Daten fitten und approximieren mit *Mathematica* und *M@th Desktop*

Exzerpt der Diplomarbeit von Markus Smole, Graz

Zusammenfassung

In diesem Artikel wird gezeigt, wie Schüler Daten von einer Kurve in parametrisierter Form fitten als auch approximieren können. Als praktische Anwendung der gefitteten Kurve wird das Bogenlängenintegral vorgestellt. Die gefittete Kurve wird verwendet, um die Länge des Radweges entlang der Saar in Ockfen zu bestimmen.

1 Einleitung

Eine Landkarte enthält eine Fülle von geometrischen Objekten: Strassen, Flüsse, Bäche, Forstwege, etc. Im allgemeinen ist es schwierig, diese Objekte analytisch zu beschreiben. Eine Möglichkeit, diese Kurven zu erfassen, bietet das CAGD [1]: Interpolationskurven, Splinekurven, Bézierkurven, etc.[5]. Die Herausforderung besteht darin, durch Datenpunkte gegebene Kurven mittels geeigneten, den Schülern bekannten Funktionen anzunähern. Dabei fördert man zwei Wissensbereiche des Schülers:

- Spielerische Modifikation von Funktionen anhand ihres Funktionsgraphen durch geschicktes Manipulieren einzelner Parameter
- Sinnvolles Einsetzen von Computeralgebrasystemen (CAS)

CAS sollen in den Mathematikunterricht integriert werden und stellen ein bedeutendes Hilfsmittel dar [1]. Diese Programme bieten dem Schüler beim Aufsuchen und Diskutieren von Zielfunktionen zu vorgegebenen Datenpunkten eine gute Möglichkeit, sich intensiv mit dem Modellieren von Funktionen zu beschäftigen.

[1] Computer Aided Geometric Design

2 Der Radweg - ein Beispiel

Die Gemeinde Ockfen[2] plant einen Radweg von der Brücke über die Saar bis zu einer als Erholungsgebiet benutzen Sandbank. Dabei verläuft der Radweg genau entlang dem Flussbett und kostet pro 100 Meter inklusive Schotterung und Asphaltierung 1200 Euro (Abb. 1).

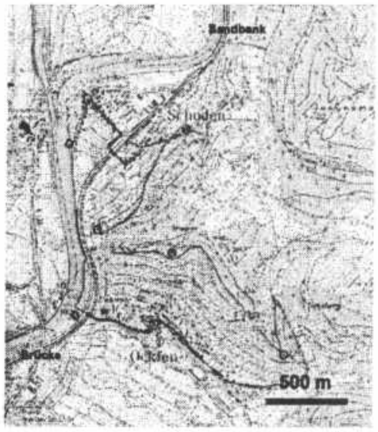

Abbildung 1: Radweg entlang der Saar

Wie teuer kommt der Gemeinde der Radweg?

2.1 Einlesen der Daten

Um überhaupt eine Kurve entlang des Flussbettes zu legen, benötigen wir zuerst Datenpunkte, die exakt dem Flusslauf entsprechen. *Mathematica* bietet dazu eine zugleich einfache wie perfekte Möglichkeit:

- Wir gehen davon aus, dass die Landkarte in digitaler Form[3] vorhanden ist. Mit dem Menü *File* → *Import* importiert man die digitalisierte Landkarte in das *Mathematica* Arbeitsblatt.
- Klickt man mit der Maus den Flusslauf entlang während man gleichzeitig die *STRG* - Taste (*COMMAND* - Taste am Apple - Macintosh) gedrückt hält, werden alle diese Punkte auf der Grafik markiert. Ihre Koordinaten sind dabei in der linken unteren Ecke des *Mathematica* Arbeitsblatt zu sehen[4].
- Durch das Drücken der Tasten *STRG + C* und danach *STRG + V* fügt man die Koordinaten in Form einer Liste in das *Mathematica* Arbeitsblatt ein. Diese Liste wird gleich unter dem Namen `river`

[2] http://home.t-online.de/home/klemens.minn/Geo_Karte_1_n.JPG

[3] etwa im Bitmapformat bmp, Breite ca. 300 Pixel

[4] *Mathematica* setzt den Koordinatenurprung in die linke untere Ecke der Grafik, der Punkt mit den Koordinaten (1, 0) wird in die rechte untere Ecke gelegt.

abgespeichert, um so immer darauf zugreifen zu können. `river` enthält 43 Messpunkte.

```
In[1]:= river = {{0.1,0.246667},
        {0.12,0.256667},
        ...
        ,{0.573333,1.07667},{0.57,1.1}};
```

- Ein Plot der Datenpunkte zeigt den Vergleich des eigentlichen Flusslaufs und Position der eingelesenen Datenpunkte (Abb. 2).

