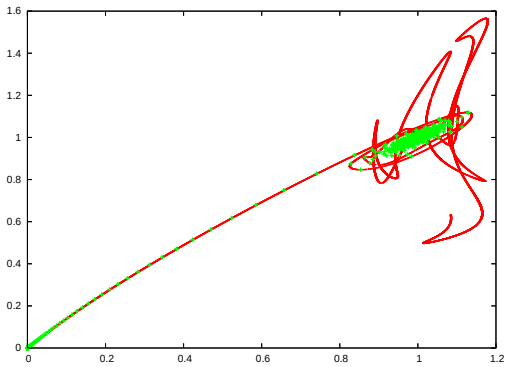
- Als nächstes wird mit einem Plot-Programm eine 3D-Grafik erstellt.

  Zum Erhalt dieser Darstellung muss das Programm das Bild wie angegeben drehen und anschließend, da unsere Monitore *nur* zweidimensionale Bilder darstellten können, auf die Betrachtungsebene projizieren.

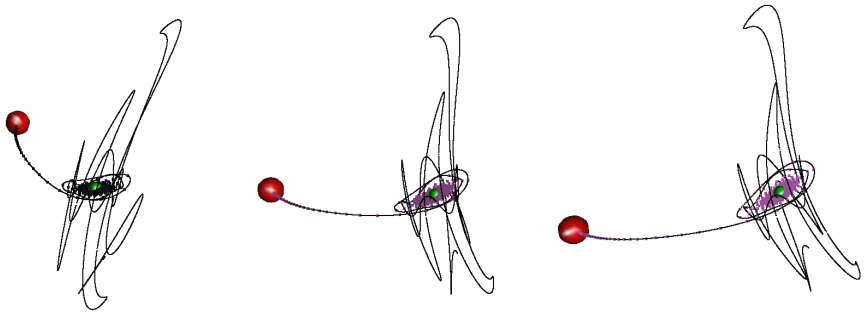


Somit stellt sich die Frage, wie Projektionen, Drehungen, Translationen und Scherungen mathematisch definiert werden.

Eine besonders einfache Projektion wird beispielsweise durch Weglassen einer Koordinate definiert.



- Einen besseren Eindruck erhält man durch Erstellen einer Animation, beispielsweise unter Verwendung der OpenGL-Bibliothek.

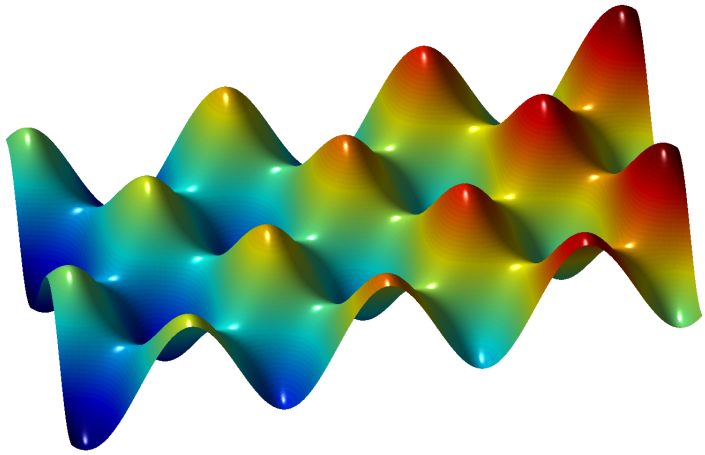


Um zu verstehen, wie diese Animation berechnet wird, ist neben den bereits angesprochenen elementaren Transformationen auch ein Verständnis der – zum Beispiel von der OpenGL-Bibliothek verwendeten – projektiven Geometrie hilfreich.

Zur Animation von 3D-Objekten müssen diese mathematisch definiert werden. In dieser Vorlesung führen wir eine formale Beschreibung (Parametrisierung) von sogenannten analytisch beschreibbaren Oberflächen ein.

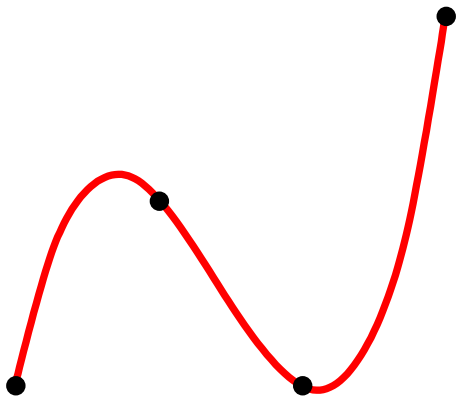
Zur realistischen Darstellung dieser Objekte wird zusätzlich ein Beleuchtungsmodell benötigt, das die Berechnung von Reflexionen erlaubt. Intern sind hierzu Normalenvektoren an die gegebenen Oberflächen zu bestimmen, mit deren Hilfe die reflektierten Lichtstahlen berechnet werden.

Durch Kenntnis der mathematischen Hintergründe können interessante Oberflächen elegant beschrieben werden.



Ein weiteres Themengebiet dieser Vorlesung sind Interpolationsprobleme. Hier untersuchen wir,

- ob zu gegebenen Datenpaaren in der Ebene ein Polynom existiert, das durch diese Daten verläuft,
- unter welchen Annahmen dieses Polynom eindeutig bestimmt ist,
- wie man solch ein Interpolationspolynom numerisch berechnen kann.

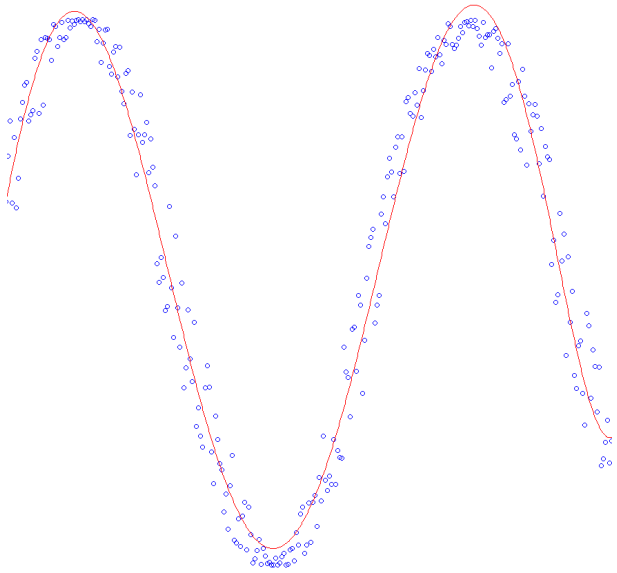


Den Interpolationsansatz verwenden wir auch, um Formeln zur numerischen Differentiation einer Funktion herzuleiten.

Im Abschnitt Minimierungs- und Ausgleichsprobleme zeigen wir, wie Extrema in höheren Raumdimensionen bestimmt werden können, z. B. für die oben auf dieser Seite gezeichnete Funktion.

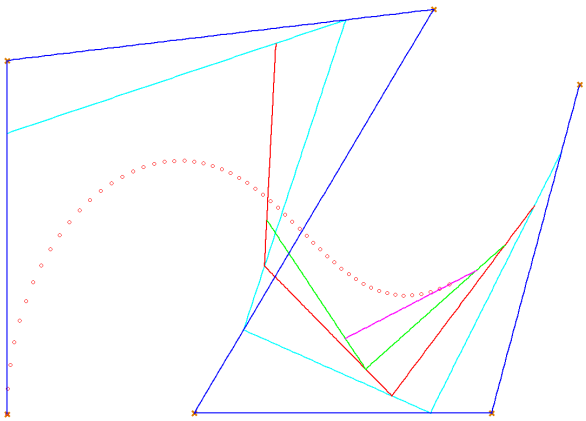
Anschließend betrachten wir sogenannte Ausgleichsprobleme. Hierbei wird eine Funktion oder speziell ein Polynom gesucht, das möglichst nah an den gegebenen Datenpaaren liegt, die beispielsweise aus einer

biologischen Messreihe stammen. Für diese Problemstellung ist der Interpolationsansatz nicht sinnvoll anwendbar, da die Ausgleichs-Funktion nicht jedem Messfehler folgen soll.



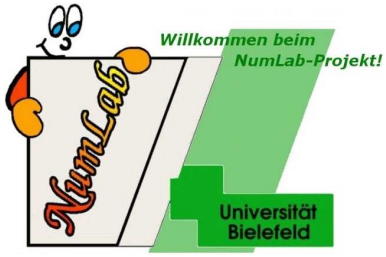
Alternativ können gegebene Daten auch mit Exponentialsummen interpoliert werden. Wir lernen in diesem Zusammenhang die diskrete Fourier-Transformation kennen, die viele Anwendungen in der Computergrafik besitzt. Beispiele hierfür sind Kompressionsverfahren oder das in der Abbildung illustrierte Fokus-Stacking.

Auch gehen wir der Frage nach, wie Kurven mit Hilfe weniger Kontrollpunkte konstruiert werden können, so dass sie der Computer schnell darstellen und transformieren kann. Diese Fragestellung führt auf die sogenannten Bézierkurven.



1.1 Software

Viele Abbildungen in diesem Skript wurden mit dem Programmpaket MATLAB, siehe [2], erstellt. Die MATLAB-Toolbox NUMLAB enthält verschiedene Programme zur Illustration numerischer Methoden und steht unter [6] zum Download bereit.



Eine kostenfreie MATLAB-Alternative ist das Programmpaket SCILAB, siehe [3], welches wir auch zum Lösen der praktischen Übungsaufgaben verwenden. Auch das Computeralgebrasystem MAPLE, siehe [1], kam bei der Erstellung dieses Skripts zum Einsatz.

1.2 Literaturhinweise

Für die einzelnen, in der Vorlesung angesprochenen Bereiche der Mathematik, bietet sich die folgende Literatur zur Vertiefung an.

- Lineare Algebra: [12], [15].
- Analysis: [4], [5], [9], [13].
- Numerische Mathematik: [11], [16], [17], [18], [20], [23], [14].
- Fourier-Transformation: [22], [17].

Kapitel 2

Lineare Algebra mit Anwendungen

In diesem Kapitel werden die grundlegenden Begriffe der linearen Algebra wiederholt und erweitert.

2.1 Lineare Transformationen im $\mathbb{R}^2$

Sei eine Matrix A und ein Vektor x gegeben:

$$A = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathbb{R}^{2,2}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2.$$

Die Abbildung

$$F: \begin{array}{ccc} \mathbb{R}^2 & \to & \mathbb{R}^2 \\ x & \mapsto & Ax = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} ax_1 + cx_2 \\ bx_1 + dx_2 \end{pmatrix} \end{array} \tag{2.1}$$

ist linear.

Zur Erinnerung: Eine Abbildung $F : \mathbb{R}^2 \to \mathbb{R}^2$ heißt **linear**, falls die folgenden Bedingungen erfüllt sind:

(1) $F(x + y) = F(x) + F(y)$ für alle $x,\ y \in \mathbb{R}^2$,

(2) $F(\lambda x) = \lambda F(x)$ für alle $\lambda \in \mathbb{R},\ x \in \mathbb{R}^2$.

Eine Abbildung der Form

$$F: \begin{array}{ccc} \mathbb{R}^2 & \to & \mathbb{R}^2 \\ x & \mapsto & Ax + v \end{array}$$

heißt **affin linear**.

Beachte, dass die Abbildung $G(x) := F(x) - F(0)$ wieder linear ist.

Mit SCILAB kann die Matrixmultiplikation sehr einfach durchgeführt werden.

```
-->A = [1  2;3  4]
 A  =
    1.    2.
    3.    4.

-->x = [5;6]
 x  =
    5.
    6.

-->b = A*x
 b  =
    17.
    39.
```

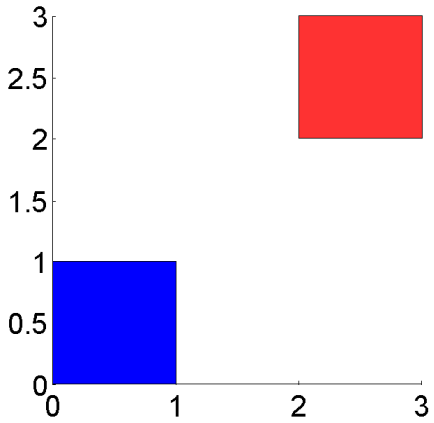
Ein Beispiel einer einfachen affin linearen Abbildung ist eine **Translation**.

$$x \mapsto x + v. \tag{2.2}$$

Im Beispiel wird der Einheitsquader

$$Q = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1 e^1 + x_2 e^2 : 0 \leq x_1, x_2 \leq 1 \right\}, \quad e^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} e^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

um den Vektor $v = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ verschoben.



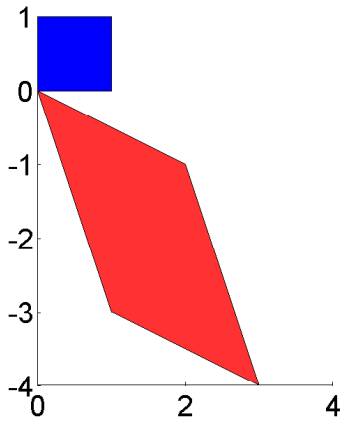
Wird mit einer linearen Abbildung (2.1) transformiert, so erhält man

$$AQ = \left\{ x_1 Ae^1 + x_2 Ae^2 = x_1 \begin{pmatrix} a \\ b \end{pmatrix} + x_2 \begin{pmatrix} c \\ d \end{pmatrix}, 0 \leq x_1, x_2 \leq 1 \right\}.$$

Ein Beispiel ist für die Matrix

$$A = \begin{pmatrix} 2 & 1 \\ -1 & -3 \end{pmatrix}$$

angegeben.



Jede affin lineare Abbildung kann als Hintereinanderausführung einer linearen Abbildung und einer Translation dargestellt werden.

Im Allgemeinen kann eine lineare Transformation:

- Längen und Flächen verändern,
- die Orientierung ändern.

Spezielle lineare Abbildungen / Matrizen, die die Länge und zum Teil auch die Orientierung nicht verändern, werden im folgenden Abschnitt vorgestellt.

2.2 Euklidischer Vektorraum und orthogonale Matrizen

Betrachte die **euklidische Norm**

$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}} = \left(x^T x \right)^{\frac{1}{2}}, \quad x \in \mathbb{R}^n.$$

Definiere das **innere Produkt (Skalarprodukt)**

$$\langle x, y \rangle := \sum_{i=1}^{n} x_i y_i = x^T y. \tag{2.3}$$

Mit einem inneren Produkt wird der $\mathbb{R}^n$ zu einem **euklidischen Vektorraum.**

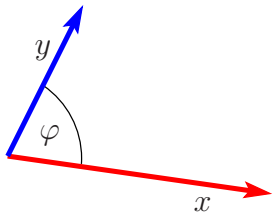
Zur Erinnerung: Ein inneres Produkt besitzt die folgenden Eigenschaften

$$\begin{aligned}
\langle x, x\rangle &> 0 \quad \forall x \in \mathbb{R}^n, x \neq 0,\\
\langle x, y\rangle &= \langle y, x\rangle \quad \forall x, y \in \mathbb{R}^n,\\
\langle ax, y\rangle &= a\langle x, y\rangle \quad \forall a \in \mathbb{R},\ x, y \in \mathbb{R}^n,\\
\langle x + y, z\rangle &= \langle x, z\rangle + \langle y, z\rangle \quad \forall x, y, z \in \mathbb{R}^n.
\end{aligned}$$

Zwischen dem in (2.3) definierten inneren Produkt und der euklidischen Norm besteht ein enger Zusammenhang; es gilt:

$$\|x\|_2^2 = \langle x, x\rangle.$$

Das innere Produkt kann wie folgt veranschaulicht werden.



Man erhält (durch Anwendung des Kosinussatzes):

$$\langle x, y\rangle = \|x\|_2 \|y\|_2 \cos\varphi.$$

Lemma 2.1 *Für* $A \in \mathbb{R}^{n,n}$ *und* $x, y \in \mathbb{R}^n$ *gilt*

$$\langle Ax, y\rangle = \langle x, A^T y\rangle.$$

Beweis: Direktes Nachrechnen liefert:

$$\begin{aligned}
\langle Ax, y\rangle &= \sum_{i=1}^{n} (Ax)_i y_i \\
&= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{ij} x_j \right) y_i \\
&= \sum_{j=1}^{n} \sum_{i=1}^{n} A_{ij} x_j y_i \\
&= \sum_{j=1}^{n} x_j \left(\sum_{i=1}^{n} \underbrace{A_{ij}}_{A^T_{ji}} y_i \right) \\
&= \sum_{j=1}^{n} x_j (A^T y)_j \\
&= \langle x, A^T y\rangle.
\end{aligned}$$

Alternativ kann das Lemma auch mit der aus der linearen Algebra bekannten Identität

$$(BC)^T = C^T B^T, \quad B \in \mathbb{R}^{n,m},\ C \in \mathbb{R}^{m,k}$$

bewiesen werden:

$$\langle Ax, y\rangle = (Ax)^T y = x^T A^T y = \langle x, A^T y\rangle.$$

■

Definition 2.2 Eine Matrix $A \in \mathbb{R}^{n,n}$ heißt

- **orthogonal**, falls

  $$\|Ax\|_2 = \|x\|_2 \quad \forall x \in \mathbb{R}^n$$

  gilt, d. h. die Länge eines Vektors wird durch die Anwendung von A nicht verändert. Diese Aussage ist äquivalent zu $A^T A = I$, siehe Übungsaufgabe.

- **symmetrisch**, falls

  $$A^T = A$$

  erfüllt ist.

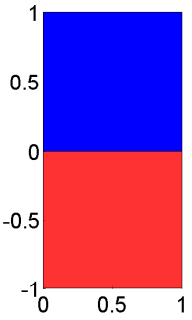
Beachte:

$$I = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix},$$

$$A = \begin{pmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nn} \end{pmatrix}, \quad A^T = \begin{pmatrix} A_{11} & \dots & A_{n1} \\ \vdots & \ddots & \vdots \\ A_{1n} & \dots & A_{nn} \end{pmatrix}.$$

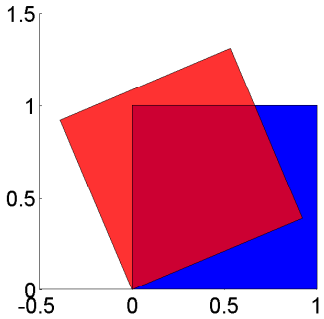
Es werden die Matrizen eingeführt, die eine Spiegelung an der x-Achse bzw. eine Drehung im $\mathbb{R}^2$ beschreiben.

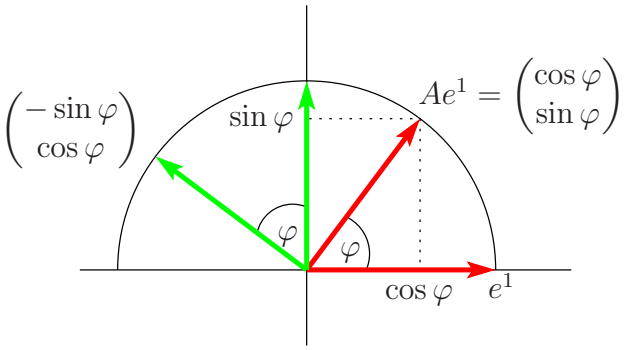
- Die Matrix
$$A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$
definiert eine Spiegelung an der x-Achse.



Beachte:
$$A = A^T = A^{-1}.$$
Somit ist diese Matrix orthogonal.

- Die Matrix
$$D_\varphi = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$
beschreibt eine **Drehung** um φ gegen den Uhrzeigersinn.





Beachte:

$$D_{-\varphi} = D_\varphi^T = \begin{pmatrix}\cos\varphi & \sin\varphi\\ -\sin\varphi & \cos\varphi\end{pmatrix} = D_\varphi^{-1},$$

denn

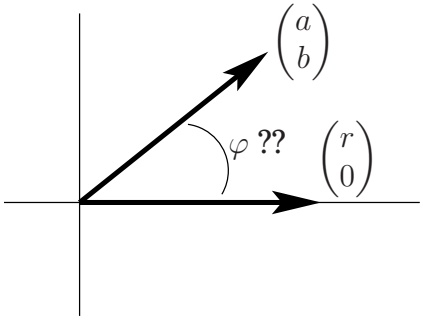
$$\begin{aligned} D_\varphi D_{-\varphi} &= \begin{pmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{pmatrix}\begin{pmatrix}\cos\varphi & \sin\varphi\\ -\sin\varphi & \cos\varphi\end{pmatrix}\\ &= \begin{pmatrix}\cos^2\varphi+\sin^2\varphi & \cos\varphi\sin\varphi-\sin\varphi\cos\varphi\\ \sin\varphi\cos\varphi-\cos\varphi\sin\varphi & \sin^2\varphi+\cos^2\varphi\end{pmatrix}\\ &= \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}. \end{aligned}$$

2.2.1 Matrix gesucht: Eine erste numerische Aufgabe

Gegeben sei der Vektor $\begin{pmatrix}a\\ b\end{pmatrix} \in \mathbb{R}^2$, $a, b \neq 0$. Finde eine Drehmatrix D_φ, so dass $D_{-\varphi}\begin{pmatrix}a\\ b\end{pmatrix}$ auf der positiven x_1-Achse liegt, also

$$D_{-\varphi}\begin{pmatrix}a\\ b\end{pmatrix} = \begin{pmatrix}r\\ 0\end{pmatrix}, \quad \text{bzw.} \quad \begin{pmatrix}a\\ b\end{pmatrix} = D_\varphi\begin{pmatrix}r\\ 0\end{pmatrix} \tag{2.4}$$

gilt.



Schreibe die Matrix $D_{-\varphi}$ in der Form

$$D_{-\varphi} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \quad c^2 + s^2 = 1, \quad \text{mit} \quad \begin{matrix} s = \sin\varphi \\ c = \cos\varphi \end{matrix}.$$

Als notwendige Bedingung an r erhalten wir

$$\left\| \begin{pmatrix} r \\ 0 \end{pmatrix} \right\|_2 = r = \left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\|_2 = (a^2 + b^2)^{\frac{1}{2}}.$$

Aus 2.4 folgt

$$D_{-\varphi} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \begin{matrix} ca + sb &=& r \\ -sa + cb &=& 0 \end{matrix} \quad \Leftrightarrow \quad \begin{matrix} sca + s^2 b &=& sr \\ -sca + c^2 b &=& 0 \end{matrix},$$

und durch Addition dieser Gleichungen folgt

$$sr = s^2 b + c^2 b = b(s^2 + c^2) = b \quad \Rightarrow s = \frac{b}{r},$$

und

$$c = \frac{sa}{b} = \frac{\frac{b}{r}a}{b} = \frac{a}{r}.$$

Somit erhalten wir die gesuchte Drehmatrix:

$$D_{-\varphi} = \frac{1}{r} \begin{pmatrix} a & b \\ -b & a \end{pmatrix}.$$

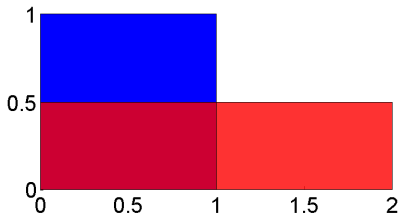
Für $\varphi \in (-\frac{\pi}{2}, \frac{\pi}{2})$ kann der Drehwinkel mit Hilfe der Tangens-Funktion bestimmt werden; es gilt

$$\tan\varphi = \frac{\sin\varphi}{\cos\varphi} = \frac{b}{a} \Longrightarrow \varphi = \arctan\left(\frac{b}{a}\right).$$

Zur Bestimmung dieses Winkels muss die Länge r nicht berechnet werden.

2.3 Skalierungen und Scherungen

- Die Matrix
$$A = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$
definiert eine **Skalierung** und im Fall $\lambda_1 = \lambda_2$ eine **Streckung**. Im Beispiel ist der Fall $\lambda_1 = 2$, $\lambda_2 = \frac{1}{2}$ angegeben.



Beachte, dass die Matrix A symmetrisch ist und

$$A^T A = \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix}$$

gilt. A ist somit genau dann orthogonal, wenn $\lambda_{1,2} \in \{\pm 1\}$ erfüllt ist. In diesem Fall beschreibt A eine Spiegelung oder eine Drehung um π oder die Identität.

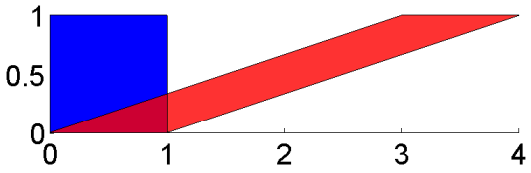
- Die Matrix

$$A = \begin{pmatrix} 1 & c \\ 0 & 1 \end{pmatrix}$$

definiert eine **Scherung**. Die Inverse wird durch

$$A^{-1} = \begin{pmatrix} 1 & -c \\ 0 & 1 \end{pmatrix}$$

gegeben. Die Abbildung zeigt den Fall $c = 3$.



Beachte: Für $b = 0$, $a \neq 0$ gilt die Darstellung:

$$\begin{pmatrix} a & c \\ 0 & d \end{pmatrix} = \underbrace{\begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}}_{\text{Skalierung}} \cdot \underbrace{\begin{pmatrix} 1 & \frac{c}{a} \\ 0 & 1 \end{pmatrix}}_{\text{Scherung}}, \tag{2.5}$$

d. h. Matrizen der Form $\begin{pmatrix} a & c \\ 0 & d \end{pmatrix}$ können in eine Skalierung und eine Scherung *zerlegt* werden.

2.4 QR-Zerlegung einer beliebigen 2×2 Matrix

Sei eine beliebige 2×2 Matrix

$$A = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \quad \text{mit} \quad \begin{pmatrix} a \\ b \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

gegeben. Wir zeigen in diesem Abschnitt, dass die Matrix A als Produkt einer Drehung, einer Skalierung und einer Scherung dargestellt werden kann.

Nach Abschnitt 2.2.1 existiert eine Drehung $D_{-\varphi}$ mit

$$D_{-\varphi} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \quad \text{und} \quad r^2 = a^2 + b^2 > 0,$$

und zusammen mit (2.5) folgt

$$D_{-\varphi} A = \begin{pmatrix} r & s \\ 0 & t \end{pmatrix} = \begin{pmatrix} r & 0 \\ 0 & t \end{pmatrix} \begin{pmatrix} 1 & \frac{s}{r} \\ 0 & 1 \end{pmatrix}. \tag{2.6}$$

Also ist

$$\begin{aligned} A &= D_{\varphi} \begin{pmatrix} r & 0 \\ 0 & t \end{pmatrix} \begin{pmatrix} 1 & \frac{s}{r} \\ 0 & 1 \end{pmatrix} \\ &= \text{Drehung} * \text{Skalierung} * \text{Scherung} \\ &= D_{\varphi} \begin{pmatrix} r & s \\ 0 & t \end{pmatrix} \qquad (2.7) \\ &= QR \quad \text{mit} \quad Q = D_{\varphi}, \quad R = \begin{pmatrix} r & s \\ 0 & t \end{pmatrix}, \end{aligned}$$

wobei Q orthogonal und R eine rechte obere Dreiecksmatrix ist. Diese Darstellung wird auch QR**-Zerlegung** der Matrix A genannt.

In SCILAB sind Routinen implementiert, die die QR-Zerlegung einer Matrix berechnen.

```
A = [1  2; 3  4]
[Q,R] = qr(A)
I = Q'* Q
```

Ergebnis:

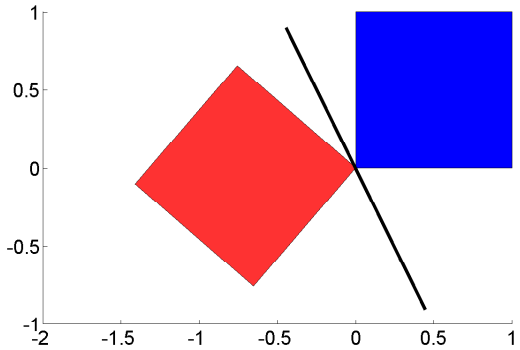
```
A  =
     1.      2.
     3.      4.

 R  =
  - 3.1622777   - 4.4271887
    0.          - 0.6324555

 Q  =
  - 0.3162278   - 0.9486833
  - 0.9486833     0.3162278

 I  =
     1.      0.
     0.      1.
```

2.5 Spiegelungen

Die Matrix

$$S_\varphi = \begin{pmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{pmatrix}$$

beschreibt eine **Spiegelung** an der Geraden, die mit der x-Achse den Winkel $\frac{\varphi}{2}$ einschließt.



Diese Aussage folgt, da

$$D_{-\varphi}S_\varphi = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

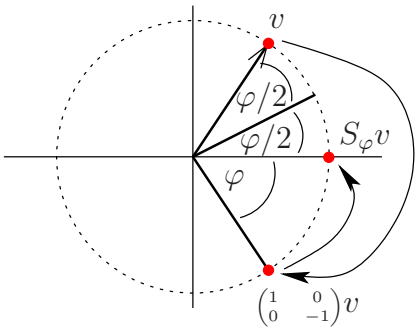
die in Abschnitt 2.2 eingeführte Spiegelung an der x-Achse ist, und somit erhält man

$$S_\varphi = D_\varphi \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.8}$$

Beachte:

$$S_{-\varphi} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ -\sin\varphi & -\cos\varphi \end{pmatrix} \quad \text{also} \quad S_\varphi^{-1} = S_\varphi = S_\varphi^T \neq S_{-\varphi}.$$

Die Abbildung zeigt, wie ein Vektor v, der mit der x-Achse den Winkel φ einschließt, mittels (2.8) abgebildet wird.



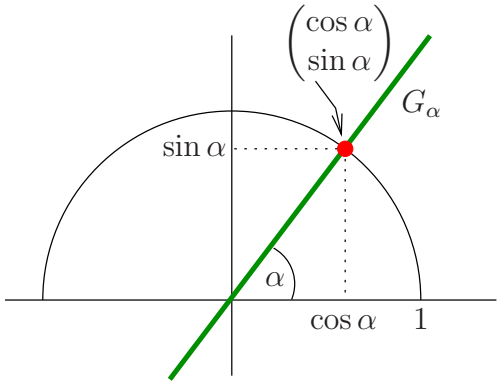
Die Spiegelungsachse kann auch analytisch bestimmt werden. Sei $\varphi \in [0, 2\pi)$. Wir suchen eine Gerade G_α, die durch S_φ auf sich selbst abgebildet wird, d. h. es gilt

$$S_\varphi x = x \quad \forall x \in G_\alpha. \tag{2.9}$$

Eine Gerade, die zur x-Achse den Winkel α besitzt, wird durch

$$G_\alpha := \left\{ \lambda \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} : \lambda \in \mathbb{R} \right\}$$

definiert.



Bestimme jetzt $\alpha \in [0, \pi)$ (unter Verwendung der Additionstheoreme) so, dass (2.9) erfüllt ist:

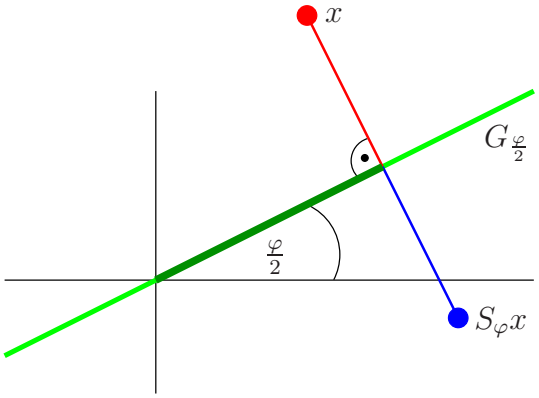
$$\begin{aligned} S_\varphi \begin{pmatrix} \lambda\cos\alpha \\ \lambda\sin\alpha \end{pmatrix} &= \lambda S_\varphi \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} = \lambda \begin{pmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{pmatrix} \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \\ &= \lambda \begin{pmatrix} \cos\varphi\cos\alpha + \sin\varphi\sin\alpha \\ \sin\varphi\cos\alpha - \cos\varphi\sin\alpha \end{pmatrix} = \lambda \begin{pmatrix} \cos(\varphi-\alpha) \\ \sin(\varphi-\alpha) \end{pmatrix} \\ &\stackrel{!}{=} \lambda \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}. \end{aligned}$$

Es folgt $\alpha = \frac{\varphi}{2}$ und wir erhalten die Spiegelungsachse

$$G_{\frac{\varphi}{2}} := \left\{ \lambda \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{pmatrix} : \lambda \in \mathbb{R} \right\}$$

Beliebige Vektoren $x \in \mathbb{R}^2$ können in einen Anteil in Richtung der Geraden $G_{\frac{\varphi}{2}}$ und einen zweiten, dazu orthogonalen Anteil zerlegt werden:

$$x = \lambda \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{pmatrix} + \mu \begin{pmatrix} -\sin\frac{\varphi}{2} \\ \cos\frac{\varphi}{2} \end{pmatrix}.$$

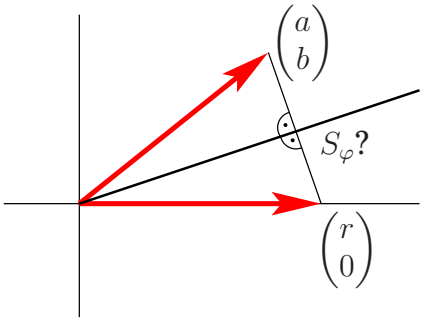


Jetzt gilt wegen der Linearität und den Additionstheoremen

$$\begin{aligned} S_\varphi x &= \lambda S_\varphi \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{pmatrix} + \mu S_\varphi \begin{pmatrix} -\sin\frac{\varphi}{2} \\ \cos\frac{\varphi}{2} \end{pmatrix} \\ &= \lambda \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{pmatrix} + \mu \begin{pmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{pmatrix} \begin{pmatrix} -\sin\frac{\varphi}{2} \\ \cos\frac{\varphi}{2} \end{pmatrix} \\ &= \lambda \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{pmatrix} + \mu \begin{pmatrix} -\cos\varphi\sin\frac{\varphi}{2} + \sin\varphi\cos\frac{\varphi}{2} \\ -\sin\varphi\sin\frac{\varphi}{2} - \cos\varphi\cos\frac{\varphi}{2} \end{pmatrix} \\ &= \lambda \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{pmatrix} + \mu \begin{pmatrix} \sin\frac{\varphi}{2} \\ -\cos\frac{\varphi}{2} \end{pmatrix}. \end{aligned}$$

2.5.1 Matrix gesucht: Eine zweite numerische Aufgabe

Gegeben sei der Vektor $\begin{pmatrix} a \\ b \end{pmatrix}$, $a, b \neq 0$.



Finde eine Spiegelung S_φ, so dass $S_\varphi \begin{pmatrix} a \\ b \end{pmatrix}$ auf der positiven x_1-Achse liegt, also

$$S_\varphi \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}, \quad r = (a^2 + b^2)^{\frac{1}{2}}, \quad \begin{matrix} s = \sin\varphi \\ c = \cos\varphi \end{matrix}$$

erfüllt ist. Analog zu den Überlegungen aus Abschnitt 2.2.1 erhalten wir das Gleichungssystem

$$\begin{matrix} ca + sb &=& r \\ sa - cb &=& 0 \end{matrix} \quad \Leftrightarrow \quad \begin{matrix} c^2a + scb &=& cr \\ s^2a - scb &=& 0 \end{matrix}.$$

Die Addition dieser Gleichungen liefert

$$c^2a + s^2a = (c^2 + s^2)a = a = cr \quad \Rightarrow \quad c = \frac{a}{r},$$

und zusätzlich gilt

$$s = \frac{cb}{a} = \frac{\frac{a}{r}b}{a} = \frac{b}{r}.$$

Es folgt

$$S_\varphi = \frac{1}{r} \begin{pmatrix} a & b \\ b & -a \end{pmatrix},$$

vgl. die Ergebnisse in Abschnitt 2.2.1.

2.6 Charakterisierung orthogonaler 2×2 Matrizen

2.6.1 Charakterisierung mit Hilfe der Determinante

Wir zeigen in diesem Abschnitt, dass Drehungen und Spiegelungen die einzigen orthogonalen – also Längen erhaltenden – linearen Abbildungen im $\mathbb{R}^2$ sind. Somit muss eine vorgegebene orthogonale 2×2 Matrix Q in einer dieser Klassen liegen.

Anhand des Vorzeichens der Determinante von Q bestimmen wir, welcher Fall vorliegt.

Lemma 2.3 *Sei Q eine orthogonale Matrix, dann gilt*

$$\det Q = \pm 1.$$

Beweis: Da Q orthogonal ist, erhalten wir

$$\begin{aligned} I &= Q^T Q \\ \Rightarrow 1 &= \det(I) = \det(Q^T Q) = \det Q^T \det Q = \det Q \det Q = (\det Q)^2, \end{aligned}$$

und somit folgt die Behauptung.

■

Bemerkung 2.4 *Es ist zu beachten, dass die Umkehrung von Lemma 2.3 im Allgemeinen **falsch** ist. Als Gegenbeispiel betrachten wir die Matrix*

$$A = \begin{pmatrix} 1 & 27 \\ 0 & 1 \end{pmatrix}.$$

*Die Bedingung $\det(A) = \pm 1$ ist erfüllt, aber offensichtlich ist die Matrix A **nicht** orthogonal.*

Lemma 2.5 *Sei Q eine orthogonale 2×2 Matrix. Dann beschreibt Q entweder eine Spiegelung oder eine Drehung.*

Beweis: Schreibe Q als Produkt einer Drehung und einer Matrix in rechter oberer Dreiecksform (vgl. (2.7)):

$$Q = D_\varphi \underbrace{\begin{pmatrix} r & s \\ 0 & t \end{pmatrix}}_{=R}.$$

Da die Matrizen Q und D_φ orthogonal sind, muss auch R orthogonal sein, d. h. $R^T R = I$. Somit folgt:

$$I = R^T R = \begin{pmatrix} r & 0 \\ s & t \end{pmatrix} \begin{pmatrix} r & s \\ 0 & t \end{pmatrix} = \begin{pmatrix} r^2 & rs \\ sr & s^2 + t^2 \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

mit $r > 0$, $s, t \in \mathbb{R}$. Also erhalten wir $r = 1$, $s = 0$, $t = \pm 1$ und somit

$$Q = D_\varphi \begin{pmatrix} 1 & 0 \\ 0 & \pm 1 \end{pmatrix}.$$

Im Fall $Q = D_\varphi \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ liegt eine Drehmatrix vor, und

im Fall $Q = D_\varphi \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ beschreibt Q eine Spiegelung, vgl. (2.8).

■

Die Determinante einer Drehmatrix ist immer identisch 1:

$$\det D_\varphi = \det \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} = \cos^2\varphi + \sin^2\varphi = 1.$$

Es folgt

$$\det Q = (\pm 1) \det D_\varphi = \pm 1.$$

Fazit:

- $\det Q = 1$: In diesem Fall ist $Q = D_\varphi$; Q ist eine Drehmatrix.
- $\det Q = -1$: In diesem Fall ist Q eine Spiegelung, vgl. (2.8).

2.6.2 Charakterisierung anhand der Dimension des Fixpunktraumes

Alternativ können orthogonale 2×2 Matrizen auch anhand der Dimension des Fixpunktraumes

$$V = \{x : Ax = x\}$$

charakterisiert werden.

Beachte: Aus der Linearität folgt für alle $\lambda \in \mathbb{R}$:

$$Ax = x \quad \Rightarrow \quad A\lambda x = \lambda A x = \lambda x.$$

Beispiel 2.6 Die Fixpunkte der orthogonalen Abbildung

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

bestimmt man mittels

$$\begin{aligned} Ax &= x \\ \Leftrightarrow \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \end{aligned}$$

Dieses lineare Gleichungssystem

$$\begin{aligned} -x_1 &= x_1 \\ x_2 &= x_2 \end{aligned}$$

besitzt die Lösungen $x_1 = 0$, x_2 beliebig und wir erhalten den Fixpunktraum

$$V = \left\{ \begin{pmatrix} 0 \\ x_2 \end{pmatrix} : x_2 \in \mathbb{R} \right\}.$$

Dieses ist ein ein-dimensionaler Unterraum des $\mathbb{R}^2$.

Da die Abbildung A in diesem Beispiel eine Spiegelung an der y-Achse beschreibt, liegt die Vermutung nahe, dass *jede* orthogonale Abbildung, die einen ein-dimensionalen Fixpunktraum besitzt, eine Spiegelung ist. Dieses wird im Folgenden systematisch untersucht.

Sei A eine orthogonale Matrix.

(a) $\underline{\dim V = 2:}$ Dann ist jeder Punkt ein Fixpunkt, d. h. $Ax = x$ für alle $x \in \mathbb{R}^2$. Also gilt $A = I$.

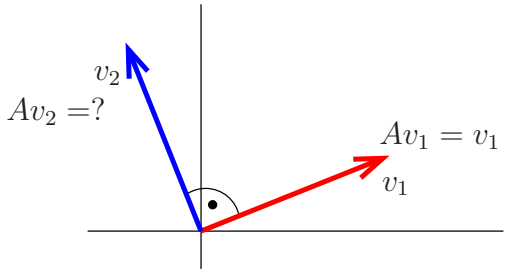
(b) $\underline{\dim V = 1:}$ Der ein-dimensionale Fixpunktraum V wird von einem Vektor $v_1 \neq 0$ aufgespannt, d. h.

$$V = \{\lambda v_1 : \lambda \in \mathbb{R}\}$$

und es gilt:

$$A\lambda v_1 = \lambda v_1 \quad \forall \lambda \in \mathbb{R}.$$

Wähle jetzt $v_2 := v_1^{\perp} \Leftrightarrow \langle v_1, v_2 \rangle = 0$, siehe Abbildung.



Mit Lemma 2.1 erhalten wir

$$\begin{aligned} \langle Av_2, v_1 \rangle &= \langle Av_2, Av_1 \rangle \\ &= \langle v_2, A^T Av_1 \rangle \\ &= \langle v_2, v_1 \rangle = \langle v_1, v_2 \rangle = 0, \end{aligned}$$

d. h. $Av_2 = \lambda v_2$ und da A längenerhaltend ist, folgt $\lambda = \pm 1$.

Im Fall $\lambda = 1$ ist v_2 wegen $Av_2 = v_2$ ein Fixpunkt; es folgt $\dim V = 2$; ein Widerspruch.

Also ist $\lambda = -1$ und es gilt $Av_2 = -v_2$.

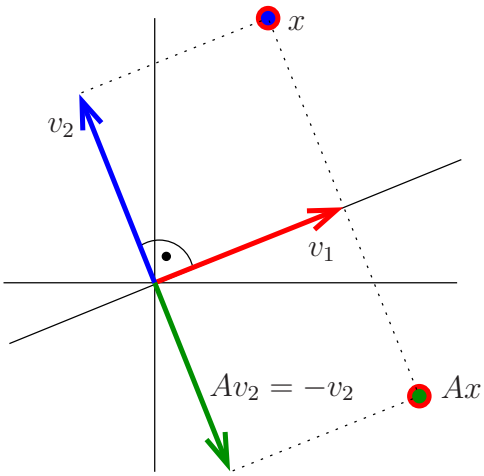
Wie in Abschnitt 2.5 zeigen wir, dass die Matrix A eine Spiegelung beschreibt. Sei x ein beliebiger Vektor im $\mathbb{R}^2$. Wir finden die Zerlegung

$$x = \lambda_1 v_1 + \lambda_2 v_2$$

und es folgt

$$\begin{aligned} Ax &= A(\lambda_1 v_1 + \lambda_2 v_2) \\ &= \lambda_1 A v_1 + \lambda_2 A v_2 \\ &= \lambda_1 v_1 - \lambda_2 v_2, \end{aligned}$$

d. h. der Vektor x wird an $V = \{\lambda_1 v_1 : \lambda_1 \in \mathbb{R}\}$ gespiegelt, wie die Abbildung zeigt.



(c) $\underline{\dim V = 0}$: Folglich ist $V = \{0\}$. A ist keine Spiegelung, weil bei einer Spiegelung die Achse, an der gespiegelt wird, immer im Fixpunktraum V liegt und somit $\dim V = 1$ gilt.

Nach Abschnitt 2.6.1 sind Drehungen und Spiegelungen die einzigen orthogonalen Abbildungen. Also kann im Fall $\dim V = 0$ nur eine Drehung vorliegen.

2.7 Eigenwerte und Eigenvektoren

In diesem Abschnitt führen wir die zentralen Begriffe „Eigenvektor“ und „Eigenwert“ ein.

Definition 2.7 Sei eine Matrix $A \in \mathbb{C}^{n,n}$ gegeben, dann ist $v \in \mathbb{C}^n$ ein **Eigenvektor** von A zum **Eigenwert** $\lambda \in \mathbb{C}$, $\lambda \neq 0$, falls

$$Av = \lambda v$$

gilt.

Der Eigenvektor v wird also durch Anwendung der linearen Abbildung A auf das λ-fache von v abgebildet.

Beispiel 2.8 Sei $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$, dann gilt

$$A \begin{pmatrix} -1 \\ 1 \end{pmatrix} = -1 \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \text{und} \quad A \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

die Matrix A besitzt also den Eigenvektor $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ zum Eigenwert -1 und den Eigenvektor $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ zum Eigenwert 3.

Eigenwerte und Vektoren können auch mit SCILAB berechnet werden:

```
A = [1 2 ; 2 1]
[v,lambda] = spec(A)
```

Ergebnis:

```
lambda  =
  - 1.    0.
    0.    3.

v  =
  - 0.7071068    0.7071068
    0.7071068    0.7071068
```

Die Eigenwerte stehen auf der Diagonalen von `lambda` und die Spalten von `v` enthalten die Eigenvektoren von A. Es ist zu beachten, dass Eigenvektoren nicht eindeutig sind. Genauer ist mit v auch $\mu \cdot v$ ein Eigenvektor für alle $\mu \in \mathbb{R}$, $\mu \neq 0$.

Wichtige Beobachtungen:

- Reelle Matrizen können komplexe Eigenwerte besitzen. Als Beispiel betrachten wir die Drehung um den Winkel $\frac{3}{2}\pi$, beschrieben durch die Matrix $D_{\frac{3}{2}\pi} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. Diese Matrix besitzt die komplex konjugierten Eigenwerte $\lambda_1 = i$ und $\lambda_2 = -i$ zu den Eigenvektoren $\begin{pmatrix} 1 \\ i \end{pmatrix}$ und $\begin{pmatrix} 1 \\ -i \end{pmatrix}$.

Die Eigenwerte der Matrix A sind die Nullstellen des **charakteristischen Polynoms**

$$p(\lambda) := \det(A - \lambda I),$$

wie das folgende Lemma zeigt.

Lemma 2.9 *Für $A \in \mathbb{C}^{n,n}$ sind die folgenden Aussagen äquivalent:*

(1) *λ ist ein Eigenwert von A.*

(2) $\det(A - \lambda I) = 0$.

Beweis:

(1) $\Rightarrow$ (2): Sei λ ein Eigenwert von A zum Eigenvektor v, $v \neq 0$, d. h. $Av = \lambda v$. Es folgt

$$(A - \lambda I)v = Av - \lambda v = \lambda v - \lambda v = 0,$$

also ist die Matrix $A - \lambda I$ nicht invertierbar und somit ist $\det(A - \lambda I) = 0$.

(2) $\Rightarrow$ (1): Sei $\det(A - \lambda I) = 0$, dann ist die Matrix $A - \lambda I$ nicht invertierbar. Folglich existiert ein $v \in \mathbb{C}^n$, $v \neq 0$ mit $(A - \lambda I)v = 0$. Diese Bedingung ist äquivalent zu $Av = \lambda v$ und somit ist v ein Eigenvektor zum Eigenwert λ.

■

Für die Matrix aus Beispiel 2.8 erhalten wir folgende Nullstellen des charakteristischen Polynoms:

$$0 = p(\lambda) = \det(A - \lambda I) = \det \begin{pmatrix} 1 - \lambda & 2 \\ 2 & 1 - \lambda \end{pmatrix} = (1 - \lambda)^2 - 4 \Leftrightarrow \lambda \in \{-1, 3\}.$$

Definition 2.10 Sei $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ und $A \in \mathbb{K}^{n,n}$. Die Menge

$$\mathrm{Eig}(A, \lambda) := \{v \in \mathbb{K}^n : Av = \lambda v\}$$

heißt **Eigenraum** von A zum Eigenwert $\lambda \in \mathbb{K}$.

In der Tat ist $\mathrm{Eig}(A, \lambda)$ ein Untervektorraum des $\mathbb{K}^n$.

In Beispiel 2.8 gilt

$$\mathrm{Eig}(A, -1) = \left\{ \mu \begin{pmatrix} -1 \\ 1 \end{pmatrix} : \mu \in \mathbb{R} \right\} \quad \text{und} \quad \mathrm{Eig}(A, 3) = \left\{ \mu \begin{pmatrix} 1 \\ 1 \end{pmatrix} : \mu \in \mathbb{R} \right\}$$

und wir erhalten $\mathbb{R}^2 = \mathrm{Eig}(A, -1) \oplus \mathrm{Eig}(A, 3)$.

Definition 2.11 Die **geometrische Vielfachheit** des Eigenwerts λ wird durch $\dim(\mathrm{Eig}(A, \lambda))$ definiert.

Die Vielfachheit der Nullstelle λ des charakteristischen Polynoms wird als **algebraische Vielfachheit** von λ bezeichnet.

Beispiel 2.12

- $A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$. Der Eigenwert 3 besitzt sowohl die geometrische als auch die algebraische Vielfachheit 2.
- $A = \begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix}$. Der Eigenwert 3 besitzt die algebraische Vielfachheit 2 und die geometrische Vielfachheit 1.

Bemerkung 2.13

- *Da eine orthogonale Matrix Q längenerhaltend ist, d. h. $\|Qv\|_2 = \|v\|_2$ für alle $v \in \mathbb{R}^n$, kann sie nur Eigenwerte vom Betrag 1 haben.*
- *Der Fixpunktraum V aus Abschnitt 2.6.2 entsprich dem Eigenraum zum Eigenwert 1.*

Lemma 2.14 *Sei $A \in \mathbb{R}^{n,n}$ eine symmetrische Matrix ($A^T = A$), und seien $v_1, v_2 \in \mathbb{R}^n$ Eigenvektoren zu den Eigenwerten $\lambda_1 \neq \lambda_2$. Dann stehen diese Eigenvektoren senkrecht aufeinander, d. h. $\langle v_1, v_2 \rangle = v_1^T v_2 = 0$.*

Beweis: Seien die obigen Voraussetzungen erfüllt. Es gilt:

$$\begin{aligned} v_1^T A v_2 &= v_1^T \lambda_2 v_2 = \lambda_2 v_1^T v_2, \\ v_1^T A v_2 &= v_1^T A^T v_2 = (A v_1)^T v_2 = (\lambda_1 v_1)^T v_2 = \lambda_1 v_1^T v_2, \end{aligned}$$

und zusammen erhalten wir

$$\lambda_2 v_1^T v_2 = \lambda_1 v_1^T v_2.$$

Es folgt aus der Voraussetzung $\lambda_1 \neq \lambda_2$ die Behauptung $v_1^T v_2 = 0$.

