FACHHOCHSCHULE
ERFURT UNIVERSITY
OF APPLIED SCIENCES
Angewandte
Informatik

Belegarbeit

in der Angewandten Informatik

Sommersemester 2012

Mathematik Aufbau

Interpolationsverfahren

Das kubische Spline-Verfahren

Tobias Schwandt

29.07.2012

Dipl.-Math. Anja Haußen

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

GUI	Graphical User Interface

1 Einleitung

Das Ziel dieses Belegs ist eine Einführung in das Thema der Interpolationsverfahren zu geben. Weil dieses Thema sehr weitreichend und in der Komplexität äußerst umfangreich ist, beschäftigt sich dieser Beleg insbesondere mit den »Spline-Interpolationen« aus den Bereich der Stückweiten Interpolations Verfahren.

In Folge der Arbeit entsteht, neben der Ausarbeitung, eine kleine Anwendung mit dessen Hilfe sich eine Interpolationskurve anzeigen lässt.

1.1 Motivation

Schon früher waren Interpolationen von besonderer Wichtigkeit. Sie wurden benötigt, um verschiedenste Probleme im Alltag zu berechnen und als hilfreiche Unterstützung im Handwerk. Heutzutage werden Interpolationen verwendet, um Wegpunkte zu berechnen, Statistiken auszuwerten oder Zwischenpunkte zu ermitteln.

Besonders im Bereich der Computerspiele werden Interpolationen dafür verwendet zwischen zwei Logikturns eine harmonische Animation zu erhalten, bzw. Einheiten realistisch im Raum zu bewegen.

2 Grundlagen

Innerhalb der folgenden Seiten, soll ein Einblick in die Thematik der Interpolation aufgezeigt werden. Hierbei wird zum einen die Geschichte der Interpolationsverfahren betrachtet aber auch aktuelle Interpolationsverfahren aufgezeigt und erklärt.

Hauptaugenmerk liegt bei dieser Belegarbeit in der Bearbeitung der »Kubischen Spline-Interpolation«. Innerhalb dieses Abschnitts wird ein detaillierter Blick in die Theorie dieser besonderen Interpolation geworfen. Die Theorie soll die Basis für die Anwendung bilden, welche als Teil dieser Arbeit erstellt wurde.

2.1 Geschichte der Spline-Interpolation

Bereits im 17. Jahrhundert wurden Interpolationsverfahren im Schiffsbau benötigt. Hier wurden Logarithmentafeln gefunden, welche dies belegen. [Hoc08]

In der frühen Seefahrt wurden Straklatten verwendet, welches eine elastische Latte ist, die zwischen zwei Punkte gespannt wurde. Durch die Spannung entstanden harmonische Kurven - die sogenannten Splines. Die Straklatten bilden eine Linie durch alle Punkte ohne hoher Biegeenergie und mit sehr kleiner Krümmung [Her11, 170].

2.2 Spline-Interpolation in der Theorie

In der Mathematik kommt es häufiger zu der Situation, dass Messdatenreihen existieren, welche Löcher bei der Messung aufweisen. Diese Löcher bilden Mittelwerte zwischen zwei verschiedenen Messpunkten. Diese sind jedoch in den meisten Fällen nicht gradlinig, sondern bilden eine Kurve, welche sich langsam dem Ergebnis anschmiegen soll. Speziell für dieses Problem bei klassischen Interpolationen sind die Splines - (engl.: biegsames Kurvenlineal) - erfunden worden. [Kup02, 5] [Ung03, 2]

Für die Berechnung und Darstellung von Splines sind verschiedene Randbedingungen erforderlich, um alle Freiheitsgrade im mathematischen Ansatz zu eliminieren. Hierbei ist insbesondere eine Definition des Kurvenanfang und Kurvenende nötig. Diese werden durch Tangenten beschrieben - die Ableitung der Funktion in den Endpunkten. Bei »Kubischen Splines« sind folgende Randbedingungen üblich:

- Angabe von Tangenten in den Endpunkten für eine harmonische Start- und Endphase der Kurve
- Festlegung der 2. Ableitung in den Endpunkten auf den Wert Null - bedeutet eine natürliche Randbedingung
- Verknüpfung der Endpunkte des Splines
- Die sogenannte *Bessel'sche Randbedingung*

[BB05, 139]

Die wichtigste Eigenschaft von Splines ist, dass sie versuchen über zwei Punkte - auch Stützstellen genannt [Loc93, 66] - zu interpolieren und nicht über alle Stützstellen einer Funktion. Man spricht hierbei auch von einer stückweisen Interpolation. Hierdurch passen sie sich individuell den Werten an und wirken dadurch glättend und besitzen eine geringe Welligkeit. Kubische Splines werden hierbei über einem Polynom 3. Grades interpoliert [Her11, 190].