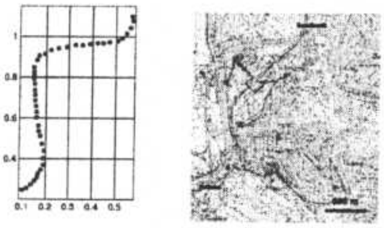

Abbildung 2: Datenpunkte und Flusslauf

3 Der Vorteil von Kurven in Parameterdarstellung

Schüler der Sekundarstufe II bringen den Begriff Kurve primär mit dem Graphen einer Funktion $y = f(x)$ in Verbindung. Jedoch sind diese Funktionsgraphen meist nur schlecht geeignet, um beliebige Datenpunkte anzunähern.
Aus diesem Grund ist es notwendig, die gegebenen Datenpunkte mittels Kurven zu interpolieren bzw. approximieren. Die Definition einer ebenen Kurve lautet dann [5]:

$$\vec{X} : [a, b] \subset \mathbb{R} \to \mathbb{R}^2 \quad wobei \quad \vec{X}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

mit dem Parameter $t \in [a, b] \subset \mathbb{R}$ und den Koordinatenfunktionen $x, y : [a, b] \to \mathbb{R}$

Ein Beispiel: Der Kreis mit dem Mittelpunkt O und dem Radius r genügt der Gleichung $x^2 + y^2 = r^2$. Auflösung nach y ergibt $y = +\sqrt{r^2 - x^2}$ und $y = -\sqrt{r^2 - x^2}$, also zwei explizite Funktionen, die jeweils auf $[-r, r]$ definiert sind. Dabei bekommt die Kreisdarstellung eine gewisse „Unhandlichkeit“. Darüber hinaus wird von Schülern oft die zweite, negative Lösung vergessen (Abb. 3, *links*).

Da der Kreis ein gut bekanntes geometrisches Objekt ist, wird den Schülern das Fehlen der zweiten Hälfte spätestens beim Zeichnen des Kreises auffallen. Bei

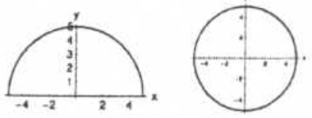

Abbildung 3: *links*: explizit; *rechts*: parametrisierte Darstellung

Funktionen, deren Graph nicht bekannt ist, besteht diese grafische Kontrolle nicht mehr. Deshalb ist es wichtig, den Schülern damit ein Problembewußtsein für das Finden *aller* Lösungen zu vermitteln.
Diese „Unhandlichkeit“ der beiden Lösungen fällt mit der Parameterdarstellung des Kreises:

$$\vec{X}(t) = \begin{pmatrix} r\sin(t) \\ r\cos(t) \end{pmatrix}$$

mit $t \in [0, 2\pi]$ (Abb. 3, *rechts*).

Man erkennt, dass mit parametrisierten Darstellungen wesentlich mehr Modellierungsspielraum vorhanden ist. Damit steht uns ein Werkzeug zu Verfügung, das es erlaubt, den Verlauf von Kurven besser zu beschreiben.

4 Koordinatenfunktionen und Datenpunkte

Wie sind die 43 Datenpunkte durch eine Kurve $\vec{X}(t)$ zu beschreiben? Dazu wählen wir folgende Vorgangsweise: Zuerst ist es notwendig, die Datenpunkte in die Form $x(t)$ und $y(t)$ zu bringen. Danach werden $x(t)$ als auch $y(t)$ von den Schülern durch spezielle Funktionen gefittet bzw. approximiert.
Die Kurve $\vec{X}(t)$ ist nun bestimmt und dient zur Berechnung der Bogenlänge. Das Integral liefert dann die Länge des Radweges

Der Parameter t ist so einfach wie möglich zu wählen. Da 43 Datenpunkte vorhanden sind, soll t das Durchlaufen der Datenpunkte beschreiben. Das ergibt einen Wertebereich $t \in [1, 43]$. Abbildung 4 visualisiert die Koordinaten eines Punktes.