■

In Beispiel 2.8 haben wir gesehen, dass SCILAB Eigenwerte und Vektoren berechnen kann. Numerisch wird hierzu das charakteristische Polynom **nicht** verwendet. Angenommen, wir wollen nur den betragsmäßig größten Eigenwert einer Matrix A berechnen, was natürlich viel einfacher, als die Bestimmung aller Eigenwerte ist, dann verwenden wir folgende Idee: Nehme einen beliebigen Vektor, und iteriere diesen wiederholt mit der Matrix A. Dann sollte sich die am schnellsten wachsende Richtung – als die gesuchte Eigenrichtung – durchsetzen. Dieser Ansatz führt auf die Potenzmethode, die im folgenden Abschnitt diskutiert wird.

2.7.1 Die Potenzmethode

Wir behandeln ein Verfahren, mit dem einzelne Eigenvektoren und Eigenwerte bestimmt werden können. Wir haben bereits gesehen, dass auch reelle Matrizen komplexe Eigenwerte besitzen können. Die folgenden Überlegungen werden somit gleich allgemein, für Matrizen über dem Körper $\mathbb{C}$ durchgeführt.

Gegeben sei eine quadratische Matrix $A \in \mathbb{C}^{n,n}$. Wir wählen ein $y^0 \in \mathbb{C}^n$ und iterieren gemäß

$$y^{k+1} = Ay^k, \; k = 0, 1, 2, \ldots \tag{2.10}$$

es gilt also $y^k = A^k \, y^0$.

Um das Verhalten von y^k für $k \to \infty$ zu analysieren, nehmen wir an, dass A diagonalisierbar ist mit Eigenwerten $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$, und zugehörigen Eigenvektoren $v_1, \ldots, v_n \in \mathbb{C}^n$, also

$$Av_i = \lambda_i v_i, \quad i = 1, \ldots, n. \tag{2.11}$$

Wir ordnen die Eigenwerte betragsmäßig und nehmen an, dass es nur einen Eigenwert mit dem größten Betrag gibt

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|. \tag{2.12}$$

Im reellen Fall $A \in \mathbb{R}^{n,n}$ ist mit λ_1 auch $\bar{\lambda}_1$ Eigenwert, so dass unter der Annahme (2.12) notwendig $\lambda_1 \in \mathbb{R}$ folgt.

Da die Eigenvektoren $v_1, \ldots, v_n$ den $\mathbb{C}^n$ aufspannen, können wir y^0 zerlegen

$$y^0 = \sum_{i=1}^{n} \alpha_i v_i, \quad \alpha_i \in \mathbb{C}$$

und erhalten für $k \in \mathbb{N}$

$$y^k = A^k y^0 = \sum_{i=1}^{n} \alpha_i A^k v_i = \sum_{i=1}^{n} \alpha_i \lambda_i^k v_i = \lambda_1^k \left(\alpha_1 v_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1} \right)^k \alpha_i v_i \right).$$

Hieraus folgt mit (2.12)

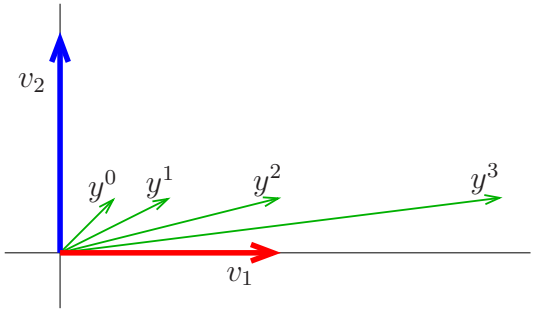
$$\|\lambda_1^{-k} y^k - \alpha_1 v_1\| \leq \sum_{i=2}^{n} \left| \frac{\lambda_i}{\lambda_1} \right|^k |\alpha_i| \|v_i\| \leq \left| \frac{\lambda_2}{\lambda_1} \right|^k \sum_{i=2}^{n} |\alpha_i| \|v_i\| \tag{2.13}$$

und insbesondere

$$\lim_{k \to \infty} \lambda_1^{-k} y^k = \alpha_1 v_1. \tag{2.14}$$

Die Vektoren y^k konvergieren also der Richtung nach gegen ein Vielfaches des Eigenvektors v_1.

Anhand der Matrix $A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ illustriert die folgende Abbildung die ersten Iterierten des Vektors $y^0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.



Da wir λ_1 nicht kennen, bilden wir anstelle von $\lambda_1^{-k} y^k$ eine geeignete „normierte“ Folge z^k. Zum Beispiel kann man ein geeignetes $\psi \in \mathbb{C}^n$ wählen und $\psi^H z^k = 1$ durch die folgende Vorschrift erzwingen

$$z^0 = y^0, \quad z^{k+1} = \frac{1}{\rho_{k+1}} A z^k \quad \text{mit} \quad \rho_{k+1} = \psi^H A z^k, \quad k = 0, 1, 2, \ldots \tag{2.15}$$

Satz 2.15 *Sei* (2.12) *vorausgesetzt und ein Startvektor* $z^0 = \sum_{i=1}^{n} \alpha_i v_i$ *gegeben mit* $\alpha_1 \neq 0$. *Der Vektor* $\psi \in \mathbb{C}^n$ *sei so gewählt, dass* $\psi^H v_1 \neq 0$ *gilt und die Folge* (2.15) *existiert, also* $\rho_k \neq 0$ *für* $k \in \mathbb{N}$ *erfüllt ist. Dann gibt es ein* $C > 0$ *mit*

$$\left\| z^k - \frac{1}{\psi^H v_1} v_1 \right\| + |\rho_{k+1} - \lambda_1| \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^k, \quad k \geq 0 \tag{2.16}$$

und es gilt

$$\lim_{k \to \infty} z^k = \frac{1}{\psi^H v_1} v_1, \quad \lim_{k \to \infty} \rho_k = \lambda_1. \tag{2.17}$$

Beweis: Nach (2.15) haben wir

$$z^k = \frac{1}{\gamma_k}\, A^k z^0 = \frac{1}{\gamma_k} y^k \quad \text{mit} \quad \gamma_k = \prod_{j=1}^{k} \rho_j$$

und

$$1 = \psi^H z^k = \frac{1}{\gamma_k} \psi^H y^k, \quad \gamma_k = \psi^H y^k.$$

Mit Hilfe von (2.13) folgt dann

$$|\gamma_k \lambda_1^{-k} - \alpha_1 \psi^H v_1| = |\psi^H (\lambda_1^{-k} y^k - \alpha_1 v_1)| \le C_1 \left|\frac{\lambda_2}{\lambda_1}\right|^k .$$

Benutzen wir $\alpha_1 \neq 0$, $\psi^H v_1 \neq 0$, so liefert dies mit (2.13)

$$\begin{aligned} \left\| z^k - \frac{1}{\psi^H v_1} v_1 \right\| &= \left\| \frac{\lambda_1^k}{\gamma_k} (\lambda_1^{-k} y^k) - \frac{1}{\alpha_1 \psi^H v_1} (\alpha_1 v_1) \right\| \\ &\le \left\| \Big(\frac{\lambda_1^k}{\gamma_k} - \frac{1}{\alpha_1 \psi^H v_1} \Big) (\lambda_1^{-k} y^k) \right\| + \left\| \frac{1}{\alpha_1 \psi^H v_1} (\lambda_1^{-k} y^k - \alpha_1 v_1) \right\| \\ &\le C_2 \left| \frac{\lambda_2}{\lambda_1} \right|^k . \end{aligned}$$

Schließlich folgt hieraus auch

$$\begin{aligned} |\rho_{k+1} - \lambda_1| &= \left| \psi^H A z^n - \psi^H \left(\frac{1}{\psi^H v_1} A v_1 \right) \right| \\ &= \left| \psi^H A \left(z^k - \frac{1}{\psi^H v_1} v_1 \right) \right| \\ &\le C_3 \left| \frac{\lambda_2}{\lambda_1} \right|^k \end{aligned}$$

und zusammen erhält man (2.16).

Zum Beweis der Aussage (2.17) bilden wir den Grenzwert von (2.16) und erhalten

$$\lim_{k\to\infty} \left\| z^k - \frac{1}{\psi^H v_1} v_1 \right\| + |\rho_{k+1} - \lambda_1| \le C \lim_{k\to\infty} \left| \frac{\lambda_2}{\lambda_1} \right|^k = 0,$$

da $\left|\frac{\lambda_2}{\lambda_1}\right| < 1$ nach (2.12). Somit folgt

$$\lim_{k\to\infty} \left\| z^k - \frac{1}{\psi^H v_1} v_1 \right\| = 0 \quad \text{und} \quad \lim_{k\to\infty} |\rho_{k+1} - \lambda_1| \le C = 0,$$

und diese Aussagen sind äquivalent zu

$$\lim_{k\to\infty} z^k = \frac{1}{\psi^H v_1} v_1 \quad \text{und} \quad \lim_{k\to\infty} \rho_k = \lambda_1.$$

■

Man bezeichnet die Iteration (2.10) und ihre normierten Versionen allgemein als **Potenzmethode** oder einfache **Vektoriteration**. Wenn es nur einen betragsmäßig größten Eigenwert gibt, so läßt er sich zusammen mit seinem Eigenvektor durch die Potenzmethode bestimmen. Die Konvergenzgeschwindigkeit hängt dabei vom Verhältnis $|\frac{\lambda_2}{\lambda_1}|$ des zweitgrößten zum größten Eigenwert ab, und sie kann für $|\frac{\lambda_2}{\lambda_1}| \approx 1$ sehr langsam sein. Die Voraussetzung $\alpha_1 \neq 0$ in Satz 2.15 ist in der Regel unkritisch, da die Iterationsvektoren durch Rundungsfehler stets Anteile in Richtung v_1 erhalten und sich somit die Richtung von v_1 praktisch immer durchsetzt.

2.8 Definitheit und Indefinitheit

Definition 2.16 Eine Matrix $A \in \mathbb{R}^{n,n}$ heißt

- **positiv definit**, falls $x^T Ax > 0$ für alle $x \in \mathbb{R}^n$, $x \neq 0$ gilt.
- **positiv semi-definit**, falls $x^T Ax \geq 0$ für alle $x \in \mathbb{R}^n$ gilt.
- **negativ definit**, falls $x^T Ax < 0$ für alle $x \in \mathbb{R}^n$, $x \neq 0$ gilt.
- **negativ semi-definit**, falls $x^T Ax \leq 0$ für alle $x \in \mathbb{R}^n$ gilt.
- **indefinit**, falls zwei Vektoren $x, y \in \mathbb{R}^n$ existieren, mit $x^T Ax > 0$ und $y^T Ay < 0$.

Beispiele:

- Die Einheitsmatrix ist positiv definit, da $x^T Ix = x^T x = \sum_{i=1}^n x_i^2 > 0$, falls $x \neq 0$.
- Die Matrix $A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ ist positiv semi-definit.
- Die Matrix $A = -I$ ist negativ definit.
- Die Matrix $A = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$ ist negativ semi-definit.
- Die Matrix $A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ ist indefinit.

Lemma 2.17 *Sei $A \in \mathbb{R}^{n,n}$ eine positiv definite (negativ definite) Matrix, die ausschließlich reelle Eigenwerte besitzt. Dann hat A nur positive (negative) Eigenwerte.*

Beweis: Sei $v \in \mathbb{R}^n$ ein Eigenvektor von A zum Eigenwert $\lambda \in \mathbb{R}$, dann gilt $Av = \lambda v$ und

$$v^T A v = v^T \lambda v = \lambda v^T v = \lambda \|v\|_2^2.$$

Es folgt

$$\lambda = \frac{v^T A v}{\|v\|_2^2} \quad \begin{cases} > 0, & \text{falls } A \text{ positiv definit ist,} \\ < 0, & \text{falls } A \text{ negativ definit ist.} \end{cases}$$

■

Bemerkung 2.18 *Die Umkehrung von Lemma 2.17 ist im Allgemeinen falsch! Als Gegenbeispiel betrachten wir die Matrix*

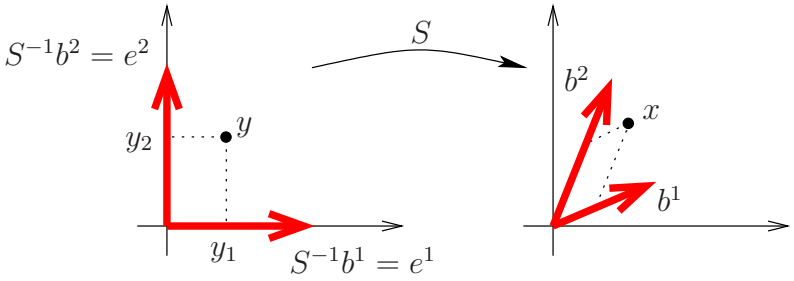
$$A = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}.$$

Diese Matrix besitzt den doppelten positiven Eigenwert 1, ist aber nicht positiv definit, denn mit $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ folgt

$$x^T A x = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -1.$$

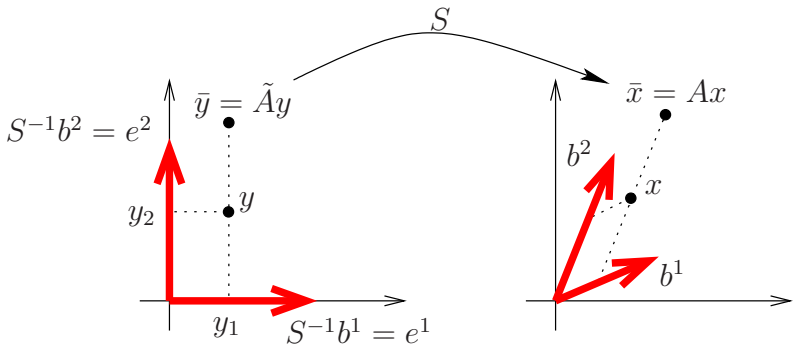
2.9 Ähnlichkeitstransformationen

Gegeben sei eine Matrix $A \in \mathbb{R}^{2,2}$. Diese Matrix beschreibt eine lineare Abbildung bezüglich der Basis $\{e^1, e^2\} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$. Sei $B := \{b^1, b^2\}$ eine weitere Basis des $\mathbb{R}^2$. Wir definieren die Matrix $S = \begin{pmatrix} b^1 & b^2 \end{pmatrix}$ und erhalten $S(e^1) = b^1$, $S(e^2) = b^2$. Zu $x \in \mathbb{R}^2$ setzen wir $y = S^{-1}x$. Den Zusammenhang zwischen dem Originalsystem (links) und dem transformierten System (rechts) veranschaulicht die Abbildung.



Wird im rechten System der Punkt x mit der Matrix A abgebildet, $\bar{x} := Ax$, so entspricht dieser Punkt im transformierten System dem Punkt $\bar{y} := S^{-1}\bar{x}$. Es folgt der Zusammenhang zwischen rechten und linken System:

$$\bar{y} = S^{-1}\bar{x} = S^{-1}Ax = S^{-1}ASy = \tilde{A}y, \quad \text{mit} \quad \tilde{A} = S^{-1}AS.$$



Die Matrizen A und $\tilde{A}$ sind zueinander **ähnlich**.

Ähnliche Matrizen besitzen identische Eigenwerte, wie das folgende Lemma zeigt.

Lemma 2.19 *Seien A und $\tilde{A} \in \mathbb{C}^{n,n}$ zwei ähnliche Matrizen. Ist $\lambda \in \mathbb{C}$ ein Eigenwert von A so ist λ auch ein Eigenwert von $\tilde{A}$.*

Beweis: Da A und $\tilde{A}$ ähnliche Matrizen sind, existiert eine invertierbare Matrix $S \in \mathbb{C}^{n,n}$ mit

$$\tilde{A} = S^{-1}AS.$$

Sei λ ein Eigenwert von A mit dem zugehörigen Eigenvektor v, d. h. $Av = \lambda v$.

Sei $w := S^{-1}v$. Wir zeigen jetzt, dass w ein Eigenvektor von $\tilde{A}$ zum Eigenwert λ ist und folglich besitzt dann, wie behauptet, $\tilde{A}$ den Eigenwert λ.

Es gilt:

$$\begin{aligned} \tilde{A}w &= S^{-1}ASw = S^{-1}ASS^{-1}v = S^{-1}Av = S^{-1}\lambda v = \lambda S^{-1}v \\ &= \lambda w. \end{aligned}$$

■

Sei $A \in \mathbb{R}^{n,n}$ eine diagonalisierbare Matrix und $B = \{v^1, \ldots, v^n\}$ eine Basis des $\mathbb{R}^n$, bestehend aus Eigenvektoren von A. Seien $\lambda_1, \ldots, \lambda_n$ die zugehörigen Eigenvektoren, so gilt $Av^i = \lambda_i v^i$ für alle $i = 1, \ldots, n$. Wir transformieren die Eigenvektoren auf die Koordinatenachsen und zeigen, dass das resultierende System eine einfache Diagonalgestalt besitzt.

Zuerst definieren wir, wie oben, die Matrix $S = \begin{pmatrix} v^1 & \cdots & v^n \end{pmatrix}$ und die zu A ähnliche Matrix $\tilde{A} := S^{-1}AS$. $\tilde{A}$ beschreibt folglich die Abbildung A bezüglich der Eigenbasis B.

Sei e^i der i-te Einheitsvektor. Dann gilt für $i = 1, \ldots, n$, dass $Se^i = v^i$ und somit $S^{-1}v^i = e^i$. Die i-te Spalte von $\tilde{A}$ hat die Form

$$\tilde{A}e^i = S^{-1}ASe^i = S^{-1}Av^i = S^{-1}\lambda_i v^i = \lambda_i S^{-1}v^i = \lambda_i e^i$$

und wir erhalten

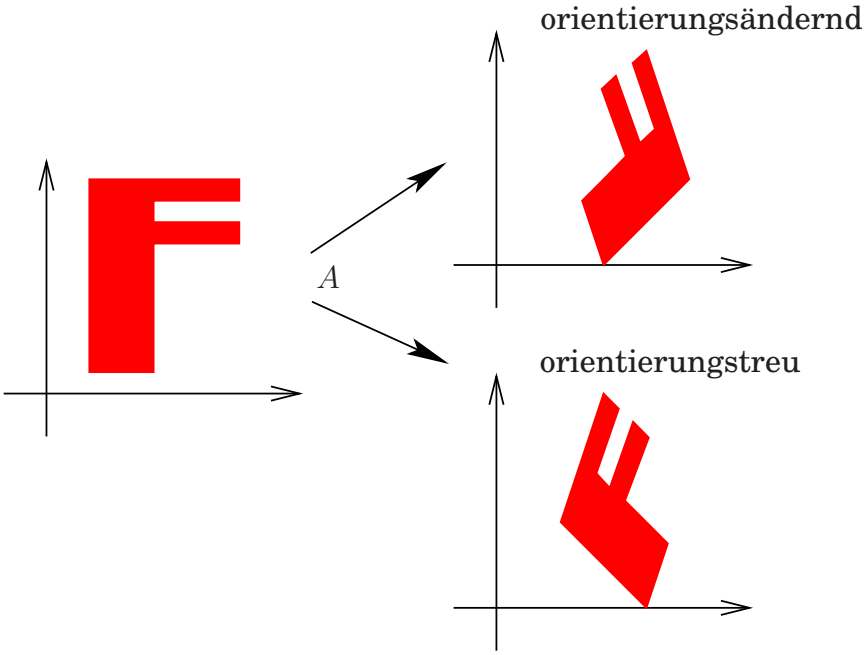
$$\tilde{A} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}.$$

2.10 Orientierung

Definition 2.20 Eine durch die Matrix $A \in \mathbb{R}^{n,n}$ beschriebene lineare Abbildung heißt **orientierungstreu**, falls $\det A > 0$ ist. Gilt $\det A < 0$, so heißt die Abbildung **orientierungsändernd**.

Aus diesen Bedingungen folgt sofort, das orientierungstreue und orientierungsändernde Matrizen invertierbar sind.

Ein Beispiel zeigt die folgende Abbildung.



Sei Q eine orthogonale 2×2-Matrix. Wir haben in Abschnitt 2.6.1 gesehen, dass in diesem Fall $\det Q = \pm 1$ gilt. Genauer haben wir gezeigt:

- $\det Q = 1 > 0$, dann beschreibt Q eine Drehung und diese Abbildung ist orientierungserhaltend.
- $\det Q = -1 < 0$, dann beschreibt Q eine Spiegelung und diese Abbildung ist orientierungsändernd.

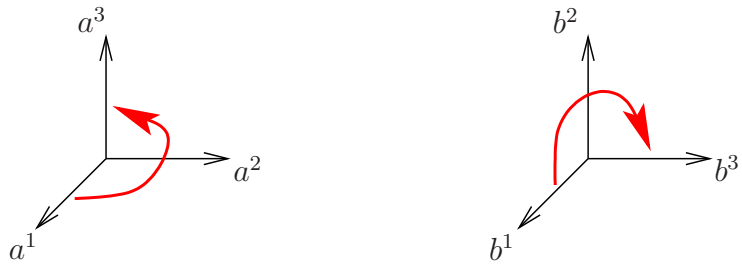
Seien $A = \{a^1, \ldots, a^n\}$ und $B = \{b^1, \ldots, b^n\}$ zwei Basen des $\mathbb{R}^n$. Dann wird durch $Ta^i = b^i$, $i = 1, \ldots, n$ eindeutig eine lineare Abbildung T definiert, die die Basisvektoren aus A in die Basisvektoren aus B überführt.

Wir nennen die Basen A und B **gleich orientiert**, falls die Transformation T orientierungserhaltend ist.

Als Beispiel betrachten wir die zwei Basen

$$A = \{a^1, a^2, a^3\} := \{e^1, e^2, e^3\} \quad \text{und} \quad B = \{b^1, b^2, b^3\} := \{e^1, e^3, e^2\}.$$

Dann gilt $T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, $\det T = -1$ und folglich sind A und B nicht gleich orientiert, wie die Abbildung zeigt.



2.11 Spiegelungen und Rotationen in $\mathbb{R}^3$; Charakterisierung orthogonaler Matrizen im $\mathbb{R}^3$

Gegeben sei eine allgemeine 3×3 Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},$$

die bekanntlich **orthogonal** ist, falls

$$A^T A = I$$

gilt.

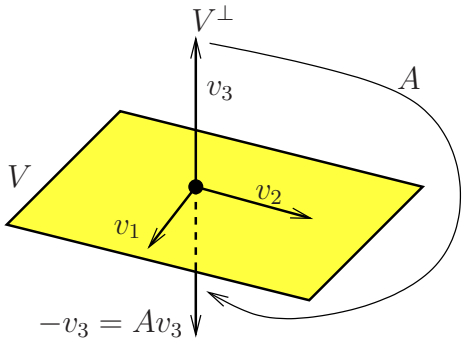
Wir haben in Abschnitt 2.7 gesehen, dass orthogonale Matrizen nur Eigenwerte vom Betrag 1 besitzen. Ähnlich, wie wir es im Fall von 2×2 Matrizen in Abschnitt 2.6.2 durchgeführt haben, kann auch eine 3×3-Matrizen A anhand der Dimension des Eigenraums V zum Eigenwert 1, definiert durch

$$V = \{x : Ax = x\},$$

charakterisiert werden. Hierbei ist zu beachten, dass V der Menge der Fixpunkte der Abbildung A entspricht. Wir diskutieren die möglichen Fälle:

(a) $\underline{\dim V = 3}$: In diesem Fall sind alle Punkte des $\mathbb{R}^3$ Fixpunkte von A, dieses gilt aber nur für $A = I$.

(b) $\underline{\dim V = 2}$:



Seien $v_1,\ v_2$ eine orthonormale Basis von V, d. h.

$$\langle v_i, v_j \rangle = \delta_{ij} \quad \text{mit} \quad \delta_{ij} = \begin{cases} 1, & \text{für } i = j \\ 0, & \text{für } i \neq j. \end{cases}$$

Da die Menge V nur aus Fixpunkten besteht, gilt insbesondere

$$Av_i = v_i, \quad i = 1, 2.$$

Wähle $v_3 \in V^\perp$ (siehe Abbildung), so gilt für $i = 1, 2$ (vgl. Lemma 2.1)

$$\langle Av_3, v_i \rangle = \langle Av_3, Av_i \rangle = \langle v_3, \underbrace{A^T A}_{=I} v_i \rangle = \langle v_3, v_i \rangle = 0.$$

Somit folgt

- $Av_3 = v_3$: dieser Fall ist ausgeschlossen, denn sonst liegt der Fall (a) mit $\dim V = 3$ vor.
- $Av_3 = -v_3$: Nur dieser Fall ist möglich; eine **Spiegelung an der Ebene V** liegt vor.

Neben der Basis $\left\{ \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix} \right\}$ haben wir mit $\{v_1, v_2, v_3\}$ eine zweite Basis des $\mathbb{R}^3$ gefunden. Sei $S = (v_1\ v_2\ v_3)$, dann erhalten wir mittels der Ähnlichkeitstransformation

$$\tilde{A} = S^{-1}AS$$

die Darstellung $\tilde{A}$ dieser Abbildung (bezüglich der Basis $\{v_1, v_2, v_3\}$). Der Vorteil der neuen Darstellung liegt in der einfachen Form der Abbildung $\tilde{A}$, wie die folgende Rechnung zeigt (vgl. auch Abschnitt 2.9):

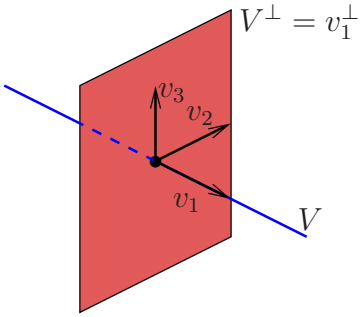
$$\begin{aligned}\tilde{A} &= S^{-1}AS = S^{-1}A(v_1\ v_2\ v_3)\\ &= S^{-1}(Av_1\ Av_2\ Av_3) = S^{-1}(v_1\ v_2\ -v_3)\\ &= \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{pmatrix}.\end{aligned}$$

Beachte, dass diese Abbildung orientierungsändernd ist, da

$$-1 = \det\tilde{A} = \det(S^{-1}AS) = \det(S^{-1})\det(A)\det(S) = \det A$$

gilt, vgl. Abschnitt 2.10.

(c) $\underline{\dim V = 1}$:



Wähle $v_1 \in V$, dann besitzt V die Darstellung

$$V = \{\lambda v_1 : \lambda \in \mathbb{R}\}.$$

Sei $v \in V^\perp$, es gilt (vgl. Fall (b))

$$\langle Av, v_1\rangle = \langle Av, Av_1\rangle = \langle v, A^TAv_1\rangle = \langle v, v_1\rangle = 0$$

und somit:

$$A(V^\perp) \subset V^\perp, \quad \dim V^\perp = 2.$$

$A_{|V^\perp}$ kann durch eine orthogonale 2×2 Matrix dargestellt werden, beschreibt also nach Bemerkung 2.6 eine Spiegelung oder eine Drehung.

- $A_{|V^\perp}$ kann keine Spiegelung beschreiben, denn sonst würden weitere Fixpunkte von A in $V^\perp$ liegen.
- Folglich beschreibt $A_{|V^\perp}$ eine Rotation.

Somit ist A eine **Rotation mit der Drehachse** V.

In der Basis v_1, v_2, v_3 besitzt diese Abbildung die Darstellung

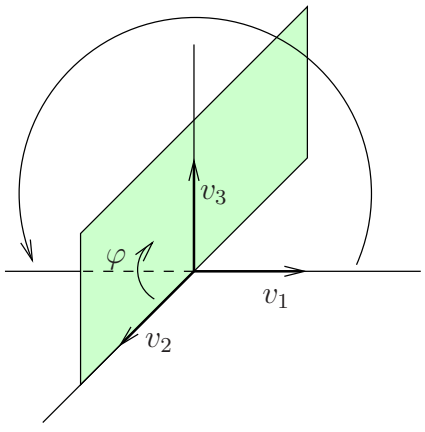
$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix}.$$

Beachte, dass Rotationen orientierungserhaltend sind, da $\det B = 1$ gilt, vgl. Abschnitt 2.10.

(d) $\underline{\dim V = 0}$: Dieser Fall wird zunächst anhand eines Beispiels analysiert. Durch die Matrix

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix}$$

wird eine **Drehspiegelung** (vgl. die Abbildung) definiert.



Offensichtlich ist $\dim V = 0$ erfüllt. Zusätzlich ist diese Abbildung orientierungsändernd.

Durch Auswahl eines geeigneten orthogonalen Koordinatensystems kann auch für eine beliebige orthogonale Matrix A, deren Fixpunktraum null-dimensional ist und die somit nicht den Eigenwert 1 besitzt, nachgewiesen werden, dass durch sie eine Drehspiegelung

beschrieben wird. An dieser Stelle skizzieren wir nur die Idee des Beweises. Zuerst ist die Existenz eines Vektors $v_1 \in \mathbb{R}^3$ nachzuweisen, der $Av_1 = -v_1$ und $\|v_1\|_2 = 1$ erfüllt. Dann zeigt man, dass der senkrecht auf dem Vektor v_1 stehende Raum $V_1^T := \{v \in \mathbb{R}^3 : \langle v_1, v\rangle = 0\}$ durch die Abbildung A wieder in sich abgebildet wird. Die auf diesen Raum eingeschränkte Abbildung $A_{|V_1^T}$ beschreibt eine Drehung, d. h. jeder Punkt aus V_1^T wird bei Anwendung der Abbildung A um die von v_1 erzeugte Gerade gedreht.

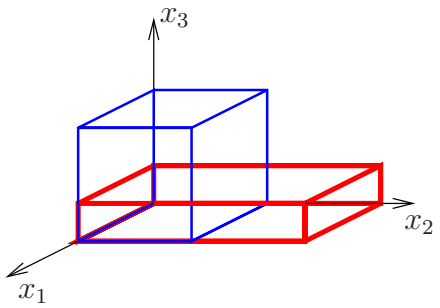
2.12 Scherungen und Skalierungen im $\mathbb{R}^3$

Wie in Abschnitt 2.3 wird durch

$$A = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{2.18}$$

eine **Skalierung** und im Spezialfall $\lambda_1 = \lambda_2 = \lambda_3$ eine **Streckung** definiert.

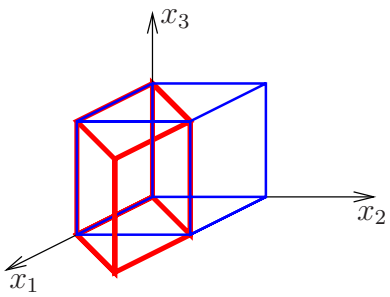
In der Abbildung ist der Fall $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = \frac{1}{3}$ dargestellt.



Die Matrix

$$A = \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix}$$

definiert eine Scherung. Den Fall $a = 1$, $b = c = 0$ zeigt die Abbildung.



Beachte, dass die in diesem Abschnitt diskutierten Abbildungen, außer in den Trivialfällen $|\lambda_1| = |\lambda_2| = |\lambda_3| = 1$ bzw. $a = b = c = 0$, keine orthogonalen Transformationen sind.

Das **Volumen** des Einheitsquaders $Q = [0,1] \times [0,1] \times [0,1]$ ist offensichtlich $\text{vol}(Q) = 1$. Für den transformierten Quader kann das Volumen mit Hilfe der Formel

$$\text{vol}(A(Q)) = |\det A|$$

berechnet werden.

Im Fall einer Skalierung erhält man $\text{vol}(A(Q)) = |\lambda_1 \lambda_2 \lambda_3|$ und dieses Volumen ist im Allgemeinen ungleich eins.

Scherungen ändern das Volumen nicht und sind orientierungserhaltend, denn

$$\det \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix} = 1.$$

2.13 QR-Zerlegung für 3×3 Matrizen

Wie im Fall einer 2×2 Matrix, deren QR-Zerlegung in Abschnitt 2.4 diskutiert wurde, ist es naheliegend, dass wir auch jede Matrix $A \in \mathbb{R}^{3,3}$ in die Form

$$A = QR, \quad Q, R \in \mathbb{R}^{3,3} \tag{2.19}$$

zerlegen können. Hierbei ist Q eine orthogonale Matrix

$$Q^T Q = I$$

und R eine rechte obere Dreiecksmatrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix}.$$

Vorausgesetzt die QR-Zerlegung existiert – dieses wird im Folgenden nachgewiesen – so erhalten wir im Fall $r_{11}, r_{22}, r_{33} \neq 0$ (falls A invertierbar ist)

$$R = \underbrace{\begin{pmatrix} r_{11} & 0 & 0 \\ 0 & r_{22} & 0 \\ 0 & 0 & r_{33} \end{pmatrix}}_{\text{Skalierung}} \underbrace{\begin{pmatrix} 1 & \frac{r_{12}}{r_{11}} & \frac{r_{13}}{r_{11}} \\ 0 & 1 & \frac{r_{23}}{r_{22}} \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Scherung}}.$$

Somit gilt:

$$A = QDS,$$

wobei Q orthogonal, D eine Skalierung und S eine Scherung ist.

Diese Zerlegung einer Matrix ist für viele – nicht nur geometrische – Aufgaben von Bedeutung:

- Lösung von Problemen der kleinsten Fehlerquadrate,
- Berechnung von Eigenwerten,
- statistische Probleme.

Wir weisen jetzt für eine beliebige 3×3 Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

die Existenz der QR-Zerlegung nach.

Nach Abschnitt 2.4, Gleichung (2.6) existiert eine Drehmatrix

$$D_{-\varphi_1} = \begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix}$$

mit

$$D_{-\varphi_1} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{pmatrix}.$$

Da $D_{-\varphi_1}$ orthogonal ist, ist auch die 'erweiterte' Matrix

$$Q_1 := \begin{pmatrix} D_{-\varphi_1} & 0 \\ 0 \quad 0 & 1 \end{pmatrix}$$

orthogonal. Es gilt

$$Q_1 A = \begin{pmatrix} D_{-\varphi_1} & 0 \\ 0 \quad 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Nun wählen wir eine zweite Drehmatrix

$$D_{-\varphi_2} = \begin{pmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{pmatrix}$$

mit

$$D_{-\varphi_2} \begin{pmatrix} b_{11} & b_{12} \\ a_{31} & a_{32} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ 0 & c_{32} \end{pmatrix}.$$

Mit der ebenfalls orthogonalen 'erweiterten' Matrix

$$Q_2 := \begin{pmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & c_2 \end{pmatrix}$$

gilt dann

$$Q_2 Q_1 A = \begin{pmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & c_2 \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ 0 & b_{22} & b_{23} \\ 0 & c_{32} & c_{33} \end{pmatrix}.$$

Schließlich finden wir noch eine dritte Drehmatrix $D_{-\varphi_3}$ mit

$$D_{-\varphi_3} \begin{pmatrix} b_{22} & b_{23} \\ c_{32} & c_{33} \end{pmatrix} = \begin{pmatrix} d_{22} & d_{23} \\ 0 & d_{33} \end{pmatrix}$$

und die orthogonale 'erweiterte' Matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & & \\ 0 & D_{-\varphi_3} & \end{pmatrix}$$

liefert

$$Q_3 Q_2 Q_1 A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & & \\ 0 & D_{-\varphi_3} & \end{pmatrix} \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ 0 & b_{22} & b_{23} \\ 0 & c_{32} & c_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ 0 & d_{22} & d_{23} \\ 0 & 0 & d_{33} \end{pmatrix} = R.$$

Somit folgt

$$A = (Q_3 Q_2 Q_1)^T R = Q_1^T Q_2^T Q_3^T R = QR \quad \text{mit} \quad Q = Q_1^T Q_2^T Q_3^T;$$

die Matrix Q ist als Produkt von drei orthogonalen Matrizen – in diesem Fall Rotationen – selbst wieder orthogonal.

Beachte:

$$\begin{aligned} A \text{ ist orthogonal} &\Rightarrow A^T \text{ ist orthogonal,} \\ A, B \text{ sind orthogonal} &\Rightarrow AB \text{ ist orthogonal.} \end{aligned}$$

Das SCILAB-Programm führt die QR-Zerlegung an einem Beispiel durch:

```
A = [1 2 4 ; 1 3 9; 1 4 16]
[Q,R] = qr(A)
I = Q'* Q
```

Ergebnis:

```
A  =
    1.    2.    4.
    1.    3.    9.
    1.    4.    16.
```

```
R  =

  - 1.7320508   - 5.1961524   - 16.743158
    0.          - 1.4142136   - 8.4852814
    0.            0.            0.8164966


Q  =
  - 0.5773503     0.7071068     0.4082483
  - 0.5773503   - 1.665D-16   - 0.8164966
  - 0.5773503   - 0.7071068     0.4082483

I  =
    1.            5.551D-17     2.776D-17
    5.551D-17     1.            5.551D-17
    2.776D-17     5.551D-17     1.
```

2.14 QR-Zerlegung einer $n \times n$-Matrix

In den Abschnitten 2.4 und 2.13 haben wir gesehen, dass sich jede 2×2 bzw. 3×3 Matrix A in das Produkt einer orthogonalen Matrix Q und einer rechten oberen Dreiecksmatrix R zerlegen lässt. Dort haben wir die QR-Zerlegung als Produkt von Rotationen, Scherungen und Skalierungen konstruiert. Wir verfolgen in diesem Abschnitt den geometrisch motivierten Zugang nicht weiter; es werden nun analytische Verfahren zur Erzeugung der QR-Zerlegung einer beliebigen $n \times n$ Matrix angegeben, die zum Beispiel auch (in abgewandelter Form) von SCILAB verwendet werden.

2.14.1 Eine Anwendung der QR-Zerlegung

Eine Anwendung der QR-Zerlegung besteht in der Lösung von linearen Gleichungssystemen der Form

$$Ay = b, \tag{2.20}$$

wobei $A \in \mathbb{R}^{n,n}$ invertierbar und $b \in \mathbb{R}^n$ gegeben sei; der Vektor $y \in \mathbb{R}^n$ ist gesucht. Die Lösung wird durch

$$y = A^{-1}b$$

bestimmt, doch diese Darstellung ist nur von theoretischem Nutzen, da die Berechnung der Inversen numerisch wesentlich aufwendiger ist, als

die Lösung der Aufgabe (2.20) mit dem im Folgenden vorgestellten Verfahren.

Angenommen, die Matrix A kann QR-zerlegt werden, dann folgt wegen der Orthogonalität der Matrix Q:

$$Ay = QRy = b \quad \Rightarrow Ry = Q^T b = \tilde{b}.$$

Somit ist zum Erhalt von $\tilde{b}$ nur eine (Matrix * Vektor)-Multiplikation durchzuführen. Schließlich kann die Gleichung

$$\begin{pmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_n \end{pmatrix}$$

durch Rückwärtsauflösung berechnet werden:

$$\begin{aligned} y_n &= \frac{\tilde{b}_n}{r_{nn}}, \\ y_{n-1} &= \frac{\tilde{b}_{n-1} - r_{n-1\ n} y_n}{r_{n-1\ n-1}}, \\ &\vdots \\ y_i &= \frac{\tilde{b}_i - \sum_{j=i+1}^{n} r_{ij} y_j}{r_{ii}}, \quad i = n-2, \dots, 1. \end{aligned}$$

Zusätzlich hat die QR-Zerlegung weitere gute Eigenschaften, wie wir in den späteren Anwendungen, insbesondere beim Ausgleichsproblem, sehen werden, vgl. Abschnitt 5.3.

2.14.2 Die Gram-Schmidtsche Methode

In diesem Abschnitt geben wir eine einfache Methode – das sogenannte **Gram-Schmidtsche Orthogonalisierungsverfahren** – zur Berechnung der QR-Zerlegung einer beliebigen $n \times n$ Matrix an.

Gegeben sei eine Matrix

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix} \in \mathbb{R}^{n,n}, \quad a_i \in \mathbb{R}^n,\ i = 1, \dots, n.$$

Gesucht ist die Zerlegung

$$A = QR \tag{2.21}$$

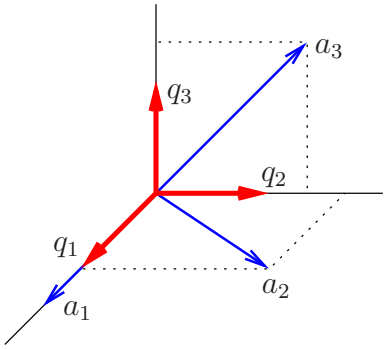
mit

$$Q = \begin{pmatrix} q_1 & q_2 & \dots & q_n \end{pmatrix},\ q_i \in \mathbb{R}^n,\ i = 1, \dots, n, \quad R = \begin{pmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix}.$$

Hierbei sei die Matrix Q orthogonal:

$$Q^T Q = I \quad \Longleftrightarrow \quad \langle q_i, q_j \rangle = \delta_{ij}, \quad i, j = 1, \ldots, n.$$

Die Abbildung zeigt die Idee dieser Konstruktion.



Unter Verwendung dieser Notation besitzt (2.21) die Form

$$\begin{pmatrix} a_1 & a_2 & \ldots & a_n \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & \ldots & q_n \end{pmatrix} \begin{pmatrix} r_{11} & \ldots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix}.$$

Folglich lautet die Gleichung für die i-te Spalte

$$a_i = \sum_{j=1}^{i} r_{ji} q_j. \tag{2.22}$$

Angenommen, die orthogonalen Vektoren $q_1, \ldots, q_{i-1}$ sind schon bekannt. Wir zeigen, dass dann r_{ji}, $j = 1, \ldots, i$ und q_i berechnet werden können und erhalten somit einen Algorithmus, der uns die QR-Zerlegung einer gegebenen Matrix liefert.

Die Multiplikation von (2.22) mit q_ν^T, $\nu = 1, \ldots, i-1$ von links liefert

$$\langle a_i, q_\nu \rangle = \sum_{j=1}^{i} r_{ji} \langle q_j, q_\nu \rangle = r_{\nu i}.$$

Somit habe wir $r_{\nu i} = \langle a_i, q_\nu \rangle$ für $\nu = 1, \ldots, i-1$ bestimmt. Aus (2.22) folgt

$$a_i = \sum_{j=1}^{i-1} r_{ji} q_j + r_{ii} q_i$$

und somit

$$r_{ii} q_i = a_i - \sum_{j=1}^{i-1} r_{ji} q_j =: z_i. \tag{2.23}$$

Beachte, dass z_i berechnet werden kann, da a_i, r_{ij} und q_j für $j = 1, \ldots, i-1$ schon bestimmt wurden.

Wegen $\|q_i\|_2 = \langle q_i, q_i \rangle = 1$ liefert schließlich die Setzung

$$\begin{aligned} r_{ii} &= \|z_i\|_2, \\ q_i &= \frac{1}{r_{ii}} z_i \end{aligned}$$

die gesuchte QR-Zerlegung.

Insgesamt erhalten wir den Algorithmus des Gram-Schmidtschen Orthogonalisierungsverfahrens, angegeben in einem *Pseudo-Code*:

$$\begin{array}{lll} i & = 1 \ldots n & \\ \left\lfloor \right. & j = 1 \ldots i-1 & \text{leer, falls } i = 1 \\ & \left\lfloor r_{ji} = \langle a_i, q_j \rangle \right. & \\ & z_i = a_i - \displaystyle\sum_{j=1}^{i-1} r_{ji} q_j & \\ & r_{ii} = \|z_i\|_2 & \\ & q_i = \dfrac{1}{r_{ii}} z_i & \left. \right\rfloor \end{array}$$

Der Algorithmus wird jetzt anhand eines 2×2 Beispiels von Hand ausgeführt. Gegeben sei die Matrix

$$A = \begin{pmatrix} 3 & 1 \\ 4 & 1 \end{pmatrix}, \quad \text{und somit} \quad a_1 = \begin{pmatrix} 3 \\ 4 \end{pmatrix},\ a_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

$i = 1$:

$$\begin{aligned} z_1 &= a_1 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \\ r_{11} &= \|z_1\|_2 = \left\| \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right\|_2 = 5, \\ q_1 &= \frac{1}{r_{11}} z_1 = \frac{1}{5} \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{3}{5} \\ \frac{4}{5} \end{pmatrix}. \end{aligned}$$

$i = 2$:

$$\begin{aligned} r_{12} &= \langle a_2, q_1 \rangle = \left\langle \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{3}{5} \\ \frac{4}{5} \end{pmatrix} \right\rangle = \frac{7}{5}, \\ z_2 &= a_2 - r_{12} q_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{7}{5} \begin{pmatrix} \frac{3}{5} \\ \frac{4}{5} \end{pmatrix} = \begin{pmatrix} \frac{4}{25} \\ -\frac{3}{25} \end{pmatrix}, \\ r_{22} &= \|z_2\|_2 = \left\| \begin{pmatrix} \frac{4}{25} \\ -\frac{3}{25} \end{pmatrix} \right\|_2 = \frac{1}{5}, \\ q_2 &= \frac{1}{r_{22}} z_2 = 5 \left\| \begin{pmatrix} \frac{4}{25} \\ -\frac{3}{25} \end{pmatrix} \right\|_2 = \begin{pmatrix} \frac{4}{5} \\ -\frac{3}{5} \end{pmatrix}. \end{aligned}$$

Also erhalten wir die Zerlegung

$$A = \begin{pmatrix} 3 & 1 \\ 4 & 1 \end{pmatrix} = QR = \begin{pmatrix} \frac{3}{5} & \frac{4}{5} \\ \frac{4}{5} & -\frac{3}{5} \end{pmatrix} \begin{pmatrix} 5 & \frac{7}{5} \\ 0 & \frac{1}{5} \end{pmatrix}.$$

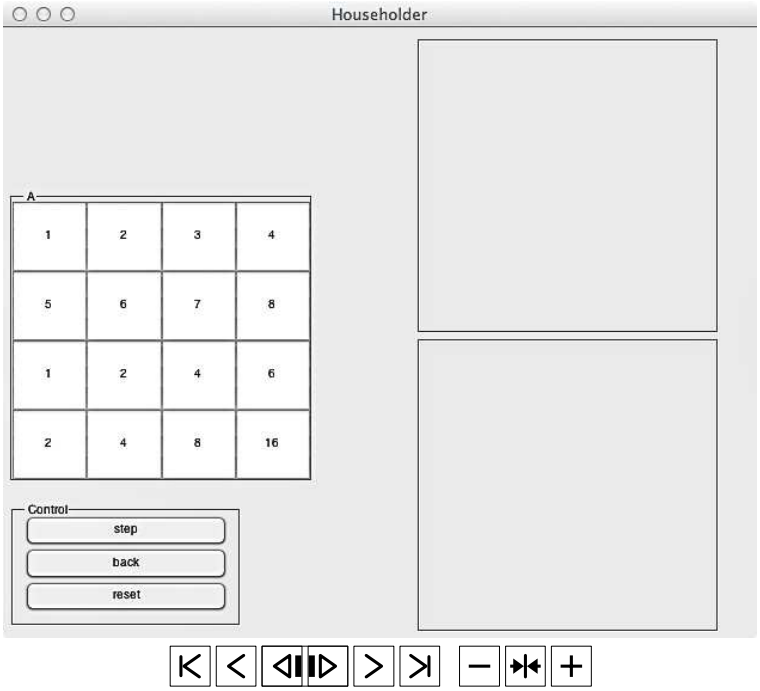
Dieser Algorithmus ist einfach zu programmieren, aber sehr anfällig gegen Rundungsfehler. Besser ist es beispielsweise, die Matrix A mit Hilfe von Spiegelungen nach Householder, die im nächsten Abschnitt beschrieben werden, zu transformieren.

2.14.3 Orthogonalisierungsverfahren nach Householder

Eine Alternative zu dem Gram-Schmidtschen Orthogonalisierungsverfahren stellt eine Methode dar, die von Alston Scott Householder (1904-1993) entwickelt wurde. Zu der gegebenen Matrix $A \in \mathbb{R}^{n,n}$ wird eine Folge von Spiegelungen $S_1, S_2, \ldots$ konstruiert, mit

$$\begin{aligned} S_1 A &= \begin{pmatrix} * & \ldots & \ldots & * \\ 0 & * & \ldots & * \\ \vdots & \vdots & \ddots & * \\ 0 & * & \ldots & * \end{pmatrix}, \\ S_2 S_1 A &= \begin{pmatrix} * & \ldots & \ldots & * \\ 0 & * & \ldots & * \\ \vdots & 0 & \ddots & * \\ 0 & 0 & \ldots & * \end{pmatrix}, \\ &\vdots \\ S_{n-1} \ldots S_2 S_1 A &= \begin{pmatrix} * & \ldots & \ldots & * \\ & * & \ldots & * \\ & & \ddots & * \\ & & & * \end{pmatrix} = R. \end{aligned}$$

Die Idee der QR-Zerlegung verdeutlicht die folgende Animation.



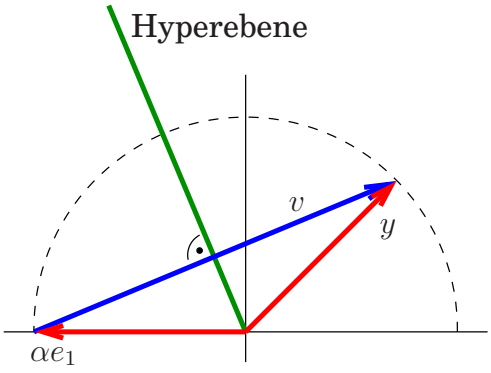
Wenn wir Spiegelungen $S_1, \ldots, S_{n-1}$ mit der obigen Eigenschaft gefunden haben (beachte $S_j = S_j^{-1}$, $j = 1, \ldots, n-1$), so erhalten wir die Zerlegung

$$A = QR \quad \text{mit } Q = S_1 \ldots S_{n-1}.$$

Zunächst konstruieren wir eine Spiegelung S an einer Hyperebene, die einen vorgegebenen Vektor $y \in \mathbb{R}^n$ auf

$$\alpha e_1 = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

abbildet, siehe Grafik.



Da S orthogonal ist, muss

$$|\alpha| = \|\alpha e_1\|_2 = \|y\|_2$$

gelten. Der Anschauung nach müssen wir an einer Hyperebene spiegeln, die senkrecht auf

$$v = y - \alpha e_1 = \begin{pmatrix} y_1 - \alpha \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

steht.