Für alle Splines $s(x)$ gelten folgende 3 Regeln:

1. $s(x_i) = y_i$ für $i = 0, 1, ..., n$ [Her11, 183]
2. s ist in $[x_0, x_n]$ zweimal stetig differenzierbar [Her11, 183]
3. die Gesamtkrümmung von s ist minimal [CK07, 385]

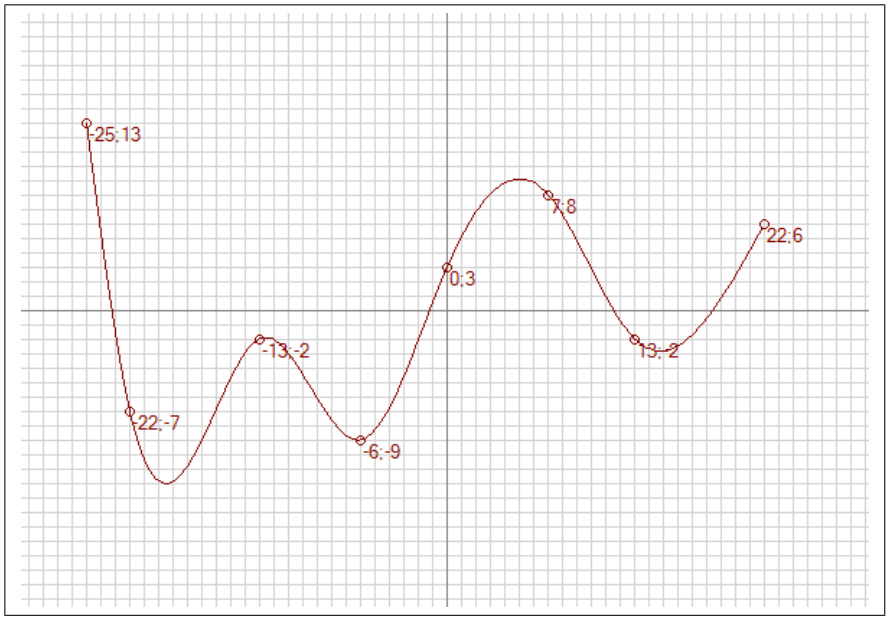

Abbildung 2.1: Beispiel für einen kubischen Spline

2.2.1 Herleitung von kubischen Splines

Gegeben:

Gegeben ist eine Menge von Stützstellen (x_i, y_i) für $i = 0, ..., n$, so dass alle Stützstellen paarweise wie folgt angeordnet sind: $a = x_0 < x_1 < ... < x_n = b$. Dies gilt für alle Stützstellen des Gitters.

Gesucht:

Gesucht ist eine Menge von kubischen Splinefunktionen $s(x)$ für die gilt, dass $s(x)$ auf dem Gitter $[x_0, x_n]$ zweimal stetig differenzierbar ist. Es muss

$$s_i(x_{i+1}) = s_{i+1}(x_{i+1}),\ s_i'(x_{i+1}) = s_{i+1}'(x_{i+1})$$

und

$$s_i''(x_{i+1}) = s_{i+1}''(x_{i+1})$$

gelten.

Lösung:

Wie bereits erwähnt sind »Kubische Splines« $s_i(x)$ eine Interpolation am Polynom 3. Grades. Gegeben sei also ein Polynom 3. Grades [CK07, 387]:

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

Dieses allgemeine Polynom muss nun nach jedem Koeffizienten umgestellt werden, um diese berechnen zu können. Hierzu, und um die Krümmung des Splines berechnen zu können, wird die 1. und 2. Ableitung des Polynoms benötigt welche wie folgt aussieht [CK07, 387]:

$$s_i'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

$$s_i''(x) = 6a_i(x - x_i) + 2b_i$$

Für die Übersicht wird im weiteren Verlauf die erste Ableitung im Punkt $P(x_i, y_i)$ als S_i bezeichnet und die zweite Ableitung im Punkt $P(x_{i+1}, y_{i+1})$ als S_{i+1}. Gleichzeitig setzen wir voraus, dass $h_i = x_{i+1} - x_i$ gilt. Mit diesen Vorraussetzungen kann nun nach den Parametern a_i, b_i, c_i, d_i umgestellt werden. [Ung03, 3 ff.]