Die Punkte der Koordinatenfunktionen bezeichnen wir mit X_i und Y_i (Abb. 5):

$$X_i = (i, x(i)) \qquad Y_i = (i, y(i))$$

Zwei kleine *Mathematica* Programme [9] schreiben die Datenpunkte in $X_i(i, x(i))$ und $Y_i(i, y(i))$ um. `riverX` beinhaltet die Punkte X_i, `riverY` die Punkte Y_i.

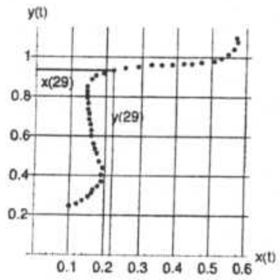

Abbildung 4: Datenpunkte $P_{29}(x(29), y(29))$ der gesuchten Kurve

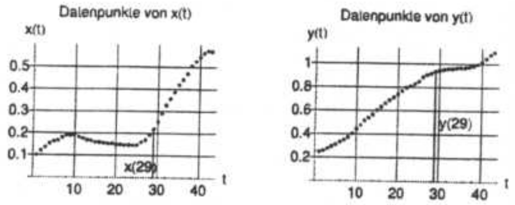

Abbildung 5: Datenpunkte der Koordinatenfunktionen der gesuchten Kurve

```
In[2]:= riverX = MDDataX[river]
Out[2]= {{1,0.1},{2,0.12},
         ...
         ,{42,0.573333},{43,0.57}}
In[3]:= riverY = MDDataY[river]
Out[3]= {{1,0.246667},{2,0.256667},
         ...
         ,{42,1.07667},{43,1.1}}
```

Zum Vergleich der Koordinaten wirde die Liste `river` nochmals angeführt.

```
In[4]:= river = {{0.1,0.246667},{0.12,0.256667},
         ...
         ,{0.573333,1.07667},{0.57,1.1}};
```

5 Funktionen *basteln*

5.1 Eine Familie von Funktionen zum Modellieren von Koordinatenfunktionen

Um eine freie Kurve mathematisch zu modellieren, geht man meist von einer bekannten, parameterabhängigen Kurve aus. Man modifiziert die Parameter um die bekannte Kurve den Datenpunkten der freien Kurve möglichst genau anzupassen. Hier verfolgen wir eine andere Strategie: Die Ausgangsfunktion $anfang(t)$, die etwa durch lineare Approximation der gegebenen Datenpunkte gefunden wird, soll nun durch Addition geeigneter Deformationsfunktionen $deform_1(t),\ldots,deform_N(t)$ verändert werden:

$$ziel(t) = anfang(t) + \sum_{i=1}^{N} deform_i(t)$$

Eine geeignete Deformationsfunktion soll bei Addition zur Ausgangsfunktion $anfang(t)$ nur einen kleinen Bereich modifizieren. Dadurch bleibt das Modellieren überschaubar. Diese Eigenschaften erfüllt folgende Funktion:

$$deform(t) = \frac{k}{l(t-a)^2+1}$$

Abbildung 6 zeigt die Funktion für $k = 1$, $l = 1$ und $a = 0$.

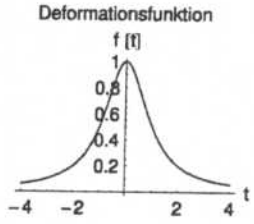

Abbildung 6: Deformationsfunktion

Diese Familie von Deformationsfunktionen $deform(t) = \frac{k}{l(t-a)^2+1}$ ist nur eine Möglichkeit. Darüber hinaus existieren noch eine Vielzahl von geeigneten Funktionen, welche hier nicht behandelt werden. Insbesondere besteht natürlich die Möglichkeit, die Kurve in zwei oder mehrere Kurvenstücke zu zerlegen. Die kreisförmigen Teile können durch die Parameterform des Kreises dargestellt werden. Beim Auffinden der Ausgangsfunktion $anfang(t)$ kann unter anderem auch die Kurvendiskussion eingesetzt werden, um eine passende Funktion zu finden.

5.2 Veränderung der Deformationsfunktion

Eine Funktion hängt von einem oder mehreren Parameteren ab. Die Auswirkung einer Änderung dieser Parameter läßt sich am besten am Bild des Funktionsgrafen studieren. M@th Desktop bietet eine Movie - Funktion an[5]: Dabei wird ein kleiner Film erzeugt, dessen einzelne Bilder durch verschiedene Werte des

[5]Menü *M@th Desktop* → *Movie erzeugen*

spezifizierten Parameters erzeugt werden.