Die Spiegelung

$$S = I - \beta v v^T, \quad \beta = \frac{2}{\|v\|_2^2}$$

besitzt die gewünschte Eigenschaft. Wähle α gemäß

$$\alpha = -\text{sign}(y_1)\|y\|_2,$$

dann erhalten wir

$$v = \begin{pmatrix} y_1 + \text{sign}(y_1)\|y\|_2 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \text{sign}(y_1)\big(|y_1| + \|y\|_2\big) \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Folglich gilt

$$\|v\|_2^2 = \langle v, v\rangle = \big(|y_1| + \|y\|_2\big)^2 + y_2^2 + \cdots + y_n^2 = 2\|y\|_2^2 + 2|y_1|\|y\|_2,$$

und somit

$$\beta = \frac{2}{\|v\|_2^2} = \frac{1}{\|y\|_2\big(|y_1| + \|y\|_2\big)}.$$

Auf die Matrix $A = \begin{pmatrix} a_1 & \dots & a_n \end{pmatrix}$ wird diese Spiegelung jetzt mit der Setzung $y = a_1$ angewandt; es folgt

$$SA = \begin{pmatrix} \alpha & & & \\ 0 & & & \\ \vdots & b_2 & \dots & b_n \\ 0 & & & \end{pmatrix},$$

wobei

$$b_i = Sa_i = (I - \beta v v^T)a_i = a_i - \beta v \langle v, a_i\rangle, \quad \text{für } i \geq 2. \tag{2.24}$$

Nun bearbeitet man den „kürzeren“ Vektor

$$\tilde{y} \in \mathbb{R}^{n-1} \quad \text{in} \quad b_2 = \begin{pmatrix} b_{12} \\ -- \\ \tilde{y} \end{pmatrix}$$

mit derselben Methode. Es wird wie oben eine Spiegelung

$$\tilde{S} = I - \tilde{\beta}\tilde{v}\tilde{v}^T$$

konstruiert und man erhält

$$\begin{pmatrix} 1 & 0 \\ 0 & \tilde{S} \end{pmatrix} SA = \begin{pmatrix} \alpha & b_{12} & * & \dots & * \\ 0 & \tilde{\alpha} & \vdots & & \vdots \\ \vdots & 0 & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & * & \dots & * \end{pmatrix}.$$

Also haben wir den folgenden Algorithmus, geschrieben in *Pseudo-Code* hergeleitet.

$$\begin{aligned}
k &= 1 \dots n-1 \\
&\left\lfloor \begin{aligned}
\sigma &= \left(\sum_{i=k}^{n} a_{ik}^2 \right)^{\frac{1}{2}} && \text{(entspricht } \|y\|_2\text{)} \\
\alpha_k &= -\operatorname{sign}(a_{kk})\sigma \\
\beta_k &= \frac{1}{\sigma(|a_{kk}| + \sigma)} \\
a_{kk} &= a_{kk} - \alpha_k \\
j &= k+1 \dots n \\
&\left\lfloor \begin{aligned}
\rho &= \beta_k \sum_{i=k}^{n} a_{ik} a_{ij} && \text{(vgl. (2.24))} \\
i &= k \dots n \\
&\lfloor a_{ij} = a_{ij} - \rho a_{ik}
\end{aligned} \right.
\end{aligned} \right.
\end{aligned}$$

Beachte, dass die Vektoren v selbst wieder in der Matrix A abgespeichert werden. Nach dem Ausführen des Algorithmus ist

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \vdots \\ & & a_{nn} \end{pmatrix}$$

die gesuchte rechte obere Dreiecksmatrix R und die Informationen über die Spiegelungen sind in

$$\begin{pmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \dots & a_{n\,n-1} & 0 \end{pmatrix} \quad \text{und} \quad \beta_1, \dots \beta_{n-1}$$

enthalten.

Der SCILAB-Befehl `qr` berechnet die QR-Zerlegung unter Verwendung des Orthogonalisierungsverfahrens nach Householder.

```
-->A=[1 1 1 1; 1 2 4 8; 1 3 9 27; 1 4 16 64]
 A  =
    1.    1.    1.    1.
    1.    2.    4.    8.
    1.    3.    9.    27.
    1.    4.    16.   64.

-->[Q,R] = qr(A)
 R  =
  - 2.  - 5.        - 15.       - 50.
    0.  - 2.236068  - 11.18034  - 46.510214
    0.    0.          2.          15.
    0.    0.          0.        - 1.3416408

 Q  =
  - 0.5     0.6708204     0.5     0.2236068
  - 0.5     0.2236068   - 0.5   - 0.6708204
  - 0.5   - 0.2236068   - 0.5     0.6708204
  - 0.5   - 0.6708204     0.5   - 0.2236068

-->Q*Q' =
    1.            3.431E-16   - 4.516E-17     1.832E-16
    3.431E-16     1.            5.267E-17   - 7.789E-17
  - 4.516E-17     5.267E-17     1.          - 1.645E-16
    1.832E-16   - 7.789E-17   - 1.645E-16     1.
```

2.15 Projektionen

Es wird zum Abschluss dieses Kapitels der Begriff der Projektion eingeführt. Projektionen werden immer dann benötigt, wenn ein Objekt,

auf eine Ebene von geringerer Dimension abgebildet werden soll. Beispielsweise können auf einem zwei-dimensionalen Bildschirm keine dreidimensionalen Objekte dargestellt werden, sondern ausschließlich zweidimensionale Projektionen dieser Objekte.

Formal nennt man eine lineare Abbildung P einen **Projektor**, falls

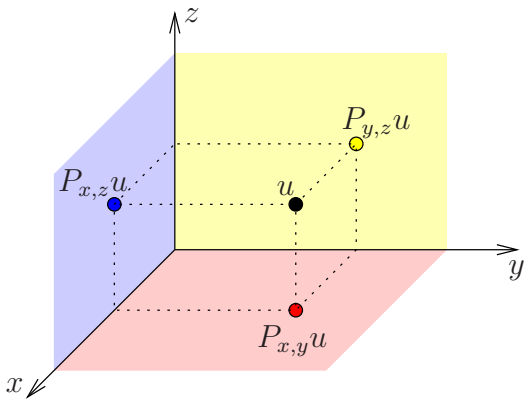
$$PP = P$$

gilt. Diese Definition entspricht der Anschauung: ein Punkt u wird durch Pu in die Projektionsebene projiziert. Durch nochmaliges Projizieren, d. h. durch Berechnung von PPu $(= Pu)$, bleibt der Punkt dann in der Projektionsebene.

Projektoren in die (x, y)- , (x, z)- bzw. (y, z)-Ebene werden durch die Matrizen

$$P_{x,y} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad P_{x,z} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad P_{y,z} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

definiert. Diese Projektoren veranschaulicht die Abbildung.



Hierbei ist zu beachten, dass ein Projektor P nur im Trivialfall $P = I$ eine orthogonale Abbildung ist.

Ein Projektor P im $\mathbb{R}^n$ wird eindeutig durch die Angabe von Bild und Kern bestimmt, wobei

$$\text{bild}(P) = \{Px : x \in \mathbb{R}^n\}, \quad \text{kern}(P) = \{x \in \mathbb{R}^n : Px = 0\}.$$

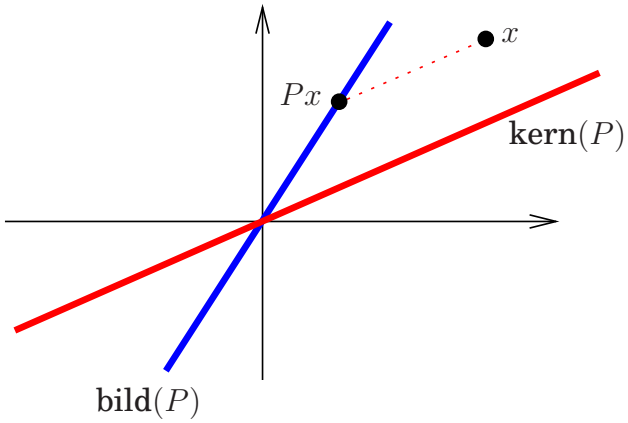
Der Projektor P bildet somit jeden Punkt $x \in \mathbb{R}^n$ – entlang seines Kerns – in sein Bild ab. Sei

$$x = a + b, \quad a \in \text{bild}(P), \quad b \in \text{kern}(P).$$

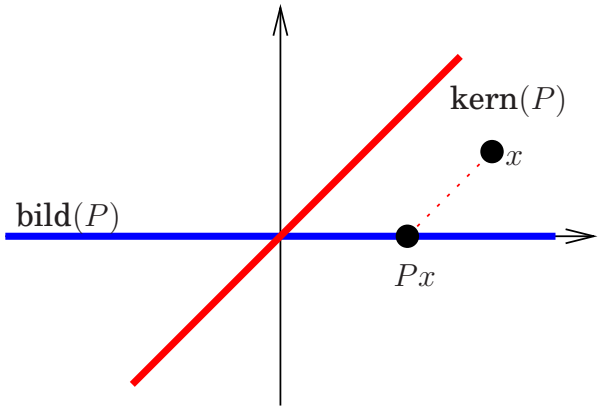
Dann gilt

$$Px = P(a + b) = Pa + Pb = Pa = a.$$

Diese Überlegung veranschaulicht auch die folgende Abbildung.



Der Projektor, der einen Punkt im $\mathbb{R}^2$ entlang der ersten Winkelhalbierenden auf die x-Achse abbildet (vgl. Abbildung), kann wie folgt konstruiert werden.



Wir bestimmen a, b, c, d so, dass $P = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ ein Projektor mit den gewünschten Eigenschaften ist. Wähle zunächst jeweils einen Vektor im Bild und im Kern von P:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \text{bild}(P), \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \text{kern}(P).$$

Löse dann das lineare Gleichungssystem

$$P\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad P\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Die Lösung $P = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$ ist der gesuchte Projektor.

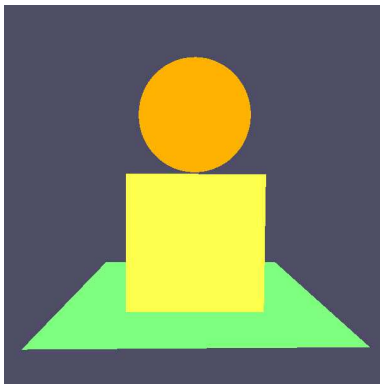
Kapitel 3

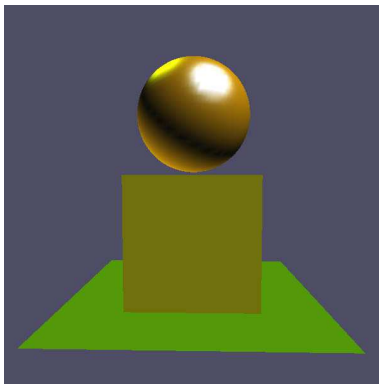
Analysis mit Anwendungen in der Computergrafik

3.1 Motivation: Tangential- und Normalenvektoren

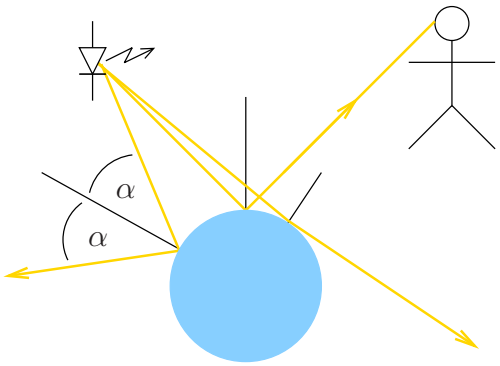
Nachdem wir in den letzten beiden Kapiteln gesehen haben, wie elementare Operationen, wie Drehungen, Streckungen, Rotationen durchgeführt werden – wir also jetzt Objekte im Raum bewegen und verändern können – betrachten wir in diesem Abschnitt die Berechnung von Reflexionen. Ohne die Darstellung von Reflexionen kann man beispielsweise eine Kugel von einer flachen Scheibe nicht unterscheiden.

Zur Illustration wird in der linken, mit der OpenGL-Bibliothek erstellten Animation, das Beleuchtungsmodell ausgeschaltet und in der rechten Animation eingeschaltet.





Zur Berechnung von Reflexionen ist die Bestimmung von Normalenvektoren an die jeweilige Oberfläche entscheidend, wie die folgende Abbildung verdeutlicht.

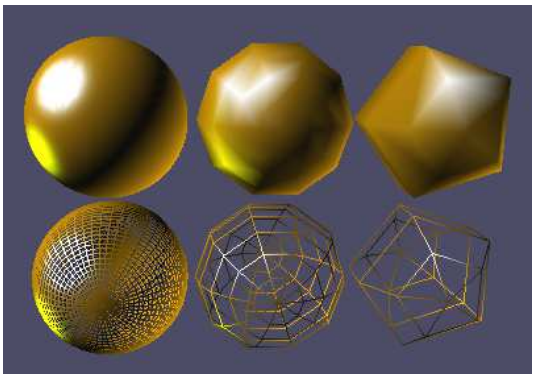


Der Eintrittswinkel des Lichts entspricht also dem Austrittswinkel, gemessen am Normalenvektor. Es darf somit nur an den Stellen der Oberfläche eine Reflexion eingezeichnet werden, an denen der Lichtstrahl wieder auf das Auge des Betrachters trifft.

Bei der Berechnung von Normalenvektoren werden in diesem Abschnitt die beiden Fälle untersucht:

- Berechnung eines Normalenvektors an eine analytisch vorgegebene Oberfläche.
- Bestimmung von Normalenvektoren an Oberflächen, die stückweise durch Polygone gebildet werden.

In der Computergrafik werden Oberflächen i. Allg. durch Polygone approximiert, siehe Abbildung. Zur Berechnung der Normalenvektoren können – wenn keine weiteren Kenntnisse bezüglich der Gestalt der Oberfläche vorliegen – auch nur diese Polygone verwendet werden.



Liegen aber, wie im Fall einer Kugel, weitere Informationen vor, so erleichtert dieses die Berechnung. Sei x ein Punkt auf der Oberfläche der Kugel, die den Mittelpunkt m besitzt. Der Normalenvektor in diesem Punkt zeigt in die Richtung der Geraden, die durch die Punkte x und m verläuft.

3.2 Ableitung einer Funktion $F : \mathbb{R} \to \mathbb{R}^n$

Bevor die Berechnung von Normalenvektoren im Detail analysiert wird, benötigen wir einige Grundkenntnisse aus der Analysis.

Sei

$$\begin{aligned} F : \mathbb{R} &\to \mathbb{R}^n \\ u &\mapsto F(u) = \begin{pmatrix} F_1(u) \\ \vdots \\ F_n(u) \end{pmatrix}. \end{aligned}$$

Hierbei sind die Komponenten-Abbildungen $F_i : \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, n$ wieder skalare Abbildungen. Sind diese Komponenten-Abbildungen stetig bzw. differenzierbar, so ist auch F stetig bzw. differenzierbar und es gilt im letzten Fall

$$F'(u) = \begin{pmatrix} F_1'(u) \\ \vdots \\ F_n'(u) \end{pmatrix}.$$

Beispiel 3.1

(a) Sei

$$F(u) = \begin{pmatrix} u \\ u^2 \end{pmatrix}, \quad F_1(u) = u, \quad F_2(x) = u^2.$$

Dann gilt

$$F'(u) = \begin{pmatrix} 1 \\ 2u \end{pmatrix}.$$

(b) Als zweites Beispiel betrachten wir die Abbildung

$$F(u) = \begin{pmatrix} \cos(u) \\ \sin(u) \end{pmatrix}, \quad F_1(u) = \cos(u), \quad F_2(u) = \sin(u)$$

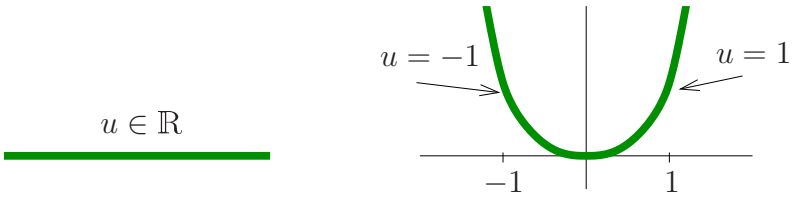
deren Ableitung durch

$$F'(u) = \begin{pmatrix} -\sin(u) \\ \cos(u) \end{pmatrix}$$

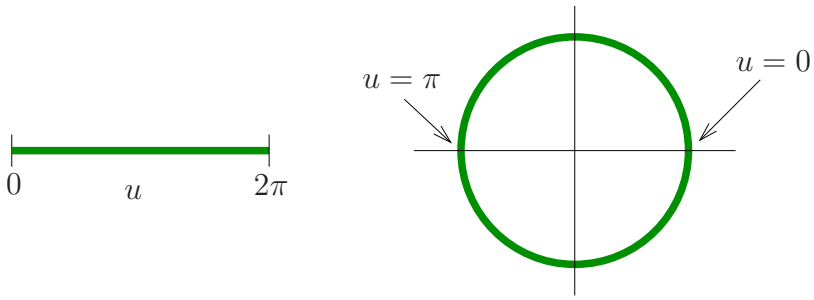
gegeben ist.

3.3 Kurven im $\mathbb{R}^n$

Wir betrachten zunächst Beispiel 3.1 genauer. Durchläuft u im Fall (a) die reellen Zahlen, so durchläuft $F(u)$ die Normalparabel, also eine Kurve im $\mathbb{R}^2$, die die folgende Abbildung zeigt.



Durchläuft u das Intervall $[0, 2\pi]$ in Beispiel 3.1 (b), so durchlaufen die Punkte $F(u) = \begin{pmatrix} F_1(u) \\ F_2(u) \end{pmatrix} = \begin{pmatrix} \cos(u) \\ \sin(u) \end{pmatrix}$ den Einheitskreis.



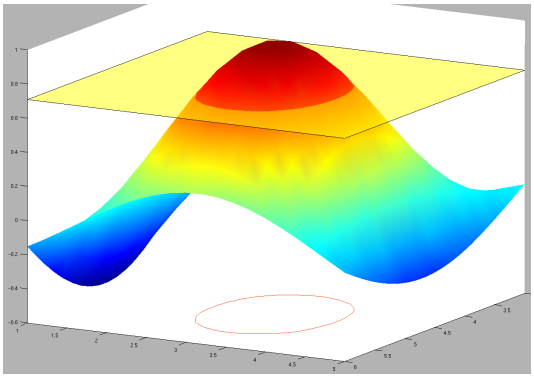
Formal versteht man unter einer **Kurve** im $\mathbb{R}^n$ eine stetige Abbildung

$$F : I \to \mathbb{R}^n,$$

wobei $I \subset \mathbb{R}$ ein Intervall bezeichnet. Es ist auch der Fall $I = \mathbb{R}$ zugelassen.

3.3.1 Kurven als Niveaulinien

In Abschnitt 3.3 haben wird Kurven explizit, durch Angabe einer Funktion F definiert. Kurven können aber auch implizit gegeben sein, z. B. als Niveau- oder Höhenlinien eines Berges.



Die Oberfläche eines Berges kann mathematisch mit einer Funktion $F : \mathbb{R}^2 \to \mathbb{R}$ beschrieben werden, die jedem Punkt auf der Erde (im $\mathbb{R}^2$) die

entsprechende Höheninformation zuordnet. In der obigen Abbildung ist der Graph der Funktion

$$F\begin{pmatrix} x \\ y \end{pmatrix} = \sin(x)\cos(y), \quad x \in [3,6], \quad y \in [1,5]$$

gezeichnet. Die eingezeichnete Niveaulinie ergibt sich als Lösung der Gleichung

$$F\begin{pmatrix} x \\ y \end{pmatrix} = 0.7.$$

Kurven, die durch Niveaulinien gegeben sind werden auch in Abschnitt 3.5.3 untersucht.

3.4 Partielle Ableitungen

Gegeben sei die Funktion

$$\begin{array}{rcl} F : \mathbb{R}^n & \to & \mathbb{R} \\ x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} & \mapsto & F(x), \end{array}$$

deren Graph im Fall $n = 2$ die Oberfläche eines Berges beschreiben kann, wie wir in Abschnitt 3.3.1 gesehen haben.

Im Folgenden soll die Differenzierbarkeit einer Funktion untersucht werden, die von mehreren Variablen $x_1, \ldots, x_n$ abhängt. Diese Fragestellung führt auf den Begriff der partiellen Ableitung. Wir wählen eine Variable aus und differenzieren nach dieser, wobei alle anderen Variablen festgehalten werden.

Definition 3.2 Sei $U \subset \mathbb{R}^n$ offen.

- Die Abbildung $F : \mathbb{R}^n \to \mathbb{R}$ heißt im Punkt $x \in U$ **partiell differenzierbar bezüglich der i-ten Koordinatenrichtung**, falls der Limes
$$\frac{\partial}{\partial x_i} F(x) := \lim_{h \to 0, h \neq 0} \frac{F(x + he^i) - F(x)}{h}$$
existiert.

- Die Abbildung F heißt im Punkt $x \in U$ **partiell differenzierbar**, falls sie für alle $i = 1, \ldots, n$ partiell differenzierbar bezüglich der i-ten Koordinatenrichtung ist.

Im Beispiel

$$\begin{aligned} F : \mathbb{R}^3 &\to \mathbb{R}, \\ F(x) &= x_1^2 + x_2 \sin(2x_3) \end{aligned}$$

erhalten wir:

$$\begin{aligned} \frac{\partial}{\partial x_1} F(x) &= 2x_1, \\ \frac{\partial}{\partial x_2} F(x) &= \sin(2x_3), \\ \frac{\partial}{\partial x_3} F(x) &= 2x_2 \cos(2x_3). \end{aligned}$$

Bemerkung 3.3 *Im Fall $n = 1$ erhalten wir für die Ableitung die äquivalenten Darstellungen*

$$F'(x) = \frac{\partial}{\partial x} F(x) = \frac{\partial F}{\partial x}(x).$$

Schließlich können wir auch die partiellen Ableitungen einer Funktion

$$\begin{aligned} F : \mathbb{R}^n &\to \mathbb{R}^m \\ x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} &\mapsto F(x) = \begin{pmatrix} F_1(x) \\ \vdots \\ F_m(x) \end{pmatrix} \end{aligned}$$

mit partiell differenzierbaren Funktionen

$$F_j : \mathbb{R}^n \to \mathbb{R}, \quad j = 1, \ldots, m$$

definieren:

$$\frac{\partial}{\partial x_i} F(x) := \begin{pmatrix} \frac{\partial}{\partial x_i} F_1(x) \\ \vdots \\ \frac{\partial}{\partial x_i} F_m(x) \end{pmatrix}, \quad i = 1, \ldots, n.$$

Im Beispiel $n = 2$, $m = 3$,

$$F \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2x_1 \cdot \mathrm{e}^{3x_2} \\ x_1^2 \cdot x_2 \\ x_2^3 \end{pmatrix}$$

gilt:

$$\frac{\partial}{\partial x_1} F(x) = \begin{pmatrix} 2\mathrm{e}^{3x_2} \\ 2x_1 \cdot x_2 \\ 0 \end{pmatrix}, \qquad \frac{\partial}{\partial x_2} F(x) = \begin{pmatrix} 6x_1 \cdot \mathrm{e}^{3x_2} \\ x_1^2 \\ 3x_2^2 \end{pmatrix}.$$

3.4.1 Der Gradient

Definition 3.4 Der **Gradient** einer partiell differenzierbaren Funktion $F : \mathbb{R}^n \to \mathbb{R}$ wird definiert durch:

$$\nabla F(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} F(x) \\ \vdots \\ \frac{\partial}{\partial x_n} F(x) \end{pmatrix}.$$

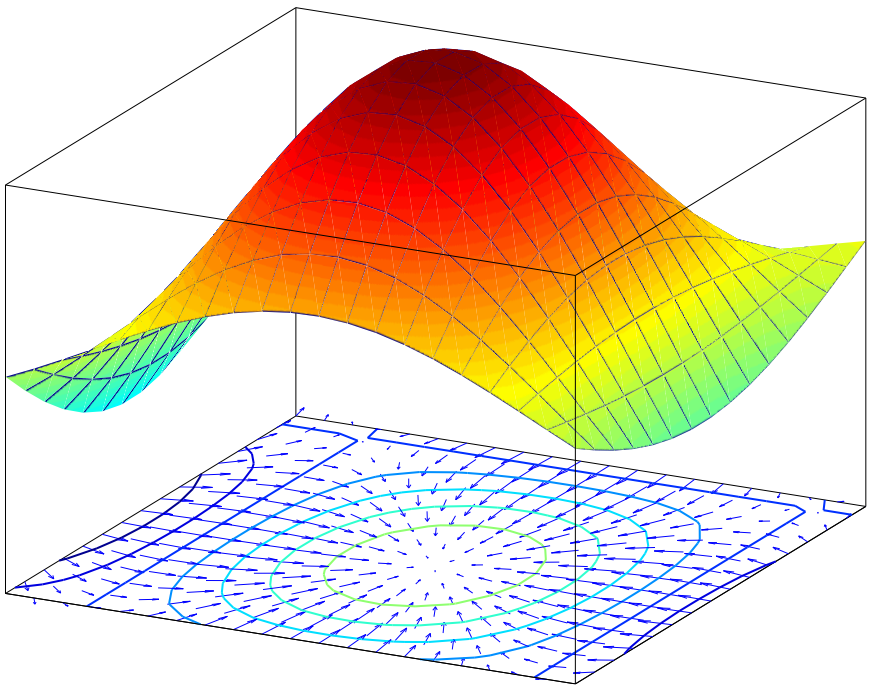
Wir betrachten erneut das Beispiel aus Abschnitt 3.3.1; es gilt

$$F(x) = \sin(x_1)\cos(x_2), \quad \nabla F(x) = \begin{pmatrix} \cos(x_1)\cos(x_2) \\ -\sin(x_1)\sin(x_2) \end{pmatrix}.$$

Die Abbildung illustriert die Berechnung. In der Grundebene sind ausgewählte Niveaulinien und Gradienten eingezeichnet.

Man erkennt,

- dass der Gradient ein Vektor ist, der in die Richtung des steilsten Anstiegs zeigt,
- dass der Gradient senkrecht auf den Niveaulinien steht.



3.5 Tangential- und Normalenvektoren an analytische Oberflächen

3.5.1 Explizite Darstellung im $\mathbb{R}^2$

Wir betrachten zunächst den Fall einer Kurve, die durch die Abbildung

$$F : \mathbb{R} \to \mathbb{R}^2,$$
$$F(x) := \begin{pmatrix} x \\ f(x) \end{pmatrix} \tag{3.1}$$

mit einer stetig differenzierbaren Funktion $f : \mathbb{R} \to \mathbb{R}$ definiert wird. Ein **Tangentialvektor** an diese Kurve wird durch

$$\frac{\partial F}{\partial x}(x) = \begin{pmatrix} 1 \\ f'(x) \end{pmatrix}$$

bestimmt. Ein **Normalenvektor** in diesem Punkt

$$n(x) = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}(x)$$

ist dadurch charakterisiert, dass er senkrecht auf dem Tangentialvektor steht, also die Bedingung

$$\left\langle n(x), \frac{\partial F}{\partial x}(x) \right\rangle = 0$$

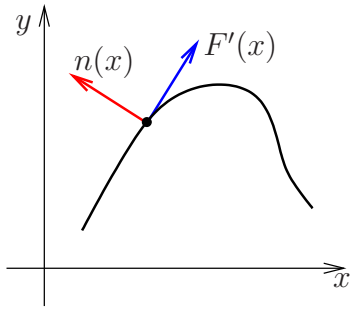
erfüllt. Somit erhalten wir die Beziehung

$$n_1(x) \cdot 1 + n_2(x) f'(x) = 0 \quad \Leftrightarrow \quad n_1(x) = -n_2(x) f'(x)$$

und folglich den Normalenvektor

$$n(x) = \begin{pmatrix} -f'(x) \\ 1 \end{pmatrix},$$

siehe Abbildung.



Tangential- und Normalenvektoren sind nicht eindeutig. Offensichtlich liefert

$$\lambda \frac{\partial F}{\partial x}(x) \quad \text{bzw.} \quad \lambda n(x) \quad \text{für } \lambda \neq 0$$

wieder einen Tangential- bzw. Normalenvektor.

Ein besonders ausgezeichneter Normalenvektor ist der sogenannte **Einheitsnormalenvektor** $\frac{n(x)}{\|n(x)\|_2}$, der auf die euklidische Länge eins normiert ist.

Diese Überlegungen illustrieren wir am Beispiel der Normalparabel. Sei

$$F(x) := \begin{pmatrix} x \\ x^2 \end{pmatrix},$$

dann gilt

$$\frac{\partial F}{\partial x}(x) = \begin{pmatrix} 1 \\ 2x \end{pmatrix}, \quad n(x) = \begin{pmatrix} -2x \\ 1 \end{pmatrix}$$

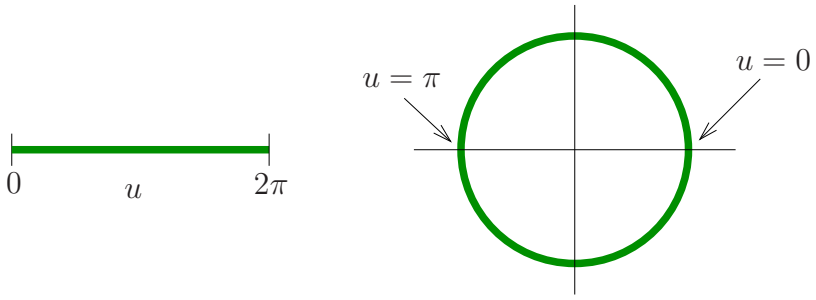
und der Einheitsnormalenvektor ist

$$\bar{n}(x) = \frac{1}{\sqrt{4x^2+1}} \begin{pmatrix} -2x \\ 1 \end{pmatrix}.$$

Allerdings kann nicht jede Kurve im $\mathbb{R}^2$ mittels (3.1) parametrisiert werden. Als Gegenbeispiel betrachten wir den Einheitskreis, der durch

$$F(u) := \begin{pmatrix} \cos(u) \\ \sin(u) \end{pmatrix}$$

definiert wird, siehe Abbildung.



Deshalb verallgemeinern wir die Darstellung (3.1) auf den Fall

$$F(u) := \begin{pmatrix} g(u) \\ f(u) \end{pmatrix}$$

mit $g, f : \mathbb{R} \to \mathbb{R}$ stetig differenzierbar. Einen Tangentialvektor im Punkt $F(u)$ liefert die Ableitung

$$\frac{\partial F}{\partial u}(u) = \begin{pmatrix} g'(u) \\ f'(u) \end{pmatrix}.$$

Den Normalenvektor erhalten wir als Lösung der Gleichung

$$\left\langle n(u), \frac{\partial F}{\partial u}(u) \right\rangle = 0;$$

es gilt

$$g'(u)n_1(u) + f'(u)n_2(u) = 0$$

und somit

$$n(u) = \begin{pmatrix} f'(u) \\ -g'(u) \end{pmatrix}.$$

Im Beispiel:

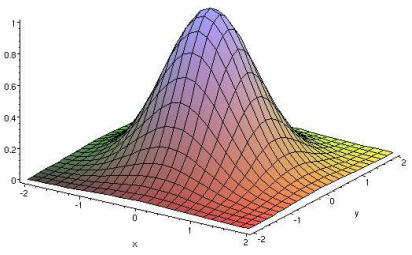
$$n(u) = \begin{pmatrix} \sin'(u) \\ -\cos'(u) \end{pmatrix} = \begin{pmatrix} \cos(u) \\ \sin(u) \end{pmatrix}.$$

3.5.2 Explizite Darstellung im $\mathbb{R}^3$

Als Nächstes wird eine Fläche untersucht, die die Darstellung

$$F(x, y) := \begin{pmatrix} x \\ y \\ f(x, y) \end{pmatrix}, \tag{3.2}$$

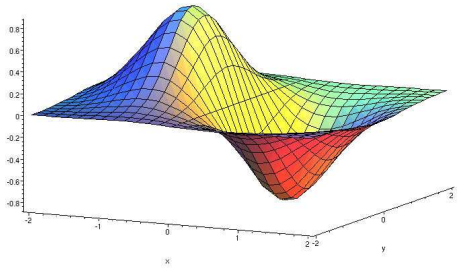
mit einer stetig differenzierbaren Funktion $f(x, y) : \mathbb{R}^2 \to \mathbb{R}$, besitzt. Als Beispiel betrachten wir die Funktion $f(x, y) = \mathrm{e}^{-x^2-y^2}$.



Zur Bestimmung eines Normalenvektors, werden zunächst Tangentialvektoren in x- bzw. in y-Richtung berechnet:

$$\frac{\partial F}{\partial x}(x, y) = \begin{pmatrix} 1 \\ 0 \\ \frac{\partial f}{\partial x}(x, y) \end{pmatrix}, \quad \frac{\partial F}{\partial y}(x, y) = \begin{pmatrix} 0 \\ 1 \\ \frac{\partial f}{\partial y}(x, y) \end{pmatrix}.$$

Die Ableitung $\frac{\partial f}{\partial x}(x, y) = -2x\mathrm{e}^{-x^2-y^2}$ zeigt die folgende Grafik:



Im obigen Beispiel liegt der Punkt $P = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, der Gipfel des Berges, auf der Fläche. Wir erhalten die partiellen Ableitungen

$$\begin{aligned} \frac{\partial F}{\partial x}(0,0) &= \begin{pmatrix} 1 \\ 0 \\ -2x\mathrm{e}^{-x^2-y^2}|_{(0,0)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \\ \frac{\partial F}{\partial y}(0,0) &= \begin{pmatrix} 0 \\ 1 \\ -2y\mathrm{e}^{-x^2-y^2}|_{(0,0)} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \end{aligned}$$

Durch diese beiden Vektoren wird die Tangentialebene im Punkt P aufgespannt. Somit müssen wir „nur“ noch einen Vektor n finden, der auf den beiden Tangentialvektoren $\frac{\partial F}{\partial x}$ und $\frac{\partial F}{\partial y}$ senkrecht steht. Diesen Vektor liefert das Vektorprodukt, definiert durch

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix} = \begin{pmatrix} \det \begin{pmatrix} a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \\ -\det \begin{pmatrix} a_1 & b_1 \\ a_3 & b_3 \end{pmatrix} \\ \det \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \end{pmatrix}.$$

Beachte (siehe Übungsaufgabe):

$$a \perp a \times b \perp b.$$

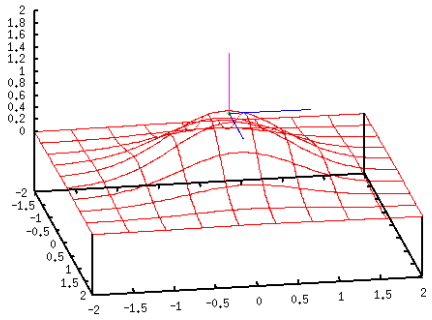
Somit folgt

$$n(x,y) = \left(\frac{\partial F}{\partial x} \times \frac{\partial F}{\partial y} \right)(x,y) = \begin{pmatrix} 1 \\ 0 \\ \frac{\partial f}{\partial x}(x,y) \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ \frac{\partial f}{\partial y}(x,y) \end{pmatrix} = \begin{pmatrix} -\frac{\partial f}{\partial x}(x,y) \\ -\frac{\partial f}{\partial y}(x,y) \\ 1 \end{pmatrix}.$$

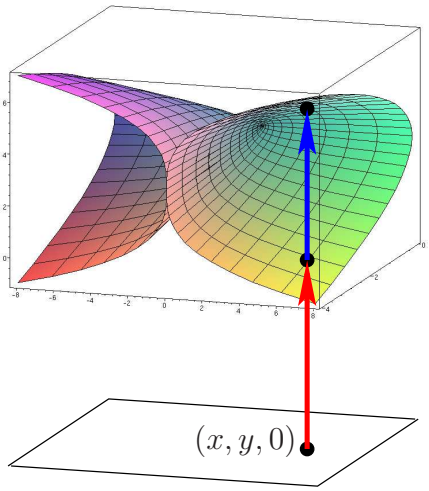
Im Beispiel gilt

$$n(x,y) = \begin{pmatrix} 2x\mathrm{e}^{-x^2-y^2} \\ 2y\mathrm{e}^{-x^2-y^2} \\ 1 \end{pmatrix}.$$

Für $x = y = 0$ erhalten wir somit den Normalenvektor $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, d. h. der Normalenvektor zeigt in Richtung der z-Achse.



Allerdings ist es nicht möglich, jede Fläche im $\mathbb{R}^3$ mit Hilfe der Darstellung (3.2) zu beschreiben. Da $f(\cdot, \cdot)$ eine Funktion ist, kann jedem Paar (x, y) nur ein Funktionswert zugeordnet werden. Somit kann eine Fläche, wie sie in der Abbildung angegeben ist, nicht mit dem obigen Ansatz beschrieben werden.



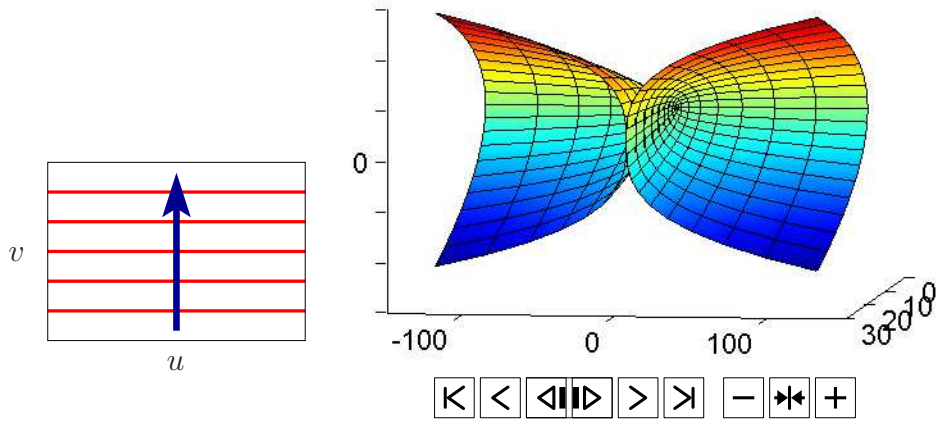
Aus diesem Grund müssen wir eine andere Parametrisierung wählen. Betrachte Oberflächen, die durch eine Funktion der Form

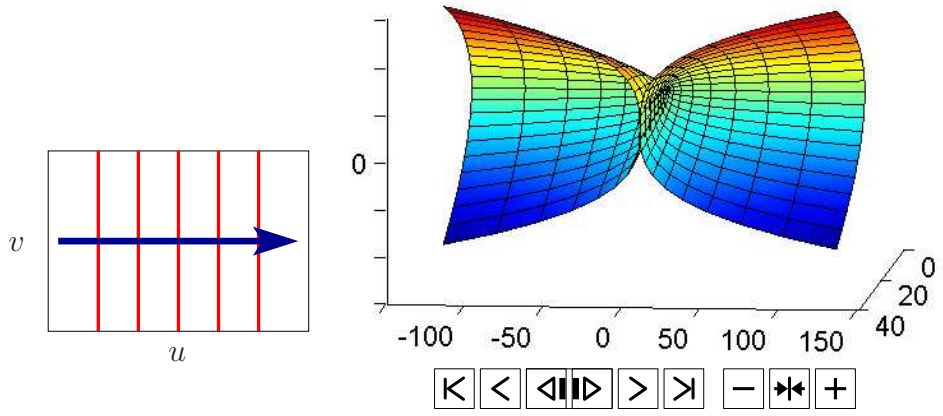
$$F(u, v) := \begin{pmatrix} X(u, v) \\ Y(u, v) \\ Z(u, v) \end{pmatrix}$$

gegeben sind, wobei $X, Y, Z : \mathbb{R}^2 \to \mathbb{R}$ stetig differenzierbare Funktionen sind. Als Beispiel wählen wir

$$F(u,v) = \begin{pmatrix} u^2 \\ v^3 \\ 3 - uv \end{pmatrix}.$$

Die beiden animierten Darstellungen verdeutlicht die Parametrisierung dieser Fläche.





Zur Bestimmung eines Normalenvektors werden zunächst Tangentialvektoren an diese Fläche in Richtung von u und v berechnet. Diese Tangentialvektoren erhält man durch Bestimmung der *partiellen Ableitungen*

$$\frac{\partial F}{\partial u} \quad \text{und} \quad \frac{\partial F}{\partial v},$$

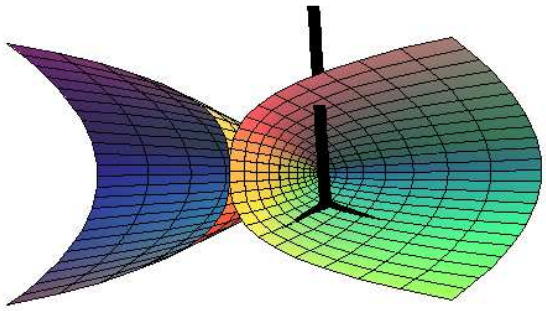
also der Ableitung von F bezüglich u bzw. v. Im obigen Beispiel erhalten wir

$$\frac{\partial F}{\partial u}(u,v) = \begin{pmatrix} 2u \\ 0 \\ -v \end{pmatrix} \quad \text{und} \quad \frac{\partial F}{\partial v}(u,v) = \begin{pmatrix} 0 \\ 3v^2 \\ -u \end{pmatrix}.$$

Für $u = 1$, $v = 1$ finden wir den Punkt $p = (1,1,2)^T$. Die obige Rechnung liefert in diesem Punkt die Tangentialvektoren

$$\frac{\partial F}{\partial u}(1,1) = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix} \quad \text{und} \quad \frac{\partial F}{\partial v}(1,1) = \begin{pmatrix} 0 \\ 3 \\ -1 \end{pmatrix}.$$

Diese Tangentialvektoren wurden mit MAPLE gezeichnet; zusätzlich ist auch der Normalenvektor angegeben.



Jeder Vektor, der senkrecht auf den beiden Tangentialvektoren $\frac{\partial F}{\partial u}$ und $\frac{\partial F}{\partial v}$ steht, ist ein Normalenvektor.

Den gesuchten Normalenvektor erhalten wir durch Berechnung von

$$\frac{\partial F}{\partial u} \times \frac{\partial F}{\partial v}.$$

Im Beispiel gilt:

$$\left(\frac{\partial F}{\partial u} \times \frac{\partial F}{\partial v}\right)(u,v) = \begin{pmatrix} 2u \\ 0 \\ -v \end{pmatrix} \times \begin{pmatrix} 0 \\ 3v^2 \\ -u \end{pmatrix} = \begin{pmatrix} 3v^3 \\ 2u^2 \\ 6uv^2 \end{pmatrix},$$

und man erhält den Normalenvektor

$$\left(\frac{\partial F}{\partial u} \times \frac{\partial F}{\partial v}\right)(1,1) = \begin{pmatrix} 3 \\ 2 \\ 6 \end{pmatrix}.$$

Schließlich ist es sinnvoll, den Einheitsnormalenvektor zu bestimmen. Beachte, dass ein Vektor a mittels $\frac{a}{\|a\|_2}$ auf die Länge eins normiert wird. Also erhalten wir

$$\left\| \begin{pmatrix} 3 \\ 2 \\ 6 \end{pmatrix} \right\|_2 = \left(3^2 + 2^2 + 6^2\right)^{\frac{1}{2}} = \sqrt{49} = 7,$$

und der Einheitsnormalenvektor ist

$$n = \begin{pmatrix} \frac{3}{7} \\ \frac{2}{7} \\ \frac{6}{7} \end{pmatrix}.$$

3.5.3 Implizite Darstellung im $\mathbb{R}^2$

Wird eine Kurve K im $\mathbb{R}^2$ durch die Nullstellenmenge einer stetig differenzierbaren Funktion $G : \mathbb{R}^2 \to \mathbb{R}$ definiert, d. h.

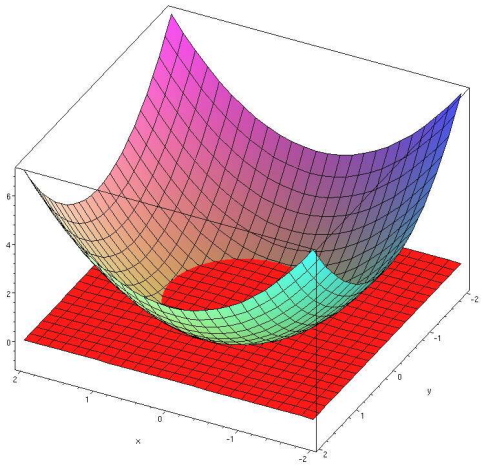
$$K = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} : G(x, y) = 0 \right\}, \tag{3.3}$$

so bezeichnet man (3.3) als **implizite Darstellung** der Kurve K.

Zunächst betrachten wir das Beispiel

$$G(x, y) := x^2 + y^2 - 1$$

und fragen, wie die Nullstellenmenge der Funktion G aussieht.



In diesem Beispiel kann y in Abhängigkeit von x beschrieben werden:

$$\begin{aligned} x^2 + y^2 - 1 &= 0 \\ \Rightarrow \quad y^2 &= -x^2 + 1 \\ \Rightarrow \quad y_\pm &= \pm\sqrt{1 - x^2}. \end{aligned} \tag{3.4}$$

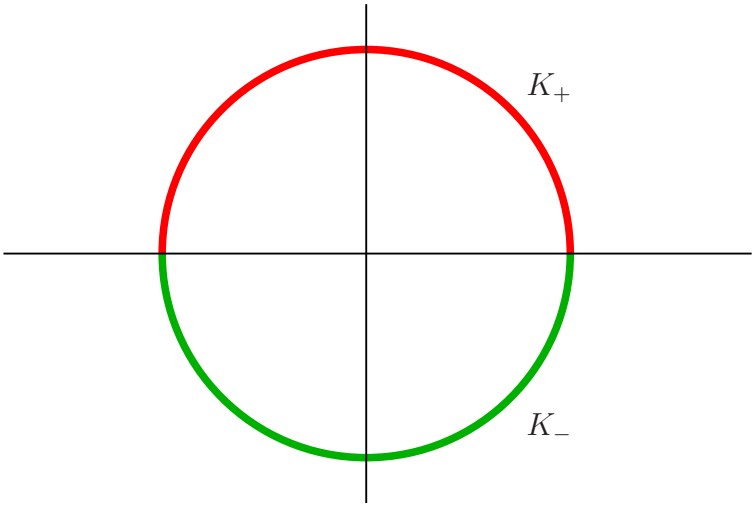
Allerdings ist (3.4) nur für $|x| \leq 1$ definiert. Außerdem kann y nicht eindeutig dargestellt werden, da sowohl der negative als auch der positive Zweig der Wurzel zu berücksichtigen sind.

In diesem Fall kann die Kurve somit mittels

$$F_{\pm}(x) := \begin{pmatrix} x \\ \pm\sqrt{1-x^2} \end{pmatrix},$$

wie in Abschnitt 3.5.1, explizit beschrieben werden. Wir erhalten die folgenden expliziten (Teil)-Darstellungen der Kurve K:

$$K_{\pm} = \{F_{\pm}(x) : x \in [-1, 1]\}.$$



Allgemein funktioniert dieser Ansatz, wenn man eine Darstellung der Form

$$y = h(x) \tag{3.5}$$

findet. In diesem Fall definieren wir

$$F(x) := \begin{pmatrix} x \\ h(x) \end{pmatrix}$$

und können einen Tangential- bzw. Normalenvektor, wie in Abschnitt 3.5.1, bestimmen.

Falls die Darstellung (3.5) nicht existiert, kann man den Normalenvektor trotzdem leicht berechnen. Der Gradient

$$n = \nabla G = \begin{pmatrix} \frac{\partial G}{\partial x} \\ \frac{\partial G}{\partial y} \end{pmatrix}$$

ist ein Normalenvektor. Dieser Vektor kann leicht berechnet werden, wohingegen es schwierig ist Punkte, die auf der Kurve liegen zu finden.

Im obigen Beispiel gilt

$$\nabla G(x,y) = \begin{pmatrix} 2x \\ 2y \end{pmatrix}.$$

3.5.4 Implizite Darstellung im $\mathbb{R}^3$

Sei $G : \mathbb{R}^3 \to \mathbb{R}$ stetig differenzierbar. Wird eine analytische Oberfläche durch die implizite Darstellung

$$M = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} : G(x,y,z) = 0 \right\} \tag{3.6}$$

beschrieben, ist die Berechnung von Normalenvektoren leicht, aber die Bestimmung von Punkten, die auf der Fläche liegen, ist schwierig.

Ein einfacher Fall liegt vor, wenn man (z. B. mit Hilfe des Satzes über implizite Funktionen) eine Darstellung der Form

$$z = h(x,y)$$

findet, also eine der Variablen (hier z) durch die Funktion h beschrieben werden kann, die nur von den beiden anderen Variablen (hier x, y) abhängt. Dann erhalten wir wieder die explizite Darstellung

$$F(x,y) := \begin{pmatrix} x \\ y \\ h(x,y) \end{pmatrix}$$

und verfahren wie in Abschnitt 3.5.2.

Zum Beispiel beschreibt die Menge

$$\left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} : x^2 + y^2 + z^2 - 1 = 0 \right\}$$

die Einheitskugel um den Ursprung mit dem Radius 1. Setze

$$G(x,y,z) = x^2 + y^2 + z^2 - 1. \tag{3.7}$$

Offensichtlich kann in diesem Fall nach z aufgelöst werden; es gilt

$$z = \pm\sqrt{1 - x^2 - y^2}, \quad \text{für } x^2 + y^2 \leq 1,$$

und Normalenvektoren können wie oben beschrieben, an Punkte auf der oberen bzw. unteren Halbkugel für die expliziten Teil-Darstellungen

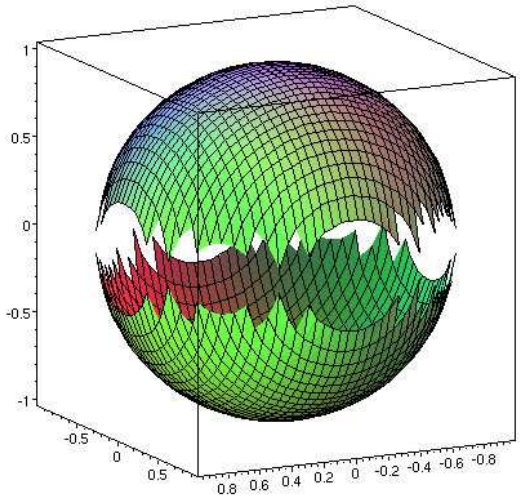
$$F_{\pm}(x,y) = \begin{pmatrix} x \\ y \\ \pm\sqrt{1-x^2-y^2} \end{pmatrix}, \quad x^2+y^2 \leq 1$$

berechnet werden.

Das MAPLE-Programm

```
plot3d({[x,y,sqrt(1-x^2-y^2)],[x,y,-sqrt(1-x^2-y^2)]},x=-1..1,y=-1..1,grid=[40,40]);
```

bei dem die Bedingung $x^2+y^2 \leq 1$ nicht explizit angegeben wird, liefert die folgende Illustration.



Ein Normalenvektor n an die implizit definierte Fläche (3.6) ist der Gradient

$$n = \nabla G = \begin{pmatrix} \frac{\partial G}{\partial x} \\ \frac{\partial G}{\partial y} \\ \frac{\partial G}{\partial z} \end{pmatrix}.$$

Die Schwierigkeit besteht somit nicht in der Berechnung des Gradienten, sondern darin, einen Punkt auf der durch G definierten Fläche zu finden.

Im Fall der Einheitskugel (3.7) gilt

$$\nabla G(x,y) = \begin{pmatrix} 2x \\ 2y \\ 2z \end{pmatrix},$$

und diese Rechnung beweist die anfangs aufgestellte Behauptung, dass der Normalenvektor immer in der Richtung der Geraden liegt, die durch den Mittelpunkt und den jeweiligen Punkt der Oberfläche definiert wird.

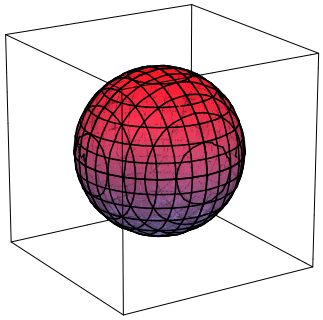
3.6 Kugeln, Ellipsoide, Hyperboloide und Sattelflächen

3.6.1 Kugeln

Wir habe bereits gesehen, dass die Einheitskugel im $\mathbb{R}^3$ (implizit) durch die Lösungsmenge der Gleichung

$$\|x\|_2^2 = x^T x = x_1^2 + x_2^2 + x_3^2 = 1 \tag{3.8}$$

definiert wird.



3.6.2 Ellipsoide

Durch Modifikation von (3.8) erhalten wir die implizite Darstellung eines **Ellipsoiden** mit den Halbachsen der Länge $a, b, c > 0$:

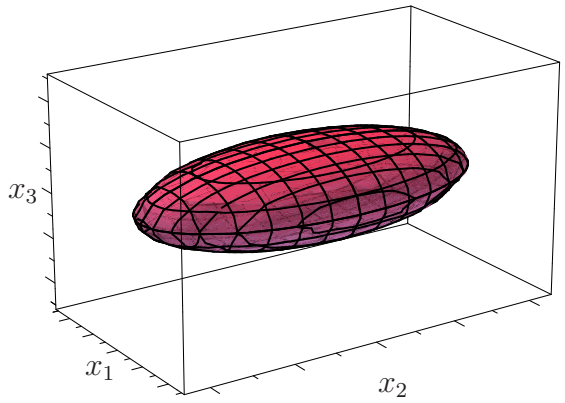
$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} + \frac{x_3^2}{c^2} = 1. \tag{3.9}$$

Äquivalent können wir diese Formel auch wie folgt darstellen:

$$x^T A x = 1, \quad \text{mit} \quad A = \begin{pmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{pmatrix}.$$

Man erkennt, dass die Wurzel aus den Inversen der Eigenwerte $\{\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{c^2}\}$ von A die Längen der Halbachsen angeben. Die Halbachsen selbst werden durch die Eigenvektoren $\{e^1, e^2, e^3\}$ von A festgelegt.

In der Abbildung ist der Fall $a = 1$, $b = 2$ und $c = \frac{1}{2}$ dargestellt.



Im Spezialfall $a = b = c = 1$ erhalten wir wieder die Einheitskugel aus Abschnitt 3.6.1.

Sei $A \in \mathbb{R}^{n,n}$ eine positiv definite Matrix, vgl. Abschnitt 2.8, dann wird durch die Lösungsmenge der Gleichung

$$x^T A x = 1$$

ein Ellipsoid in $\mathbb{R}^n$ definiert.

3.6.3 Hyperboloide

Hyperboloide erhalten wir durch eine weitere Modifikation von (3.9):

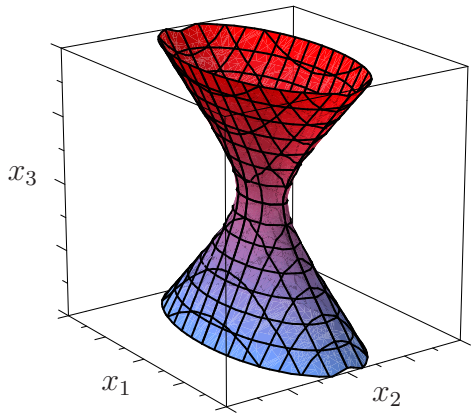
$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - \frac{x_3^2}{c^2} = 1. \tag{3.10}$$

Alternativ erhalten wir die Darstellung

$$x^T A x = 1, \quad \text{mit} \quad A = \begin{pmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & -\frac{1}{c^2} \end{pmatrix}.$$

Hierbei ist zu beachten, dass die Matrix A indefinit ist.

Die Abbildung zeigt den Fall $a = 1$, $b = \frac{1}{2}$ und $c = 1$.



3.6.4 Sattelflächen

Schließlich erhalten wir eine **Sattelfläche** als Lösungsmenge der Gleichung

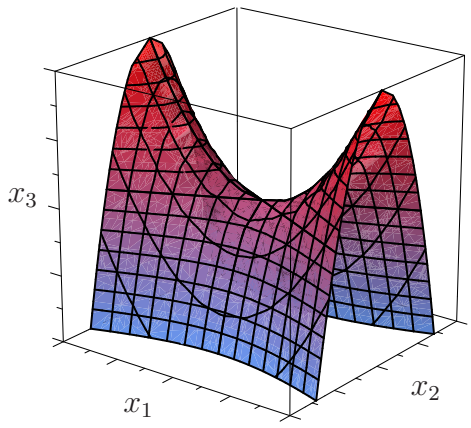
$$\frac{x_1^2}{a^2} - \frac{x_2^2}{b^2} - x_3 = 0. \tag{3.11}$$

Diese Lösungsmenge wird auch als hyperbolisches Paraboloid bezeichnet, das uns erneut in Abschnitt 7.2 begegnen wird. Alternativ erhalten wir die Darstellung

$$x_3 = \begin{pmatrix} x_1 & x_2 \end{pmatrix} A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \text{mit} \quad A = \begin{pmatrix} \frac{1}{a^2} & 0 \\ 0 & -\frac{1}{b^2} \end{pmatrix}.$$

Die Matrix A besitzt den positiven Eigenwert $\frac{1}{a^2}$ mit Eigenvektor $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und den negativen Eigenwert $-\frac{1}{b^2}$ mit Eigenvektor $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Genauer bestimmt a das Ansteigen des Sattels in Richtung der x_1-Achse, wogegen b das Abklingen in Richtung der x_2-Achse bestimmt.

Die Abbildung zeigt den Fall $a = 2$, $b = 1$.



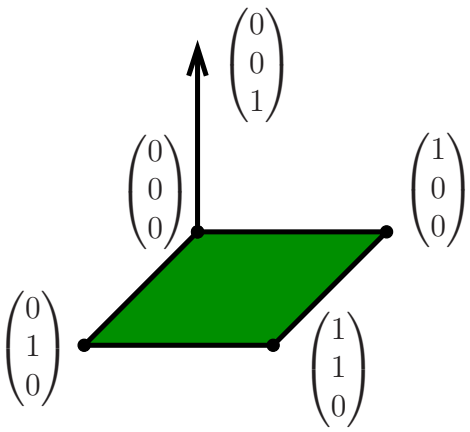
3.7 Normalenvektoren an Polygone

Wie bereits erwähnt, werden Oberflächen in der Computergrafik durch Polygone approximiert. An ein ebenes Polygon P kann der Normalenvektor mit Hilfe des Vektorprodukts berechnet werden. Seien a, b, c drei Punkte aus P, die nicht auf einer Geraden liegen. Dann liefert

$$(b - a) \times (c - a)$$

einen Vektor, der senkrecht auf P steht. Eine Normierung auf die Länge 1 liefert den gesuchten Normalenvektor.

Den Fall, dass P das Einheitsquadrat in der (x, y)-Ebene ist, zeigt die Abbildung.

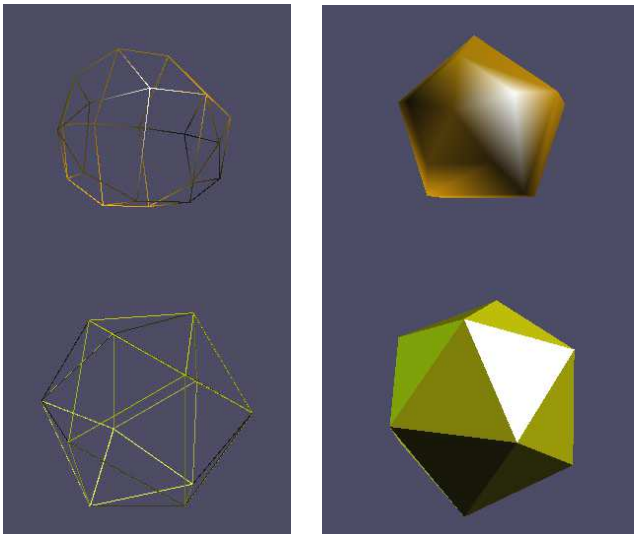


Werden die Normalenvektoren an alle Polygone, die eine Oberfläche approximieren mit diesem Ansatz berechnet, so erhält man keine glatte

Oberfläche, da sich die Richtung der Normalenvektoren an den Kanten sprungartig verändert. Allerdings ist dieser Effekt erwünscht, wenn ein Objekt mit scharfen Kanten dargestellt werden soll, z. B. ein Ikosaeder.

In der Abbildung ist eine grob durch Polygone approximierte Kugel und ein Ikosaeder dargestellt.

Zum Erhalt glatter Übergänge wird zwischen den Normalenvektoren benachbarter Segmente der Durchschnitt gebildet und die Länge dieses neuen Vektors wird wieder auf die Länge 1 normiert. Enthält ein Objekt sowohl „weiche“ und „scharfe“ Kanten, so darf diese Durchschnittsbildung nur an den „weichen“ Kanten durchgeführt werden.



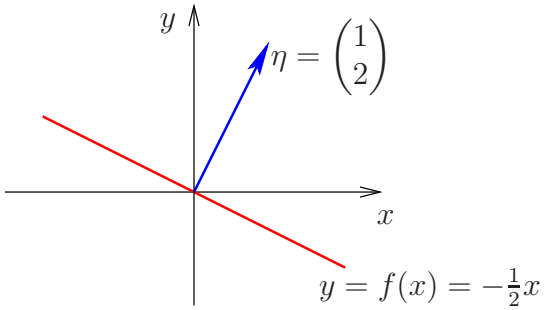
3.8 Transformation von Normalenvektoren

Gegeben sei die Gerade G und ein zugehöriger Normalenvektor η. Wird die Gerade G mit einer Matrix M transformiert, so liegt die **falsche** Vermutung nahe, dass $M\eta$ ein Normalenvektor der transformierten Ebene $M(G)$ ist. Zur Verdeutlichung wird ein Gegenbeispiel angegeben.

Betrachte die Gerade, definiert durch

$$f(x) = -\frac{1}{2}x.$$

Ein Normalenvektor wird durch $\eta = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ gegeben, siehe Abbildung.



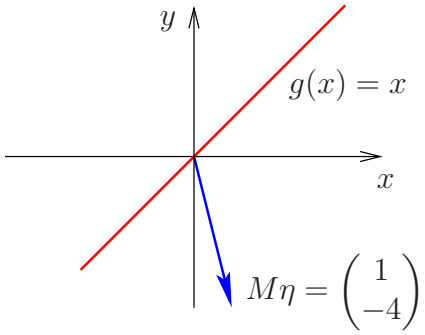
Werden jetzt alle Punkte des $\mathbb{R}^2$ mit der Matrix

$$M = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}$$

transformiert, so erhalten wir:

$$\begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} x \\ -\frac{1}{2}x \end{pmatrix} = \begin{pmatrix} x \\ x \end{pmatrix}, \quad M\eta = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ -4 \end{pmatrix},$$

also wird die Gerade auf die Winkelhalbierende g transformiert.



Offensichtlich haben wir durch Berechnen von $M\eta$ **keinen** Normalenvektor gefunden.

Sei jetzt z ein Vektor auf der Geraden G, die den Normalenvektor η besitze. Es gilt:

$$\langle \eta, z \rangle = \eta^T z = 0.$$

Wird z mit der invertierbaren Matrix M transformiert, so folgt

$$\left\langle (M^{-1})^T \eta, Mz \right\rangle = \left((M^{-1})^T \eta \right)^T Mz = \eta^T M^{-1} Mz = \eta^T z = 0,$$

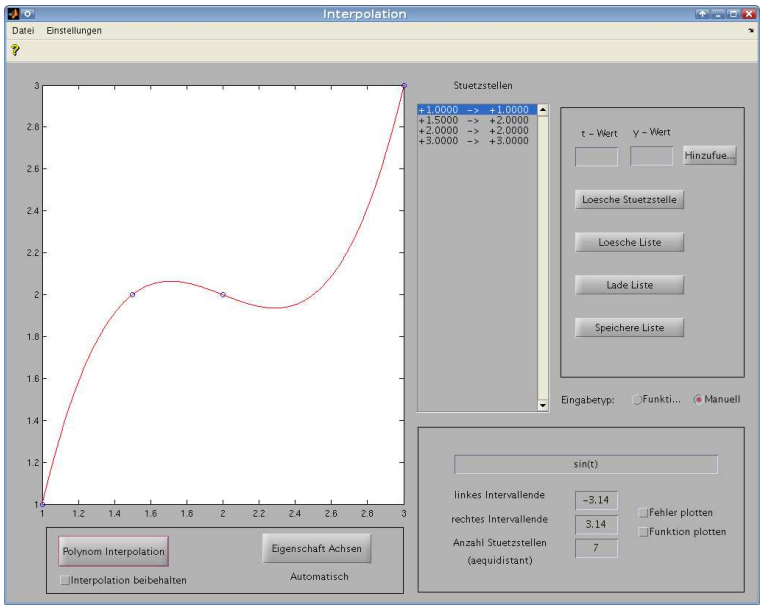
und folglich liefert $(M^{-1})^T \eta$ die gesuchte Transformation des Normalenvektors.

Im obigen Beispiel erhalten wir somit

$$(M^{-1})^T \eta = \begin{pmatrix} 1 & 0 \\ 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Kapitel 4

Interpolation und Anwendungen



Dieses Kapitel spricht zwei Ansätze zur Lösung des in der Einleitung bereits diskutierten Interpolationsproblems an.

Wir beginnen mit einem Beispiel:

4.1 Interpolationspolynom durch 3 Punkte

Wir betrachten den Fall, dass 3 Punkte $(1,1)$, $(2,1)$, $(3,5)$ gegeben sind.

Gesucht ist ein Polynom vom $\mathrm{Grad} \leq 2$, das durch diese Punkte verläuft. Dieses Polynom besitzt die Form

$$p(t) = a\,t^2 + b\,t + c,$$

wobei die Koeffizienten $a,\ b,\ c$ so zu bestimmen sind, dass die Interpolationsbedingungen

$$\begin{aligned} p(1) &= 1, \\ p(2) &= 1, \\ p(3) &= 5 \end{aligned}$$

erfüllt sind. Wir erhalten folglich das lineare Gleichungssystem

$$\begin{array}{rcrcrcl} a & + & b & + & c & = & 1 \\ 4a & + & 2b & + & c & = & 1 \\ 9a & + & 3b & + & c & = & 5 \end{array}$$

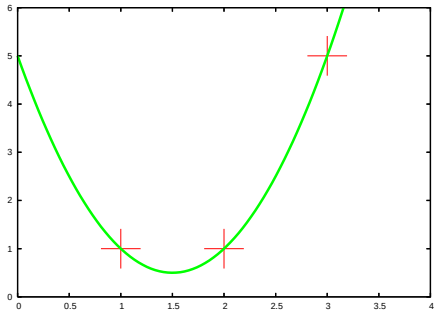
bzw. in Matrixform die Aufgabe

$$\begin{pmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 5 \end{pmatrix}.$$

Als Lösung ergibt sich $a = 2$, $b = -6$, $c = 5$ und somit ist

$$p(t) = 2t^2 - 6t + 5 \tag{4.1}$$

das gesuchte Interpolationspolynom, siehe Abbildung.



4.2 Allgemeine Fragestellung

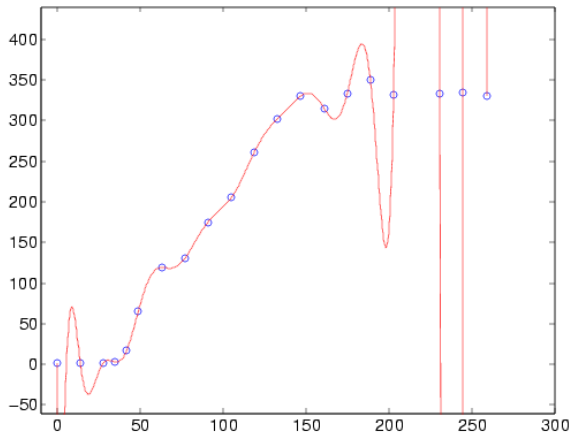
Die obige Fragestellung wird jetzt wie folgt verallgemeinert: Gegeben sei eine Menge $(m+1)$ von reellen Datenpaaren

$$(t_i, s_i), \quad i = 0, \ldots, m.$$

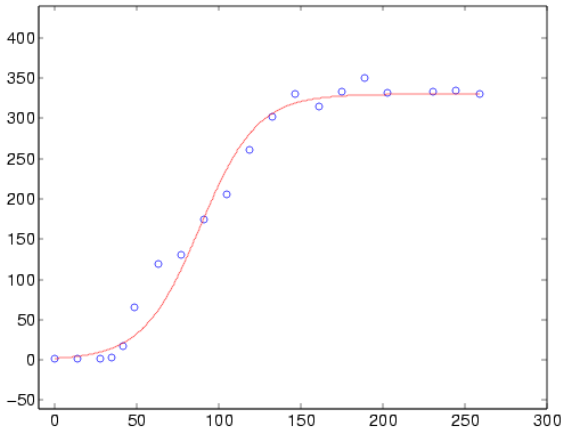
Gesucht wird eine reelle Funktion p, die durch diese Punkte verläuft, d. h.

$$p(t_i) = s_i, \quad i = 0, \ldots, m.$$

Ein Beispiel zeigt die folgende Grafik.



Stammen die Daten von Messreihen, z. B. aus der Biologie, Physik oder Chemie, so ist es fraglich, eine Kurve zu suchen, die durch diese Messwerte verläuft. Stattdessen ist es in diesen Fällen geschickter, eine glatte Kurve zu bestimmen, die möglichst nahe an den einzelnen Punkten liegt. Das zugehörige Ausgleichsproblem werden wir in Kapitel 5 studieren.



Sinnvoll ist der Interpolationsansatz nur, wenn bekannt ist, dass die gegebenen Datenpaare Stützstellen einer gesuchten Funktion sind.

4.3 Polynome

Eine Funktion der Form

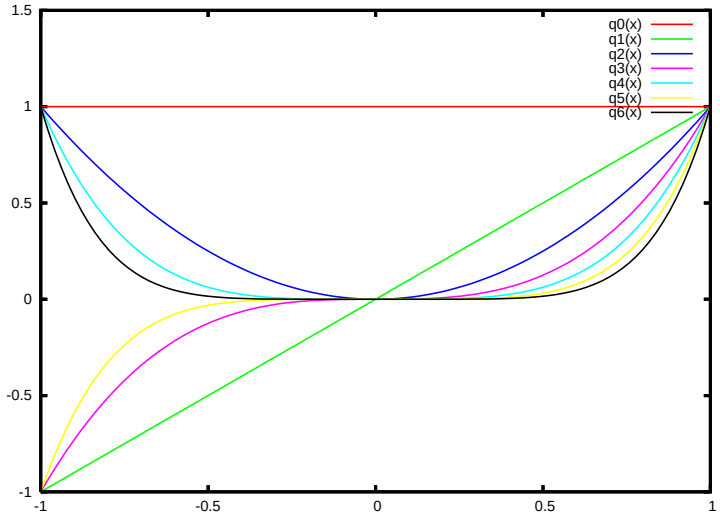
$$\begin{aligned} p(t) &:= a_m t^m + a_{m-1} t^{m-1} + \cdots + a_1 t + a_0, \\ &= \sum_{i=0}^{m} a_i t^i, \quad t \in \mathbb{R},\ a_0, \ldots, a_m \in \mathbb{R}, a_m \neq 0 \end{aligned} \tag{4.2}$$

definiert ein **reelles Polynom** vom Grad m mit den Koeffizienten $a_0, \ldots, a_m$.

Genauer gilt, dass die Menge der Polynome von Grad $\leq m$ einen Vektorraum bilden, den wir mit $\mathcal{P}_m$ bezeichnen. Dieser Vektorraum besitzt die Dimension $m+1$, und die **Monome**

$$q_j(t) := t^j, \quad j = 0, \ldots, m$$

bilden eine Basis von $\mathcal{P}_m$.



4.4 Das Horner-Schema

Als nächstes wird die Auswertung von Polynomen behandelt. Durch sukzessives Ausklammern erhalten wir aus (4.2)

$$p(t) = (\ldots(((\underbrace{a_m}_{b_m} t + a_{m-1})\underbrace{t + a_{m-2}}_{b_{m-1}})\underbrace{t + a_{m-3}}_{b_{m-2}})t + \cdots + a_1)t + a_0.$$

Im Beispiel:

$$\begin{aligned} p(t) &= 4t^3 + 3t^2 + 2t + 1 \\ &= (4t^2 + 3t + 2)t + 1 \\ &= \big((4t + 3)t + 2\big)t + 1. \end{aligned}$$

Dieser Ansatz führt zu dem **Horner-Schema** für die Auswertung von $p(\bar{t})$:

$$\begin{aligned} b_m &= a_m, \\ b_j &= b_{j+1}\bar{t} + a_j \quad \textbf{für} \quad j = m-1, \ldots, 0. \end{aligned} \tag{4.3}$$

Beachte, dass wir die Bezeichnung $\bar{t}$ verwenden, da für t der feste Wert $t = \bar{t}$ in (4.2) eingesetzt wird.

Nach der Durchführung dieser Iteration haben wir

$$b_0 = p(\bar{t}).$$

In unserem Beispiel ($a_0 = 1,\ a_1 = 2,\ a_2 = 3,\ a_3 = 4$) liefert diese Rechenvorschrift an der Stelle $\bar{t} = 2$:

$$\begin{aligned} b_3 &= a_3 = 4, \\ b_2 &= b_3\bar{t} + a_2 = 4 \cdot 2 + 3 = 11, \\ b_1 &= b_2\bar{t} + a_1 = 11 \cdot 2 + 2 = 24, \\ b_0 &= b_1\bar{t} + a_0 = 24 \cdot 2 + 1 = 49. \end{aligned}$$

Beachte, dass bei dieser Form der Polynomauswertung mit Hilfe des Horner-Schemas nur Additionen und Multiplikationen verwendet werden; es wird nicht potenziert!

Wir können (4.3) durch ein sogenanntes **Pseudoprogramm** realisieren, wobei nur ein Speicherplatz für die a_j benötigt wird.

$$\begin{aligned} b &= a_m \\ j &= m-1, \ldots, 0 \\ &\lfloor b = \bar{t} * b + a_j. \end{aligned}$$

Der Algorithmus benötigt m Multiplikationen und m Additionen, also m flops (**floating point operations**). Dabei ist

1 flop = (1 Multiplikation und 1 Addition)

gesetzt. Eine Multiplikation verlangt normalerweise einen größeren Zeitaufwand als eine Addition. Jedoch hat sich das Verhältnis in den letzten Jahren soweit angeglichen, dass neuerdings schon Additionen und Multiplikationen gleichberechtigt als ein flop gezählt werden (siehe die neueste Ausgabe von [15]).

Bei der naiven Auswertung von $p(\bar{t})$ würde man entsprechend dem folgenden Pseudoprogramm vorgehen.

$$\begin{aligned} b &= 1 \\ s &= a_0 \\ j &= 1, \ldots, m \\ &\left| \begin{array}{ll} b = b * \bar{t} & (= \bar{t}^j) \\ s = s + a_j * b. & \end{array} \right. \end{aligned}$$

Zur Durchführung dieses Algorithmus werden $2m$ Multiplikationen und m Additionen benötigt, also ist das Horner-Schema vorzuziehen.

Präziser stellt sich das Ergebnis in Form eines Satzes wie folgt dar:

Satz 4.1 *Berechnet man* $b_j,\ j = 0, \ldots, m$ *nach dem Horner-Schema, so gilt*

$$p(t) = \left(\sum_{i=0}^{m-1} b_{i+1} t^i \right) (t - \bar{t}) + b_0 \quad \forall\, t \in \mathbb{R}.$$

Insbesondere ist $p(\bar{t}) = b_0$.

Beweis: Aus der Konstruktion des Horner-Schemas (4.3) folgt $a_m = b_m$ und $a_i = b_i - \bar{t}\, b_{i+1}$ für $i = 0, \ldots, m-1$. Somit gilt:

$$\begin{aligned}
& \left(\sum_{i=0}^{m-1} b_{i+1} t^i \right) (t - \bar{t}) + b_0 \\
= & \sum_{i=0}^{m-1} b_{i+1} t^{i+1} - \sum_{i=0}^{m-1} b_{i+1} t^i\, \bar{t} + b_0 \\
= & \sum_{i=1}^{m} b_i t^i - \bar{t} \sum_{i=0}^{m-1} b_{i+1} t^i + b_0 \\
= & \sum_{i=1}^{m-1} \underbrace{(b_i - \bar{t}\, b_{i+1})}_{=a_i} t^i + \underbrace{b_m}_{=a_m} t^m \underbrace{-\bar{t}\, b_1 + b_0}_{=a_0} \\
= & \sum_{i=0}^{m} a_i t^i \\
= & \ p(t).
\end{aligned}$$

Schließlich erhalten wir

$$p(\bar{t}) = \left(\sum_{i=0}^{m-1} b_{i+1} \bar{t}^i \right) (\bar{t} - \bar{t}) + b_0 = b_0.$$

■

Die Koeffizienten $b_j,\ j = 1, \ldots, m$ des Horner-Schemas bestimmen also das durch Abspalten des Linearfaktors $(t - \bar{t})$ aus $p(t) - p(\bar{t})$ entstehende Polynom.

Praktische Anwendung des Horner-Schemas im Steuerrecht:
EStG §32a Einkommensteuertarif (2002) (inzwischen weggefallen)
(3) Die zur Berechnung der tariflichen Einkommensteuer erforderlichen

Rechenschritte sind in der Reihenfolge auszuführen, die sich nach dem Horner- Schema ergibt. Dabei sind die sich aus den Multiplikationen ergebenden Zwischenergebnisse für jeden weiteren Rechenschritt mit drei Dezimalstellen anzusetzen; die nachfolgenden Dezimalstellen sind fortzulassen. Der sich ergebende Steuerbetrag ist auf den nächsten vollen Euro-Betrag abzurunden.

4.5 Das Interpolationspolynom

Wir kommen jetzt auf die eingangs gestellte Frage zurück: Finde zu den vorgegebenen Punkten (t_i, s_i), $i = 0, \ldots, m$ ein Polynom p vom $\mathrm{Grad} \leq m$, das durch jeden dieser Punkte verläuft.

Wir wählen den folgenden Ansatz:

$$p(t_i) = \sum_{j=0}^{m} a_j t_i^j = s_i, \quad i = 0, \ldots, m \tag{4.4}$$

wobei die Koeffizienten $a_0, \ldots, a_m$ so zu bestimmen sind, dass (4.4) gilt. Man beachte, dass ein Polynom durch die Angabe seiner Koeffizienten $a_0, \ldots, a_m$ eindeutig bestimmt ist.

Diese Fragestellung führt auf ein **lineares Interpolationsproblem**, denn die Koeffizienten können durch Lösen eines linearen Gleichungssystems bestimmt werden.

Hierzu definieren wir die Matrix

$$A = \big(A_{ij}\big)_{i=0,\ldots m,\ j=0,\ldots m} \quad \text{mit} \quad A_{ij} = t_i^j, \quad i = 0, \ldots m,\ j = 0, \ldots m$$

und

$$\alpha = \begin{pmatrix} a_0 \\ \vdots \\ a_m \end{pmatrix}, \quad s = \begin{pmatrix} s_0 \\ \vdots \\ s_m \end{pmatrix},$$

dann ist (4.4) äquivalent zu

$$A\alpha = s. \tag{4.5}$$

Die gesuchten Koeffizienten sind die Lösung des linearen Gleichungssystems (4.5).

Numerisch ist diese Vorgehensweise ineffizient. Einen effizienteren Algorithmus zur Lösung des Interpolationsproblems diskutieren wir in Abschnitt 4.8.

Zunächst sei noch bemerkt, dass der Ansatz (4.4) auf beliebige Ansatzfunktionen verallgemeinert werden kann. (In (4.4) sind die Monome die verwendeten Ansatzfunktionen.)

4.6 Das allgemeine Interpolationsproblem

Wir verallgemeinern (4.4) zu

$$p(t_i) = \sum_{j=0}^{m} a_j p_j(t_i) = s_i \tag{4.6}$$

mit **Ansatzfunktionen** p_j. Einige, oft verwendete Ansatzfunktionen werden im Folgenden aufgelistet.

1. **Polynominterpolation:**

   $$p_j(t) = t^j, \quad j = 0, \ldots, m,$$

   in diesem Fall reduziert sich (4.6) auf (4.4).

2. **Interpolation mit Exponentialsummen:**

   $$p_j(t) = e^{-\mu_j t}, \quad j = 0, \ldots, m$$

   mit gegebenen Exponenten $\mu_j \in \mathbb{R}$.

3. **trigonometrische Interpolation:**
   Sei $m = 2k$ gerade,

   $$\begin{aligned} p_{2j}(t) &= \cos\left(\frac{2\pi j t}{T}\right), \quad j = 0, \ldots, k \\ p_{2j-1}(t) &= \sin\left(\frac{2\pi j t}{T}\right), \quad j = 1, \ldots, k. \end{aligned}$$

Die gesuchten Koeffizienten $a_0, \ldots, a_m$ in (4.6) finden wir wieder durch Lösen des linearen Gleichungssystems (4.5), wobei die Matrix A durch

$$A_{ij} = p_j(t_i), \quad i = 0, \ldots m, \; j = 0, \ldots m$$

definiert wird.

4.7 Lagrangesche Darstellung

Wir geben eine explizite Formel für das gesuchte Polynom an. Dazu bilden wir zunächst mit den Stützstellen $t_i, \; i = 0, \ldots, m$ die sogenannten **Lagrangeschen Basispolynome**

$$L_j(t) = \frac{\omega_j(t)}{\omega_j(t_j)}, \quad \omega_j(t) = \prod_{\substack{i=0 \\ i \neq j}}^{m} (t - t_i), \quad j = 0, \ldots, m.$$

Lemma 4.2 *Die Langrangeschen Basispolynome zu den Stützstellen* t_i, $i = 0, \ldots, m$ *besitzen die folgenden Eigenschaften:*

(i)

$$L_j(t_k) = \delta_{jk} := \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases},$$

(ii)

$$\mathrm{Grad}(L_j) = m \quad \textit{für alle} \quad j = 0, \ldots, m.$$

Beweis: Sei $j \in \{0, \ldots, m\}$ fest gewählt. Es gilt

$$L_j(t_j) = \frac{\omega_j(t_j)}{\omega_j(t_j)} = 1$$

und für $k \in \{0, \ldots, m\}$, $k \neq j$ erhalten wir

$$\omega_j(t_k) = \prod_{\substack{i=0 \\ i \neq j}}^{m} (t_k - t_i) = 0,$$

da $(t_k - t_k)$ ein Faktor dieses Produktes ist. Folglich ist auch $L_j(t_k) = 0$ und (i) ist bewiesen.

Auch der Beweis von (ii) ergibt sich unmittelbar: Sei $j \in \{0, \ldots, m\}$, dann folgt

$$\mathrm{Grad}(L_j) = \mathrm{Grad}(w_j) = m,$$

da

$$w_j(t) = \prod_{\substack{i=0 \\ i \neq j}}^{m} (t - t_i) = t^m + \cdots .$$

■

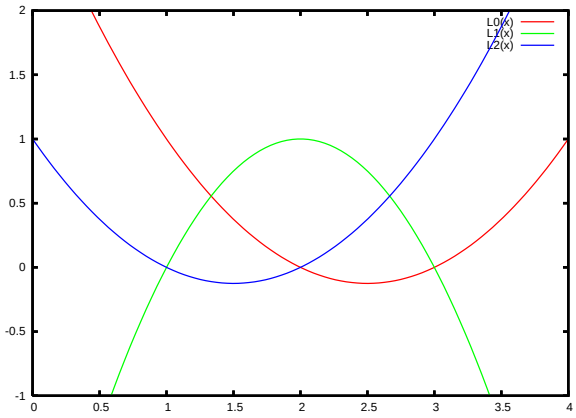
Für das Beispiel aus Abschnitt 4.1 erhält man $m = 2$,

$$\begin{aligned} t_0 &= 1, & s_0 &= 1, \\ t_1 &= 2, & s_1 &= 1, \\ t_2 &= 3, & s_2 &= 5, \end{aligned}$$

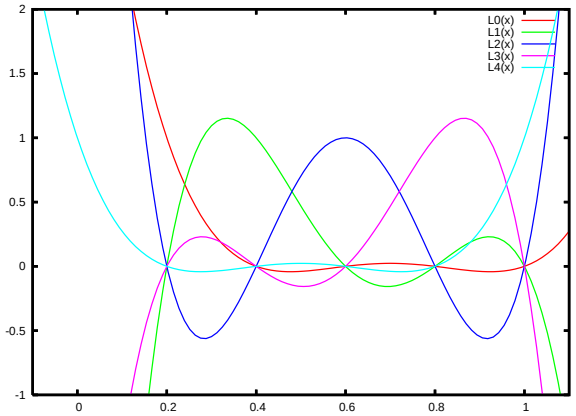
$$\begin{aligned} w_0(t) &= (t-2)(t-3) = t^2 - 5t + 6, \\ w_1(t) &= (t-1)(t-3) = t^2 - 4t + 3, \\ w_2(t) &= (t-1)(t-2) = t^2 - 3t + 2, \end{aligned}$$

$$
\begin{aligned}
L_0(t) &= \frac{t^2-5t+6}{1-5+6} = \frac{1}{2}t^2 - \frac{5}{2}t + 3, \\
L_1(t) &= \frac{t^2-4t+3}{4-8+3} = -t^2 + 4t - 3, \\
L_2(t) &= \frac{t^2-3t+2}{9-9+2} = \frac{1}{2}t^2 - \frac{3}{2}t + 1.
\end{aligned}
$$

Diese Basispolynome zeigt die Abbildung.



Die Lagrangeschen Basispolynome für $m = 4$ und für die Stützstellen $t_i = \frac{i+1}{5},\ i = (0, \cdots, 4)$ sind in der folgenden Grafik angegeben.



Zum Erhalt eines Interpolationspolynoms setzen wir jetzt

$$p(t) := \sum_{j=0}^{m} s_j L_j(t). \tag{4.7}$$

Satz 4.3 *Seien Datenpaare* (t_i, s_i)*,* $i = 0, \ldots, m$ *gegeben mit* $t_i \neq t_j$ *für alle* $i \neq j$*,* $i, j \in \{0, \ldots, m\}$ *(*t_i *sind paarweise verschieden).*

Dann besitzt das in (4.7) *definierte Polynom den* $\mathrm{Grad} \leq m$ *und* p *erfüllt die Interpolationsbedingung*

$$p(t_i) = s_i \quad \textit{für alle} \quad i = 0, \ldots, m.$$

Beweis: Da L_j nach Lemma 4.2 ein Polynom vom $\mathrm{Grad} \leq m$ ist, gilt dies auch für die Summe dieser Polynome; also $\mathrm{Grad}(p) \leq m$.

Zusätzlich verläuft dieses Polynom durch die Stützstellen (t_i, s_i), $i = 0, \ldots, m$, denn es gilt

$$p(t_k) = \sum_{j=0}^{m} s_j L_j(t_k) = \sum_{j=0}^{m} s_j \delta_{jk} = s_k, \quad \text{für } k = 0, \ldots, m.$$

■

In unserem Beispiel erhalten wir

$$\begin{aligned} p(t) &= 1 \cdot \left(\frac{1}{2}t^2 - \frac{5}{2}t + 3\right) + 1 \cdot \left(-t^2 + 4t - 3\right) + 5 \cdot \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \\ &= 2t^2 - 6t + 5, \end{aligned}$$

vgl. (4.1).

Damit ist die Existenz eines geeigneten Polynoms gesichert. Zusätzlich ist dieses Polynom auch eindeutig bestimmt, wie der folgende Satz zeigt.

Satz 4.4 *Zu* $m+1$ *Paaren* $(t_i, s_i),\ i = 0, \ldots, m$ *mit paarweise verschiedenen* t_i *gibt es genau ein Polynom* p *vom* $\mathrm{Grad} \leq m$ *mit*

$$p(t_i) = s_i,\ i = 0, \ldots, m.$$

Das Polynom p heißt das **Interpolationspolynom** zu den Daten (t_i, s_i), $i = 0, \ldots, m$.

Beweis: Die Existenz des Interpolationspolynoms liefert Satz 4.3.

- Die Eindeutigkeit kann man mit Hilfe der linearen Algebra so erhalten: Wir haben gezeigt, dass es zu jedem $s \in \mathbb{R}^{m+1}$ eine Lösung $\alpha = \begin{pmatrix} a_0 \\ \vdots \\ a_m \end{pmatrix} \in \mathbb{R}^{m+1}$ von (4.5) gibt. Also ist der Rang von A gleich $m+1$, d. h. A ist invertierbar und die Koeffizienten $a_0, \ldots, a_m$ sind eindeutig bestimmt.

- Ein alternativer Beweis der Eindeutigkeit kann wie folgt geführt werden:
  Sind p und q zwei Polynome vom $\mathrm{Grad} \leq m$ mit $p(t_i) = q(t_i) = s_i$, $i = 0, \ldots, m$, so hat das Polynom $p - q$ vom $\mathrm{Grad} \leq m$ mindestens $m + 1$ verschiedene Nullstellen $t_i,\ i = 0, \ldots, m$. Es muss also nach einem Satz der Analysis das Nullpolynom sein (dies kann man zum Beispiel mit dem Satz von Rolle einsehen).

■

Leider erweist sich die Lagrangesche Darstellung numerisch als ineffizient; einen besseren Ansatz erhalten wir mit der Newtonschen Darstellung, die im folgenden Abschnitt vorgestellt wird.

4.8 Newtonsche Darstellung

Einen schnellen Algorithmus zur Polynominterpolation liefert die sogenannte **Newtonsche Darstellung**

$$\begin{aligned} p(t) &= a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1) + \ldots + a_m(t - t_0) \cdot \ldots \cdot (t - t_{m-1}) \\ &= \sum_{j=0}^{m} a_j \prod_{i=0}^{j-1} (t - t_i). \end{aligned} \tag{4.8}$$

In unserem Beispiel aus Abschnitt 4.1 erhalten wir das Polynom

$$p(t) = a_0 + a_1(t - 1) + a_2(t - 1)(t - 2).$$

Angenommen, wir haben das Interpolationspolynom in die Form (4.8) gebracht, dann erhalten wir durch sukzessives Ausklammern die Darstellung

$$p(t) = (\ldots(\underbrace{\underbrace{(\underbrace{a_m}_{b_m}(t - t_{m-1}) + a_{m-1}}_{b_{m-1}})(t - t_{m-2}) + a_{m-2}}_{b_{m-2}})\ldots)(t - t_0) + a_0,$$

die mit dem folgenden Horner-artigen Schema

$$\begin{aligned} b_m &= a_m \\ j &= m - 1, \ldots, 0 \\ &\lfloor b_j = b_{j+1}(\bar{t} - t_j) + a_j \end{aligned}$$

ausgewertet werden kann. Es gilt dann offensichtlich $b_0 = p(\bar{t})$.

Wie können wir nun die gesuchten Koeffizienten $a_j,\ j = 0, \ldots, m$ bestimmen? Zunächst wird diese Rechnung anhand des Beispiels aus Abschnitt 4.1 durchgeführt. Es gilt

$$\begin{aligned} 1 = p(1) = a_0 &\Rightarrow a_0 = 1, \\ 1 = p(2) = 1 + a_1(2-1) &\Rightarrow a_1 = 0, \\ 5 = p(3) = 1 + a_2(3-1)(3-2) &\Rightarrow a_2 = 2. \end{aligned}$$

Also erhalten wir (wieder) das Polynom

$$p(t) = 1 + 2(t-1)(t-2) = 1 + 2(t^2 - 3t + 2) = 2t^2 - 6t + 5,$$

vgl. (4.1).

Zum Erhalt eines systematischen Algorithmus werden jetzt für drei beliebige Punkte (t_0, s_0), (t_1, s_1), (t_2, s_2) die Koeffizienten a_0, a_1 und a_2 bestimmt.

Wir nehmen an, dass das Interpolationspolynom p die Newtonsche Darstellung

$$p(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1)$$

besitzt. In der Tat kann jedes Polynom so dargestellt werden.

Durch Einsetzen von $t_0, t_1, t_2, \ldots$ folgt

$$s_0 = p(t_0) = a_0,$$

$$s_1 = p(t_1) = a_0 + a_1(t_1 - t_0) \Rightarrow a_1 = \frac{s_1 - s_0}{t_1 - t_0},$$

$$\begin{aligned} s_2 &= p(t_2) = a_0 + a_1(t_2 - t_0) + a_2(t_2 - t_0)(t_2 - t_1) \\ \Rightarrow a_2 &= \frac{s_2 - a_0 - a_1(t_2 - t_0)}{(t_2 - t_0)(t_2 - t_1)} \\ &= \frac{s_2 - a_0 - a_1(t_1 - t_0 + t_2 - t_1)}{(t_2 - t_0)(t_2 - t_1)} \\ &= \frac{s_2 \overbrace{-a_0 - a_1(t_1 - t_0)}^{-s_1} - a_1(t_2 - t_1)}{(t_2 - t_0)(t_2 - t_1)} \\ &= \frac{s_2 - s_1 - a_1(t_2 - t_1)}{(t_2 - t_0)(t_2 - t_1)} \\ &= \frac{\frac{s_2 - s_1}{t_2 - t_1} - \frac{s_1 - s_0}{t_1 - t_0}}{t_2 - t_0}. \end{aligned}$$

Den Zusammenhang der Koeffizienten verdeutlicht das folgende Dreiecksschema, das auch als Schema der **dividierten Differenzen** bezeichnet wird:

$$\begin{array}{llllll} t_0 & s_0 = d_{00} & & & & \\ & & \searrow & & & \\ t_1 & s_1 = d_{10} & \rightarrow & d_{11} & & \\ & & \searrow & & \searrow & \\ t_2 & s_2 = d_{20} & \rightarrow & d_{21} & \rightarrow & d_{22}, \end{array}$$

mit

$$\begin{aligned} d_{11} &= \frac{d_{10} - d_{00}}{t_1 - t_0} = \frac{s_1 - s_0}{t_1 - t_0} = a_1, \\ d_{21} &= \frac{d_{20} - d_{10}}{t_2 - t_1} = \frac{s_2 - s_1}{t_2 - t_1}, \\ d_{22} &= \frac{d_{21} - d_{11}}{t_2 - t_0} = \frac{\frac{s_2 - s_1}{t_2 - t_1} - \frac{s_1 - s_0}{t_1 - t_0}}{t_2 - t_0} = a_2. \end{aligned}$$

Bei der Berechnung der Koeffizienten eines Interpolationspolynoms durch $n + 1$ Punkte, erhält man entsprechend das folgende Schema der **dividierten Differenzen** (siehe zum Beispiel [17], [20]):

$$\begin{array}{llllllll} t_0 & s_0 = d_{00} & & & & & & \\ & & \searrow & & & & & \\ t_1 & s_1 = d_{10} & \rightarrow & d_{11} & & & & \\ & & \searrow & & \searrow & & & \\ t_2 & s_2 = d_{20} & \rightarrow & d_{21} & \rightarrow & d_{22} & & \\ \vdots & \vdots & & \vdots & & \ddots & & \\ \vdots & \vdots & & \vdots & & & \ddots & \\ & & \searrow & & \searrow & & & \searrow \\ t_m & s_m = d_{m0} & \rightarrow & d_{m1} & \rightarrow & d_{m2} & \dots \quad \dots & \rightarrow \quad d_{mm}, \end{array}$$

welches durch den folgenden Algorithmus definiert wird

$$\begin{array}{ll} i & = 0, \dots, m \\ & \lfloor d_{i0} = s_i \\ j & = 1, \dots, m \\ & \left\lfloor \begin{array}{ll} i & = j, \dots, m \\ & \lfloor d_{ij} = \frac{d_{i,j-1} - d_{i-1,j-1}}{t_i - t_{i-j}} \end{array} \right. \end{array}$$

Die Diagonalelemente $a_j = d_{jj}$ sind dann die gesuchten Koeffizienten.

4.9 Interpolation einer gegebenen Funktion durch ein Polynom

Das Interpolationspolynom vom $\mathrm{Grad} \leq m$ wird nach Satz 4.4 eindeutig durch die Angaben von $m+1$ Datenpaare (t_i, s_i), $i = 0, \ldots, m$ bestimmt. Diese Datenpaare können Messwerte eines biologischen Experiments sein. Alternativ kann aber auch eine Wertetabelle einer Funktion $f : \mathbb{R} \to \mathbb{R}$ vorliegen. In diesem Fall stellt sich die Frage, ob das Interpolationspolynom p auch nahe an der Funktion f liegt. Insbesondere ist von Interesse, wie viele Datenpaare benötigt werden, um eine gute Übereinstimmung zu erzielen.

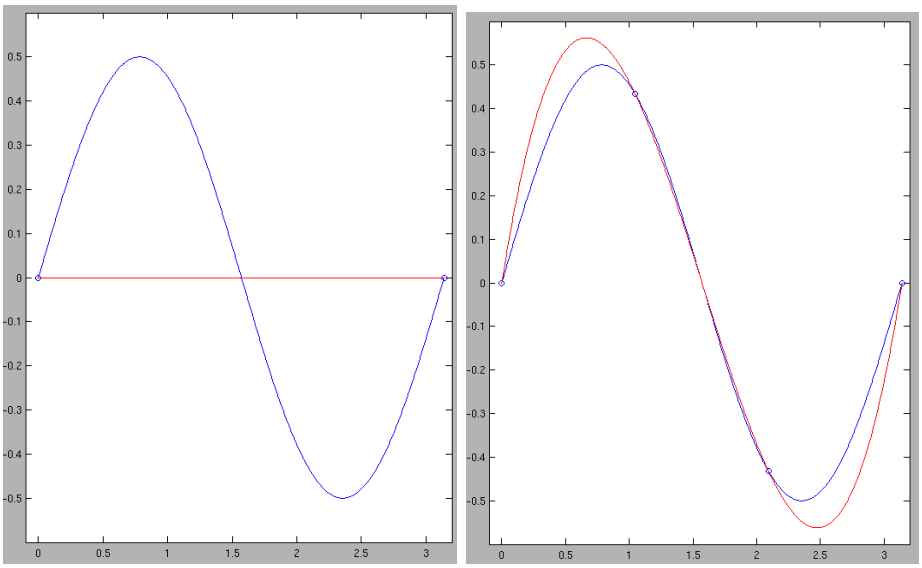
Als Beispiel betrachten wir die Funktion

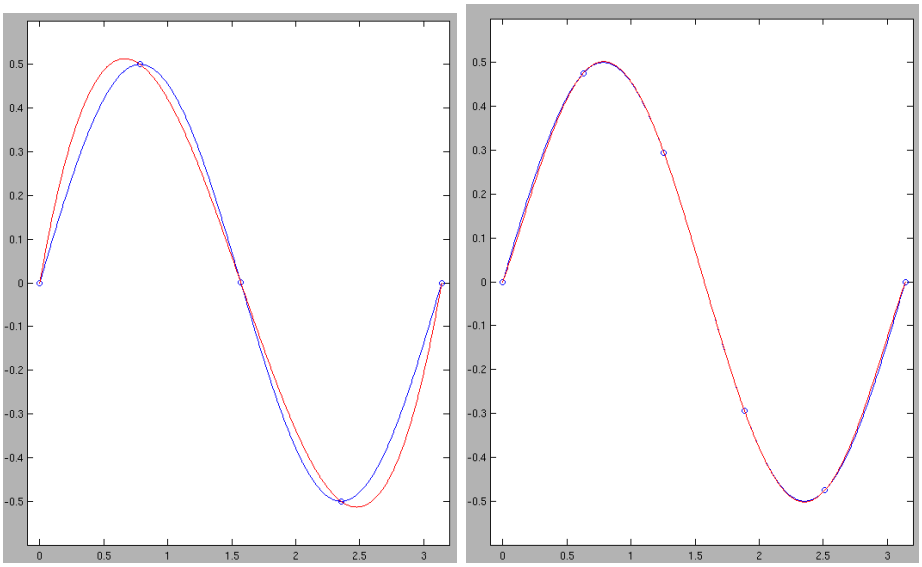
$$f(t) = \sin(t)\cos(t), \quad t \in [0, \pi]$$

und wählen für die Polynominterpolation $m+1$ äquidistant verteilte Stützstellen

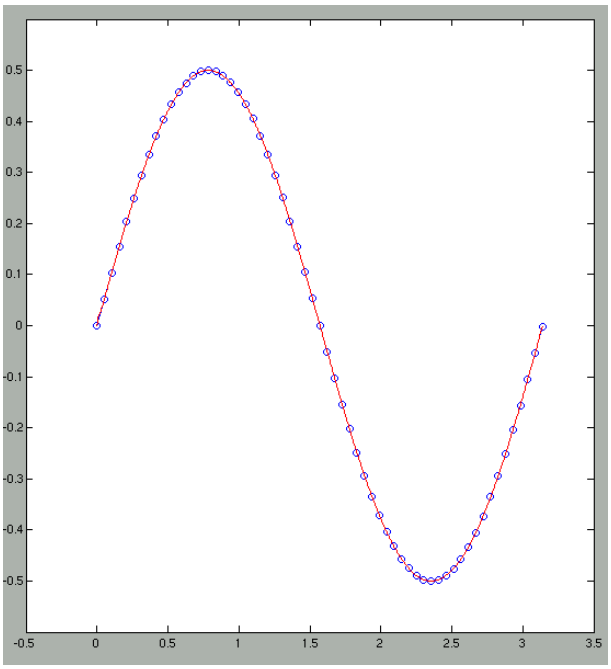
$$t_i = i\frac{\pi}{m}, \quad s_i = f(t_i), \quad i = 0, \cdots, m.$$

Die Abbildungen zeigen jeweils den Graphen der Funktion f (in blau) und den Graphen des Interpolationspolynoms (in rot) für $m \in \{1, 3, 4, 5\}$.