Verschiebung von *deform*(t) längs der t-Achse

Die Notwendigkeit der lokalen Veränderung von *anfang*(t) verlangt, den Hochpunkt der Funktion *deform*(t) bei $t = 0$ an eine bestimmte t-Stelle a zu verschieben. Dies gelingt, für die Schüler leicht verständlich, indem man t durch $(t - a)$ ersetzt. Die Funktion hat dann folgende Form ($k = l = 1$):

$$deform(t) = \frac{1}{(t-a)^2 + 1}$$

Ein Movie visualisiert den Verschiebungseffekt.

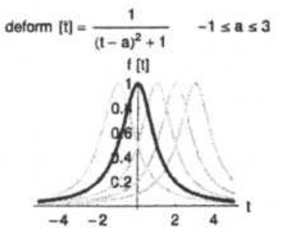

Abbildung 7: Verschiebung längs der t-Achse

Damit ist es den Schülern durch das Verschieben nun möglich, die Funktion *anfang*(t) an jeder Stelle zu modellieren.

Veränderung der Höhe und der Steigung

In analoger Weise kann der Lehrer auch die Auswirkungen der anderen beiden Parameter k und l zeigen (Abb. 8).

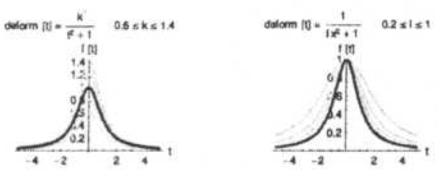

Abbildung 8: Veränderung der Höhe bzw. Steigung

Mit dieser Deformationsfunktion und ihren „Klonen" können sich Schüler nun spielerisch daran versuchen, die Datenpunkte der Koordinatenfunktionen zu approximieren.

6 Der Radweg - das hat die Gemeinde zu zahlen

Fassen wir zuerst einmal die bereits erfolgten Rechenschritte und gewonnenen Erkenntnisse zusammen:

- In Abschnitt 2.1 haben wir die Datenpunkte in *Mathematica* eingelesen und unter `river` abgespeichert.
- In Abschnitt 4 wurden die Datenpunkte der Koordinatenfunktionen bestimmt und unter `riverX` bzw. `riverY` gespeichert.
- Eine Modellierungsmethode mit Deformationsfunktionen *deform*(t) wurde in Abschnitt 5.1 vorgestellt und genauer untersucht.

6.1 Generieren der Lösungskurve

Koordinatenfunktion der y-Achse: $y(t)$

Gemäß Abschnitt 5.1 sucht man zuerst eine Funktion, welche die Punkte annähert. Hier verwenden wir eine lineare Funktion `anfang[t]`, die durch graphisches Probieren oder durch lineare Approximation gefunden wird. Dabei ist jedoch zu beachten, dass die Ausgangsfunktion `anfang[t]` während des Modellierens eventuell verändert werden muß, um Verschiebungen längs der vertikalen Koordinatenachse, die bei Addition der Deformationsfunktionen `deform[t]` entstehen, auszugleichen.

Nun bleibt es den Schülern überlassen, jene Bereiche, in denen eine zu starke Abweichung vorliegt, durch geeignete Deformationsfunktionen `deform[t]` zu korrigieren. Dabei kann zusätzlich zu der graphischen Kontrolle ein Fehlerquadratsummentest (LST) durchgeführt werden. Ein den Schülern vorgegebener Fehlerwert, den die Zielfunktion unterschreiten soll, beschränkt ein langes Herumprobieren. Darüberhinaus lassen sich so auch kleine „Wettrennen" organisieren. Für unser konretes Beispiel geben wir uns einen Fehlerwert von 0.003 vor. Wendet man den LST auf die Ausgangsfunktion `anfang[t]`, so ergibt das:

```
In[5]:= FehlerquadratSumme[anfangY[t],riverY]
Out[5]= 0.409236
```

Sechs verschiedene Funktionen `deform1[t]`,..., `deform6[t]` werden verwendet, um eine hinreichend kleine Abweichung zu bekommen. Man beginnt mit der Funktion `deform1[t]`, die den *Höcker* bei $t = 29$ erzeugt und fährt mit den weiteren Deformationen analog fort:

In[6]:= `anfang[t_] := 0.019 t + 0.225;`

$$\text{deform1[t\_]} := \frac{0.22}{0.01\,(t-29)^2+1};$$

⋮

$$\text{deform6[t\_]} := +\frac{0.025}{0.08\,(t-0)^2+1};$$

```
y[t_] := anfang[t] + deform1[t]
+deform2[t] + deform3[t]+
deform4[t] + deform5[t]+
deform6[t];
```

Ein Plot der Kontrollpunkte und der Zielfunktion `y[t]` zeigt den Schülern die Qualität der Ergebnisse (Abb. 9).

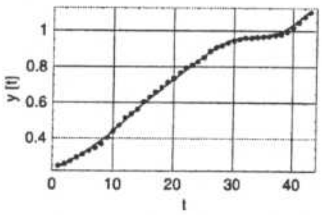

Abbildung 9: Kontrollpunkte mit Zielfunktion `y[t]`

Darüber hinaus ergibt der LST einen Fehler von 0.0029.

Koordinatenfunktion der x-Achse: $x(t)$

Die Schüler sollen nun auch die in *Mathematica* implemtierten Algorithmen kennen lernen. Neben der manuellen Methode zur Funktionsfindung bietet *Mathematica* nämlich auch interne Programme an, welche dies automatisch erledigen.

Die Funktionen heißt `PolynomialFit`. Sie berechnet ein approximierendes Polynom vom Grad n. Dabei ist zu beachten, dass die Erhöhung des Polynomgrades nicht immer zu einem besseren Ergebnis führt, da Polynome für grosse n leicht oszillieren.

Ein Plot der Datenpunkte und der Zielfunktion `x[t]` (Abb. 10) zeigt den Schülern die grafische Qualität des Ergebnisses.

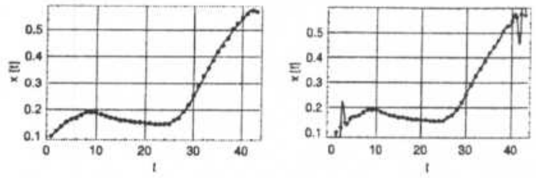

Abbildung 10: links: $x(t)$ mit $n = 10$, rechts: $n = 30$

Kombinieren der Zielfunktionen zur Kurve in Parameterdarstellung

Im letzten Schritt sollen die Schüler die beiden im vorangegangenen Kapitel berechneten Funktionen mittels eines parametrischen Plots anzeigen lassen (Abb. 11) und mit den Datenpunkten vergleichen. Dabei befindet sich die Brücke, wo der Radweg beginnt, beim Parameterwert 1 und das Ende des Radwegs, die Sandbank, beim Wert 41. Die beiden zusätzlichen Punkte P_{42} und P_{43} helfen beim Modellieren.

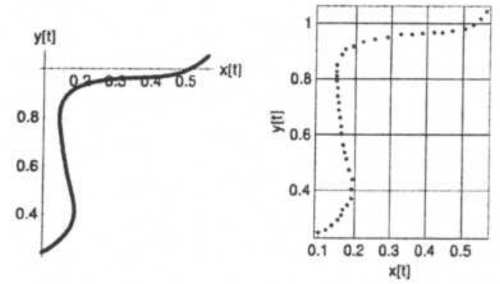

Abbildung 11: *links:* Fertigen Modell, *rechts:* Messdaten

Den Schülern ist nun die Kurve bekannt. Sie können die Länge des radweges ausrechnen.

6.2 Bogenlänge zur Berechnung der Flusslänge

Der Radweg kostet pro 100m Euro 1200.-. Somit müssen die Schüler die Länge des Flusses von der Brücke bis zur Sandbank bestimmen. Mathematisch entspricht das dem Berechnen der Bogenlänge l einer parametrisierten Funktion in den Grenzen $t \in [a, b]$. Die Formel dafür lautet [6]:

$$l(a,b) = \int_a^b \sqrt{x'(t)^2 + y'(t^2)}\,dt$$

M@th Desktop bietet dafür die vorgefertigte Palette Bogenlänge[6] (Abb. 12), mit der man die Bogenlängenberechnung schnell und einfach durchführen kann.