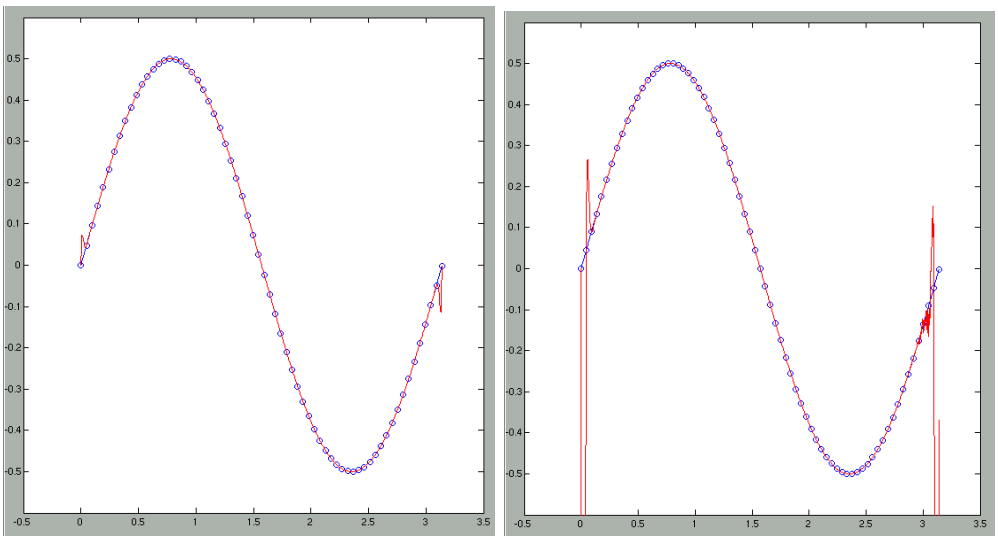


Für $m = 60$ stimmt das Interpolationspolynom gut mit der Funktion überein, wie die Abbildung zeigt.



Somit liegt der **falsche** Schluss nah, dass das Interpolationspolynom für $m \to \infty$ gegen die gesuchte Funktion konvergiert.

In Wirklichkeit zeigt das Interpolationspolynom starke Oszillationen, wenn m weiter vergrößert wird, vgl. die beiden Abbildungen für $m = 65$ und $m = 70$.



4.10 Bemerkung zur Eindeutigkeit des Interpolationspolynoms

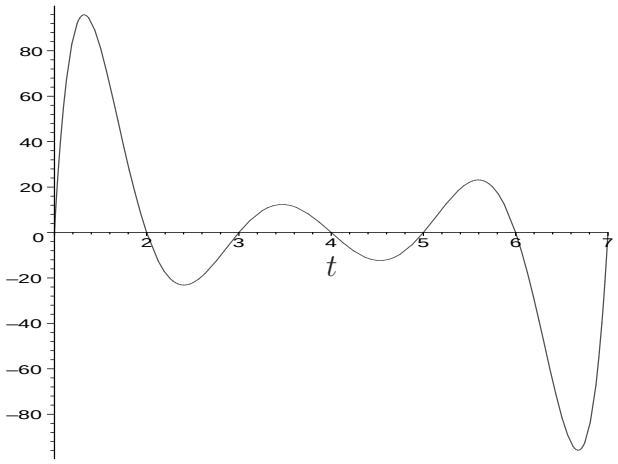
Betrachte die auf der x-Achse liegenden Stützstellen $(1,0)$, $(2,0)$, $(3,0)$, $(4,0)$, $(5,0)$, $(6,0)$, $(7,0)$.

Offensichtlich liegen diese Punkte auf der Geraden $f(x) = 0$. Insbesondere liefert die Eindeutigkeit des Interpolationspolynoms (siehe Satz 4.4) – es besitzt in diesem Beispiel den $\mathrm{Grad} \leq 6$ – dass die Funktion $f(t) = 0$ das gesucht Interpolationspolynom ist.

Die Eindeutigkeit ist natürlich nicht mehr gegeben, wenn Polynome höheren Grades betrachtet werden. So kann ein Interpolationspolynom vom $\mathrm{Grad}\ m = 7$ einfach bestimmt werden, das durch die gegebenen Datenpaare läuft:

$$\begin{aligned} p(t) &= (t-1)(t-2)(t-3)(t-4)(t-5)(t-6)(t-7) \\ &= t^7 - 28t^6 + 322t^5 - 1960t^4 + 6769t^3 - 13132t^2 + 13068t - 5040. \end{aligned}$$

In der Abbildung ist der Graph dieser Funktion angegeben.



Es ist zu erkennen, dass die y-Werte zwischen den Stützstellen extrem groß werden; dieser Effekt verstärkt sich, je mehr Stützstellen gegeben sind.

4.11 Die Taylor-Entwicklung

In diesem Abschnitt leiten wir die sogenannte Taylor-Entwicklung einer Funktion her.

Wir beginnen mit einem Beispiel. Sei

$$f(t) = t^2 + 2t + 1,$$

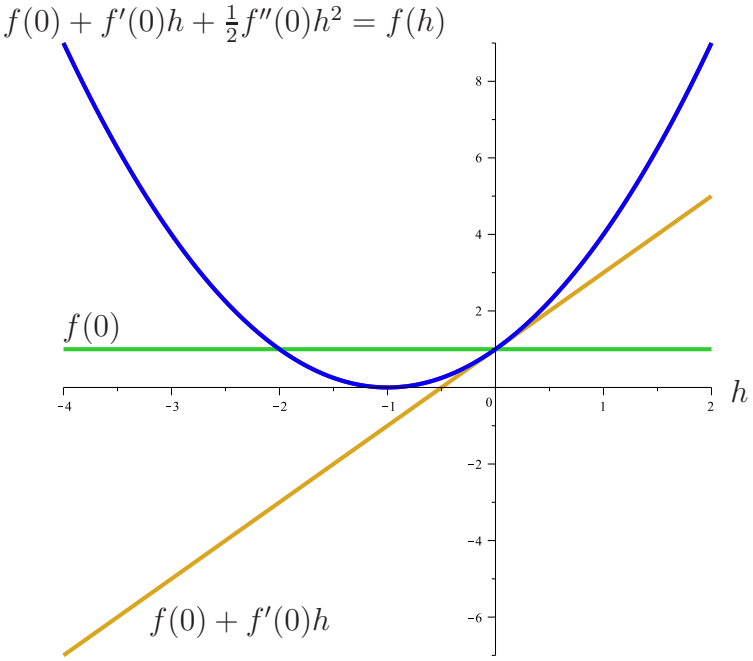
dann gilt

$$f'(t) = 2t + 2, \quad f''(t) = 2, \quad f^{(i)}(t) = 0 \text{ für alle } i \geq 3.$$

Somit erhalten wir für beliebige $h \in \mathbb{R}$ die Darstellung

$$\begin{aligned} f(t+h) &= (t+h)^2 + 2(t+h) + 1 \\ &= t^2 + 2ht + h^2 + 2t + 2h + 1 \\ &= (t^2 + 2t + 1) + \frac{1}{1!}(2t+2)h + \frac{1}{2!}2h^2 \\ &= f(t) + \frac{1}{1!}f'(t)h + \frac{1}{2!}f''(t)h^2. \end{aligned}$$

Diese Rechnung illustriert die folgende Abbildung im Fall $t = 0$.



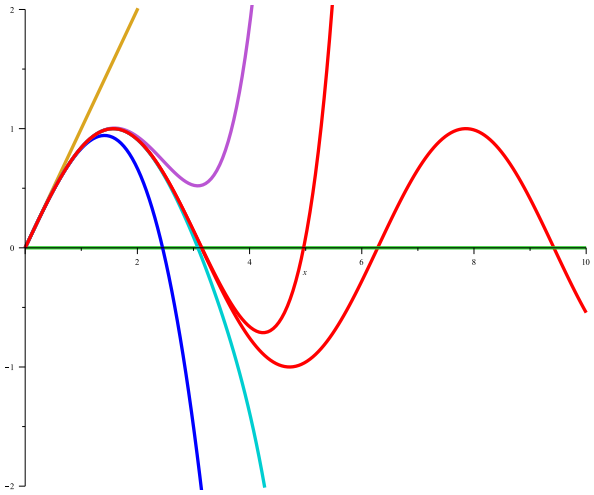
Für eine hinreichend oft stetig differenzierbare Funktion erhalten wir

$$f(t+h) = f(t) + f'(t)h + \frac{1}{2!}f''(t)h^2 + \frac{1}{3!}f'''(t)h^3 + \dots.$$

Am Beispiel der Sinus-Funktion $f(t) = \sin(t)$ finden wir an der Stelle $t = 0$ die Darstellung

$$\sin(h) = h - \frac{1}{3!}h^3 + \frac{1}{5!}h^5 - \frac{1}{7!}h^7 + - \dots.$$

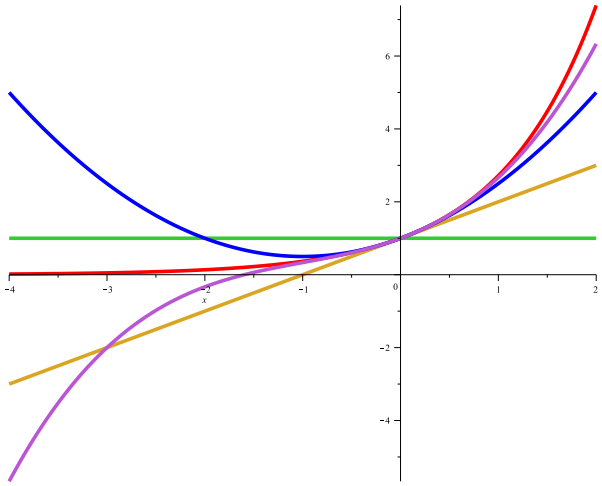
Die Abbildung zeigt die Partialsummen.



Im Fall der Exponentialfunktion erhalten wir mit $t = 0$ die Entwicklung

$$\begin{aligned}\exp(h) &= \exp(0) + \exp'(0)h + \frac{1}{2!}\exp''(0)h^2 + \frac{1}{3!}\exp'''(0)h^3 + \dots \\ &= 1 + h + \frac{1}{2!}h^2 + \frac{1}{3!}h^3 + \dots \\ &= \sum_{i=0}^{\infty} \frac{h^i}{i!},\end{aligned}$$

also die, aus der Analysis bekannte Reihendarstellung. Die Partialsummen zeigt die folgende Abbildung.



Wir zeigen die folgende **Taylor-Formel**:

Satz 4.5 *Sei $I \subset \mathbb{R}$ ein abgeschlossenes Intervall und $f : I \to \mathbb{R}$ eine $(k+1)$-mal stetig differenzierbare Funktion. Seien t und $t + h \in I$, dann gilt*

$$f(t+h) = f(t) + \frac{1}{1!}f'(t)h + \frac{1}{2!}f''(t)h^2 + \dots + \frac{1}{k!}f^{(k)}(t)h^k + R_{k+1}(t,h)$$

mit

$$R_{k+1}(t,h) = \frac{1}{k!}\int_t^{t+h} (t+h-\tau)^k f^{(k+1)}(\tau)d\tau.$$

Beweis: Der Beweis wird mit vollständiger Induktion über k geführt.

- Induktionsanfang: $k = 0$:

Nach dem Hauptsatz der Differential- und Integralrechnung gilt

$$\begin{aligned} f(t+h) &= f(t) + \int_t^{t+h} f'(\tau)d\tau \\ &= f(t) + \frac{1}{0!}\int_t^{t+h} (t+h-\tau)^0 f^{(0+1)}(\tau)d\tau \\ &= f(t) + R_1(t,h). \end{aligned}$$

- Induktionsschritt: $k-1 \to k$:
  Nach Induktionsannahme gilt

$$\begin{aligned} f(t+h) &= f(t) + \tfrac{1}{1!}f'(t)h + \tfrac{1}{2!}f''(t)h^2 + \cdots + \tfrac{1}{(k-1)!}f^{(k-1)}(t)h^{k-1} \\ &\quad + R_k(t,h) \end{aligned}$$

  und mit partieller Integration erhalten wir

$$\begin{aligned} R_k(t,h) &= \frac{1}{(k-1)!}\int_t^{t+h} (t+h-\tau)^{k-1} f^{(k)}(\tau)d\tau \\ &= -\int_t^{t+h} \frac{\partial}{\partial\tau}\left(\frac{(t+h-\tau)^k}{k!}\right) f^{(k)}(\tau)d\tau \\ &= -f^{(k)}(\tau)\frac{(t+h-\tau)^k}{k!}\Big|_{\tau=t}^{\tau=t+h} + \int_t^{t+h} \frac{(t+h-\tau)^k}{k!} f^{(k+1)}(\tau) \\ &= \frac{1}{k!}f^{(k)}(t)h^k + R_{k+1}(t,h). \end{aligned}$$

  Zusammen ergibt dies

$$\begin{aligned} f(t+h) &= f(t) + \tfrac{1}{1!}f'(t)h + \tfrac{1}{2!}f''(t)h^2 + \cdots + \tfrac{1}{(k-1)!}f^{(k-1)}(t)h^{k-1} \\ &\quad + \tfrac{1}{k!}f^{(k)}(t)h^k + R_{k+1}(t,h). \end{aligned}$$

■

Jetzt leiten wir eine Abschätzung des Restgliedes $R_{k+1}(t,h)$ her. Da die stetige Funktion $f^{(k+1)}$ ihr Maximum in dem abgeschlossenen Intervall I annimmt, d. h. es existiert eine Konstante $C > 0$ mit

$$\max_{t\in I} |f^{(k+1)}(t)| \leq C,$$

folgt

$$\begin{aligned} |R_{k+1}(t,h)| &\leq \frac{1}{k!}\left|\int_t^{t+h} |(t+h-\tau)^k||f^{(k+1)}(\tau)|d\tau\right| \\ &\leq \frac{C}{k!}\left|\int_t^{t+h} (t+h-\tau)^k d\tau\right| \\ &= \frac{C}{k!}\left|\frac{(t+h-\tau)^{k+1}}{k+1}\Big|_{\tau=t}^{\tau=t+h}\right| \\ &= \frac{C}{(k+1)!}|h|^{k+1}. \end{aligned}$$

4.12 Numerische Differentiation

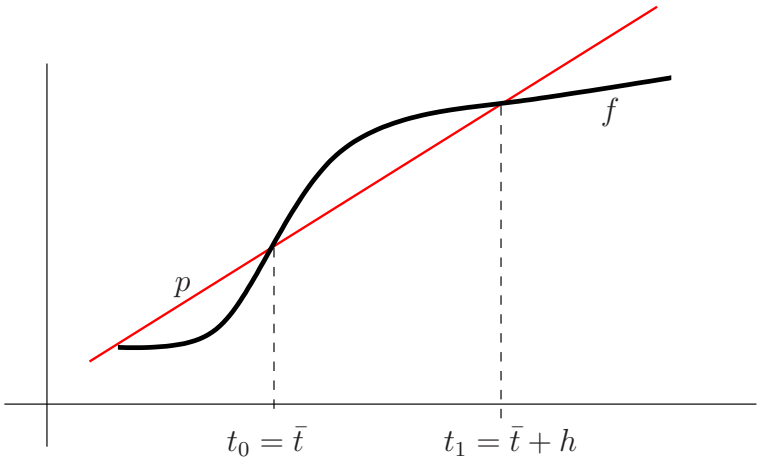
In Abschnitt 4.9 haben wir gesehen, dass eine gegebene Funktion durch ein Interpolationspolynom mit hoher Genauigkeit approximiert werden kann. Wir wollen diesen Ansatz jetzt verwenden, um auch eine gute Approximation der Ableitung zu erhalten.

Idee zur numerischen Berechnung der Ableitung: Approximiere die gegebene Funktion f mit Hilfe des Interpolationspolynoms p und leite das Interpolationspolynom p ab. Die Ableitung von p ist dann auch eine gute Approximation der Ableitung von f.

Sei eine stetig differenzierbare Abbildung $f : \mathbb{R} \to \mathbb{R}$ gegeben, deren Ableitung wir an der Stelle $\bar{t} \in \mathbb{R}$ berechnen wollen. Als erstes wählen wir $m+1$ Stützstellen $t_0 < \cdots < t_m$ in der Nähe von $\bar{t}$. Dann berechnen wir das Interpolationspolynom zu den Datenpaaren $(t_i, f(t_i))$, $i = 0, \ldots, m$. Unter Verwendung der Lagrangeschen Darstellung des Interpolationspolynoms aus Abschnitt 4.7 erhalten wir

$$p(t) = \sum_{j=0}^{m} f(t_j)L_j(t) = \sum_{j=0}^{m} f(t_j)\frac{\omega_j(t)}{\omega_j(t_j)} = \sum_{j=0}^{m} f(t_j)\prod_{\substack{i=0\\ i\neq j}}^{m}\frac{t-t_i}{t_j-t_i}.$$

Im Fall $m = 1$ und für die Wahl $t_0 = \bar{t}$, $t_1 = \bar{t} + h$ illustriert die Abbildung diese Konstruktion. Es ist zu beachten, dass für $h \to 0$ die Steigung von p am Punkt $\bar{t}$ gegen die Steigung von f konvergiert.



Da p ein Polynom ist, kann diese Funktion einfach (automatisch) abgeleitet werden:

$$p'(\bar{t}) = \sum_{j=0}^{m} f(t_j) L_j'(\bar{t}) = \sum_{j=0}^{m} f(t_j) \frac{\partial}{\partial t} \left(\prod_{\substack{i=0 \\ i \neq j}}^{m} \frac{t - t_i}{t_j - t_i} \right)_{|t=\bar{t}}$$

und wird erhalten die allgemeine Darstellung:

$$p'(\bar{t}) = \sum_{j=0}^{m} f(t_j) \sum_{\substack{k=0 \\ k \neq j}}^{m} \left(\frac{1}{t_j - t_k} \prod_{\substack{i=0 \\ i \neq k \\ i \neq j}}^{m} \frac{\bar{t} - t_i}{t_j - t_i} \right).$$

Konkret liefert dieser Ansatz im Fall $m = 1$ mit der obigen Setzung

$$\begin{aligned} L_0(t) &= \frac{t - t_1}{t_0 - t_1} \quad \Rightarrow \quad L_0'(t) = \frac{1}{t_0 - t_1}, \\ L_1(t) &= \frac{t - t_0}{t_1 - t_0} \quad \Rightarrow \quad L_1'(t) = \frac{1}{t_1 - t_0}. \end{aligned}$$

Mit $t_0 = \bar{t}$, $t_1 = \bar{t} + h$ erhalten wir somit

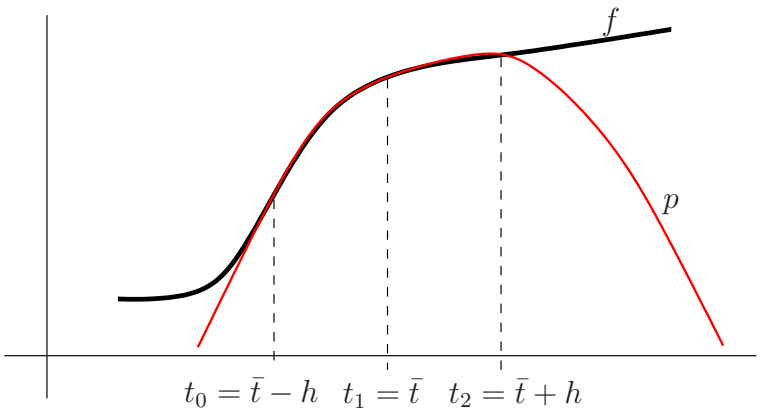
$$\begin{aligned} L_0'(\bar{t}) &= \frac{1}{\bar{t} - \bar{t} - h} = \frac{1}{-h}, \\ L_1'(\bar{t}) &= \frac{1}{\bar{t} + h - \bar{t}} = \frac{1}{h} \end{aligned}$$

und zusammen ergibt sich als Approximation $p'(\bar{t})$ der Ableitung $f'(\bar{t})$ die Formel

$$f'(\bar{t}) \approx p'(\bar{t}) = \sum_{j=0}^{1} f(t_j) L_j'(\bar{t}) = f(\bar{t}) \frac{1}{-h} + f(\bar{t} + h) \frac{1}{h} = \frac{f(\bar{t} + h) - f(\bar{t})}{h}.$$

Diese Formel ist der bekannte **vorwärtsgenommene Differenzenquotient**.

Für $m = 2$, $t_0 = \bar{t} - h$, $t_1 = \bar{t}$ und $t_2 = \bar{t} + h$ rechnen wir auch $p'(\bar{t})$ aus, siehe Abbildung.



Es gilt:

$$\begin{aligned}
L_0(t) &= \frac{(t-\bar{t})(t-\bar{t}-h)}{(\bar{t}-h-\bar{t})(\bar{t}-h-\bar{t}-h)} = \frac{1}{2h^2}(t-\bar{t})(t-\bar{t}-h),\\
L_1(t) &= \frac{(t-\bar{t}+h)(t-\bar{t}-h)}{(\bar{t}-\bar{t}+h)(\bar{t}-\bar{t}-h)} = -\frac{1}{h^2}(t-\bar{t}+h)(t-\bar{t}-h),\\
L_2(t) &= \frac{(t-\bar{t}+h)(t-\bar{t})}{(\bar{t}+h-\bar{t}+h)(\bar{t}+h-\bar{t})} = \frac{1}{2h^2}(t-\bar{t}+h)(t-\bar{t}).
\end{aligned}$$

Direktes Nachrechnen zeigt:

$$\begin{aligned}
L_0'(\bar{t}) &= -\frac{1}{2h},\\
L_1'(\bar{t}) &= 0,\\
L_2'(\bar{t}) &= \frac{1}{2h}
\end{aligned}$$

und zusammen erhalten wir

$$p'(\bar{t}) = -\frac{1}{2h}f(\bar{t}-h) + 0 \cdot f(\bar{t}) + \frac{1}{2h}f(\bar{t}+h) = \frac{f(\bar{t}+h) - f(\bar{t}-h)}{2h}.$$

Diese Formel wird auch als **zentraler Differenzenquotient** bezeichnet.

Für den vorwärtsgenommenen Differenzenquotienten und den zentralen Differenzenquotienten beweisen wir die folgenden Fehlerabschätzungen.

Satz 4.6 *Sei $f : [a,b] \to \mathbb{R}$ zwei-mal bzw. drei-mal stetig differenzierbar und seien $\bar{t}, \bar{t}+h, \bar{t}-h \in [a,b]$, dann gilt:*

$$d_1(h) := \left| f'(\bar{t}) - \frac{f(\bar{t}+h) - f(\bar{t})}{h} \right| \leq \frac{h}{2} \max_{t\in[a,b]} |f''(t)|, \tag{4.9}$$

$$d_2(h) := \left| f'(\bar{t}) - \frac{f(\bar{t}+h) - f(\bar{t}-h)}{2h} \right| \leq \frac{h^2}{6} \max_{t\in[a,b]} |f'''(t)|. \tag{4.10}$$

Beweis: Nach der in Abschnitt 4.11 hergeleiteten Taylorentwicklung erhalten wir:

$$f(\bar{t}+h) = f(\bar{t}) + hf'(\bar{t}) + R_2(\bar{t}, h)$$

und es folgt

$$\frac{f(\bar{t}+h) - f(\bar{t})}{h} - f'(\bar{t}) = \frac{R_2(\bar{t}, h)}{h}.$$

Unter Verwendung der Abschätzung des Restgliedes ergibt sich

$$d_1(h) \leq \left| \frac{R_2(\bar{t}, h)}{h} \right| \leq \frac{1}{2!} \max_{t\in[a,b]} |f''(t)| \frac{h^2}{h} = \frac{h}{2} \max_{t\in[a,b]} |f''(t)|$$

und somit ist (4.9) bewiesen.

Zum Beweis von (4.10) verwenden wir die Taylor-Entwicklungen:

$$\begin{aligned} f(\bar{t}+h) &= f(\bar{t}) + f'(\bar{t})h + \frac{1}{2}f''(\bar{t})h^2 + R_3(\bar{t}, h), \\ f(\bar{t}-h) &= f(\bar{t}) - f'(\bar{t})h + \frac{1}{2}f''(\bar{t})h^2 + R_3(\bar{t}, -h). \end{aligned}$$

Die Subtraktion der zweiten Gleichung von der Ersten liefert

$$f(\bar{t}+h) - f(\bar{t}-h) = 2f'(\bar{t})h + R_3(\bar{t}, h) - R_3(\bar{t}, -h)$$

und es folgt

$$\frac{f(\bar{t}+h) - f(\bar{t}-h)}{2h} - f'(\bar{t}) = \frac{1}{2h}(R_3(\bar{t}, h) - R_3(\bar{t}, -h)).$$

Mit $|R(\bar{t}, \pm h)| \leq \frac{1}{6} \max_{t\in[a,b]} |f'''(t)|$ folgt schließlich die behauptete Abschätzung (4.10).

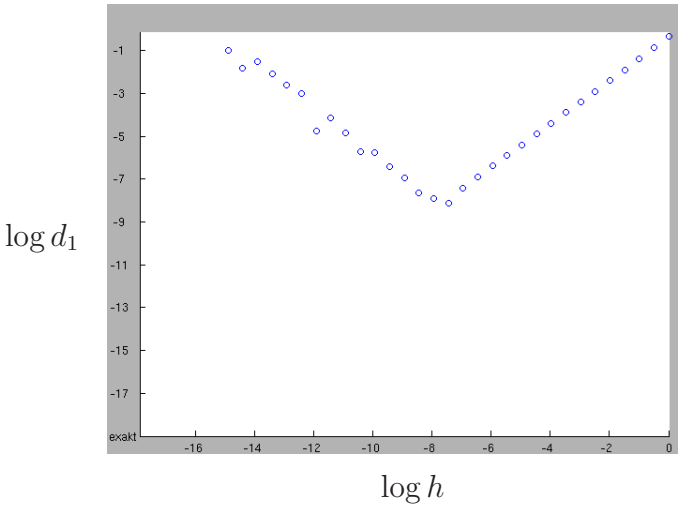
■

Diese Abschätzungen zeigen, dass bei gleicher Wahl von h, vom zentralen Differenzenquotienten eine höhere Genauigkeit zu erwarten ist.

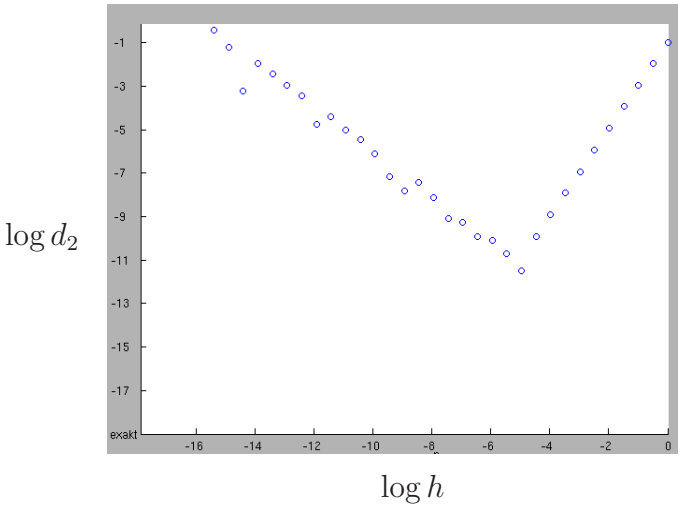
Es stellt sich schließlich die Frage, wie das optimale h in einem Programm zur Approximation der Ableitung zu wählen ist. Zum einen zeigen (4.9) und (4.10), dass die Approximation für kleiner werdendes h immer besser wird. Aber diese Abschätzungen berücksichtigen keine Rundungsfehler! Rundungsfehler werden immer größer, je kleiner h gewählt wird.

Die optimale Wahl von h untersuchen wir jetzt an einem Beispiel numerisch; wir leiten $f(t) = \sin(t)$ numerisch ab und vergleichen mit der exakten Lösung. Den Fehler $\log d_1(h)$ bzw. $\log d_2(h)$ plotten wir über $\log h$ in einem doppelt-logarithmischen Diagramm.

Numerisch werden diese Berechnungen mit einer Genauigkeit von 16 Nachkommastellen durchgeführt; die **Maschinengenauigkeit** beträgt folglich $\Delta = 10^{-16}$.



Diese Abbildung zeigt, dass beim vorwärtsgenommenen Differenzenquotienten der optimale Wert von h bei etwa $10^{-8} = \sqrt{\Delta}$ liegt. Der minimale Fehler der Ableitung beträgt $10^{-8} = \sqrt{\Delta}$ und somit kann die Ableitung mit diesem Verfahren „nur“ auf 8 Nachkommastellen genau approximiert werden.



Beim zentralen Differenzenquotienten liegt die optimale Wahl von h bei etwa $10^{-5} \approx \Delta^{\frac{1}{3}}$ und der minimale Approximationsfehler hat die Ord-

nung $10^{-12} \approx \Delta^{\frac{2}{3}}$, d. h. $\frac{2}{3}$ der Nachkommastellen können korrekt berechnet werden.

4.13 Partielle Ableitungen

Da partielle Ableitungen nichts anderes sind als gewöhnliche Ableitungen bei festgehaltenen übrigen Variablen, können wir den zentralen Differenzenquotienten einfach verwenden. Für eine stetig differenzierbare Funktion

$$f : \begin{array}{c} \mathbb{R}^2 \to \mathbb{R} \\ (x,y) \to f(x,y) \end{array}$$

approximieren wir

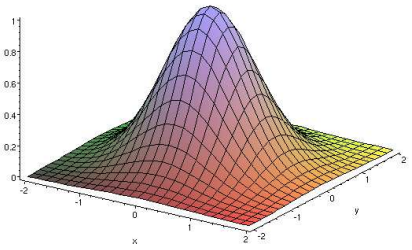
$$\begin{aligned}
\frac{\partial f}{\partial x}(x,y) &\sim \frac{1}{2h}(f(x+h,y) - f(x-h,y)), \\
\frac{\partial f}{\partial y}(x,y) &\sim \frac{1}{2h}(f(x,y+h) - f(x,y-h)), \\
\frac{\partial^2 f}{\partial x \partial x}(x,y) &\sim \frac{1}{4h^2}(f(x-2h,y) - 2f(x,y) + f(x+2h,y)), \\
\frac{\partial^2 f}{\partial x \partial y}(x,y) &\sim \frac{1}{4h^2}(f(x+h,y+h) - f(x+h,y-h) \\
&\quad -f(x-h,y+h) + f(x-h,y-h)), \\
\frac{\partial^2 f}{\partial y \partial y}(x,y) &\sim \frac{1}{4h^2}(f(x,y-2h) - 2f(x,y) + f(x,y+2h)).
\end{aligned}$$

Kapitel 5

Minimierungs- und Ausgleichsprobleme

5.1 Berechnung lokaler Extrema

Gegeben sei eine Funktion $f : D \subset \mathbb{R}^n \to \mathbb{R}$. Im Fall $n = 2$ haben wir das Beispiel $f(x) = \mathrm{e}^{-x_1^2 - x_2^2}$ bereits in Abschnitt 3.5.2 untersucht.



Im Folgenden diskutieren wir die Berechnung lokaler Minima bzw. Maxima. Die Definition lokaler Extrema entspricht der Definition im ein-dimensionalen Fall.

Definition 5.1 Die Abbildung $f : D \subset \mathbb{R}^n \to \mathbb{R}$ besitzt im Punkt $\bar{x}$ ein

- **lokales Maximum**, falls eine Umgebung $U \subset D$ von $\bar{x}$ existiert, mit $f(x) \leq f(\bar{x})$ für alle $x \in U$.
- **lokales Minimum**, falls eine Umgebung $U \subset D$ von $\bar{x}$ existiert, mit $f(x) \geq f(\bar{x})$ für alle $x \in U$.

Im Fall $n = 1$ ist der folgende Satz bekannt:

Satz 5.2 *Sei $f : D \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion, wobei D ein offenes Intervall ist.*

(i) Liegt bei $\bar{x} \in D$ ein lokales Extremum vor, so gilt $f'(\bar{x}) = 0$.

(ii) Gilt zusätzlich zu (i) $f''(\bar{x}) < 0$, so liegt bei $\bar{x}$ ein lokales Maximum vor.

(iii) Gilt zusätzlich zu (i) $f''(\bar{x}) > 0$, so liegt bei $\bar{x}$ ein lokales Minimum vor.

Satz 5.2 kann entsprechend im höherdimensionalen Fall $n \geq 2$ formuliert werden. Die Ableitung f' entspricht dann dem, in Abschnitt 3.4.1 eingeführten Gradienten und die Bedingungen an die zweiter Ableitung werden mit Hilfe der sogenannten Hesse-Matrix ausgedrückt.

5.1.1 Die Hesse-Matrix

Definition 5.3 Sei $f : \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar. Die Matrix

$$\mathrm{Hess}(f)(x) := \begin{pmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(x) & \dots & \frac{\partial^2}{\partial x_1 \partial x_n} f(x) \\ \vdots & & \vdots \\ \frac{\partial^2}{\partial x_n \partial x_1} f(x) & \dots & \frac{\partial^2}{\partial x_n \partial x_n} f(x) \end{pmatrix}$$

heißt **Hessematrix** von f.

Beispiel 5.4 Als Beispiel betrachten wir die Abbildung $f(x) = x_1^3 \sin(x_2)$. Dann gilt

$$\begin{aligned} \frac{\partial}{\partial x_1} f(x) &= 3x_1^2 \sin(x_2), \\ \frac{\partial}{\partial x_2} f(x) &= x_1^3 \cos(x_2), \\ \frac{\partial^2}{\partial x_1 \partial x_1} f(x) &= 6x_1 \sin(x_2), \\ \frac{\partial^2}{\partial x_1 \partial x_2} f(x) &= 3x_1^2 \cos(x_2), \\ \frac{\partial^2}{\partial x_2 \partial x_1} f(x) &= 3x_1^2 \cos(x_2), \\ \frac{\partial^2}{\partial x_2 \partial x_2} f(x) &= -x_1^3 \sin(x_2) \end{aligned}$$

und es folgt

$$\mathrm{Hess}(f)(x) = \begin{pmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(x) & \frac{\partial^2}{\partial x_1 \partial x_2} f(x) \\ \frac{\partial^2}{\partial x_2 \partial x_1} f(x) & \frac{\partial^2}{\partial x_2 \partial x_2} f(x) \end{pmatrix} = \begin{pmatrix} 6x_1 \sin(x_2) & 3x_1^2 \cos(x_2) \\ 3x_1^2 \cos(x_2) & -x_1^3 \sin(x_2) \end{pmatrix}.$$

Es ist zu beachten, dass

$$\frac{\partial^2}{\partial x_i \partial x_j} f(x) = \frac{\partial^2}{\partial x_j \partial x_i} f(x) \quad \text{für alle} \quad i, j \in \{1, \ldots, n\}.$$

Folglich ist die Hessematrix symmetrisch, d. h. $\mathrm{Hess}(f)(x)^T = \mathrm{Hess}(f)(x)$.

5.1.2 Lokale Extrema von $f : \mathbb{R}^n \to \mathbb{R}$

Mit Hilfe der Hessematrix können wird Satz 5.2 auf höherdimensionale Abbildungen $f : \mathbb{R}^n \to \mathbb{R}$ verallgemeinern.

Satz 5.5 *Sei $f : D \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion, wobei $D \subset \mathbb{R}^n$ eine offene Teilmenge des $\mathbb{R}^n$ ist.*

(i) Liegt bei $\bar{x} \in D$ ein lokales Extremum vor, so gilt $\nabla(f(\bar{x})) = 0 \in \mathbb{R}^n$.

(ii) Gilt zusätzlich zu (i), dass $\mathrm{Hess}(f)(\bar{x})$ negativ definit ist, so liegt bei $\bar{x}$ ein lokales Maximum vor.

(iii) Gilt zusätzlich zu (i), dass $\mathrm{Hess}(f)(\bar{x})$ positiv definit ist, so liegt bei $\bar{x}$ ein lokales Minimum vor.

Bemerkung 5.6 *Positiv bzw. negativ definite Matrizen wurden in Abschnitt 2.8 eingeführt.*

Man beachte, dass Satz 5.5 im Spezialfall $n = 1$ mit dem Satz 5.2 übereinstimmt. In diesem Fall gilt:

$$\nabla f(\bar{x}) = f'(\bar{x})$$

und

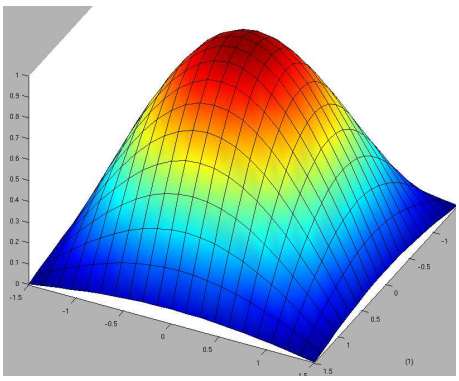
$$\mathrm{Hess}(f)(\bar{x}) = \frac{\partial^2}{\partial x \partial x} f(x) = f''(x).$$

Ist die 1×1 Matrix $f''(x)$ negativ definit, d. h. $yf''(\bar{x})y = y^2 f''(\bar{x}) < 0$ für alle $y \in \mathbb{R}$, $y \neq 0$, so folgt $f''(\bar{x}) < 0$.

5.2 Minimierungsprobleme

In diesem Abschnitt diskutieren wir die Lösung des Minimierungsproblems an einem Beispiel. Hierzu betrachten wir die Abbildung

$$f(x) = \cos(x_1)\cos(x_2). \tag{5.1}$$



Es gilt

$$\nabla f(x) = \begin{pmatrix} -\sin(x_1)\cos(x_2) \\ -\cos(x_1)\sin(x_2) \end{pmatrix}$$

und $\nabla f \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, folglich liegt bei $\bar{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ein Extremum vor.

Zur Analyse, ob ein Maximum oder ein Minimum vorliegt, betrachten wir die Hessematrix

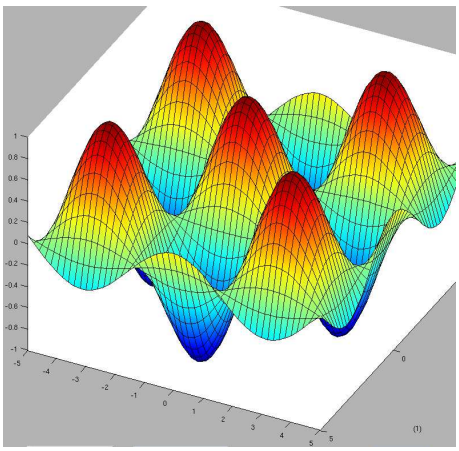
$$\begin{aligned} \mathrm{Hess}(f)(x) &= \begin{pmatrix} -\cos(x_1)\cos(x_2) & \sin(x_1)\sin(x_2) \\ \sin(x_1)\sin(x_2) & -\cos(x_1)\cos(x_2) \end{pmatrix}, \\ \mathrm{Hess}(f)(\bar{x}) &= \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned}$$

Es folgt, dass

$$y^T \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} y = -y^T y = -\|y\|_2^2 < 0 \quad \text{für alle } y \neq 0$$

und folglich ist $\mathrm{Hess}(f)(\bar{x})$ negativ definit und es liegt ein lokales Maximum vor.

Dieses Maximum ist nicht global, wie die folgende Abbildung zeigt.



Den Gradienten und die Hesse-Matrix können wir auch numerisch, unter Verwendung der Methoden aus Abschnitt 4.13 bestimmen. Eine Nullstelle des Gradienten kann numerisch mit dem Newton-Verfahren bestimmt werden, das in dieser Vorlesung nicht näher vorgestellt wird, vgl. [17].

Im obigen Beispiel war es einfach, die Hessematrix auf positive bzw. negative Definitheit zu testen.

Im Allgemeinen kann diese Eigenschaft einer Matrix nicht sofort angesehen werden. Da die Hessematrix symmetrisch ist, kann man zeigen, dass die Umkehrung von Lemma 2.17 gilt und somit positive (negative) Definitheit vorliegt, wenn die Hessematrix ausschließlich positive (negative) Eigenwerte besitzt.

Der Nachteil dieser Methode liegt auf der Hand: Numerisch ist die Berechnung aller Eigenwerte einer Matrix sehr aufwändig. Beachte: Unser Ansatz mit der Potenzmethode in Abschnitt 2.7.1 liefert nur den größten Eigenwert!

Einen einfachen Test zur Überprüfung einer symmetrischen $n \times n$-Matrix auf positive Definitheit liefert der folgende Satz, siehe [12].

Satz 5.7 *Sei $A \in \mathbb{R}^{n,n}$ eine symmetrische Matrix.*

- *A ist genau dann positiv definit, wenn alle führenden Hauptunterdeterminanten*

$$\det \begin{pmatrix} A_{11} & \dots & A_{1k} \\ \vdots & & \vdots \\ A_{k1} & \dots & A_{kk} \end{pmatrix} \neq 0, \quad k = 1, \dots, n,$$

  positiv sind.

- *A ist genau dann negativ definit, wenn die Matrix $-A$ positiv definit ist.*

5.2.1 Der Sintflutalgorithmus

Wir betrachten wieder das Minimierungsproblem aus Abschnitt 5.2. Für hochdimensionale Probleme ist es oft recht aufwändig, ein Maximum als Nullstelle des Gradienten zu bestimmen. Deshalb stellen wir ein einfaches heuristisches Verfahren zur Bestimmung eines Maximums vor, das keine Zusatzinformationen aus dem Gradienten verwendet, sondern durch Versuch und Irrtum das Maximum bestimmt.

Der Algorithmus zur Berechnung eines Maximums der Funktion $f : \mathbb{R}^n \to \mathbb{R}$ besitzt die folgende Form:

(1) Wähle einen Startvektor $x \in \mathbb{R}^n$ und eine Startschrittweite $\tau > 0$.

(2) Wähle einen Zufallsvektor $z \in \{y \in \mathbb{R}^n : |y_i| \leq 1 \; \forall i = 1, \ldots, n\}$.

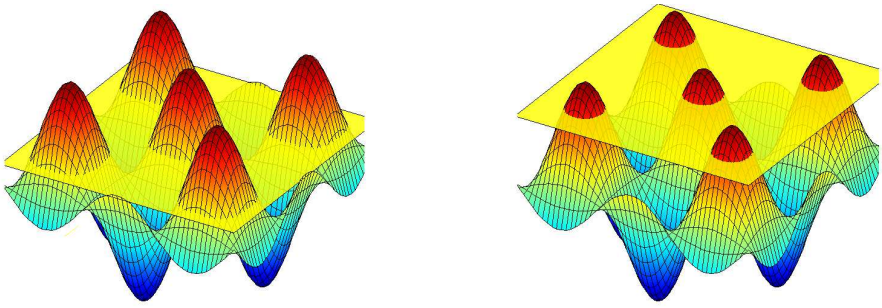
(3) Setze $y := x + \tau z$.

(4) Gilt, dass $f(y) > f(x)$?

(4a) Ja, dann setze $x := y$ und gehe zu Schritt (2).

(4b) Nein, dann behalte das alte x und gehe zu Schritt (2).

Man erkennt, dass nur ein Anstieg in Richtung des Maximums möglich ist, und eine einmal erreichte Höhe nie wieder unterschritten werden kann. Diese Eigenschaft begründet die Namensgebung „Sintflut-“ oder „Bergsteiger-Algorithmus“.

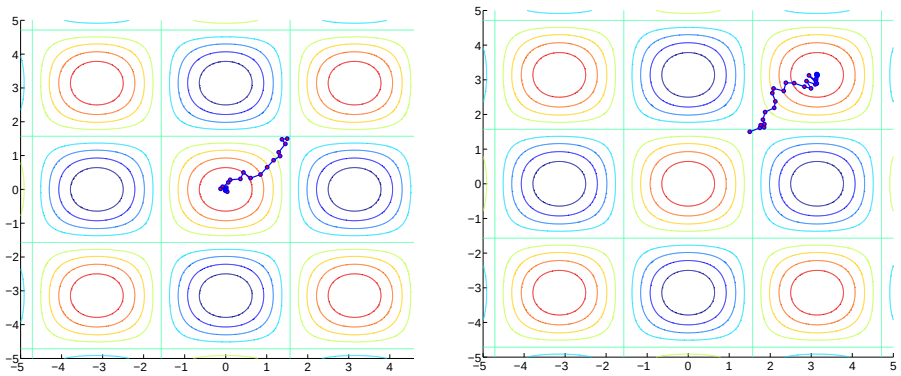


Folgendes ist bei der Umsetzung des Algorithmus zu beachten:

(a) Die Genauigkeit hängt von der gewählten Schrittweite τ ab. Ist diese zu groß, kann keine Verbesserung mehr erreicht werden. Somit ist es ratsam, τ zu verkleinern, wenn nach einer gewissen Anzahl von Schritten kein neues Maximum gefunden wurde.

(b) Auch eine Abbruchbedingung ist noch zu implementieren. Beispielsweise kann die Berechnung abgebrochen werden, wenn die Schrittweite aus (a) sehr weit heruntergesteuert wurde, etwa auf 10^{-10}.

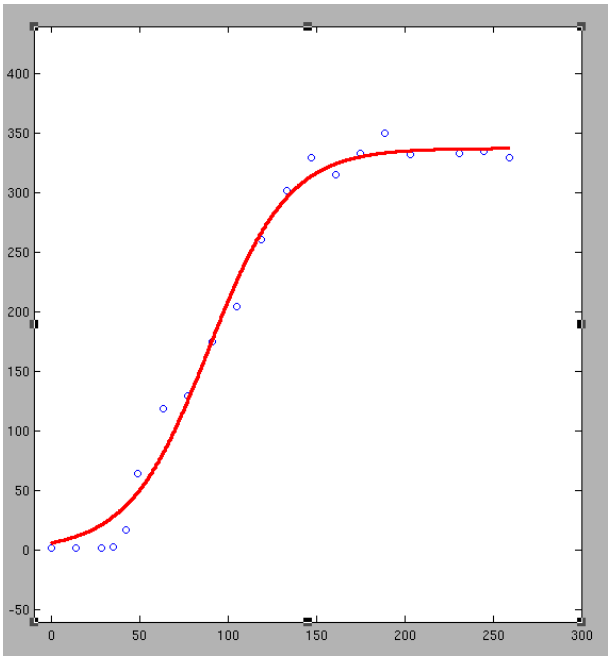
Die Abbildung zeigt die Niveaulinien unseres Beispiels (5.1) zusammen mit der Ausgabe des obigen Algorithmus zum Startpunkt $\begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}$. Wegen der Wahl von Zufallsvektoren ist a-priori nicht klar, welches Maximum der Algorithmus ansteuert.



Insbesondere erkennt man, dass dieser Ansatz im Allgemeinen kein globales Maximum findet. Hat man erst einen kleinen Berg bestiegen und ist von Wasser umgeben, kann man (ohne Modifikation des Algorithmus) nicht mehr auf den höheren Nachbargipfel wechseln.

5.3 Ausgleichsprobleme

Im Gegensatz zu der in Kapitel 4 betrachteten Interpolation von Daten ist es insbesondere bei Messdaten sinnvoll, eine Kurve zu suchen, die möglichst nahe an den Daten liegt, nicht aber durch jeden der gegebenen Punkte verläuft. Wird diese Methode angewendet, so ist der Einfluss von Messfehlern nicht so dramatisch und wird sogar ausgeglichen.



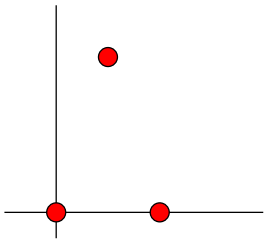
5.4 Fitten von 3 Punkten durch eine Gerade

Gegeben seien drei Punkte in der Ebene. Wir suchen eine Gerade, die möglichst nahe an den gegebenen Punkten liegt.

Als entscheidendes Hilfsmittel zum Lösen dieser Aufgabe werden wir die aus den Abschnitten 2.4, 2.13 und 2.14 bekannte QR-Zerlegung einer Matrix verwenden. Diese Zerlegung hat nicht nur eine geometrische Interpretation (Zerlegung einer linearen Abbildung in orthogonale Transformationen, Skalierungen und Scherungen), sie ist auch nützlich beim Fitten von Daten.

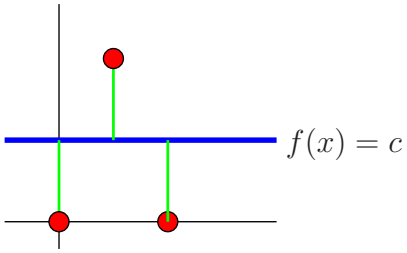
5.4.1 Wahl einer geeigneten Minimierungsmethode

Seien drei Punkte $(0,0)$, $(1,3)$ und $(2,0)$ gegeben.



Im einfachsten Fall suchen wir eine Gerade $f(x) = c$, die parallel zur x-Achse verläuft und möglichst nahe an den gegebenen Daten liegt. Hierbei ist jedoch zunächst nicht klar, wie der umgangssprachliche Ausdruck "möglichst nahe" zu formalisieren ist.

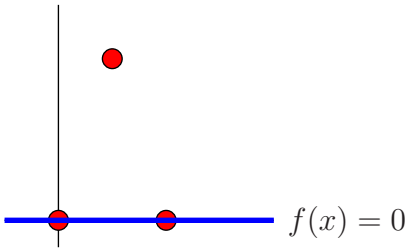
1. **Ansatz:** Minimiere die Summe der Fehler zwischen der Geraden und den gegebenen Punkten (d. h. die Summe der Längen der grünen Strecken in der folgenden Abbildung).



   Wir erhalten in unserem Beispiel

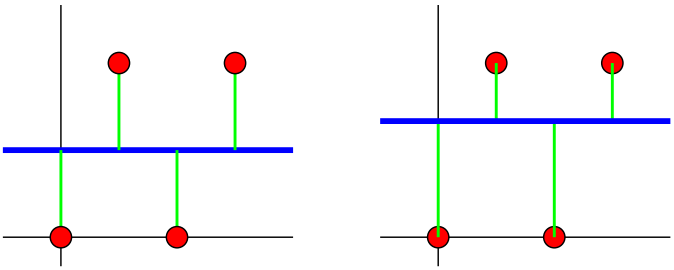
$$|f(0) - 0| + |f(1) - 3| + |f(2) - 0| = c + 3 - c + c = 3 + c,$$

   und das Minimum dieses Ausdrucks für $c \in [0,3]$, wird bei $c = 0$ angenommen, siehe Abbildung.



Offensichtlich ist dieses Ergebnis nicht zufriedenstellend, denn eine Gerade, die möglichst nahe an den gegebenen Punkten liegt, sollte vom mittleren Punkt nach oben gezogen werden.

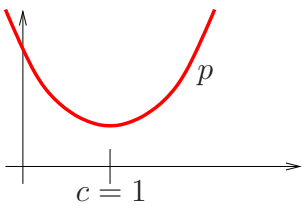
Weiter ist zu beachten, dass der erste Ansatz im Allgemeinen keine eindeutige Lösung besitzt und folglich unbrauchbar ist. Erweitern wir unser Beispiel um den vierten Punkt $(3, 3)$, so liefert jedes $c \in [0, 3]$ ein Minimum.



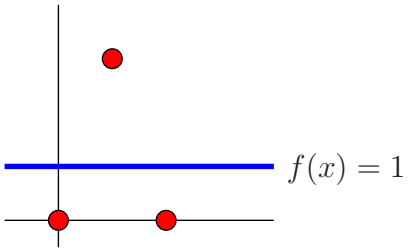
2. Ansatz: Jetzt minimieren wir die Summe der Fehlerquadrate (d. h. die Summe der Quadrate der Längen der grünen Strecken). Bei diesem Ansatz werden kleine Abweichungen toleriert, größere fallen jedoch stärker ins Gewicht. In unserem Beispiel gilt

$$|f(0)-0|^2+|f(1)-3|^2+|f(2)-0|^2 = c^2+(3-c)^2+c^2 = 3c^2-6c+9 =: p(c).$$

Die Parabel p besitzt ihr Minimum bei $c = 1$, siehe Abbildung.



Folglich erhalten wir im zweiten Ansatz die Gerade $f(x) = 1$. Wie die Abbildung zeigt, ist diese Gerade eine gute Lösung des Ausgleichsproblems.



Zusätzlich liefert dieser Ansatz eine eindeutige Lösung!

5.4.2 Minimierung der Summe der Fehlerquadrate

Wir suchen zunächst eine Gerade

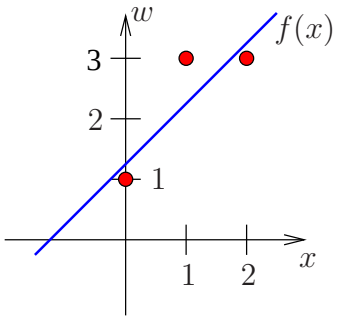
$$f(x) = a_1 x + a_0,$$

die möglichst nahe an den vorgegebenen Daten verläuft.

Konkret betrachten wir das Beispiel

$$(x_1, w_1) = (0, 1), \quad (x_2, w_2) = (1, 3), \quad (x_3, w_3) = (2, 3),$$

siehe Abbildung.



Eine Gerade, die durch alle Punkte verläuft, gibt es offensichtlich nicht. Sie müsste

$$f(x_i) = a_0 + a_1 x_i = w_i \quad \text{für } i = 1, 2, 3 \tag{5.2}$$

erfüllen. Dieses sind 3 Gleichungen durch die nur 2 Unbekannte bestimmt werden. Somit ist diese Aufgabe im Allgemeinen nicht lösbar.

Also minimieren wir wieder die **Summe der Fehlerquadrate**

$$F(a_0, a_1) := \big(f(x_1) - w_1\big)^2 + \big(f(x_2) - w_2\big)^2 + \big(f(x_3) - w_3\big)^2. \tag{5.3}$$

Die Gleichung (5.3) schreiben wir in der Form

$$\begin{aligned} F(a_0, a_1) &= \sum_{i=1}^{3} \big(f(x_i) - w_i\big)^2 \\ &= \sum_{i=1}^{3} \big(a_0 + a_1 x_i - w_i\big)^2 \\ &= \|z - w\|_2^2 \end{aligned} \tag{5.4}$$

mit den Vektoren

$$w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, \quad z = a_0 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + a_1 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = Xa.$$

Mit der 3×2 Matrix

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{pmatrix} \overset{\text{im Beispiel}}{=} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

und dem Vektor

$$a = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$$

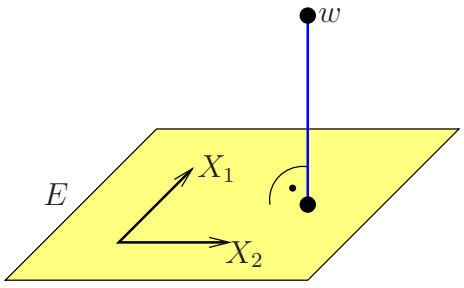
können wir (5.4) schreiben als

$$F(a) = \|Xa - w\|_2^2. \tag{5.5}$$

Dieses Minimierungsproblem kann auch graphisch interpretiert werden. Seien (X_1, X_2) die Spalten der Matrix X. Dann durchläuft Xa für $a \in \mathbb{R}^2$ die Ebene

$$E = \{a_0 X_1 + a_1 X_2 : a_0, a_1 \in \mathbb{R}\} \subset \mathbb{R}^3,$$

siehe Abbildung. Um $F(a)$ zu minimieren, müssen wir also den Punkt Xa in der Ebene E bestimmen, der von w den kleinsten Abstand besitzt.



Angenommen, wir haben eine QR-Zerlegung der nicht-quadratischen Matrix X gefunden:

$$\underbrace{X}_{3\times 2} = \underbrace{Q}_{3\times 3}\underbrace{R}_{3\times 2} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}\begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \end{pmatrix} \tag{5.6}$$

mit $r_{11} \neq 0 \neq r_{22}$, dann gilt

$$\|Xa - w\|_2^2 = \|QRa - QQ^T w\|_2^2 = \|Ra - Q^T w\|_2^2. \tag{5.7}$$

Ein Minimum liegt offensichtlich vor, wenn wir ein $a = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$ finden mit

$$\underbrace{\begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}}_{\tilde{R}} a = \begin{pmatrix} (Q^T w)_1 \\ (Q^T w)_2 \end{pmatrix}.$$

Das Minimum wird dann durch die dritte Zeile von (5.7) festgelegt:

$$F_{\min} = (Q^T w)_3^2.$$

Die Zerlegung (5.6) erhalten wir mit dem im Abschnitt 2.14.3 beschriebenen Householder-Verfahren. Es ist zu beachten, dass nur zwei statt 3 Spiegelungen durchzuführen sind, da die Matrix X nicht quadratisch ist.

Alternativ kann auch die in Abschnitt 2.14.2 beschriebene Gram-Schmidtsche-Methode angewandt werden. Dann bekommen wir das kleinere System

$$X = \tilde{Q}\tilde{R}, \quad \text{mit } \tilde{Q} = \begin{pmatrix} q_1 & q_2 \end{pmatrix} \in \mathbb{R}^{3,2}, \quad \tilde{Q}^T\tilde{Q} = I_{2\times 2}.$$

In diesem Fall kann aber eine dritte Spalte q_3 so gewählt werden, dass (5.6) wieder erfüllt ist. Also erhalten wir a durch Lösen von

$$\tilde{R}a = \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \langle q_1, w\rangle \\ \langle q_2, w\rangle \end{pmatrix} = \tilde{Q}^T w. \tag{5.8}$$

Durch Rückwärtsauflösen folgt

$$\begin{aligned} a_1 &= \frac{1}{r_{22}}\langle q_2, w\rangle, \\ a_0 &= \frac{1}{r_{11}}\big(\langle q_2, w\rangle - r_{12}a_1\big). \end{aligned}$$

Auch kann man (5.8) in eine Gleichung umschreiben, die nur von X, a und w abhängt:

$$Q^T QRa = \begin{pmatrix} \tilde{Q}^T w \\ 0 \end{pmatrix},$$

und die Multiplikation mit R^T liefert

$$R^T Q^T Q R a = \begin{pmatrix} \tilde{R}^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{Q}^T w \\ 0 \end{pmatrix} = R^T Q^T w,$$

also die sogenannte **Normalgleichung**

$$X^T X a = X^T w. \tag{5.9}$$

Wir kommen jetzt auf das eingangs angesprochene Beispiel zurück. Im Fall von 3 Punkten (x_i, w_i), $i = 1, 2, 3$ lautet die Normalengleichung

$$\begin{aligned} & \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \\ \iff & \begin{pmatrix} 3 & \sum_{i=1}^3 x_i \\ \sum_{i=1}^3 x_i & \sum_{i=1}^3 x_i^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^3 w_i \\ \sum_{i=1}^3 x_i w_i \end{pmatrix}. \end{aligned}$$

Mit den konkreten Zahlen erhalten wir folglich das lineare Gleichungssystem

$$\begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 7 \\ 9 \end{pmatrix}$$

und somit die Lösung

$$a_0 = \frac{4}{3}, \quad a_1 = 1.$$

Folglich wird die gesuchte **Regressionsgerade** durch

$$f(x) = \frac{4}{3} + x$$

definiert.

5.5 Fitten von n Punkten durch eine Gerade

Wird die Regressionsgerade durch n Punkte (x_i, w_i), $i = 1 \ldots, n$ gesucht, so kann der oben verfolgte Ansatz direkt übertragen werden. Wieder gilt die Gleichung (5.9), die ausgeschrieben die Form

$$\begin{aligned} & \begin{pmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} \\ \iff & \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n w_i \\ \sum_{i=1}^n x_i w_i \end{pmatrix} \end{aligned}$$

besitzt, d. h. die Koeffizienten $a_0,\ a_1$ erhält man durch Lösen eines linearen Gleichungssystems der Dimension 2.

5.6 Fitten von n Punkten durch ein Polynom vom Grad p

Abschließend suchen wir ein Polynom vom Grad $p < n-1$

$$f(x) = a_p x^p + a_{p-1}x^{p-1} + \cdots + a_1 x + a_0 = \sum_{j=0}^{p} a_j x^j,$$

das die gegebenen Punkte (x_i, w_i), $i = 1, \ldots, n$ möglichst gut approximiert.

Da eine Lösung a_j, $j = 0, \ldots, p$ des Interpolationsproblems

$$f(x_i) = w_i, \quad i = 1, \ldots, n$$

i. Allg. nicht existiert ($p < n-1$), betrachten wir das Minimierungsproblem

$$\begin{aligned} F(a_0, \ldots, a_p) &= \sum_{i=1}^{n} \big(f(x_i) - w_i\big)^2 \\ &= \sum_{i=1}^{n} \left(\sum_{j=1}^{p} a_j (x_i)^j - w_i \right)^2 \\ &= \|z - w\|_2^2 \end{aligned}$$

mit

$$\begin{aligned} w = \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}, \quad z &= a_0 \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + a_1 \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + \cdots + a_p \begin{pmatrix} x_1^p \\ \vdots \\ x_n^p \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & x_1 & \ldots & x_1^p \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \ldots & x_n^p \end{pmatrix}}_{X} \underbrace{\begin{pmatrix} a_0 \\ \vdots \\ a_p \end{pmatrix}}_{a} = Xa. \end{aligned}$$

Eine Minimallösung kann mit Hilfe der Normalgleichung (5.9)

$$X^T X a = X^T w$$

bestimmt werden. Ausgeschrieben erhalten wir das $p+1$ dimensionale

lineare Gleichungssystem

$$\begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \\ \vdots & & \vdots \\ x_1^p & \dots & x_n^p \end{pmatrix} \begin{pmatrix} 1 & x_1 & \dots & x_1^p \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^p \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \\ \vdots & & \vdots \\ x_1^p & \dots & x_n^p \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}$$

$$\begin{pmatrix} n & \sum x_i & \dots & \sum x_i^{p-1} & \sum x_i^p \\ \sum x_i & \sum x_i^2 & \dots & \sum x_i^p & \sum x_i^{p+1} \\ \vdots & \vdots & & \vdots & \vdots \\ \sum x_i^p & \sum x_i^{p+1} & \dots & \sum x_i^{2p-1} & \sum x_i^{2p} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \sum w_i \\ \sum x_i w_i \\ \vdots \\ \sum x_i^p w_i \end{pmatrix}.$$

An dieser Stelle soll nicht verschwiegen werden, dass das Ausgleichsproblem auch allgemeiner angesetzt werden kann. So kann die Ausgleichsfunktion mittels

$$f(x) := \sum_{j=0}^{p} a_j g_j(x)$$

definiert werden, wobei g_j geeignete Ansatzfunktionen sind.

In diesem Abschnitt wurde der Fall

$$g_j(x) = x^j, \quad j = 0, \dots, p$$

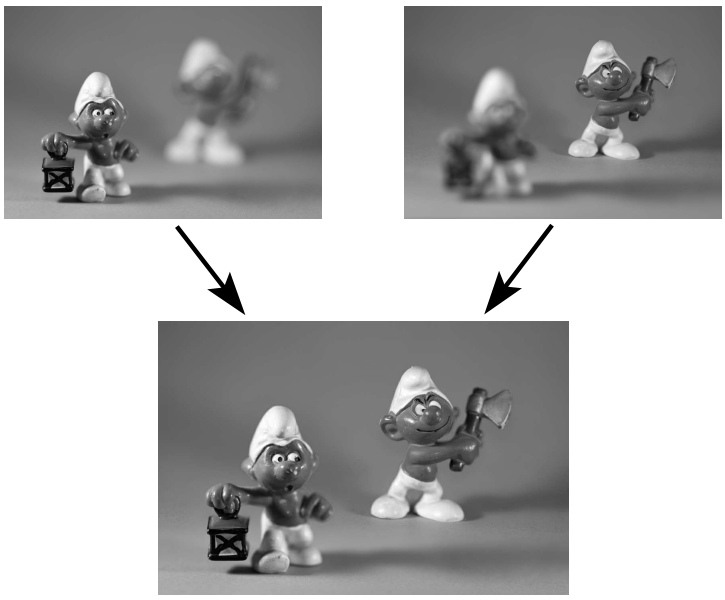
betrachtet.

Kapitel 6

Die schnelle Fourier-Transformation

Angenommen, ein diskretes Signal soll in sein Frequenzspektrum zerlegt werden, dann ist die schnelle Fourier-Transformation, kurz FFT (fast Fourier transform), die Methode der Wahl. Dieses Verfahren wurde 1965 von James Cooley und John W. Tukey veröffentlicht. Einen entsprechenden Algorithmus verwendete bereits 1805 Carl Friedrich Gauß zur Berechnung von Asteroiden-Flugbahnen.

Anwendungen sind beispielsweise das Entrauschen von Signalen, mp3- und jpg-Kompressionsverfahren[1] und sogenanntes Fokus-Stacking, das in der Makrofotografie ein Zusammensetzen von Bildern mit unterschiedlichen Schärfenebenen zu einem scharfen Bild erlaubt.



[1] Genauer wird hierbei die nahe verwandte diskrete Kosinus-Transformation verwendet.

6.1 Die Exponentialfunktion in den komplexen Zahlen

Bevor wir die Idee der diskreten Fourier-Transformation kennenlernen, werden grundlegende Aussagen über komplexe Zahlen und die e-Funktion angegeben.

- Sei $z \in \mathbb{C}$, $z = x + iy$ dann wird die konjugiert komplexe Zahl $\bar{z}$ definiert durch $\bar{z} = x - iy$.

  Seien $z_1, z_2 \in \mathbb{C}$, dann gelten die folgenden Rechenregeln:

$$\begin{aligned} \bar{z}_1 + \bar{z}_2 &= \overline{z_1 + z_2}, \\ \bar{z}_1 \cdot \bar{z}_2 &= \overline{z_1 \cdot z_2}. \end{aligned}$$

- Der Betrag einer komplexen Zahl wird definiert durch

$$|z| = \sqrt{z \cdot \bar{z}}.$$

  Es gilt

$$|z|^2 = z \cdot \bar{z} = (x + iy)(x - iy) = x^2 + y^2$$

  und wir erhalten folglich die euklidische Länge des Vektors $\begin{pmatrix} x \\ y \end{pmatrix}$.

- Die komplexe e-Funktion wird über die Reihendarstellung

$$\mathrm{e}^z = \sum_{j=0}^{\infty} \frac{z^j}{j!}, \quad z \in \mathbb{C}$$

  definiert.

  Es gilt: $\mathrm{e}^{\bar{z}} = \overline{e^z}$, denn

$$\mathrm{e}^{\bar{z}} = \sum_{j=0}^{\infty} \frac{\bar{z}^j}{j!} = \sum_{j=0}^{\infty} \frac{\overline{z^j}}{j!} = \overline{\sum_{j=0}^{\infty} \frac{z^j}{j!}} = \overline{e^z}.$$

- Sei $x \in \mathbb{R}$, dann gilt die Euler-Formel

$$\mathrm{e}^{ix} = \cos(x) + i \sin(x).$$

  Hierbei ist $\cos(x) = \mathrm{Re}(\mathrm{e}^{ix})$ und $\sin(x) = \mathrm{Im}(\mathrm{e}^{ix})$ eine zulässige Definition der reellen Kosinus- und Sinus-Funktion.

- Sei $x \in \mathbb{R}$, dann folgt

$$|\mathrm{e}^{ix}|^2 = \mathrm{e}^{ix} \cdot \overline{e^{ix}} = \mathrm{e}^{ix} \cdot \mathrm{e}^{\overline{ix}} = \mathrm{e}^{ix} \cdot \mathrm{e}^{-ix} = \mathrm{e}^{ix-ix} = \mathrm{e}^0 = 1.$$

- Weitere Rechenregeln sind:

$$\mathrm{e}^{2\pi i} = \cos(2\pi) + i \sin(2\pi) = 1 + i \cdot 0 = 1, \tag{6.1}$$

$$\mathrm{e}^{\pi i k} = \cos(\pi k) + i \sin(\pi k) = (-1)^k + i \cdot 0 = (-1)^k, \quad k \in \mathbb{Z}. \tag{6.2}$$

6.2 Interpolation mit Exponentialsummen

Wir verwenden den in Abschnitt 4.6 beschriebenen Interpolationsansatz, um gegebene Datenpaare (t_k, s_k), $k = 0, \ldots, m$ zu interpolieren.

Sei

$$p(t) = \sum_{j=0}^{m} a_j p_j(t) \tag{6.3}$$

mit Ansatzfunktionen

$$p_j(t) = \tfrac{1}{\sqrt{m+1}} \mathrm{e}^{2\pi i j t}.$$

Wie in Kapitel 4 sind die Koeffizienten $a_0, \ldots, a_m$ so zu bestimmen, dass die Interpolationsbedingung

$$p(t_k) = s_k, \quad k = 0, \ldots, m \tag{6.4}$$

erfüllt ist.

Im Folgenden seien die Stützstellen äquidistant verteilt:

$$t_k = \frac{k}{m+1}, \quad k = 0, \ldots, m. \tag{6.5}$$

Mit dieser Setzung besitzt die Interpolationsbedingung (6.4) die Form

$$\begin{aligned} s_k &= p(t_k) = \sum_{j=0}^{m} a_j \tfrac{1}{\sqrt{m+1}} \mathrm{e}^{2\pi i j t_k} = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \mathrm{e}^{2\pi i j \frac{k}{m+1}} \\ &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \left(\mathrm{e}^{\frac{2\pi i}{m+1}}\right)^{jk} = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j w^{jk}, \quad k = 0, \ldots, m \end{aligned}$$

mit $w = \mathrm{e}^{\frac{2\pi i}{m+1}}$.

Insgesamt erhalten wir das lineare Gleichungssystem

$$\begin{aligned} s_0 &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j w^0 = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j, \\ s_1 &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j w^j, \\ s_2 &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j w^{2j}, \\ &\vdots \\ s_m &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j w^{mj}, \end{aligned}$$

welches in Matrixform die Darstellung

$$\begin{pmatrix} s_0 \\ \vdots \\ \vdots \\ \vdots \\ s_m \end{pmatrix} = \frac{1}{\sqrt{m+1}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^m \\ 1 & w^2 & w^4 & \dots & w^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^m & w^{2m} & \dots & w^{m^2} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ \vdots \\ \vdots \\ a_m \end{pmatrix} \tag{6.6}$$

hat. In Kurzschreibweise besitzt (6.6) die Form

$$s = M_m \cdot a \tag{6.7}$$

mit

$$s = \begin{pmatrix} s_0 \\ \vdots \\ \vdots \\ \vdots \\ s_m \end{pmatrix}, \; M_m = \frac{1}{\sqrt{m+1}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^m \\ 1 & w^2 & w^4 & \dots & w^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^m & w^{2m} & \dots & w^{m^2} \end{pmatrix} \text{ und } a = \begin{pmatrix} a_0 \\ \vdots \\ \vdots \\ \vdots \\ a_m \end{pmatrix}. \tag{6.8}$$

Die gesuchten Koeffizienten a können somit durch Lösen dieses Systems bestimmt werden.

6.2.1 Reduktion des Rechenaufwands – Teil 1

Unter Ausnutzung der speziellen Struktur kann die Inverse von M_m sehr einfach berechnet werden.

Lemma 6.1 *Mit den obigen Bezeichungen gilt:*

$$\overline{M_m} \cdot M_m = I.$$

Beweis: Für die komplex konjugierte Matrix gilt

$$\overline{M_m} = \frac{1}{\sqrt{m+1}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \overline{w} & \overline{w}^2 & \dots & \overline{w}^m \\ 1 & \overline{w}^2 & \overline{w}^4 & \dots & \overline{w}^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \overline{w}^m & \overline{w}^{2m} & \dots & \overline{w}^{m^2} \end{pmatrix} \quad \text{mit} \quad \overline{w} = \mathrm{e}^{-\frac{2\pi i}{m+1}}.$$

Durch Ausschreiben der Matrix-Multiplikation[2] folgt für $j,k=0,\ldots,m$

$$\begin{aligned}\left(\overline{M_m}\cdot M_m\right)_{j+1,k+1} &= \sum_{n=1}^{m+1}(\overline{M_m})_{j+1,n}\cdot(M_m)_{n,k+1} = \tfrac{1}{m+1}\sum_{n=1}^{m+1}\overline{w}^{j(n-1)}w^{k(n-1)}\\ &= \tfrac{1}{m+1}\sum_{n=0}^{m}\overline{w}^{jn}w^{kn} = \tfrac{1}{m+1}\sum_{n=0}^{m}\mathrm{e}^{-\frac{2\pi ijn}{m+1}}\mathrm{e}^{\frac{2\pi ink}{m+1}}\\ &= \tfrac{1}{m+1}\sum_{n=0}^{m}\mathrm{e}^{\frac{2\pi i(k-j)n}{m+1}} = \tfrac{1}{m+1}\sum_{n=0}^{m}w^{(k-j)n}.\end{aligned}$$

Im Fall $k=j$ gilt

$$\left(\overline{M_m}\cdot M_m\right)_{k+1,k+1} = \tfrac{1}{m+1}\sum_{n=0}^{m}w^{(k-k)n} = \tfrac{1}{m+1}\sum_{n=0}^{m}1 = \tfrac{m+1}{m+1} = 1.$$

Im Fall $k\neq j$ liefert die Anwendung der Formel für die geometrische Summe $\sum_{n=0}^{m}a^n=\frac{1-a^{m+1}}{1-a}$ mit $a=w^{k-j}$

$$\begin{aligned}(m+1)\left(\overline{M_m}\cdot M_m\right)_{j+1,k+1} &= \sum_{n=0}^{m}w^{(k-j)n} = \frac{1-w^{(k-j)(m+1)}}{1-w^{(k-j)}}\\ &= \frac{1-\mathrm{e}^{\frac{2\pi i(k-j)(m+1)}{m+1}}}{1-w^{(k-j)}} = \frac{1-\mathrm{e}^{2\pi i(k-j)}}{1-w^{(k-j)}}\\ &= \frac{1-\left(\mathrm{e}^{2\pi i}\right)^{(k-j)}}{1-w^{(k-j)}} = \frac{1-1}{1-w^{(k-j)}} = 0,\end{aligned}$$

da nach (6.1) $\mathrm{e}^{2\pi i}=1$ erfüllt ist.

Zusammengefasst haben wir die Beziehung

$$\left(\overline{M_m}\cdot M_m\right)_{j+1,k+1} = \begin{cases}1, & \text{für } j=k,\\ 0, & \text{für } j\neq k\end{cases}$$

bewiesen und somit auch die Behauptung $\overline{M_m}\cdot M_m = I$.

■

Durch Multiplikation von (6.7) mit der Matrix $\overline{M_m}$ erhalten wir unter Verwendung von Lemma 6.1

$$\overline{M_m}\cdot s = \overline{M_m}\cdot M_m\cdot a = a. \tag{6.9}$$

Die Koeffizienten a lassen sich direkt mit einer Matrix-Vektor-Multiplikation berechnen; es muss **kein** lineares Gleichungssystem gelöst werden!

[2] $(M_m)_{j,k}$ bezeichnet den j,k-ten Eintrag der Matrix M_m.

Das System (6.9) besitzt ausgeschrieben die Darstellung

$$\tfrac{1}{\sqrt{m+1}}\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \overline{w} & \overline{w}^2 & \dots & \overline{w}^m \\ 1 & \overline{w}^2 & \overline{w}^4 & \dots & \overline{w}^{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \overline{w}^m & \overline{w}^{2m} & \dots & \overline{w}^{m^2} \end{pmatrix} \begin{pmatrix} s_0 \\ \vdots \\ \vdots \\ \vdots \\ s_m \end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ \vdots \\ \vdots \\ a_m \end{pmatrix}$$

und komponentenweise berechnet erhalten wir für $j = 0, \dots, m$ die Koeffizienten

$$a_j = \tfrac{1}{\sqrt{m+1}} \sum_{k=0}^{m} s_k \overline{w}^{kj} =: \mathcal{F}_j(s_0, \dots, s_m). \tag{6.10}$$

6.2.2 Reduktion des Rechenaufwands – Teil 2

Die folgende Überlegung erlaubt eine weitere Reduktion des Aufwands zur Berechnung von (6.10). Wir illustrieren die zugrunde liegende Idee im Fall $m = 7$.

Sei $j \in \{0, \dots, m\}$ fest gewählt. Wie werden sehen, dass die Berechnung von $\mathcal{F}_j(s_0, \dots, s_7)$ in zwei Teilprobleme aufteilbar ist:

$$\begin{array}{ccc} & \mathcal{F}_j(s_0, \dots s_7) & \\ \nearrow & & \nwarrow \\ \mathcal{F}_j(s_0, s_2, s_4, s_6) & & \mathcal{F}_j(s_1, s_3, s_5, s_7). \end{array}$$

Diese Teilprobleme lassen sich wieder in kleinere Probleme zerlegt:

$$\begin{array}{ccccccccccccccc} & & & & & & \mathcal{F}_j(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7) & & & & & & & & \\ & & & & & \nearrow & & & & \nwarrow & & & & & \\ & & & & \mathcal{F}_j(s_0, s_2, s_4, s_6) & & & & & & \mathcal{F}_j(s_1, s_3, s_5, s_7) & & & & \\ & & & \nearrow & & \nwarrow & & & & \nearrow & & \nwarrow & & & \\ & & \mathcal{F}_j(s_0, s_4) & & & & \mathcal{F}_j(s_2, s_6) & & & & \mathcal{F}_j(s_1, s_5) & & & F_j(s_3, s_7) & \\ & \nearrow & & \nwarrow & & \nearrow & & \nwarrow & & \nearrow & & \nwarrow & & \nearrow & & \nwarrow \\ \mathcal{F}_j(s_0) & & \mathcal{F}_j(s_4) & & \mathcal{F}_j(s_2) & & \mathcal{F}_j(s_6) & & \mathcal{F}_j(s_1) & & \mathcal{F}_j(s_5) & & \mathcal{F}_j(s_3) & & \mathcal{F}_j(s_7). \end{array}$$

Die letzten Einträge sind sehr leicht anzugeben, denn es gilt:

$$\mathcal{F}_j(s_\ell) = \tfrac{1}{\sqrt{1}} \sum_{k=0}^{0} s_\ell = s_\ell, \quad \text{für alle } \ell = 0, \dots, m.$$

Den durch die Pfeile im obigen Diagramm dargestellten Zusammenhang formalisiert die folgende Rechnung. Sei m ungerade (es liegt folglich eine gerade Anzahl von Datenpaaren vor). Setze $N = \frac{m+1}{2} - 1 = \frac{m-1}{2}$, dann gilt

$$\begin{aligned}
\mathcal{F}_j(s_0,\ldots,s_m) &= \tfrac{1}{\sqrt{m+1}} \sum_{k=0}^{m} s_k \overline{w}^{kj} \\
&= \tfrac{1}{\sqrt{m+1}} \sum_{k=0}^{m} s_k \mathrm{e}^{-\frac{2\pi i k j}{m+1}} \\
&= \tfrac{1}{\sqrt{m+1}} \left(\sum_{k=0,\ k \text{ gerade}}^{m} s_k \mathrm{e}^{-\frac{2\pi i k j}{m+1}} + \sum_{k=0,\ k \text{ ungerade}}^{m} s_k \mathrm{e}^{-\frac{2\pi i k j}{m+1}} \right) \\
&= \tfrac{1}{\sqrt{m+1}} \left(\sum_{k=0}^{\frac{m-1}{2}} s_{2k} \mathrm{e}^{-\frac{2\pi i (2k) j}{m+1}} + \sum_{k=0}^{\frac{m-1}{2}} s_{2k+1} \mathrm{e}^{-\frac{2\pi i (2k+1) j}{m+1}} \right) \\
&= \tfrac{1}{\sqrt{2(N+1)}} \left(\sum_{k=0}^{N} s_{2k} \mathrm{e}^{-\frac{2\pi i 2kj}{2(N+1)}} + \sum_{k=0}^{N} s_{2k+1} \mathrm{e}^{-\frac{2\pi i 2kj}{2(N+1)}} \mathrm{e}^{-\frac{2\pi i j}{2(N+1)}} \right) \\
&= \tfrac{1}{\sqrt{2}} \left(\tfrac{1}{\sqrt{N+1}} \sum_{k=0}^{N} s_{2k} \mathrm{e}^{-\frac{2\pi i k j}{N+1}} + \tfrac{1}{\sqrt{N+1}} \sum_{k=0}^{N} s_{2k+1} \mathrm{e}^{-\frac{2\pi i k j}{N+1}} \mathrm{e}^{-\frac{\pi i j}{N+1}} \right) \\
&= \tfrac{1}{\sqrt{2}} \left(\mathcal{F}_j(s_0, s_2, \ldots, s_{m-1}) + \mathrm{e}^{-\frac{\pi i j}{N+1}} F_j(s_1, s_3, \ldots, s_m) \right).
\end{aligned}$$

Der Rechenaufwand kann, durch Ausnutzung der Identität[3]

$$\mathcal{F}_{N+1+j}(s_0,\ldots,s_m) = \tfrac{1}{\sqrt{2}} \left(\mathcal{F}_j(s_0, s_2, \ldots, s_{m-1}) - \mathrm{e}^{-\frac{\pi i j}{N+1}} \mathcal{F}_j(s_1, s_3, \ldots, s_m) \right)$$

für $j = 0, \ldots, N$, halbiert werden.

6.3 FFT – ein rekursiver Algorithmus

Der Lohn der bisherigen Vorarbeiten ist ein schneller und sehr effizienter Algorithmus – die sogenannte schnelle Fourier-Transformation – zur Berechnung der Parameter $a_0, \ldots, a_m$. Im Folgenden geben wir eine leicht zu programmierende rekursive Version dieses Algorithmus an.[4]

Gegeben seien $m + 1$ Datenpaare $s = (s_0, \ldots, s_m)$, wobei $m + 1$ eine Zweierpotenz ist.

[3] Der Beweis ist eine Übungsaufgabe.

[4] Eine iterative Version dieses Algorithmus ist wesentlich effizienter, aber auch viel technischer in der Darstellung.

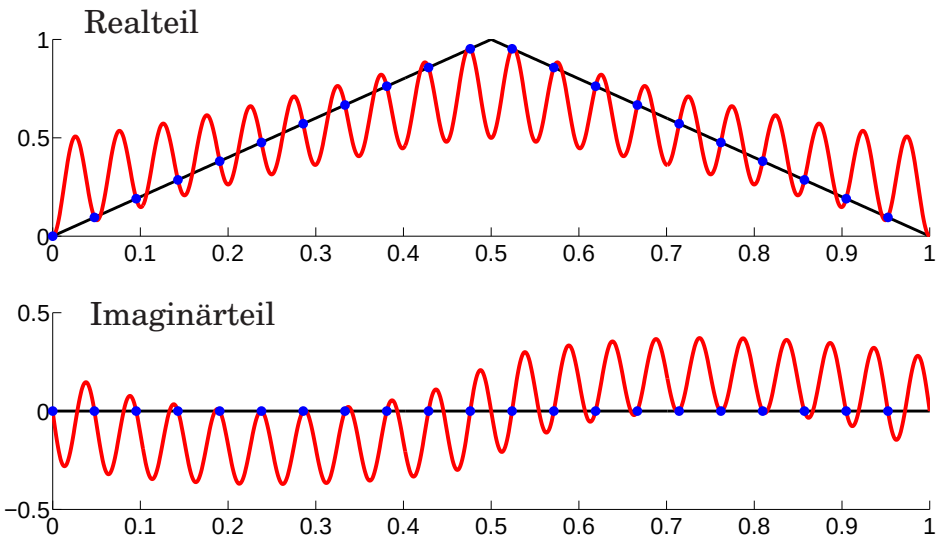
```
function FFT(m, s)
    if m = 0
        return s
    else
        g = FFT((m−1)/2, (s_0, s_2, ..., s_{m−1}))
        h = FFT((m−1)/2, (s_1, s_3, ..., s_m))
        for j = 0, ..., (m−1)/2
            a_j = 1/√2 ( g_j + h_j · e^{−πij/((m+1)/2)} )
            a_{j+(m+1)/2} = 1/√2 ( g_j − h_j · e^{−πij/((m+1)/2)} )
        return a
```

6.4 Auftreten starker Oszillationen

Gegeben sei die Dreiecks-Funktion

$$f(x) = \begin{cases} 2x, & \text{für } 0 \le x \le \frac{1}{2}, \\ 2 - 2x & \text{für } \frac{1}{2} < x \le 1. \end{cases}$$

Für $m = 20$ zeigt die Abbildung Real- und Imaginärteil der Funktion f (schwarz), die 21 Stützstellen (blau) und die Interpolation mit der Exponentialsumme (6.3) (rot).



Den Grund für das Auftreten dieser Oszillationen diskutieren wir in Abschnitt 6.6. Zunächst beschreiben wir einen Ansatz zur Vermeidung dieses Problems, der einen modifizierten Interpolationsansatz verwendet.

Bisher betrachteten wir die Exponentialsumme

$$p(t) = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \mathrm{e}^{2\pi i j t} \tag{6.11}$$

die jetzt wie folgt abgeändert wird:

$$r(t) = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \mathrm{e}^{2\pi i \left(j - \frac{m+1}{2}\right) t}. \tag{6.12}$$

Der FFT-Algorithmus kann unverändert zur Bestimmung der Koeffizienten $a_0, \ldots, a_m$ verwendet werden, denn zwischen p und r besteht folgender Zusammenhang

$$r(t) = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \mathrm{e}^{2\pi i j t} \cdot \mathrm{e}^{-2\pi i \frac{m+1}{2} t} = p(t) \cdot \mathrm{e}^{-\pi i (m+1) t}.$$

Durch Einsetzen von $t_k = \frac{k}{m+1}$ für $k = 0, \ldots, m$ ergibt sich

$$\begin{aligned} r(t_k) &= p(t_k) \cdot \mathrm{e}^{-\pi i (m+1) t_k} = p(t_k) \cdot \mathrm{e}^{-\pi i (m+1) \frac{k}{m+1}} = p(t_k) \cdot \mathrm{e}^{-\pi i k} \\ &= p(t_k) \cdot (-1)^k, \end{aligned}$$

da nach (6.2) $\mathrm{e}^{-\pi i k} = (-1)^k$ gilt.

Zusammengefasst sehen wir, dass zur Berechnung von

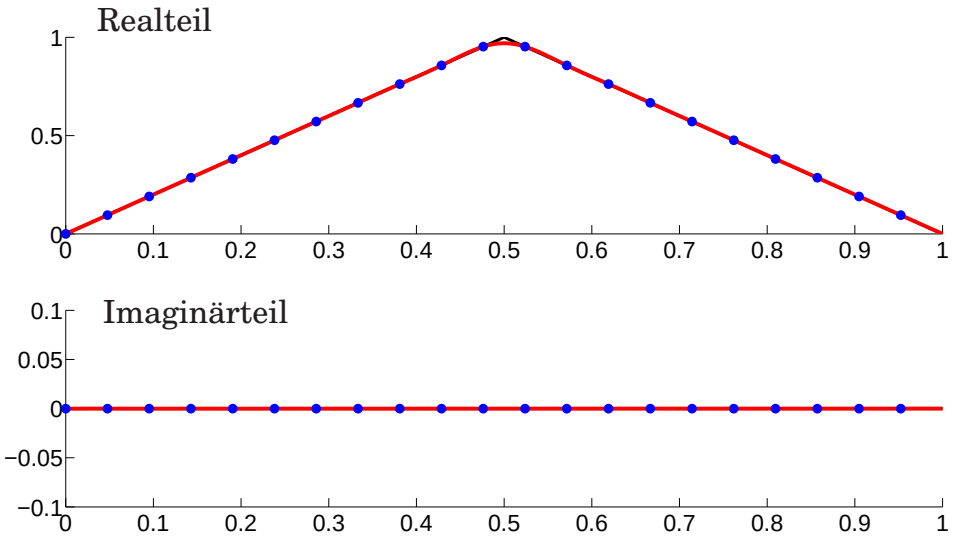
$$r(t_k) = s_k, \quad k = 0, \ldots, m$$

die Interpolationsaufgabe $r(t_k) = s_k = p(t_k) \cdot (-1)^k$ und somit

$$p(t_k) = (-1)^k \cdot s_k, \quad k = 0, \ldots, m \tag{6.13}$$

zu lösen ist. Nach diesen Umformungen ist der FFT-Algorithmus direkt anwendbar; es muss nur die rechte Seite mit $(-1)^k$ multipliziert werden.

Das Ergebnis dieser Berechnungen im obigen Beispiel zeigt die folgende Abbildung.



6.5 Zusammenfassung und Kurzschreibweise

Zusammengefasst erhalten wir zu den Datenpaaren $(t_k = \frac{k}{m+1}, s_k)$, $k = 0, \ldots, m$ die Koeffizienten $a_0, \ldots, a_m$

- von p, siehe (6.11), durch Berechnung von[5,6]

$$a = \mathbf{FFT}(s) = \overline{M}_m \cdot s.$$

- von r, siehe (6.12), durch Berechnung von

$$a = \mathbf{FFT}_{\text{opt}}(s) := \mathbf{FFT}(V \cdot s) = \overline{M}_m \cdot V \cdot s. \tag{6.14}$$

  Hierbei bezeichnet

$$V = \begin{pmatrix} 1 & & & & \\ & -1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & -1 \end{pmatrix}$$

  die Vorzeichenmatrix, die den in (6.13) beschriebenen Zusammenhang zwischen p und r realisiert.

[5] Die folgenden Abkürzungen wurden in Formel (6.8) eingeführt.

[6] Formal ist die Darstellung $\mathrm{FFT}(s) = \overline{M}_m \cdot s$ korrekt, effizient berechnet wird die schnelle Fourier-Transformation mit dem Algorithmus aus Abschnitt 6.3.

6.6 Interpretation der Koeffizienten

Wir betrachten die Koeffizienten der Exponentialsumme (6.12) am Beispiel

$$f(t) = \cos(2\pi t), \quad t \in [0,1]. \tag{6.15}$$

Wird zur Interpolation der Ansatz

$$p(t) = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \mathrm{e}^{2\pi i j t} = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \left(\cos(2\pi j t) + i \sin(2\pi j t)\right)$$

verwendet, erkennt man, dass a_0 die Amplitude der langsamsten Schwingung ist, wogegen a_m die Amplitude der höchstfrequenten Schwingung beschreibt.

Wir bestimmen für ein $m \in \mathbb{N}$, und Stützstellen $t_k = \frac{k}{m+1}$ die Koeffizienten a_j, $j = 0, \ldots, m$ so, dass diese die Interpolationsbedingung

$$f(t_k) = \cos\left(\tfrac{2\pi k}{m+1}\right) = p(t_k) = \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \left(\cos\left(\tfrac{2\pi j k}{m+1}\right) + i \sin\left(\tfrac{2\pi j k}{m+1}\right)\right) \tag{6.16}$$

für $k = 0, \ldots, m$ erfüllen.

Unter Verwendung der Gleichungen

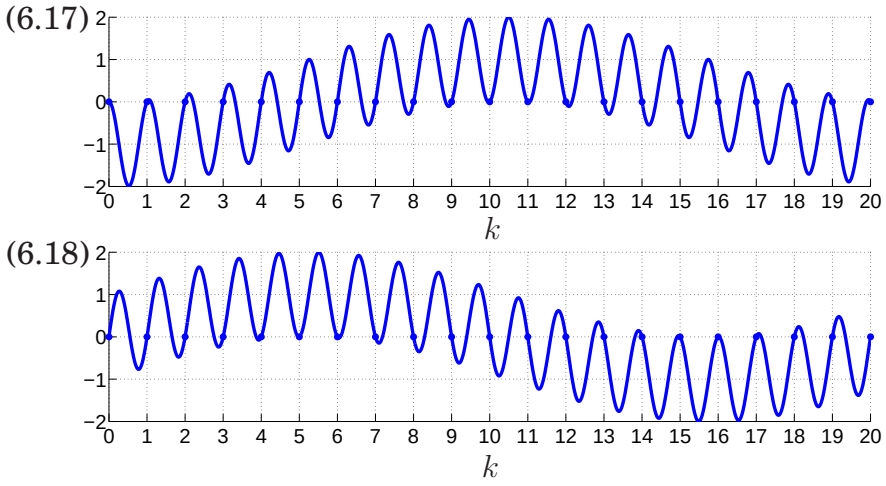
$$0 = \cos\left(\tfrac{2\pi k}{m+1}\right) - \cos\left(\tfrac{2\pi k m}{m+1}\right), \tag{6.17}$$

$$0 = \sin\left(\tfrac{2\pi k}{m+1}\right) + \sin\left(\tfrac{2\pi k m}{m+1}\right), \quad k = 0, \ldots, m \tag{6.18}$$

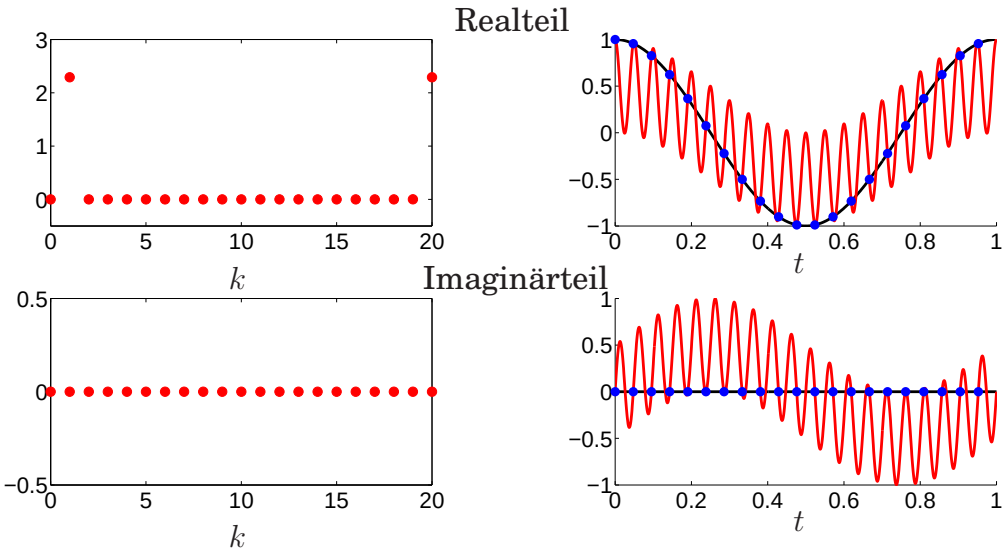
folgt die Setzung

$$a_j = \begin{cases} \frac{\sqrt{m+1}}{2}, & \text{für } j \in \{1, m\}, \\ 0, & \text{sonst} \end{cases}$$

als eindeutige Wahl der Koeffizienten von (6.16). Hierbei ist zu beachten, dass (6.17) und (6.18) nur für ganzzahlige k gelten, wie die folgende Abbildung im Fall $m = 20$ zeigt.



Die Exponentialsumme besitzt Anteile in der langsamsten Schwingungsrichtung, aber auch ein hochfrequenter Anteil liegt vor! Dieser Anteil führt bei einer Auswertung von p an Zwischenstellen zu starken Oszillationen. Die Abbildung zeigt links die Real- und Imaginärteile der berechneten Koeffizienten von p. Das rechte Bild illustriert die Graphen von f (schwarz) und p (rot) sowie die Stützstellen (blau).



Beim verbesserten Ansatz (6.3) verwenden wir den Interpolationsansatz

$$\begin{aligned} r(t) &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \mathrm{e}^{2\pi i \left(j - \frac{m+1}{2}\right) t} \\ &= \tfrac{1}{\sqrt{m+1}} \sum_{j=0}^{m} a_j \left(\cos(2\pi \left(j - \tfrac{m+1}{2}\right) t) + i \sin(2\pi \left(j - \tfrac{m+1}{2}\right) t)\right) \end{aligned} \tag{6.19}$$

und erkennen, dass die Amplitude der langsamsten Schwingung durch den Koeffizienten $a_{\frac{m+1}{2}}$ beschrieben wird, falls m ungerade ist. Die Amplituden der höchstfrequenten Schwingungen beschreiben die Koeffizienten a_0 und a_m.

Betrachten wir erneut das Beispiel (6.15) und wählen ein ungerades $m \in \mathbb{N}$, so liefert die Setzung

$$a_j = \begin{cases} \frac{\sqrt{m+1}}{2}, & \text{für } j \in \left\{\frac{m+1}{2} - 1, \frac{m+1}{2} + 1\right\}, \\ 0, & \text{sonst,} \end{cases} \tag{6.20}$$

wie die folgende Rechnung zeigt, eine exakte Interpolation, d. h.

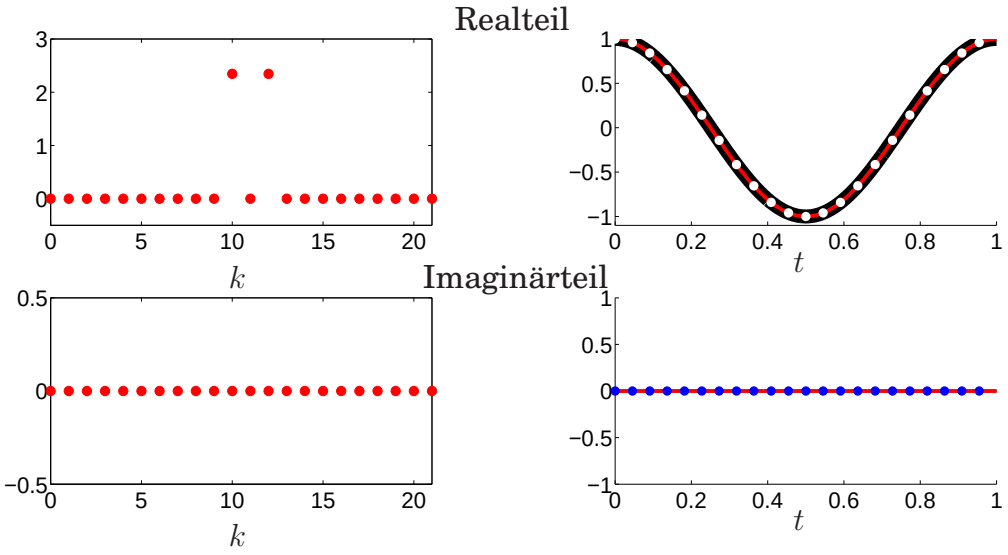
$$r(t) = f(t) \quad \text{für alle } t \in \mathbb{R}. \tag{6.21}$$

Dieses ist eine wesentliche Verbesserung des ersten Ansatzes, in dem nur die Stützstellen $t_k = \frac{k}{m+1}$, $k = 0, \ldots, m$ exakt interpoliert werden, an allen anderen Stellen liegen – wegen der Oszillationen – Interpolationsfehler vor.

Durch Einsetzen von (6.20) in (6.19) folgt (6.21):

$$\begin{aligned} r(t) &= \frac{1}{\sqrt{m+1}}\Big(\frac{\sqrt{m+1}}{2}\Big[\cos\Big(2\pi\Big(\frac{m+1}{2}-1-\frac{m+1}{2}\Big)t\Big)+i\sin\Big(2\pi\Big(\frac{m+1}{2}-1-\frac{m+1}{2}\Big)t\Big)\Big] \\ &\quad +\frac{\sqrt{m+1}}{2}\Big[\cos\Big(2\pi\Big(\frac{m+1}{2}+1-\frac{m+1}{2}\Big)t\Big)+i\sin\Big(2\pi\Big(\frac{m+1}{2}+1-\frac{m+1}{2}\Big)t\Big)\Big]\Big) \\ &= \tfrac{1}{2}\big[\cos(-2\pi t)+i\sin(-2\pi t)\big]+\tfrac{1}{2}\big[\cos(2\pi t)+i\sin(2\pi t)\big] \\ &= \cos(2\pi t) = f(t) \quad \text{für alle } t\in\mathbb{R}. \end{aligned}$$

Das Ergebnis der numerischen Rechnung für $m = 21$ zeigt die folgende Abbildung. Im rechten Bild ist die gegebene Funktion f (schwarz) mit hoher Linienbreite eingezeichnet, da diese sonst von der Interpolationsfunktion r (rot) vollständig verdeckt wird (es tritt bekanntlich in diesem Beispiel kein Approximationsfehler auf).



6.7 Die zweidimensionale Fourier-Transformation

Sei $S \in R^{m+1,m+1}$ eine Matrix, deren Einträge z. B. die Grauwerte eines schwarz-weiß-Bildes beschreiben. Das Frequenzspektrum dieser Daten kann mit der zweidimensionalen Fourier-Transformation bestimmt werden.