In[7]:= $\text{NIntegrate}\left[\sqrt{x'[t]^2 + y'[t]^2},\ \{t, 1, 41\}\right]$ //Chop

Out[7]= 1.13629

Berücksichtigung des Maßstabes

[6]Menü *MD Integr* → *Bogenlänge*

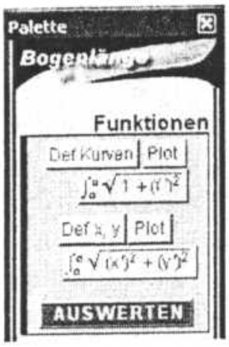

Abbildung 12: Bogenlänge Palette

Da die Länge von 500 Metern auf der Quellgrafik eingezeichnet ist, braucht man nur die Länge dieser Strecke bezogen auf das Bildkoordinatensystem von *Mathematica* bestimmen, um dann den Maßstab zu berechnen. `dist` sind der Anfangspunkt und Endpunkt der 500 Meter Markierung.

Damit wissen die Schüler, dass den 500 Meter 0.22 Einheiten des *Mathematica* Koordinatensystems entsprechen. Durch Umrechnung der berechneten Bogenlänge ergibt sich schlussendlich eine Flusslänge von 2582 Metern. Der Radweg schlägt sich somit mit Euro 30948.- zu Buche.

7 Weiterführendes und Ergänzungen

In der Praxis wird man bei dicht gesetzten Datenpunkten einfach die Summe der Differenz benachbarter Datenpunkte bilden, um zur Kurvenlänge zu kommen.

Insbesondere sei hier auch vermerkt, dass die in *Mathematica* implementierten Algorithmen zur Interpolation / Approximation von Punkten im allgemeinen genauer und vorallem schneller arbeiten als dies mit der händischen Methode möglich ist. Dies soll jedoch nicht davon abschrecken, sich mit weiteren Deformationsfunktionen zu beschäftigen, da Schüler erst durch manuelles Rechnen Sicherheit im Umgang mit Funktionen bekommt. Nähere Informationen zu *Mathematica* findet man unter `www.wolfram.com`

M@th Desktop ist eine Mathematiklernsoftware für die Sekundarstufe II und baut auf *Mathematica* auf. Nähere Informationen findet man im Internet unter `www.deltasoft.at`.

Die *Mathematica*-Programme `MDDataX[]`, `MDDataY[]` und `Fehlerquadratsumme[]` werden Interessierten gerne zugesandt: markussmole@gmx.at

Literatur

[1] FUCHS, Karl Josef: Computeralgebra - Neue Perspektiven im Mathematikunterricht; Habilitationsschrift an der naturwissenschaftlichen Fakultät der Universität Salzburg zur Erlangung der Venia Docendi aus Didaktik der Mathematik

[2] REICHEL, MÜLLER, HANISCH, LAUB: Lehrbuch der Mathematik 7-8, Verlag Hölder-Pichler-Tempsky, Wien 1992

[3] VOGLER, Hans: Skriptum zur Vorlesung Differentialgeometrie, TU Graz, 2001

[4] M@th Desktop: Zusatzmodul für *Mathematica*; Helpbrowser M@th Desktop Daten fitten, Graz 2003, `www.deltasoft.at`

[5] RÖSCHEL, Otto: Skriptum zur Vorlesung Freiformkurven und Freiformflächen, TU Graz 2002

[6] BRONSTEIN, SEMENDJAJEW, MUSIOL, MÜHLIG: Taschenbuch der Mathematik, Verlag Harri Deutsch, 1995

[7] SIMONOVITS, Reinhard: Projekt M@th Desktop. Öserreichische Mathematische Gesellschaft, Heft 32, Dezember 2000

[8] SILLER, Hans-Stefan: Auf Mathematica basierende Lerneinheiten zur fundamentalen Idee der Modellbildung, illustriert an Extremwertbeispielen und Beispielen der Integralrechnung mit M@th Desktop, Diplomarbeit, Graz, 2002

[9] SMOLE, Markus: Auf Mathematica und M@th Desktop basierende unterrichtssequenzen zur Approximation von Radwegen, sowie zur Regression und Korrelation, Diplomarbeit, Graz, 2003, `www.uni-graz.at/imawww diplomarbeiten/index.html`

Markus Smole, Frühlingstrasse 22, A-8053 Graz, ist Diplomand am Institut für Mathematik an der Karl-Franzens-Universität Graz. Email: markussmole@gmx.at