Zuerst definieren wir die Fourier-Transformation der Matrix S (unter Beibehaltung der Notation FFT) mittels

$$\mathbf{FFT}\begin{pmatrix} S_{0,0} & \dots & S_{0,m} \\ \vdots & \ddots & \vdots \\ S_{m,0} & \dots & S_{m,m} \end{pmatrix} = \begin{pmatrix} \mathbf{FFT}\begin{pmatrix} S_{0,0} \\ \vdots \\ S_{m,0} \end{pmatrix} & \dots & \mathbf{FFT}\begin{pmatrix} S_{0,m} \\ \vdots \\ S_{m,m} \end{pmatrix} \end{pmatrix}.$$

Die zweidimensionale Fourier-Transformation ist durch die Formel

$$\mathbf{FFT}_2(S) := \mathbf{FFT}(\mathbf{FFT}(S)^T)^T$$

definiert. Folglich gilt die folgende Darstellung[7]

$$\begin{aligned} \mathbf{FFT}_2(S) &= \mathbf{FFT}((\overline{M_m}S)^T)^T = \mathbf{FFT}(S^T\overline{M_m})^T \\ &= (\overline{M_m}S^T\overline{M_m})^T = \overline{M_m}S\overline{M_m}. \end{aligned}$$

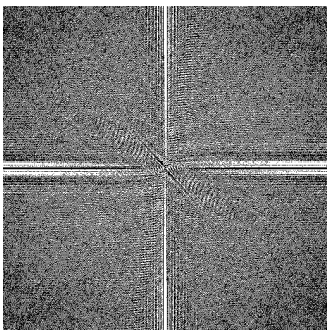
Wieder werden Artefakte durch Verwendung von $\mathbf{FFT}_{\text{opt}}$ vermieden, siehe (6.14). Es gilt[8]:

$$\begin{aligned} \mathbf{FFT}_{2\text{opt}}(S) &:= \mathbf{FFT}_{\text{opt}}(\mathbf{FFT}_{\text{opt}}(S)^T)^T \\ &= \mathbf{FFT}_{\text{opt}}(\overline{M_m}VS)^T)^T = \mathbf{FFT}_{\text{opt}}(S^TV\overline{M_m})^T \\ &= (\overline{M_m}VS^TV\overline{M_m})^T = \overline{M_m}VSV\overline{M_m} \\ &= \mathbf{FFT}_2(VSV). \end{aligned}$$

Die optimierte Version der zweidimensionalen diskreten Fourier-Transformation kann mit Hilfe der FFT-Funktion berechnet werden.

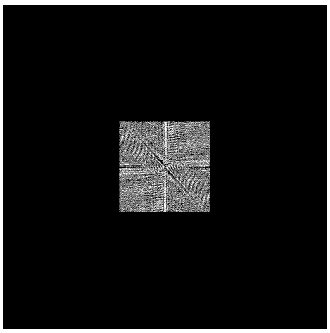
6.7.1 Anwendung: Komprimierung von Bildern

In der Bildverarbeitung kann die Fourier-Transformation zur Bildkompression verwendet werden, indem zum Beispiel hochfrequente Anteile vernachlässigt werden. Die Abbildung zeigt links das Originalfoto und dessen Fourier-Koeffizienten, berechnet mit $\mathbf{FFT}_{2\text{opt}}$.



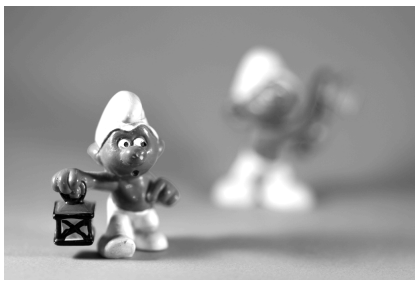
[7] Beachte, dass $\overline{M_m}^T = \overline{M_m}$ gilt.
[8] Beachte, dass $V^T = V$ gilt.

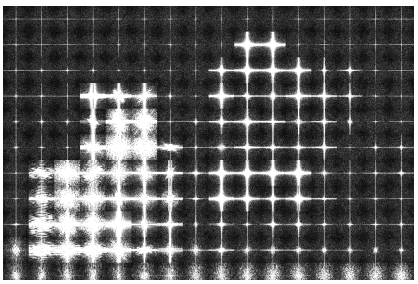
In der folgenden Abbildung wurden die hohen Frequenzen entfernt und ein stark komprimiertes Bild erzeugt.

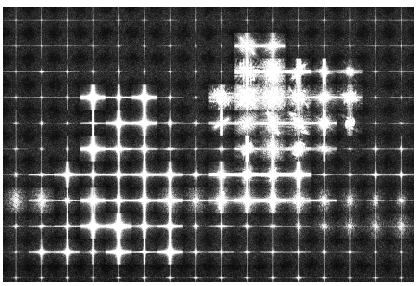


6.7.2 Anwendung: Fokus-Stacking

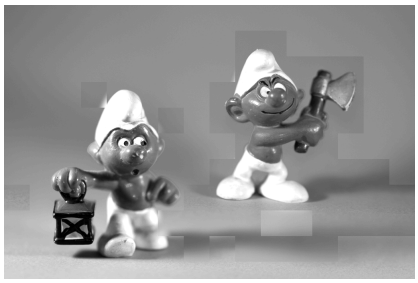
Auch das in der Einleitung beschriebene Fokus-Stacking kann mit der zweidimensionalen Fourier-Transformation realisiert werden. Zunächst unterteilen wir die beiden Bilder in Felder der Größe 64×64 und berechnen für diese Teilbereiche jeweils die Fourier-Koeffizienten unter Verwendung von $\mathrm{FFT}_{2\mathrm{opt}}$.

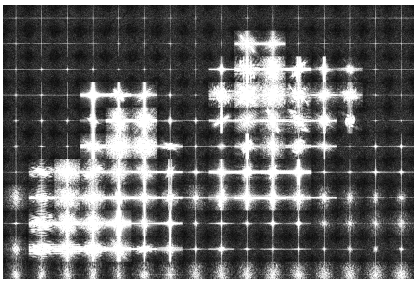






Das Auftreten von hochfrequenten Schwingungen werten wir als Indikator für eine scharfe Abbildung. Die entsprechenden Bereiche setzen wir zu einem Bild zusammen.

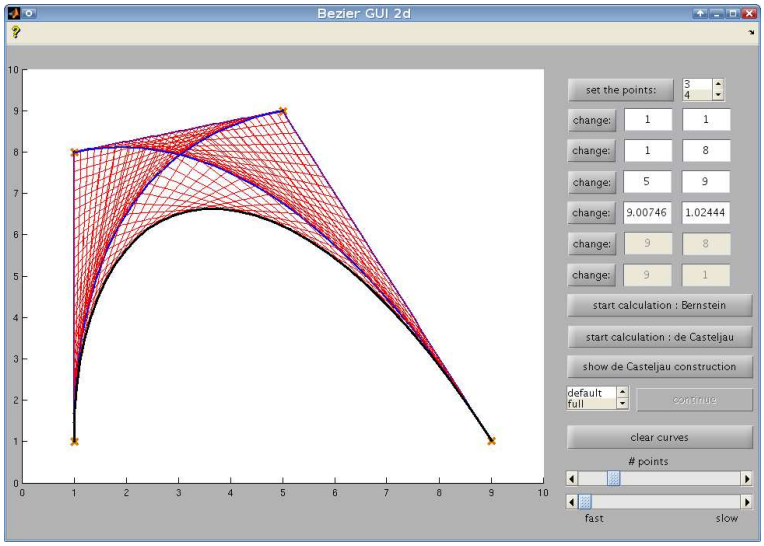




Die hier beschriebenen Ansätze besitzen noch viel Potential für Verbesserungen, beruhen aber auf der diskreten Fourier-Transformation, die die Grundlage vielfältiger Algorithmen ist.

Kapitel 7

Bézier-Kurven und Bézier-Flächen



Mit zunehmender Bedeutung der Computergrafik wurden in der Industrie Verfahren zur Darstellung von Kurven und Flächen benötigt, die es erlaubten, Objekte schnell zu zeichnen und zu manipulieren.

In diesem Abschnitt beschäftigen wir uns mit der Berechnung glatter Kurven – den sogenannten Bézier-Kurven – die durch wenige Punkte definiert werden, aber im Gegensatz zur Interpolation nur durch den Anfangs- und Endpunkt verlaufen. Dieser Ansatz wird anschließend auf die Berechnung von Bézier-Flächen verallgemeinert. Als weiterführende Literatur empfehlen sich [10] und [8].

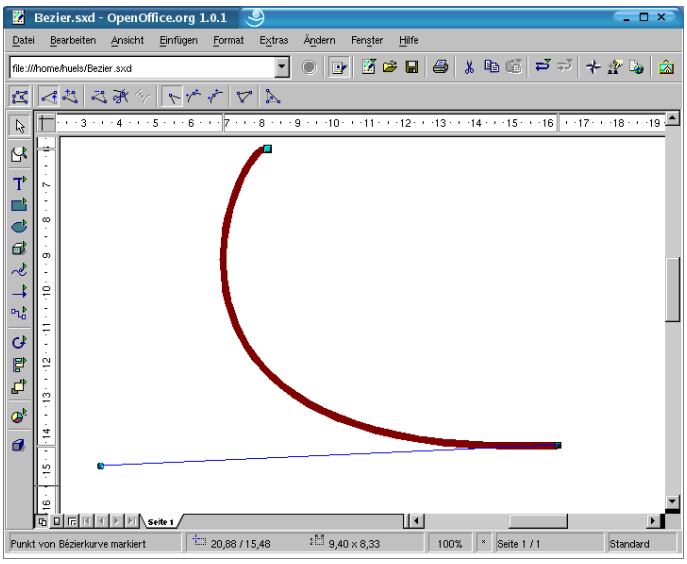
Zuerst wurden diese Kurven bzw. Flächen von Paul de Faget de Casteljau (geb. am 19. November 1930) entwickelt, der 1958 von dem Automobilhersteller Citroën eingestellt wurde. Seine Aufgabe bestand darin, eine mathematische Beschreibung einer Oberfläche – z. B. von der

Karosserie eines Autos – zu entwickeln. Zu jener Zeit stellte man in der Automobilindustrie von den Zeichnungen der Designer zunächst ein *Master-Modell* der Oberfläche her, das als Grundlage für die Serienproduktion diente. Aber die Qualität eines Master-Modells hing sehr von den Fähigkeiten der Modellschreiner und der Formsetzer ab, und die Abweichungen zwischen den Zeichnungen und den Modellen führten häufig zu Diskussionen und Produktionsverzögerungen. Der Produktionsprozess wurde revolutioniert, als de Casteljau eine einfache mathematische Beschreibung für diese Oberflächen fand. Allerdings wurden diese Ergebnisse von Citroën zunächst geheim gehalten und erst 1967 veröffentlicht, vgl. auch den autobiographischen Artikel von de Casteljau [7].

Bei Renault war der Ingenieur Pierre Étienne Bézier (geb. am 1. September 1910, gestorben am 25. November 1999), ebenfalls mit der Entwicklung einer mathematischen Beschreibung beschäftigt. Er entwickelte unabhängig von de Casteljau ein ähnliches mathematisches Konzept. Da Renault eine sofortige Veröffentlichung dieser Ergebnisse erlaubte, wurden die Kurven nach Bézier benannt.

7.1 Bézier-Kurven

Bézier-Kurven sind in Grafikprogrammen ein nützliches Tool, wenn z. B. eine Parabel zu zeichnen ist, siehe Abbildung.



Im folgenden Abschnitt geben wir einen Algorithmus zur Konstruktion einer Parabel aus drei gegebenen Punkten an. Mit einer Verallgemeinerung dieses Ansatzes erhalten wir dann die Bézier-Kurven.

7.1.1 Parabeln

Gegeben seien drei Punkte b_0, b_1 und b_2 im $\mathbb{R}^2$. Definiere die Geraden durch b_0, b_1 bzw. durch b_1, b_2 mittels

$$\begin{aligned} b_0^1(t) &:= (1-t)b_0 + tb_1, \\ b_1^1(t) &:= (1-t)b_1 + tb_2. \end{aligned}$$

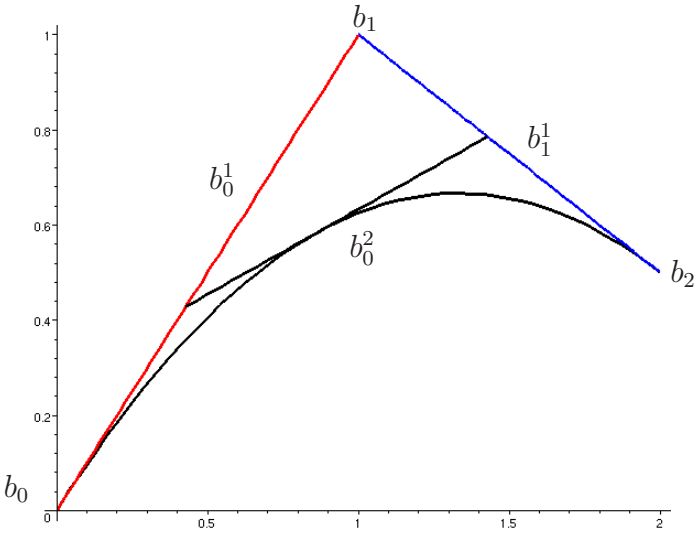
Bilde dann

$$\begin{aligned} b_0^2(t) &:= (1-t)b_0^1(t) + tb_1^1(t) \\ &= (1-t)\big((1-t)b_0 + tb_1\big) + t\big((1-t)b_1 + tb_2\big) \\ &= (1-t)^2 b_0 + 2(1-t)tb_1 + t^2 b_2. \end{aligned}$$

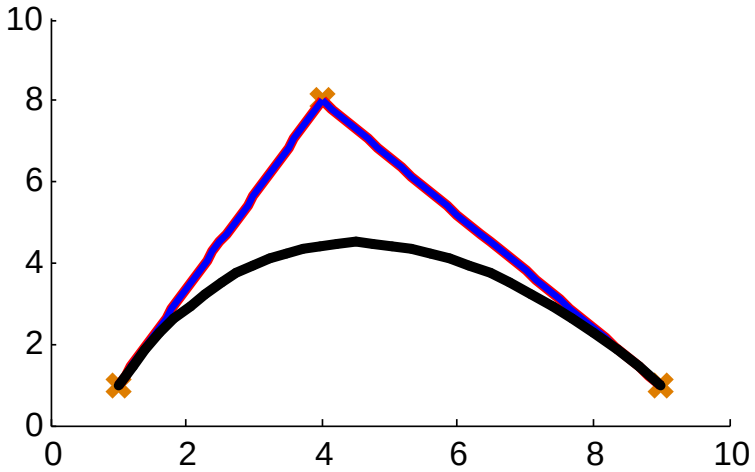
Diese Funktion ist ein quadratischer Ausdruck in t; in der Tat wird hierdurch die gesuchte Parabel beschrieben. Beachte, dass durch den oberen Index immer der Grad des Polynoms angegeben wird. Die Koeffizienten $b_i^r(t)$ können, wie bei den dividierten Differenzen, wieder mit einem Dreiecksschema beschrieben werden:

$$\begin{array}{ccccc} b_0 & \rightarrow & b_0^1 & \rightarrow & b_0^2 \\ & \nearrow & & \nearrow & \\ b_1 & \rightarrow & b_1^1 & & \\ & \nearrow & & & \\ b_2 & & & & \end{array}$$

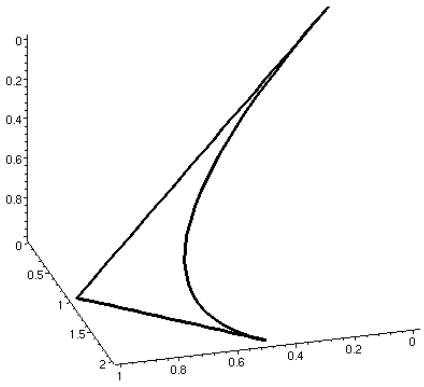
Eine Veranschaulichung dieser Konstruktion, die auch als wiederholte lineare Interpolation bezeichnet wird, zeigt die Abbildung



und die Animation.



Dieser Algorithmus ist nicht auf den $\mathbb{R}^2$ beschränkt, in der Tat ist die obige Konstruktion genau so im $\mathbb{R}^n$ ausführbar. Die Grafik zeigt die Berechnung im $\mathbb{R}^3$. An diesem Beispiel ist auch gut erkennbar, dass die berechnete Parabel keine echte Raumkurve ist; sie liegt in der durch die Vektoren $b_0 - b_1$ und $b_0 - b_2$ aufgespannten Ebene (vorausgesetzt die drei Vektoren liegen nicht zufällig auf einer Geraden). Da aber Raumkurven in der Computergrafik benötigt werden, verallgemeinern wir diesen Ansatz zunächst auf eine Kurve, die durch vier Punkte bestimmt wird.



7.1.2 Der Algorithmus von de Casteljau für 4 Punkte

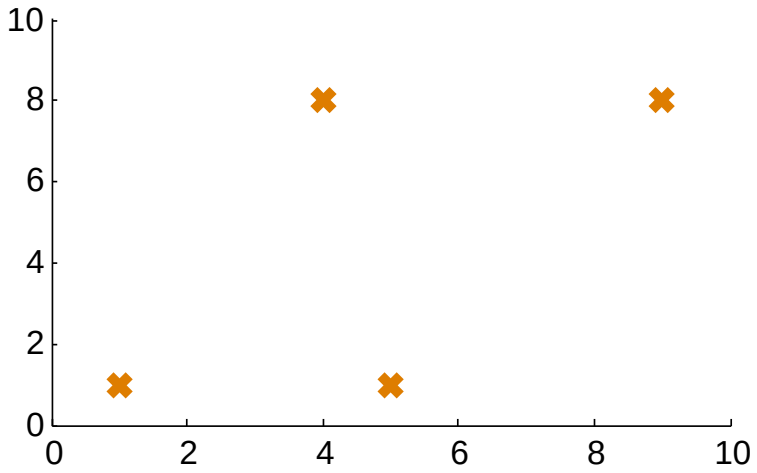
Gegeben seien die vier Punkte b_0, b_1, b_2 und b_3 im $\mathbb{R}^n$. Wie bei der Approximation der Parabeln im letzten Abschnitt definierten wir rekursiv:

$$\begin{aligned} b_0^1(t) &:= (1-t)b_0 + tb_1, \\ b_1^1(t) &:= (1-t)b_1 + tb_2, \\ b_2^1(t) &:= (1-t)b_2 + tb_3, \end{aligned}$$

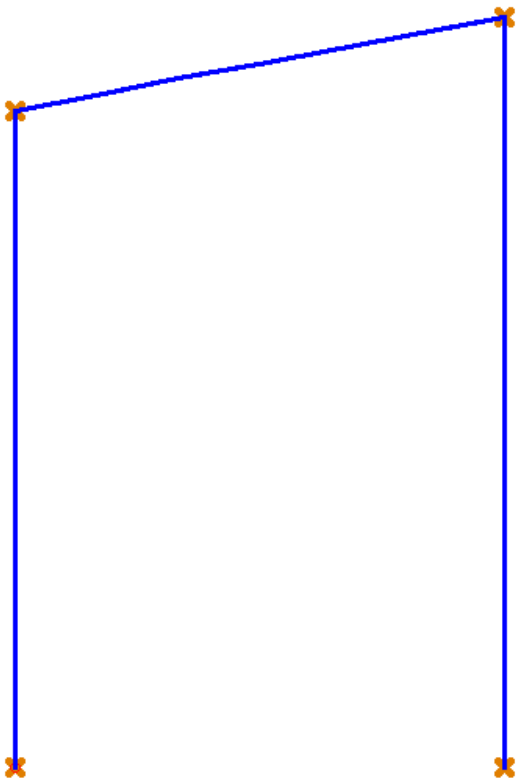
$$\begin{aligned} b_0^2(t) &:= (1-t)b_0^1(t) + tb_1^1(t), \\ b_1^2(t) &:= (1-t)b_1^1(t) + tb_2^1(t), \end{aligned}$$

$$b_0^3(t) := (1-t)b_0^2(t) + tb_1^2(t).$$

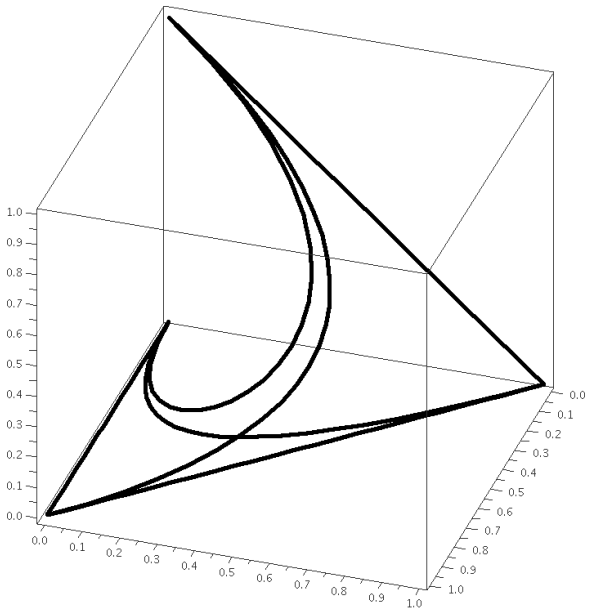
Die Animation veranschaulicht diese Konstruktion an einem Beispiel.



Eine alternative Visualisierung zeigt die folgende Animation.



Eine „echte“ dreidimensionale Kurve zeigt die folgende Abbildung.



Der Zusammenhang zwischen den Koeffizienten kann wieder anhand

eines Dreiecksschemas verdeutlicht werden.

$$\begin{array}{ccccccc}
b_0 & \rightarrow & b_0^1 & \rightarrow & b_0^2 & \rightarrow & b_0^3 \\
 & \nearrow & & \nearrow & & \nearrow & \\
b_1 & \rightarrow & b_1^1 & \rightarrow & b_1^2 & & \\
 & \nearrow & & \nearrow & & & \\
b_2 & \rightarrow & b_2^1 & & & & \\
 & \nearrow & & & & & \\
b_3 & & & & & &
\end{array}$$

7.1.3 Der de Casteljau-Algorithmus

Nach diesen Überlegungen ist es offensichtlich, wie der obige Algorithmus zu verallgemeinern ist, wenn $k+1$ Punkte $b_0,\ b_1, \ldots, b_k$ gegeben sind. Setze in diesem Fall

$$b_i^r(t) := (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t), \quad \left\{ \begin{array}{rcl} r & = & 1, \ldots, k \\ i & = & 0, \ldots, k-r. \end{array} \right. \tag{7.1}$$

Gestartet wird dieser Algorithmus mit der Setzung

$$b_i^0 := b_i, \quad i = 0, \ldots, k.$$

Die Kurve

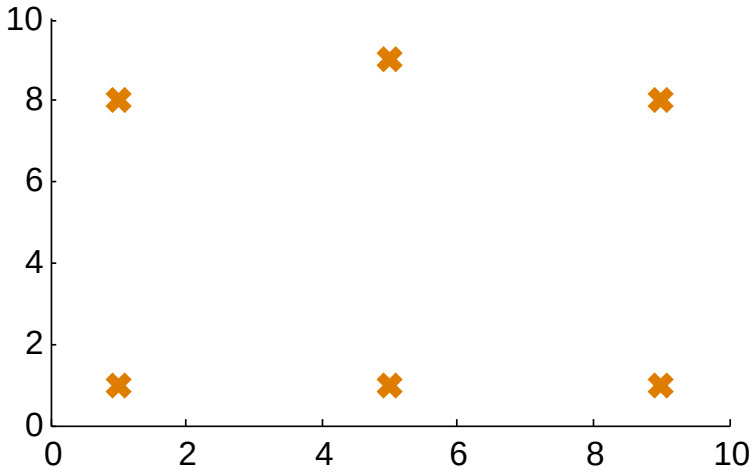
$$b(t, b_0, \ldots, b_k) := b_0^k(t)$$

die der obige Algorithmus bei Eingabe der Punkte $b_0, \ldots, b_k$ berechnet heißt **Bézier-Kurve**.

Den Zusammenhang zwischen den Koeffizienten zeigt das Dreiecksschema

$$\begin{array}{ccccccccc}
b_0 = b_0^0 & \rightarrow & b_0^1 & \rightarrow & b_0^2 & \cdots & \cdots & \rightarrow & b_0^k \\
 & \nearrow & & \nearrow & & & & \nearrow & \\
b_1 = b_1^0 & \rightarrow & b_1^1 & & \vdots & & \rightarrow & & \\
 & \nearrow & & & & & \nearrow & & \\
b_2 = b_2^0 & & \vdots & & \vdots & \rightarrow & & & \\
 & & & & & \nearrow & & & \\
\vdots & & \vdots & \rightarrow & b_{k-2}^2 & & & & \\
 & & & \nearrow & & & & & \\
\vdots & \rightarrow & b_{k-1}^1 & & & & & & \\
 & \nearrow & & & & & & & \\
b_k = b_k^0 & & & & & & & &
\end{array}$$

Eine Realisierung dieses Algorithmus für 6 Punkte zeigt die folgende Animation.



Das Polygon P, welches durch die Punkte $b_0, \ldots, b_k$ beschrieben wird, wird **Bézier-Polygon** oder **Kontrollpolygon** der Kurve b^k genannt. Entsprechend heißen die vorgegebenen Punkte $b_0, \ldots, b_k$ **Kontrollpunkte** oder **Bézier-Punkte**.

7.1.4 Eigenschaften von Bézier-Kurven

Wir wollen jetzt einige Eigenschaften von Bézier-Kurven auflisten.

- **Affine Invarianz**. In der Computergrafik ist es entscheidend, die betrachteten Objekte schnell verschieben, skalieren oder drehen zu können. Die Anwendung dieser Operationen auf eine beliebige Kurve ist i. A. sehr aufwändig. Für Bézier-Kurven lassen sich linear affine Transformationen $T(x) = Ax + v$ mit $A \in \mathbb{R}^{n,n}$, $v \in \mathbb{R}^n$ einfach ausführen, denn die folgenden beiden Verfahren führen zum gleichen Ergebnis:

  – Berechne zuerst die Bézier-Kurve der gegebenen Punkte und transformiere diese Kurve anschließend mit der Transformation T.

  – Transformiere zunächst die gegebenen Punkte mit der Abbildung T und berechne dann die Bézier-Kurve der transformierten Punkte.

  Formal gilt:

  $$b(\cdot, T(b_0), \ldots, T(b_k)) = T(b(\cdot, b_0, \ldots, b_k)).$$

  Offensichtlich liefert die Äquivalenz dieser beiden Aussagen einen schnellen Algorithmus, um Bézier-Kurven mit linear affinen Abbildungen zu transformieren, denn das transformieren der weni-

gen Kontrollpunkte und die anschließende Berechnung der Bézier-Kurve ist wesentlich einfacher als die „naive“ Transformation der Kurve.

- Der obige Algorithmus (7.1) liefert eine Kurve

$$b_0^k : [0,1] \to \mathbb{R}^n.$$

Allerdings ist die Einschränkung auf das Intervall $[0,1]$ nur eine bequeme Konvention. Durch die Einführung von $u := t(b-a)+a$ erhalten wir den zu (7.1) entsprechenden Algorithmus

$$b_i^r(u) := \frac{b-u}{b-a}\, b_i^{r-1}(u) + \frac{u-a}{b-a}\, b_{i+1}^{r-1}(u), \quad \left\{ \begin{array}{rcl} r & = & 1,\ldots,k \\ i & = & 0,\ldots,k-j, \end{array} \right.$$

der eine Bézier-Kurve

$$b_0^k : [a,b] \to \mathbb{R}^n$$

liefert.

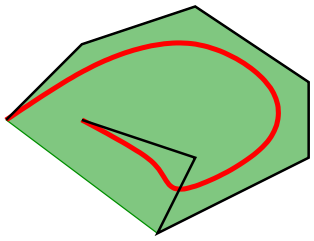
- **Konvexe Hülle**. Eine Bézier-Kurve liegt in der konvexen Hülle des Kontrollpolygons, d. h.

$$b(\cdot, b_0, \ldots, b_k) \subset \mathrm{convex}(b_0, \ldots, b_k).$$

Hierbei wird die konvexe Hülle eines Polygons, das durch die Eckpunkte $b_0, \ldots, b_k$ gegeben ist, durch

$$\mathrm{convex}(b_0, \ldots, b_k) = \left\{ \sum_{j=0}^{k} \alpha_j b_j \text{ mit } \alpha_j \in [0,1],\ \sum_{j=0}^{k} \alpha_j = 1 \right\}$$

definiert. In der Abbildung ist das Kontrollpolygon in schwarz, die Bézier-Kurve in rot und die konvexe Hülle in grün eingezeichnet.



Diese Eigenschaft ist in der Computergrafik von Bedeutung, wenn überprüft werden soll, ob sich Kurven schneiden. Eine präzise Berechnung aller Schnittpunkte ist sehr aufwändig. Überlappen sich aber die konvexen Hüllen der Kontrollpolygone nicht, kann ein

Schnitt der entsprechenden Kurven ausgeschlossen werden. Folglich können sich nur Kurven schneiden, bei denen sich die konvexen Hüllen der Kontrollpolygone überlappen. Die Überprüfung dieser Eigenschaft ist sehr einfach.

- **Endpunkt-Interpolation.** Wie man dem Algorithmus (7.1) entnehmen kann, muss die Beziehung

$$b(0, b_0, \ldots, b_k) = b_0, \quad b(1, b_0, \ldots, b_k) = b_k$$

gelten. Dieses bedeutet, dass die Bézier-Kurve die beiden Endpunkte exakt interpoliert.

7.1.5 Bernsteinpolynome

Wir zeigen jetzt, dass Bézier-Kurven mit Hilfe von **Bernsteinpolynomen** direkt beschrieben werden können, d. h. es wird eine nicht-rekursive Darstellung der Bézier-Kurve angegeben. Bernsteinpolynome werden durch die Formel

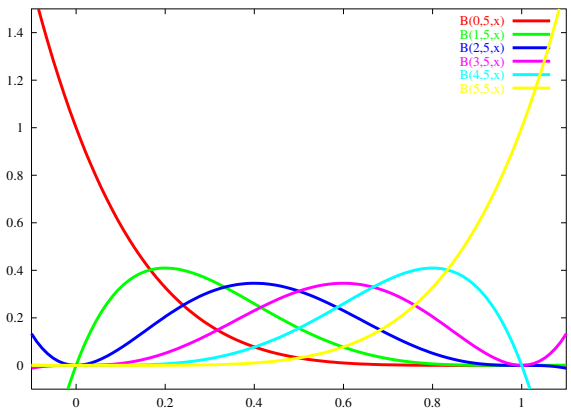
$$B_i^k(t) := \binom{k}{i} t^i (1-t)^{k-i} \tag{7.2}$$

definiert, wobei die Binomialkoeffizienten durch

$$\binom{k}{i} = \begin{cases} \dfrac{k!}{i!(k-i)!} & \text{für } 0 \le i \le k \\ 0, & \text{sonst} \end{cases}$$

bestimmt werden. Diese Polynome wurden 1911 von dem sowjetischen Mathematiker Sergei Natanowitsch Bernstein (∗5.3.1880, †26.10.1968) entwickelt.

Die Bernsteinpolynome $B_i^5(t)$, $i = 0, \ldots, 5$ sind in der Abbildung angegeben.



Bernsteinpolynome besitzen wichtige Eigenschaften, die wir im Folgenden auflisten.

- **Basis von** $\mathcal{P}_n$. Die Bernsteinpolynome $\{B_i^n\}_{i=0,\ldots,n}$ bilden eine Basis des Vektorraums $\mathcal{P}_n$ der Polynome vom Grad $\leq n$.
- **Nullstellen** der Bernsteinpolynome. $\bar{t} = 0$ ist eine i-fache Nullstelle von B_i^k und $\bar{t} = 1$ ist eine $(k-i)$-fache Nullstelle von B_i^k. Diese Aussagen folgen unmittelbar aus (7.2).
- **Symmetrie**. Es gilt die folgende Aussage

$$B_i^k(t) = B_{k-i}^k(1-t) \quad \text{für } i = 0, \ldots, k,\ t \in \mathbb{R}.$$

- **Rekursionsformel**. Es gilt:

$$B_i^k(t) = (1-t)B_i^{k-1}(t) + t\ B_{i-1}^{k-1}(t) \quad \text{für } i = 1, \ldots, k,\ t \in \mathbb{R} \tag{7.3}$$

mit

$$B_0^0(t) = 1 \quad \text{für } t \in \mathbb{R}, \tag{7.4}$$

und

$$B_j^k(t) = 0 \quad \text{für } j \notin \{0, \ldots, k\},\ t \in \mathbb{R}. \tag{7.5}$$

Beachte, dass die Rekursionsformel (7.3) und der Algorithmus von de Casteljau (7.1) die gleiche Struktur haben.

Der Beweis von (7.3) bis (7.5) kann direkt geführt werden. Zunächst gilt

$$B_0^0(t) = \binom{0}{0} t^0 (1-t)^0 = 1$$

und die Aussage (7.5) folgt aus $\binom{k}{j} = 0$ für $j \notin \{0, \ldots, k\}$.

Sei $i \in \{1, \ldots, k\}$. Unter Verwendung der Formel

$$\binom{k}{i} = \binom{k-1}{i} + \binom{k-1}{i-1}$$

erhalten wir

$$\begin{aligned}
B_i^k(t) &= \binom{k}{i} t^i (1-t)^{k-i} \\
&= \binom{k-1}{i} t^i (1-t)^{k-i} + \binom{k-1}{i-1} t^i (1-t)^{k-i} \\
&= (1-t) \underbrace{\binom{k-1}{i} t^i (1-t)^{k-1-i}}_{B_i^{k-1}(t)} + t \underbrace{\binom{k-1}{i-1} t^{i-1} (1-t)^{k-1-(i-1)}}_{B_{i-1}^{k-1}(t)} \\
&= (1-t) B_i^{k-1}(t) + t\ B_{i-1}^{k-1}(t).
\end{aligned}$$

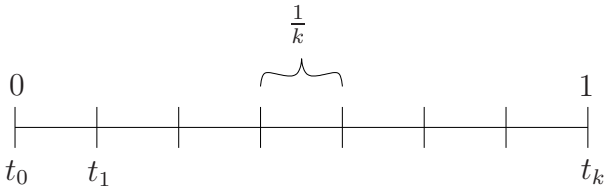
- **Teilung der Eins**. Eine weitere Eigenschaft von Bernsteinpolynomen ist die sogenannte Teilung der Eins:

$$\sum_{j=0}^{k} B_j^k(t) = 1 \quad \text{für alle } t \in \mathbb{R}. \tag{7.6}$$

Mit Hilfe des Binomischen Lehrsatzes folgt (7.6) sofort:

$$1 = 1^k = \big(t + (1-t)\big)^k = \sum_{j=0}^{k} \binom{k}{j} t^j (1-t)^{k-j} = \sum_{j=0}^{k} B_j^k(t).$$

- **Summierbarkeitseigenschaften.**
  - Sei $t_i = \frac{i}{k}$ für $i = 0, \ldots, k$.



Wir behaupten für alle $t \in \mathbb{R}$

$$\sum_{i=0}^{k} t_i B_i^k(t) = t.$$

Zum Beweis dieser Gleichung beweisen wir zunächst die Identität

$$t_i B_i^k(t) = t B_{i-1}^{k-1}(t) \quad \text{für alle } i \geq 0,\ k \geq i.$$

Es gilt

$$\begin{aligned} t_i B_i^k(t) &= \underbrace{\frac{i}{k}}_{=t_i} \binom{k}{i} t^i (1-t)^{k-i} = \frac{i}{k} \cdot \frac{k!}{i!(k-i)!}\, t^i (1-t)^{k-i} \\ &= \frac{(k-1)!}{(i-1)!((k-1)-(i-1))!}\, t \cdot t^{i-1} (1-t)^{(k-1)-(i-1)} \\ &= t \binom{k-1}{i-1} t^{i-1} (1-t)^{(k-1)-(i-1)} = t B_{i-1}^{k-1}(t). \end{aligned}$$

Jetzt folgt unter Verwendung der Teilung der Eins-Eigenschaft

$$\sum_{i=0}^{k} t_i B_i^k(t) = \underbrace{\sum_{i=0}^{k}}_{i=1} t B_{i-1}^{k-1}(t) = t \underbrace{\sum_{i=0}^{k-1} B_i^{k-1}(t)}_{=1} = t.$$

■

– Sei $t_i = \frac{i}{k}$ für $i = 0, \ldots, k$, dann gilt für alle $t \in \mathbb{R}$

$$\sum_{i=0}^{k} (t - t_i)^2 B_i^k(t) = \frac{t(1-t)}{k}.$$

Der Beweis dieser Gleichung kann mit Hilfe der vorangegangenen Ergebnisse, sowie der Teilung der Eins-Eigenschaft geführt werden.

$$\begin{aligned}
\sum_{i=0}^{k} (t - t_i)^2 B_i^k(t) &= t^2 \underbrace{\sum_{i=0}^{k} B_i^k(t)}_{=1} - 2t \underbrace{\sum_{i=0}^{k} t_i B_i^k(t)}_{=t} + \sum_{i=0}^{k} t_i^2 B_i^k(t) \\
&= -t^2 + \sum_{i=0}^{k} \underbrace{t_i}_{=\frac{i}{k}} \underbrace{t_i B_i^k(t)}_{=tB_{i-1}^{k-1}(t)} \\
&= -t^2 + \frac{1}{k} \sum_{i=0}^{k} i \cdot t B_{i-1}^{k-1}(t) \\
&= -t^2 + \frac{1}{k} t \sum_{i=1}^{k} i B_{i-1}^{k-1}(t) \\
&= -t^2 + \frac{1}{k} t \sum_{i=0}^{k-1} (i+1) B_i^{k-1}(t) \\
&= -t^2 + \frac{1}{k} t \left(\sum_{i=0}^{k-1} i B_i^{k-1}(t) + \underbrace{\sum_{i=0}^{k-1} B_i^{k-1}(t)}_{=1} \right) \\
&= -t^2 + \frac{1}{k} t + \frac{1}{k} t(k-1) \underbrace{\sum_{i=0}^{k-1} \frac{i}{k-1} B_i^{k-1}(t)}_{=t} \\
&= -t^2 + \frac{1}{k} t + \frac{1}{k} t^2 (k-1) \\
&= \frac{1}{k}(-kt^2 + t + t^2(k-1)) = \frac{1}{k} t(1-t).
\end{aligned}$$

■

- **Ableitung der Bernsteinpolynome.** Es gilt für alle $t \in \mathbb{R}$

$$\frac{\partial}{\partial t} B_i^k(t) = k \left(B_{i-1}^{k-1}(t) - B_i^{k-1}(t) \right).$$

Durch Ableiten der Bernsteinpolynome erhalten wir

$$\begin{aligned}
\frac{\partial}{\partial t}B_i^k(t) &= \frac{\partial}{\partial t}\binom{k}{i}t^i(1-t)^{k-i} \\
&= \binom{k}{i}i\cdot t^{i-1}(1-t)^{k-i} - \binom{k}{i}(k-i)t^i(1-t)^{k-i-1} \\
&= \frac{k!i}{i!(k-i)!}t^{i-1}(1-t)^{k-i} - \frac{k!(k-i)}{i!(k-i)!}t^i(1-t)^{k-i-1} \\
&= k\left(\frac{(k-1)!}{(i-1)!((k-1)-(i-1))!}t^{i-1}(1-t)^{(k-1)-(i-1)}\right. \\
&\qquad \left. -\frac{(k-1)!}{i!(k-1-i)!}t^i(1-t)^{k-1-i}\right) \\
&= k\left(B_{i-1}^{k-1}(t) - B_i^{k-1}(t)\right).
\end{aligned}$$

■

7.1.6 Bézier-Kurven und Bernsteinpolynome

Der Zusammenhang zwischen den Bernsteinpolynomen und den (mit dem Algorithmus von de Casteljau berechneten) Bézier-Kurven zeigt die Formel

$$b_i^r(t) = \sum_{j=0}^{r} b_{i+j}B_j^r(t), \quad \begin{cases} r \in \{0,\ldots,k\} \\ i \in \{0,\ldots,k-r\}. \end{cases} \tag{7.7}$$

Mit dieser Gleichung erhalten wir eine explizite Darstellung der Zwischenpunkte im Algorithmus von de Casteljau; die neue Darstellung hängt nur von den gewählten Kontrollpunkten $b_0,\ldots,b_k$ ab. Die größte Bedeutung besitzt diese Darstellung für $r = k$, $i = 0$, denn in diesem Fall liefert (7.7) eine explizite Form der Bézier-Kurve, die nur von den Kontrollpunkten abhängt:

$$b(t,b_0,\ldots,b_k) = b_0^k(t) = \sum_{j=0}^{k} b_j B_j^k(t). \tag{7.8}$$

Der Beweis von (7.7) wird mit vollständiger Induktion über r geführt. Für den Induktionsanfang betrachten wir zunächst den Fall $r = 0$, $i \in \{0,\ldots,k\}$:

$$b_i^0 = \sum_{j=0}^{0} b_{i+j}B_j^0(t) = b_iB_0^0 = b_i \quad ✔$$

nach Aussage (7.4).

Somit können wir im Induktionsschritt annehmen, dass (7.7) für b_i^l, $l \leq r-1$, $i \in \{0, \ldots, k-l\}$ erfüllt ist. Die Aussage (7.7) ist zum Abschluss des Beweises noch für b_i^r, $i \in \{0, \ldots, k-r\}$ nachzuweisen.

Der Algorithmus von de Casteljau liefert

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t).$$

Nach Induktion darf für $b_i^{r-1}(t)$ und $b_{i+1}^{r-1}(t)$ die Darstellung (7.7) eingesetzt werden:

$$b_i^r(t) = (1-t)\sum_{j=0}^{r-1} b_{i+j}B_j^{r-1}(t) + t\sum_{j=0}^{r-1} b_{i+1+j}B_j^{r-1}(t).$$

Durch eine Indexverschiebung folgt

$$b_i^r(t) = (1-t)\sum_{j=i}^{i+r-1} b_j B_{j-i}^{r-1}(t) + t\sum_{j=i+1}^{i+r} b_j B_{j-i-1}^{r-1}(t).$$

Da nach (7.5) sowohl $B_{i+r-i}^{r-1} = B_r^{r-1} = 0$ als auch $B_{i-i-1}^{r-1} = B_{-1}^{r-1} = 0$ gilt, dürfen die obigen Summen trivial durch 0-Summanden erweitert und dann zusammengefasst werden:

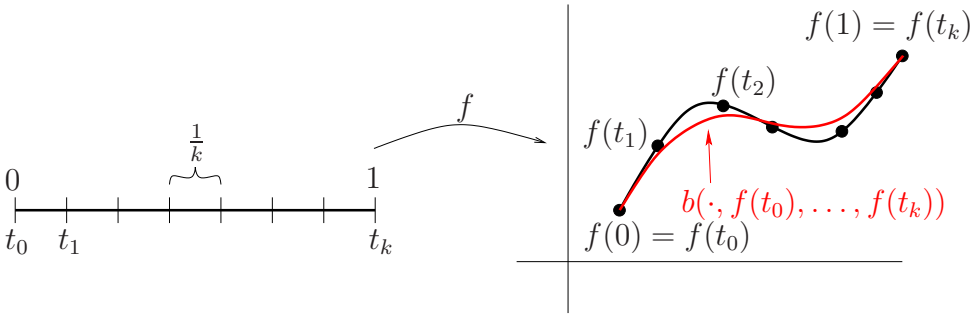
$$\begin{aligned} b_i^r(t) &= (1-t)\sum_{j=i}^{i+r} b_j B_{j-i}^{r-1}(t) + t\sum_{j=i}^{i+r} b_j B_{j-i-1}^{r-1}(t) \\ &= \sum_{j=i}^{i+r} b_j \underbrace{\left((1-t)B_{j-i}^{r-1}(t) + t\ B_{j-i-1}^{r-1}(t)\right)}_{=B_{j-i}^r(t) \text{ nach (7.3)}}. \end{aligned}$$

Eine erneute Indexverschiebung liefert somit die Behauptung:

$$b_i^r(t) = \sum_{j=0}^{r} b_{i+j}B_j^r(t). \quad \checkmark$$

7.1.7 Approximationseigenschaften

Sei $f : [0,1] \to \mathbb{R}^n$ eine stetige Funktion und sei $t_i = \frac{i}{k}$ für $i = 0, \ldots, k$. Wir zeigen, dass die Bézier-Kurve zu den Punkten $f(t_0), \ldots, f(t_k)$, für $k \to \infty$ gleichmäßig gegen die gegebene Funktion f konvergiert, vgl. die folgende Abbildung.



Satz 7.1 *Sei $f : [0,1] \to \mathbb{R}^n$ stetig und sei $t_i = \frac{i}{k}$ für $i = 0, \ldots, k$. Dann existiert zu jedem $\varepsilon > 0$ ein $k \in \mathbb{N}$ mit*

$$\|f - b(\cdot, f(t_0), \ldots, f(t_k))\|_{[0,1]} \leq \varepsilon.$$

Hierbei bezeichnet $\|\cdot\|_{[0,1]}$ die Supremumsnorm auf dem Intervall $[0,1]$, d. h.

$$\|g\|_{[0,1]} = \sup_{t \in [0,1]} \|g(t)\|,$$

wobei $\|\cdot\|$ eine Norm im $\mathbb{R}^n$ ist.

Bemerkung 7.2

- *Konvergenz in der Supremumsnorm wird auch als **gleichmäßige Konvergenz** bezeichnet.*
- *Die Bézier-Kurve $b_0^k : [0,1] \to \mathbb{R}^n$ ist ein Polynom vom Grad $\leq k$.*
- *Satz 7.1 besagt also, dass die gegebene Funktion f gleichmäßig durch ein Polynom (mit beliebig hoher Genauigkeit) approximiert werden kann.*
- *Satz 7.1 ist eine alternative Formulierung des Weierstaßschen Approximationssatzes, siehe [19, 16].*

Beweis: Sei $\varepsilon > 0$ und sei $t \in [0,1]$ beliebig gewählt, dann gilt:

$$\begin{aligned}
\|f(t) - b(t, f(t_0), \ldots, f(t_k))\| &= \Big\| \underbrace{\sum_{i=0}^{k} B_i^k(t)}_{=1} f(t) - \sum_{i=0}^{k} f(t_i) B_i^k(t) \Big\| \\
&= \Big\| \sum_{i=0}^{k} \big(f(t) - f(t_i)\big) B_i^k(t) \Big\| \\
&\leq \sum_{i=0}^{k} \|f(t) - f(t_i)\| \underbrace{B_i^k(t)}_{\geq 0 \ \forall t \in [0,1]} . \qquad (7.9)
\end{aligned}$$

Da f auf dem kompakten Intervall $[0,1]$ gleichmäßig stetig ist, existiert ein $\delta > 0$, so dass $\|f(t) - f(t_i)\| \leq \frac{\varepsilon}{2}$ für alle $t, t_i \in [0,1]$ mit $|t - t_i| \leq \delta$ gilt. Hierbei ist zu beachten, dass die Konstante δ, wegen der gleichmäßigen Stetigkeit, nicht von der Stelle t abhängt.

Setze für das beliebige aber feste $t \in [0,1]$

$$\begin{aligned} I &:= \big\{i \in \{0,\ldots,k\} : |t - t_i| \leq \delta\big\}, \\ J &:= \big\{i \in \{0,\ldots,k\} : |t - t_i| > \delta\big\}. \end{aligned}$$

Schließlich sei $c = \max_{t\in[0,1]} \|f(t)\|$, dann folgt aus (7.9):

$$\|f(t) - b(t, f(t_0), \ldots, f(t_k))\| \leq S_1 + S_2,$$

mit

$$\begin{aligned} S_1 &= \sum_{i\in I} \|f(t) - f(t_i)\| B_i^k(t), \\ S_2 &= \sum_{i\in J} \|f(t) - f(t_i)\| B_i^k(t). \end{aligned}$$

Wir zeigen, dass die beiden Ausdrücke S_1 und S_2 für hinreichend große k durch $\frac{\varepsilon}{2}$ abgeschätzt werden können und erhalten so die Behauptung

$$\|f(t) - b(t, f(t_0), \ldots, f(t_k))\| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

- **Abschätzung von S_1.** Nach Konstruktion der Indexmenge I gilt

$$S_1 \leq \sum_{i\in I} \frac{\varepsilon}{2} B_i^k(t) \leq \frac{\varepsilon}{2} \underbrace{\sum_{i=0}^{k} B_i^k(t)}_{=1} = \frac{\varepsilon}{2}.$$

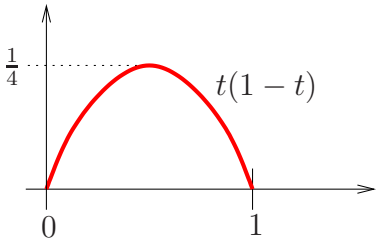
- **Abschätzung von S_2.** Aus der Konstruktion von J folgt

$$|t - t_i|^2 > \delta^2 \quad \text{und somit} \quad 1 < \frac{|t - t_i|^2}{\delta^2}.$$

Die Anwendung der zweiten Summationseigenschaft der Bernsteinpolynome liefert

$$\begin{aligned} S_2 &= \sum_{i\in J} \underbrace{\|f(t) - f(t_i)\|}_{\leq 2c} B_i^k(t) \leq \sum_{i\in J} 2c B_i^k(t) < \sum_{i\in J} 2c \underbrace{\frac{|t - t_i|^2}{\delta^2}}_{>1} B_i^k(t) \\ &\leq \frac{2c}{\delta^2} \sum_{i=0}^{k} (t - t_i)^2 B_i^k(t) = \frac{2c}{\delta^2} \cdot \frac{t(1-t)}{k}. \end{aligned}$$

Eine Kurvendiskussion der Funktion $g(t) = t(1-t)$ ergibt $\max_{t\in[0,1]} t(1-t) = \frac{1}{4}$, siehe Abbildung.



Zusammen erhalten wir

$$S_2 \leq \frac{c}{2\delta^2} \cdot \frac{1}{k} \leq \frac{\varepsilon}{2} \quad \text{für } k \text{ hinreichend groß, d. h.} \quad k \geq \frac{c}{\delta^2 \varepsilon}.$$

■

7.1.8 Untersuchung der Konvergenzgeschwindigkeit

Abschließend untersuchen wir, wie schnell die Bézier-Kurve gegen die gegebene Kurve f konvergiert, wenn die Anzahl der Stützstellen k erhöht wird.

Wir setzen Lipschitz-Stetigkeit der gegebenen Funktion f auf dem Intervall $[0,1]$ voraus, d. h. es existiert eine Konstante $L > 0$, mit

$$\|f(t_1) - f(t_2)\| \leq L|t_1 - t_2| \quad \text{für alle} \quad t_1, t_2 \in [0,1].$$

Im Beweis von Satz 7.1 haben wir mit der ε-δ-Definition der gleichmäßigen Stetigkeit gearbeitet. Für Lipschitz-stetige Funktionen gilt der folgende Zusammenhang zwischen ε und δ

$$\|f(t_1) - f(t_2)\| \leq L|t_1 - t_2| \leq \frac{\varepsilon}{2}, \text{ für } |t_1 - t_2| \leq \delta; \text{ wähle also } \delta = \frac{\varepsilon}{2L}.$$

Zur Untersuchung der Konvergenzgeschwindigkeit sind also S_1 und S_2 (aus dem Beweis von Satz 7.1) in Abhängigkeit von k abzuschätzen. Die Abschätzung von S_1 ist unabhängig von k. Somit muss nur S_2 betrachtet werden. Aus der Konstruktion der Indexmenge J und der zweiten Summationseigenschaft folgt

$$\begin{aligned} S_2 &= \sum_{i \in J} \underbrace{\|f(t) - f(t_i)\|}_{\leq L|t-t_i|} B_i^k(t) \leq \sum_{i \in J} L|t - t_i| \underbrace{\frac{|t-t_i|}{\delta}}_{>1} B_i^k(t) \\ &\leq \frac{L}{\delta} \sum_{i=0}^{k} (t-t_i)^2 B_i^k(t) = \frac{L}{\delta} \cdot \frac{\overbrace{t(1-t)}^{\leq \frac{1}{4}}}{k} \leq \frac{L}{4 \underbrace{\delta}_{= \frac{\varepsilon}{2L}}} \cdot \frac{1}{k} = \frac{L^2}{2\varepsilon} \cdot \frac{1}{k} \overset{!}{=} \frac{\varepsilon}{2} \end{aligned}$$

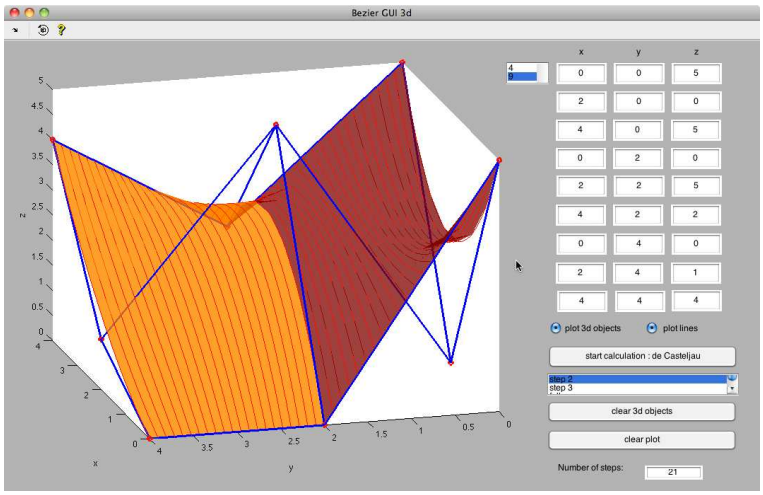
für $k = \frac{L^2}{\varepsilon^2}$. Aufgelöst nach ε ergibt sich $\varepsilon = L \cdot \frac{1}{\sqrt{k}}$.

Folglich kann die Konvergenzgeschwindigkeit der Bézier-Kurve $b(\cdot, f(t_0), \ldots f(t_k))$ gegen die gegebene Kurve f durch $C \cdot \frac{1}{\sqrt{k}}$ abgeschätzt werden:

$$\|f - b(\cdot, f(t_0), \ldots, f(t_k))\|_{[0,1]} \leq C\frac{1}{\sqrt{k}} = Ck^{-\frac{1}{2}}.$$

In anderen Worten kann die gegebene Kurve beliebig genau durch ein Polynom (die Bézier-Kurve) approximiert werden, die Konvergenz ist aber sehr langsam ($\frac{1}{\sqrt{k}}$).

7.2 Bézier-Flächen



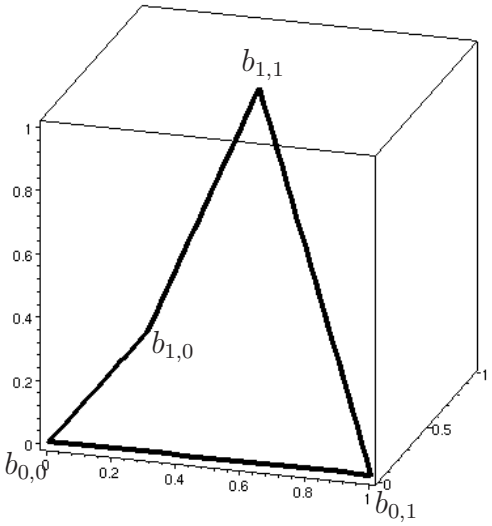
In diesem Abschnitt verallgemeinern wir den in Abschnitt 7.1.3 eingeführten Algorithmus von de Casteljau auf die Konstruktion von Bézier-Flächen. Hierbei ist die entscheidende Idee, die wiederholte *lineare Interpolation*, mit deren Hilfe in 7.1 die Bézier-Kurven konstruiert wurden, auf eine sogenannte *bilineare Interpolation* zu erweitern.

7.2.1 Hyperbolisches Paraboloid

Seien vier Punkte $b_{0,0}$, $b_{0,1}$, $b_{1,0}$, $b_{1,1}$ gegeben. Diese Punkte liegen im Allgemeinen nicht in einer Ebene. Wir betrachten das Beispiel

$$b_{0,0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad b_{0,1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad b_{1,0} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad b_{1,1} = \begin{pmatrix} 0.5 \\ 0.5 \\ 1 \end{pmatrix},$$

siehe Abbildung.



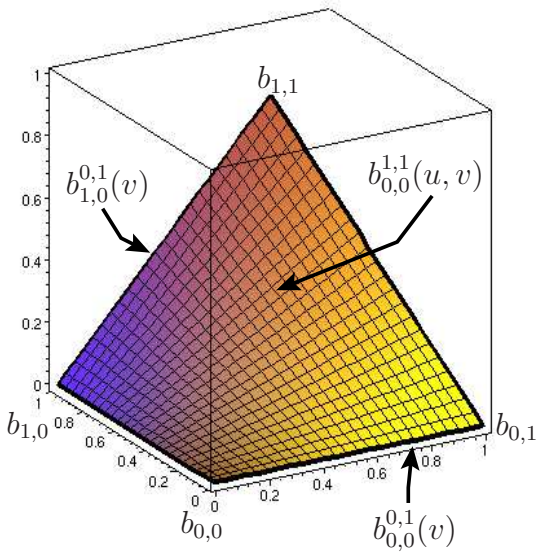
Das Ziel ist nun die Konstruktion einer Fläche, die durch diese Punkte verläuft. Wir wählen den folgenden Ansatz (vgl. Abschnitt 7.1.1):

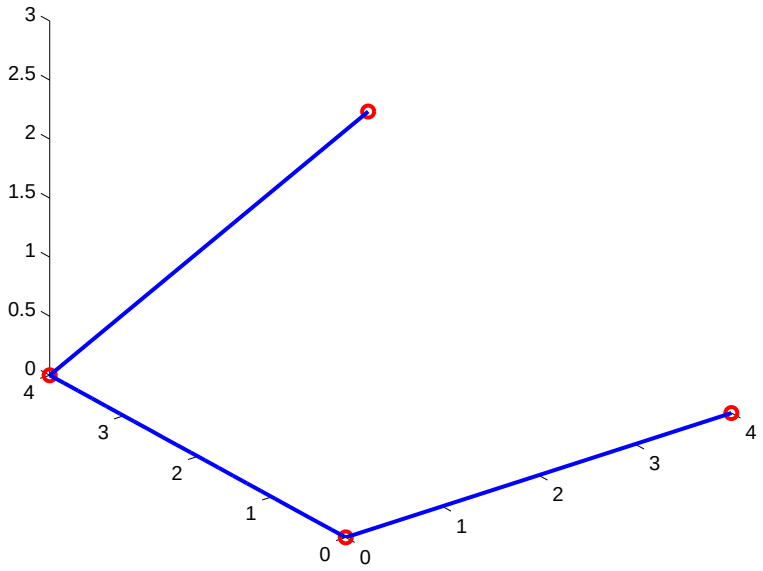
$$\begin{aligned} b_{0,0}^{0,1}(v) &:= (1-v)b_{0,0} + vb_{0,1}, \\ b_{1,0}^{0,1}(v) &:= (1-v)b_{1,0} + vb_{1,1}, \end{aligned}$$

und bilden hieraus die Bézier-Fläche mittels

$$b_{0,0}^{1,1}(u,v) := (1-u)b_{0,0}^{0,1}(v) + ub_{1,0}^{0,1}(v).$$

Diese Berechnung wird auch als *bilineare Interpolation* bezeichnet, siehe Abbildung und Animation.

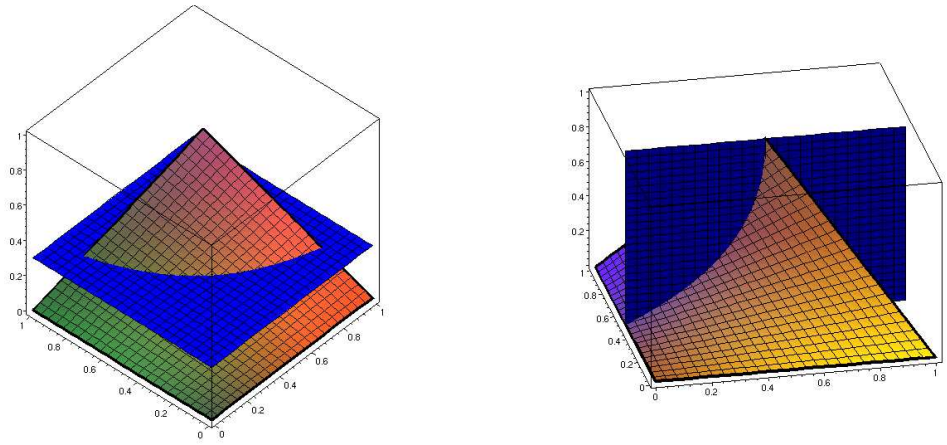




In dem Beispiel erhält man also

$$
\begin{aligned}
b_{0,0}^{0,1}(v) &= (1-v)\begin{pmatrix}0\\0\\0\end{pmatrix} + v\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}v\\0\\0\end{pmatrix}, \\
b_{1,0}^{0,1}(v) &= (1-v)\begin{pmatrix}0\\1\\0\end{pmatrix} + v\begin{pmatrix}0.5\\0.5\\1\end{pmatrix} = \begin{pmatrix}0.5v\\1-0.5v\\v\end{pmatrix}, \\
b_{0,0}^{1,1}(u,v) &= (1-u)\begin{pmatrix}v\\0\\0\end{pmatrix} + u\begin{pmatrix}0.5v\\-0.5v+1\\v\end{pmatrix} = \begin{pmatrix}v-0.5uv\\u-0.5uv\\uv\end{pmatrix}.
\end{aligned}
$$

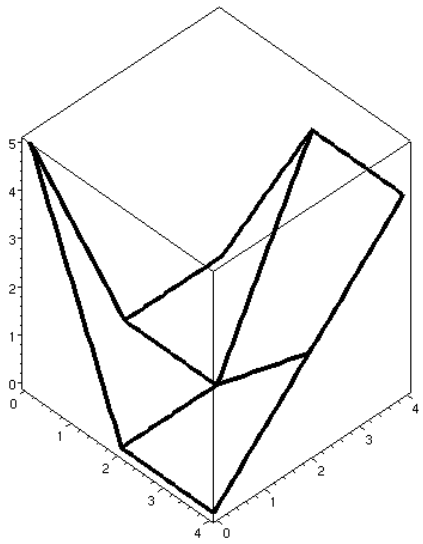
Die durch $b_{0,0}^{1,1}$ parametrisierte Fläche wird auch als *hyperbolisches Paraboloid* bezeichnet. Diese Namensgebung stammt aus der analytischen Geometrie und ist wie folgt motiviert. Schneiden wir die Fläche mit einer zur (x, y) Ebene parallelen Ebene, so ist die Schnittkurve eine Hyperbel (vgl. linke Abbildung). Aber die Schnittkurve dieser Fläche mit einer Ebene, welche die z-Achse beinhaltet, ist eine Parabel (vgl. rechte Abbildung).



7.2.2 Der Algorithmus von de Casteljau

Bézier-Kurven haben wir durch wiederholte lineare Interpolation gewonnen. Bei Bézier-Flächen funktioniert der gleiche Ansatz, nur muss bilinear interpoliert werden.

Gegeben seinen $(n+1)^2$ Punkte $p_{i,j}$, $0 \leq i,j \leq n$, die ein Gitter aufspannen. In der Abbildung ist ein Beispiel im Fall $n = 2$ angegeben.



Wir führen wiederholt die folgende bilineare Interpolation aus. Man berechne für

$$\begin{aligned} r &= 1, \ldots, n \\ i,j &= 0, \ldots, n-r \end{aligned}$$

$$b_{i,j}^{r-1,r}(u,v) = (1-v)b_{i,j}^{r-1,r-1}(u,v) + vb_{i,j+1}^{r-1,r-1}(u,v), \tag{7.10}$$

$$b_{i+1,j}^{r-1,r}(u,v) = (1-v)b_{i+1,j}^{r-1,r-1}(u,v) + vb_{i+1,j+1}^{r-1,r-1}(u,v), \tag{7.11}$$

$$b_{i,j}^{r,r}(u,v) = (1-u)b_{i,j}^{r-1,r}(u,v) + ub_{i+1,j}^{r-1,r}(u,v), \tag{7.12}$$

wobei

$$b_{i,j}^{0,0}(u,v) := p_{i,j}, \quad 0 \le i,j \le n$$

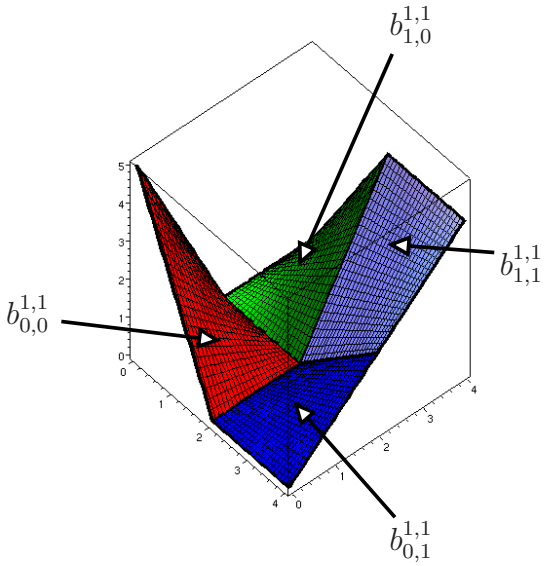
gesetzt wird. Nach Durchführung dieses Algorithmus ist $b_{0,0}^{n,n}$ die gesuchte Parametrisierung der Bézier-Fläche.

Die Formeln (7.10), (7.11) und (7.12) können auch mittels der Kurzschreibweise

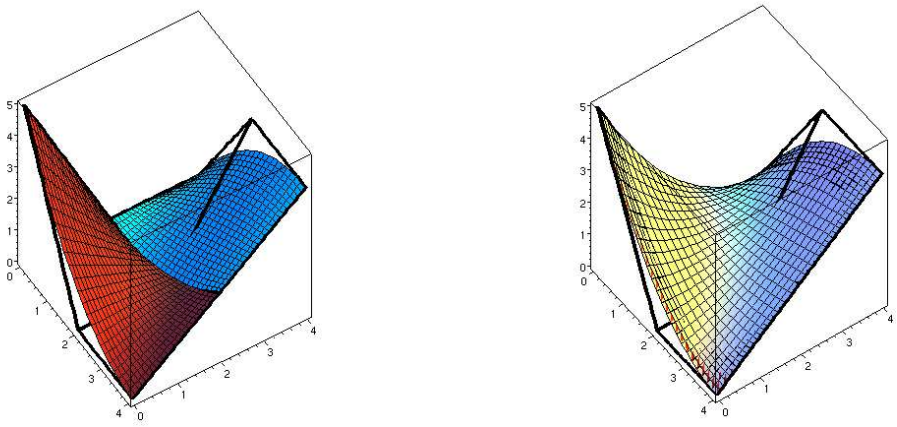
$$b_{i,j}^{r,r} = \begin{pmatrix} 1-u & u \end{pmatrix} \begin{pmatrix} b_{i,j}^{r-1,r-1} & b_{i,j+1}^{r-1,r-1} \\ b_{i+1,j}^{r-1,r-1} & b_{i+1,j+1}^{r-1,r-1} \end{pmatrix} \begin{pmatrix} 1-v \\ v \end{pmatrix}$$

zusammengefasst werden.

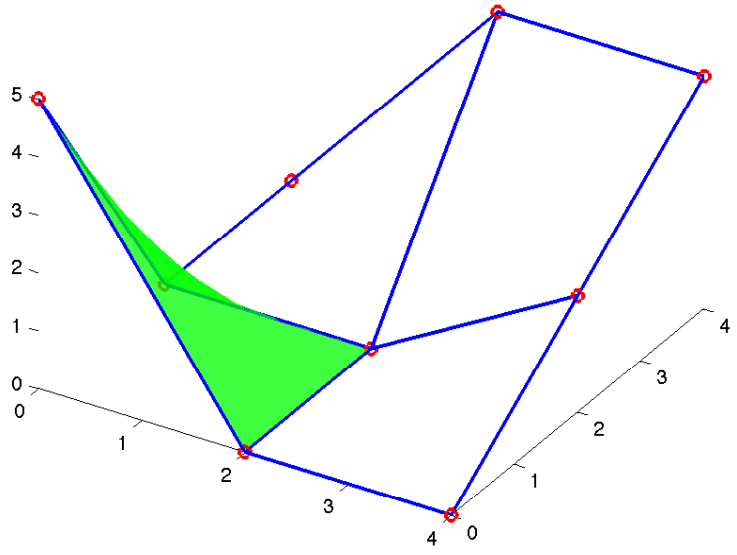
In unserem Beispiel, im Fall $n = 2$ werden im ersten Schritt ($r = 1$) des Algorithmus die einzelnen Gittervierecke bilinear interpoliert, wie die folgende Abbildung zeigt.



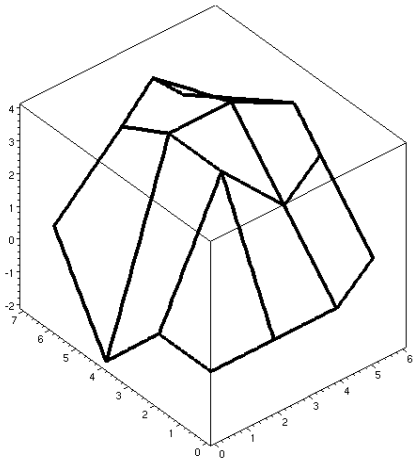
Im nächsten Schritt ($r = 2$) werden mittels (7.10) und (7.11) zwei benachbarte Gitterquadrate zusammengefasst (linke Abbildung). Hieraus wird dann durch (7.12) die Bézier-Fläche bestimmt (rechte Abbildung).



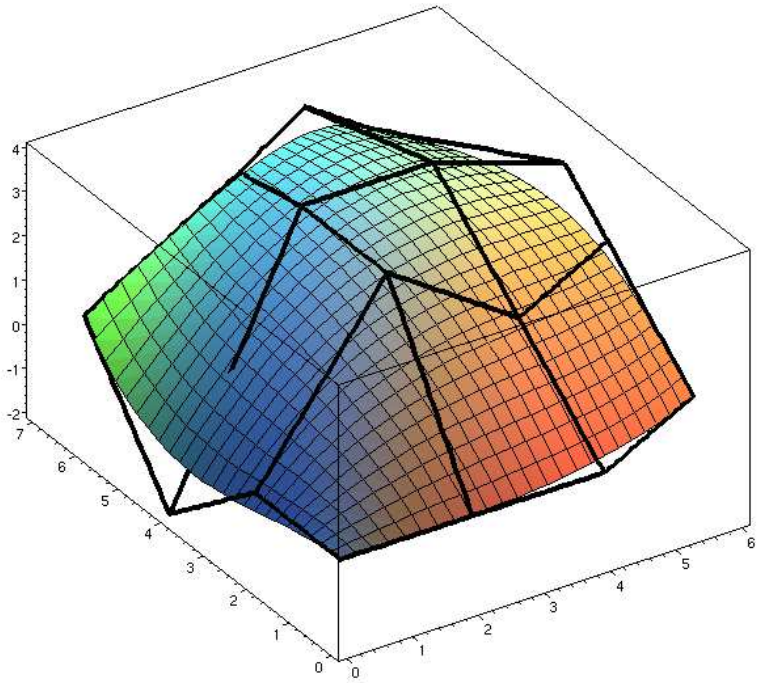
Diese Konstruktion illustriert die folgende Animation.



Entsprechend liefert der obige Algorithmus im Fall $n = 3$ für das Gitter



die folgende Bézier-Fläche.



Kapitel 8

Projektive Geometrie

Euklidische Koordinaten sind allgemein bekannt. Aber es können auch weitere Koordinatensysteme zur Beschreibung von Punkten im Raum definiert werden. Ein alternatives System, die sogenannten **homogenen Koordinaten**, die auch als **projektive Koordinaten** bezeichnet werden, führen wir in diesem Kapitel ein. Dieses neue Koordinatensystem ist von praktischer Relevanz, denn die OpenGL-Grafik-Bibliothek verwendet beispielsweise 4-dimensionale homogene Koordinaten zur internen Darstellung der Objekte. Auch können in diesen Koordinaten affin lineare Transformationen sowie Zentralprojektionen durch Matrixmultiplikationen realisiert werden, vgl. [21].

Zunächst werden wir die Einführung homogener Koordinaten geometrisch motivieren. Es werden ein, zwei und drei-dimensionale Euklidische Koordinaten in die entsprechenden zwei, drei bzw. vier-dimensionalen homogenen Koordinaten umgeschrieben. Hierbei ist zu beachten, dass die im Folgenden beschriebene **projektive Geometrie** in beliebigen Dimensionen existiert, also keineswegs auf niedrige Dimensionen beschränkt ist.

8.1 Geometrisch motivierte Einführung homogener Koordinaten

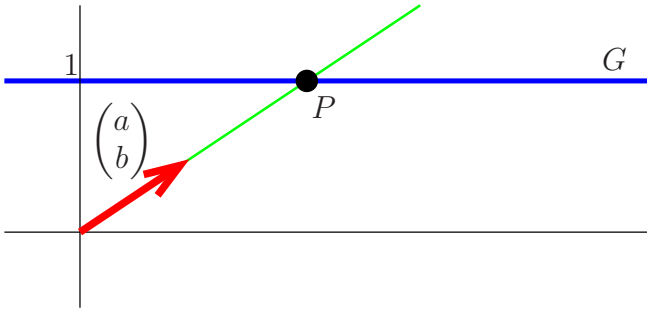
Der Vorteil homogener Koordinaten besteht darin, dass *uneigentliche* Punkte, z. B. $\begin{pmatrix} 1 \\ 1 \\ \infty \end{pmatrix}$ eine explizite Darstellung haben. Eine Repräsentation dieser Punkte wird beispielsweise benötigt, wenn eine Lichtquelle definiert werden soll, deren Licht – unabhängig von der Position im Raum – immer aus der gleichen Richtung kommt. Durch die Positionierung der

Lichtquelle im Unendlichen kann diese Eigenschaft sichergestellt werden.

8.1.1 Homogene Koordinaten für Punkte im $\mathbb{R}$

Es ist das Ziel eine Koordinatenschreibweise zu finden, in der die *uneigentlichen Punkten* ∞ und $-\infty$ dargestellt werden können.

Hierzu wählen wir den folgenden Ansatz, vgl. Abbildung.



Seien $a, b \in \mathbb{R}$ mit $b > 0$ gegeben. Dieser Vektor $\begin{pmatrix} a \\ b \end{pmatrix}$ erzeugt den Strahl

$$S = \left\{ r \begin{pmatrix} a \\ b \end{pmatrix} : r \in \mathbb{R}^+ \right\} = \left\{ \begin{pmatrix} ra \\ rb \end{pmatrix} : r \in \mathbb{R}^+ \right\},$$

der in der Abbildung in grün eingezeichnet ist. Hierbei ist zu beachten, dass der Vektor $\begin{pmatrix} ra \\ rb \end{pmatrix}$ für $r > 0$ den selben Strahl definiert. In diesem Sinne sind zwei Vektoren mit gleicher Richtung äquivalent. Wir verwenden hierfür die Schreibweise

$$\begin{pmatrix} a \\ b \end{pmatrix} \cong \begin{pmatrix} ra \\ rb \end{pmatrix}.$$

Dieser Strahl schneidet die in blau eingezeichnete Gerade

$$G := \left\{ \begin{pmatrix} x \\ 1 \end{pmatrix} : x \in \mathbb{R} \right\}$$

für

$$\begin{aligned} r \begin{pmatrix} a \\ b \end{pmatrix} &= \begin{pmatrix} x \\ 1 \end{pmatrix} \\ \Leftrightarrow r &= \frac{1}{b}, \\ x &= \frac{a}{b}; \end{aligned}$$

der Schnittpunkt P besitzt also die Koordinaten $P = \begin{pmatrix} a/b \\ 1 \end{pmatrix}$.

Die grundlegende Idee besteht darin, dem Vektor in homogenen Koordinaten $\begin{pmatrix} a \\ b \end{pmatrix}$ die Zahl $\frac{a}{b}$, also die x-Komponente des Schnittpunktes P, zuzuordnen. Für diese Zuordnung verwenden wir das Symbol

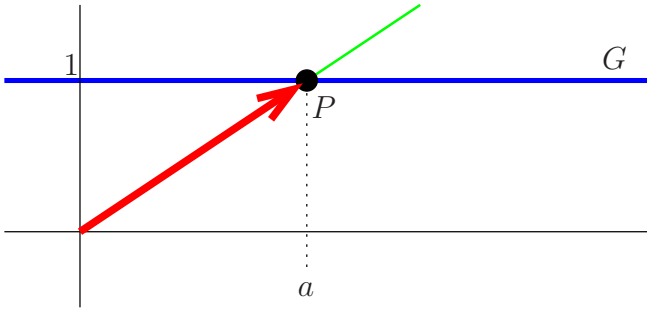
$$\begin{pmatrix} a \\ b \end{pmatrix} \rightsquigarrow \frac{a}{b}.$$

Beachte:

$$\begin{pmatrix} ra \\ rb \end{pmatrix} \rightsquigarrow \frac{ra}{rb} = \frac{a}{b},$$

d. h. nur die Richtung des Vektors wird berücksichtigt.

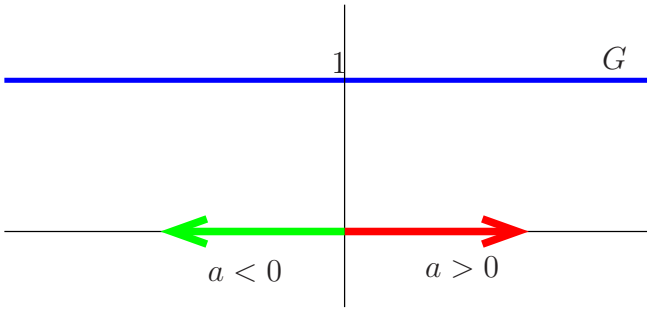
Sei umgekehrt eine Zahl $a \in \mathbb{R}$ gegeben, dann erzeugt der Vektor $\begin{pmatrix} a \\ 1 \end{pmatrix}$ einen Strahl, der die 1-Gerade G bei $\begin{pmatrix} a \\ 1 \end{pmatrix}$ schneidet, siehe Abbildung.



Zusammen erhalten wir

$$\frac{a}{b} = \frac{a/b}{1} \leftrightsquigarrow \begin{pmatrix} a/b \\ 1 \end{pmatrix} \cong \begin{pmatrix} a \\ b \end{pmatrix} \cong \begin{pmatrix} ra \\ rb \end{pmatrix} \leftrightsquigarrow \frac{ra}{rb} = \frac{a}{b}.$$

Abschließend ist noch der Fall $b = 0$ zu betrachten, siehe Abbildung.



Offensichtlich schneidet der zugehörige Strahl die Gerade G nicht. Betrachte die Folge von Vektoren $\begin{pmatrix} a \\ 10^{-n} \end{pmatrix}$ in homogenen Koordinaten. Für diese Folge erhalten wir die Zuordnung

$$u_1 = \begin{pmatrix} a \\ 0.1 \end{pmatrix} \leftrightsquigarrow \frac{a}{0.1} = 10a, \quad u_2 = \begin{pmatrix} a \\ 0.01 \end{pmatrix} \leftrightsquigarrow \frac{a}{0.01} = 100a,$$

$$u_3 = \begin{pmatrix} a \\ 0.001 \end{pmatrix} \leftrightsquigarrow \frac{a}{0.001} = 1000a, \quad u_n = \begin{pmatrix} a \\ 10^{-n} \end{pmatrix} \leftrightsquigarrow \frac{a}{10^{-n}} = 10^n a,$$

die abhängig vom Vorzeichen von a gegen $+\infty$ oder gegen $-\infty$ konvergiert.

Es folgt:

$$\begin{pmatrix} a \\ 0 \end{pmatrix} \leftrightsquigarrow \begin{cases} +\infty & \text{für } a > 0, \\ -\infty & \text{für } a < 0, \end{cases}$$

und wir haben so eine Darstellung von $\pm\infty$ in homogenen Koordinaten gefunden.

8.1.2 Homogene Koordinaten für Punkte im $\mathbb{R}^2$

Zum Erhalt der homogenen Koordinaten für 2-dimensionale Euklidische Koordinaten, wird der obige Ansatz um eine Dimension erweitert.

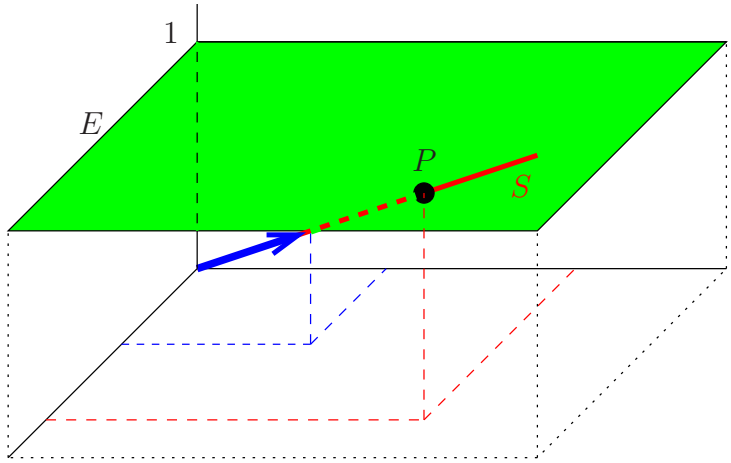
Seien $a, b, c \in \mathbb{R}$ mit $c > 0$ gegeben. Wir betrachten den Schnittpunkt P des Strahls

$$S = \left\{ r \begin{pmatrix} a \\ b \\ c \end{pmatrix} r \in \mathbb{R}^+ \right\}$$

mit der 1-Ebene

$$E = \left\{ \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} : x, y \in \mathbb{R} \right\},$$

siehe Abbildung.



Zur Berechnung des Schnittpunktes P wählen wir den Ansatz

$$\begin{pmatrix} ra \\ rb \\ rc \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

und erhalten

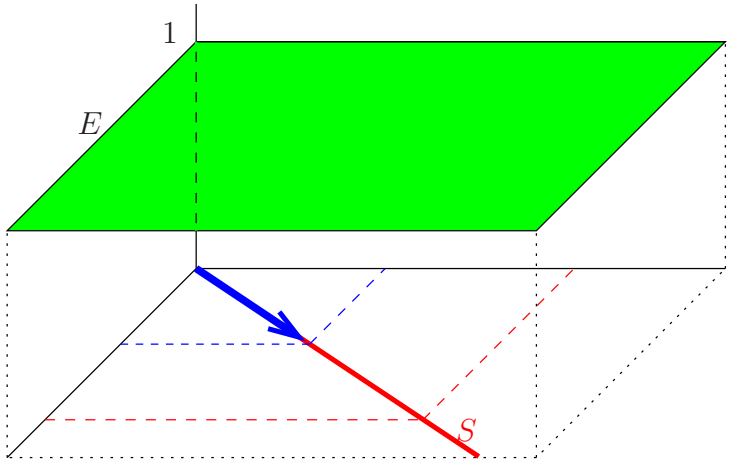
$$P = \begin{pmatrix} a/c \\ b/c \\ 1 \end{pmatrix}.$$

Somit ordnen wir dem Vektor $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ in homogenen Koordinaten den Vektor $\begin{pmatrix} a/c \\ b/c \end{pmatrix}$ in euklidischen Koordinaten zu:

$$\begin{pmatrix} ra \\ rb \\ rc \end{pmatrix} \cong \begin{pmatrix} a \\ b \\ c \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} a/c \\ b/c \end{pmatrix}.$$

Umgekehrt ordnen wir dem euklidischen Vektor $\begin{pmatrix} a \\ b \end{pmatrix}$ den Vektor $\begin{pmatrix} a \\ b \\ 1 \end{pmatrix}$ in homogenen Koordinaten zu.

Wieder existiert im Fall $c = 0$ kein Schnittpunkt des durch $\begin{pmatrix} a \\ b \\ 0 \end{pmatrix}$ erzeugten Strahls mit der 1-Ebene E, siehe Abbildung.



Dieser Vektor entspricht in euklidischen Koordinaten dem Punkt im Unendlichen, der in Richtung des Vektors $\begin{pmatrix} a \\ b \end{pmatrix}$ liegt, wie die folgende Überlegung verdeutlicht. Betrachte die Folge $u_n := \begin{pmatrix} a \\ b \\ 10^{-n} \end{pmatrix}$. Es gilt die folgende Zuordnung zwischen homogenen und euklidischen Koordinaten:

$$u_1 = \begin{pmatrix} a \\ b \\ 0.1 \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} 10a \\ 10b \end{pmatrix}, \quad \begin{pmatrix} a \\ b \\ 0.01 \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} 100a \\ 100b \end{pmatrix}, \quad \begin{pmatrix} a \\ b \\ 10^{-n} \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} 10^n a \\ 10^n b \end{pmatrix}.$$

8.1.3 Homogene Koordinaten für Punkte im $\mathbb{R}^3$

In diesem Abschnitt zeigen wir, dass jeder Vektor des $\mathbb{R}^3$ mit Hilfe von 4 Koordinaten $(x, y, z, w)^T$ dargestellt werden kann, wobei $w \geq 0$ gesetzt wird.

Sei $\begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3$ gegeben. In den homogenen Koordinaten besitzt dieser Punkt die Darstellung $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$.

Ferner gilt (wie beim Kürzen eines Bruches)

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \cong \begin{pmatrix} ax \\ ay \\ az \\ aw \end{pmatrix}, \quad \text{mit } a > 0.$$

Es ist zu beachten, dass zwei Punkte in homogenen Koordinaten (durch das Symbol $\cong$) identifiziert werden, wenn sie die gleiche Darstellung in euklidischen Koordinaten besitzen.

Somit definieren wir für $w > 0$ die folgende Entsprechung zwischen euklidischen und homogenen Koordinaten:

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \end{pmatrix}. \tag{8.1}$$

Offensichtlich ist ein Punkt mit $w = 0$ kein Punkt des $\mathbb{R}^3$; die Darstellung (8.1) besitzt dort eine Singularität. Im euklidischen Raum liegt dieser Punkt somit im Unendlichen. An einem Beispiel verdeutlichen wir diese Überlegung.

Betrachte die Folge

$$u_0 = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 1 \end{pmatrix}, \quad u_1 = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0.1 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0.01 \end{pmatrix} \quad , \ldots , \quad u_n = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 10^{-n} \end{pmatrix}.$$

In euklidischen Koordinaten haben diese Punkte die Darstellung

$$u_0 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \quad u_1 = \begin{pmatrix} 10 \\ 20 \\ 0 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 100 \\ 200 \\ 0 \end{pmatrix} \quad , \ldots , \quad u_n = \begin{pmatrix} 1 \cdot 10^n \\ 2 \cdot 10^n \\ 0 \end{pmatrix}.$$

Die Folge u_n konvergiert somit auf der Geraden $2x = y$ gegen Unendlich. Der Grenzwert dieser Folge $\begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \end{pmatrix}$ ist also der Punkt im Unendlichen, der in Richtung dieser Geraden liegt.

Auch Punkte im $\mathbb{R}^2$ werden von OpenGL in drei-dimensionale homogene Koordinaten umgewandelt. Zum Beispiel wird der Punkt

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \text{in homogenen Koordinaten zu} \quad \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

8.2 Transformationen

Die in Kapitel 2 diskutierten Transformationen werden jetzt für homogene Koordinaten angegeben. Da diese Transformationen die zusätzli-

che w-Komponente berücksichtigen, werden 4×4 Matrizen zur Beschreibung dieser Abbildung benötigt.

8.2.1 Translationen

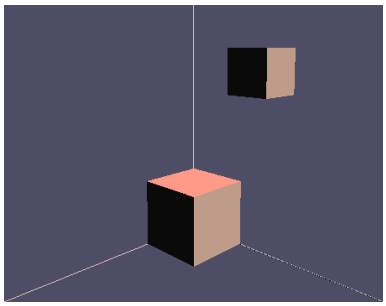
Die in (2.2) definierte Translation $x \mapsto x + v$ kann in homogenen Koordinaten (nicht aber in euklidischen) mit Hilfe einer linearen Abbildung dargestellt werden:

$$T = \begin{pmatrix} 1 & 0 & 0 & v_1 \\ 0 & 1 & 0 & v_2 \\ 0 & 0 & 1 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T^{-1} = \begin{pmatrix} 1 & 0 & 0 & -v_1 \\ 0 & 1 & 0 & -v_2 \\ 0 & 0 & 1 & -v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

denn es gilt:

$$\begin{pmatrix} 1 & 0 & 0 & v_1 \\ 0 & 1 & 0 & v_2 \\ 0 & 0 & 1 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} x + v_1 w \\ y + v_2 w \\ z + v_3 w \\ w \end{pmatrix}.$$

In OpenGL werden diese Matrizen bei Aufruf des Befehls `glTranslate*(`v_1, v_2, v_3`)` erzeugt. Im OpenGL Beispiel wird der Befehl `glTranslatef(`$-1, 2, -3$`)` auf den Einheitsquader angewandt.

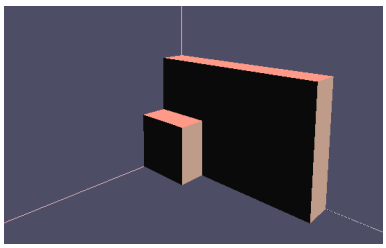


8.2.2 Skalierung

Die Skalierung wird entsprechend zu (2.18) durch

$$A = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} \frac{1}{\lambda_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\lambda_2} & 0 & 0 \\ 0 & 0 & \frac{1}{\lambda_3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

definiert. OpenGL erzeugt die Matrix A mit dem Befehl `glScale*(`$\lambda_1, \lambda_2, \lambda_3$`)`. Den Fall `glScalef(3,2,0.5)` veranschaulicht die Abbildung.



8.2.3 Rotation

Rotationen um die x, y bzw. z Achse werden durch die Matrizen

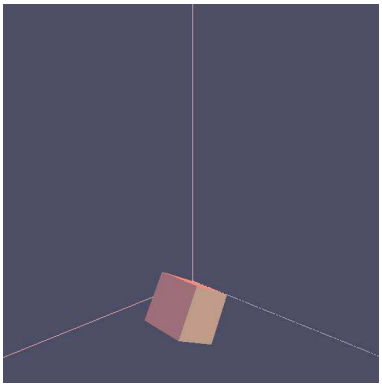
$$B^x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad B^y = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

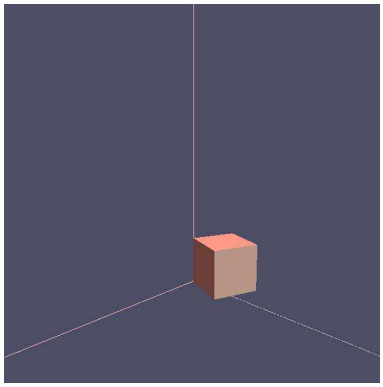
bzw.

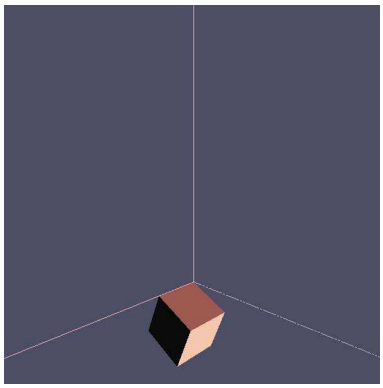
$$B^z = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

definiert. Beachte, dass Rotationen immer orthogonal sind, die inverse Matrix also der transponierten Matrix entspricht.

In OpenGL ist eine Funktion `glRotate*(`φ`,x,y,z)` implementiert, die eine Rotationsmatrix mit dem Winkel φ um die durch (x, y, z) bestimmte Gerade erzeugt. Die Spezialfälle B^x =`glRotate*(`φ`,1,0,0)`, B^y = `glRotate*(`φ`,0,1,0)` und B^z = `glRotate*(`φ`,0,0,1)` zeigen die Animationen.







8.3 Projektionen

Zunächst werden die in Abschnitt 2.15 eingeführten Parallelprojektionen in homogenen Koordinaten betrachtet. Insbesondere kann man so auch auf Ebenen projizieren, die nicht durch den Ursprung des Koordinatensystems verlaufen. Anschließend wird gezeigt, dass auch Zentralprojektionen durch den Übergang in homogene Koordinaten mit einer Matrix beschreibbar sind.

8.3.1 Parallelprojektion

Parallelprojektionen auf eine beliebige Ebene im Raum können wie folgt konstruiert werden: Sei P ein Projektor im $\mathbb{R}^3$, vgl. Abschnitt 2.15, und sei $\eta \in \operatorname{kern}(P)$, d. h. $P\eta = 0$, dann ist

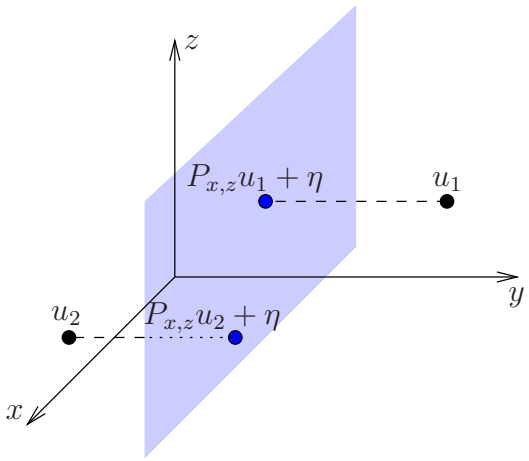
$$Q := \begin{pmatrix} P & \eta \\ 0 & 1 \end{pmatrix}$$

ein Projektor in homogenen Koordinaten. Zum Verständnis der Arbeitsweise dieses Projektors betrachten wir einen Punkt $u \in \mathbb{R}^3$. Es gilt

$$u \leftrightsquigarrow \begin{pmatrix} u \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} P & \eta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ 1 \end{pmatrix} = \begin{pmatrix} Pu + \eta \\ 1 \end{pmatrix} \leftrightsquigarrow Pu + \eta.$$

Die Abbildung illustriert diese Konstruktion anhand des Beispiels

$$P_{x,z} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \eta = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}.$$



8.3.2 Zentralprojektion

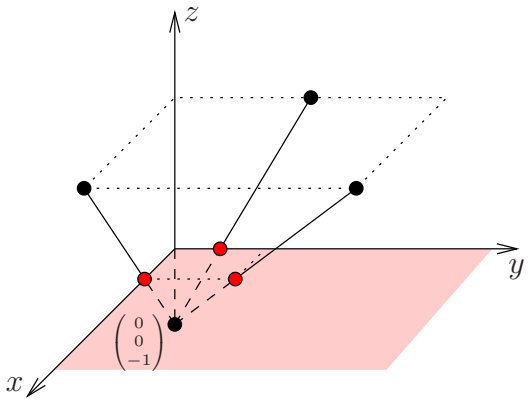
In homogenen Koordinaten können auch Zentralprojektionen mit Matrizen beschrieben werden. Als Beispiel betrachten wir die Abbildung

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Offensichtlich gilt $PP = P$; es handelt sich bei dieser Abbildung um eine Projektion. Sei $u \in \mathbb{R}^3$, dann gilt

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ 1 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ 0 \\ u_3 + 1 \end{pmatrix} \leftrightsquigarrow \begin{pmatrix} \frac{u_1}{u_3+1} \\ \frac{u_2}{u_3+1} \\ 0 \end{pmatrix}.$$

In der Abbildung wird eine Veranschaulichung dieser Projektion gegeben.



Durch diese Projektion wird ein Punkt u entlang des Strahls, der durch u und den Punkt $\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$ definiert wird, in die (x, y)-Ebene projiziert.

Eine beliebige Zentralprojektion kann durch vorgeschaltete Streckungen, Drehungen und Translationen konstruiert werden.

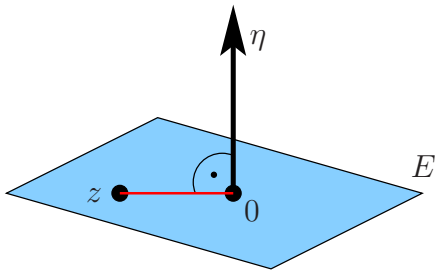
8.4 Normalenvektoren in homogenen Koordinaten

Eine Ebene E im $\mathbb{R}^3$, die durch den Ursprung verläuft, wird eindeutig durch die Angaben eines Normalenvektors η definiert. Wie wir in Kapitel 3 gesehen haben, ist der Normalenvektor nicht eindeutig, denn jeder Vektor der Form $\lambda \cdot \eta$ mit $\lambda \in \mathbb{R} \setminus \{0\}$ ist auch ein Normalenvektor. Zur Beschreibung beliebiger Ebenen im $\mathbb{R}^3$ ist zusätzliche noch die Angabe eines Punktes, der in dieser Ebene liegt, erforderlich.

Im Fall einer Ebene durch den Ursprung liegt ein Punkt z in dieser Ebene E, falls η senkrecht auf z steht, also die Beziehung

$$\langle \eta, z \rangle = \eta^T z = 0$$

gilt, siehe Abbildung.



Die gleiche Überlegung gilt auch für homogene Koordinaten. Wieder wird eine Ebene, durch den Ursprung, eindeutig durch den Normalenvektor $\eta = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_w \end{pmatrix}$ charakterisiert, wobei zumindest eine der Koordinaten $\eta_1, \eta_2, \eta_3, \eta_w$ ungleich Null ist. Somit liegt ein Vektor z genau dann in dieser Ebene, falls

$$\langle \eta, z \rangle = \begin{pmatrix} \eta_1 & \eta_2 & \eta_3 & \eta_w \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_w \end{pmatrix} = \eta_1 z_1 + \eta_2 z_2 + \eta_3 z_3 + \eta_w z_w = 0$$

erfüllt ist.

Beachte, dass zur Beschreibung einer *euklidischen Ebene* mindestens eine der Koordinaten η_1, η_2, η_3 von Null verschieden sein muss.

Falls $\eta_1 = \eta_2 = \eta_3 = 0$ und $\eta_w > 0$ gilt, so wird eine Ebene im Unendlichen beschrieben, denn nur Vektoren der Form $\begin{pmatrix} a \\ b \\ c \\ 0 \end{pmatrix}$ stehen senkrecht auf diesem Normalenvektor:

$$\left\langle \begin{pmatrix} 0 \\ 0 \\ 0 \\ \eta_w \end{pmatrix}, \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \right\rangle = 0 \cdot a + 0 \cdot b + 0 \cdot c + d \cdot \eta_w \stackrel{!}{=} 0 \quad \Rightarrow \quad d = 0.$$

Somit liegen nur Punkte im Unendlichen in dieser Ebene.

Literaturverzeichnis

[1] MAPLE, 2015. www.maplesoft.com.

[2] MATLAB, 2015. www.mathworks.de.

[3] SCILAB, 2015. www.scilab.org.

[4] H. Amann and J. Escher. *Analysis. I*. Grundstudium Mathematik. [Basic Study of Mathematics]. Birkhäuser Verlag, Basel, 2006.

[5] H. Amann and J. Escher. *Analysis. II*. Grundstudium Mathematik. [Basic Study of Mathematics]. Birkhäuser Verlag, Basel, 2008.

[6] W.-J. Beyn and T. Hüls. NUMLAB, 2015. www.math.uni-bielefeld.de/~huels/lehre_de.html.

[7] P. de Faget de Casteljau. De Casteljau's autobiography: My time at Citroën. *Computer Aided Geometric Design*, 16:583–586, 1999.

[8] P. Deuflhard and A. Hohmann. *Numerische Mathematik. 1.* de Gruyter Lehrbuch. [de Gruyter Textbook]. Walter de Gruyter & Co., Berlin, fourth edition, 2008. Eine algorithmisch orientierte Einführung. [An algorithmically oriented introduction].

[9] G. Engeln-Müllges, K. Niederdrenk, and R. Wodicka. *Numerik-Algorithmen: Verfahren, Beispiele, Anwendungen (Xpert.press)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2010.

[10] G. Farin. *Kurven und Flächen im Computer Aided Geometric Design*. Vieweg Verlag, 1994.

[11] D. W. Fellner. *Computergrafik*. BI-Wissenschaftsverlag, 1992.

[12] G. Fischer. *Lineare Algebra*. Vieweg-Studium : Grundkurs Mathematik. Vieweg, 2009.

[13] O. Forster. *Analysis 1 und 2*. Vieweg Verlag, neuste Auflage.

[14] R. W. Freund and R. H. W. Hoppe. *Stoer/Bulirsch: Numerische Mathematik 1.* Springer-Verlag Berlin Heidelberg, 2007.

[15] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[16] G. Hämmerlin and K.-H. Hoffmann. *Numerische Mathematik*. Springer-Lehrbuch. Springer-Verlag, Berlin, fourth edition, 1994.

[17] M. Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Vieweg + Teubner, Wiesbaden, third edition, 2009.

[18] F. Locher. *Numerische Mathematik für Informatiker*. Springer Verlag, 1993.

[19] J. Naas and W. Tutschke. *Große Sätze und schöne Beweise der Mathematik*, volume 63 of *Deutsch-Taschenbücher [Deutsch Paperbacks]*. Verlag Harri Deutsch, Thun, 1989. Identität des Schönen, Allgemeinen, Anwendbaren. [Identity of the beautiful, the general, and the applicable], Reprint of the 1986 original.

[20] G. Opfer. *Numerische Mathematik für Anfänger*. Vieweg Verlag, 2008.

[21] B. Pareigis. *Analytische und projektive Geometrie für die Computer-Graphik*. B. G. Teubner, Stuttgart, 1990.

[22] W. Rudin. *Reelle und komplexe Analysis*. R. Oldenbourg Verlag, Munich, 1999. Translated from the third English (1987) edition by Uwe Krieg.

[23] H. Schwarz and N. Köckler. *Numerische Mathematik*. B. G. Teubner, Stuttgart, 2011.