$$d_i = y_i$$

$$s_i''(x_i) = S_i = 6a_i(x_i - x_i) + 2b_i = 2bi$$

$$b_i = \frac{S_i}{2}$$

$$s_i''(x_{i+1}) = S_{i+1} = 6a_i(x_{i+1} - x_i) + 2b_i = 6a_ih_i + 2b_i = 6A_ih_i + s\frac{S_i}{2} = 6a_ih_i + S_i$$

$$a_i = \frac{S_{i+1} - S_i}{6h_i}$$

$$s_i(x_{i+1}) = y_{i+1} = a_ih_i^3 + b_ih_i^2 + c_ih_i + d_i$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_iS_i + h_iS_{i+1}}{6}$$

Aus den gegebenen Parametern lassen sich unterschiedliche Matrizen herleiten, je nach der Bedingung die erfüllt sein soll. Es wird grundlegend unterschieden zwischen »Natürlichen Splines«, »Clamped Spline« und »Not-A-Knot« [CK07, 387].

Zur weiteren Beschreibung werden »Natürliche Splines« verwendet. Nun müssen die beiden unbekannten S_i und S_{i+1} berechnet werden. Bekannt ist aus [Ung03, 4], dass die Steigung des rechtssteitigen Polynoms gleich dem linksseitigen Polynom im Punkte $P(x_i, y_i)$ ist. Aus den Ableitungen im Punkt P ergeben sich:

$$s_i'(x_i) = 3a_i(x_i - x_i)^2 + 2b_i(x_i - x_i) + c_i = c_i$$

$$s_{i-1}(x_i) = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} = 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1}$$

Unter der Vorraussetzung, dass

$$s_i'(x_i) = s_{i-1}'(x_i)$$

gültig ist und die Gleichungen von a_i, b_i und c_i eingesetzt werden ergibt sich folgendes Gleichungssystem:

$$c_i = 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1}$$

eingesetzt und vereinfacht

$$h_{i-1}S_{i-1} + (2h_{i-1} + 2h_i)S_i + h_i Si + 1 = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)$$

Folgendes Gesamtkonzept entsteht zum Berechnen der Koeffizienten:

$$\begin{bmatrix} 2(h_0+h_1) & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_3 & & \\ & h_2 & 2(h_2+h_3) & h_3 & \\ & & \cdot & & \\ & & & \cdot & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \cdot \\ \cdot \\ S_{n-1} \end{bmatrix} = 6 \begin{bmatrix} \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \cdot \\ \cdot \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \end{bmatrix}$$

2.2.2 Beispiel in der Theorie

Für ein Beispiel zur Berechnung eines kubischen Splines nehmen wir die folgenden Stützstellen aus der Tabelle 2.1 an. Die Werte für das Beispiel sind aus [Ung03, 7-8] entnommen.

x	1	3	4	5	6	7	8	9
y	2	6	3	4	-7	3	5	3

Tabelle 2.1: Stützstellen innerhalb eines Koordinatensystems

x_i	y_i	h_i	$f[x_i, x_{i+1}]$
1	2	1	1
2	3	1	3
3	6	1	-3
4	3	1	1
5	4	1	-11
6	-7	1	10
7	3	1	2
8	5	1	-2
9	3		

Tabelle 2.2: Berechnung von h_i und den Funktionswerten

Folgende Rechnung ergibt sich bei der Annahme, dass die Kurve außerhalb der Stützpunkt linear verlaufen kann $S_0 = 0$ und $S_n = 0$ gesetzt werden.

$$\begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} 12 \\ -36 \\ 24 \\ -72 \\ 126 \\ -48 \\ -24 \end{bmatrix}$$

$S_0 = 0, S_1 = 6.8223, S_2 = -15.2894, S_3 = 18.3352, S_4 = -34.0515,$

$S_5 = 45.8710, S_6 = -23.4323, S_7 = -0.1419, S_8 = 0$

Nun können aus den gegeben Werte die Parameter a_i, b_i, c_i und d_i berechnet werden. Die Tabelle 2.3 zeigt die berechneten Werte.

a_i	b_i	c_i	d_i
1,137	0	-0,137	2
-3,685	3,411	3,274	3
5,604	-7,644	-0,959	6
-8,731	9,167	0,564	3
13,320	-17,026	-7,295	4
-11,551	22,935	-1,385	-7
3,882	-11,716	9,834	3
0,024	-0,071	-1,953	5

Tabelle 2.3: Parameter für die Berechnung der Polynomteile

Es sind acht verschiedene Polynome 3. Grades entstanden. Mithilfe dieser Menge an Polynomfunktionen können die Funktionswerte zwischen den Stützstellen berechnet werden. Grafisch stellt sich der »Kubische Spline« wie in Abbildung 2.2 dar.

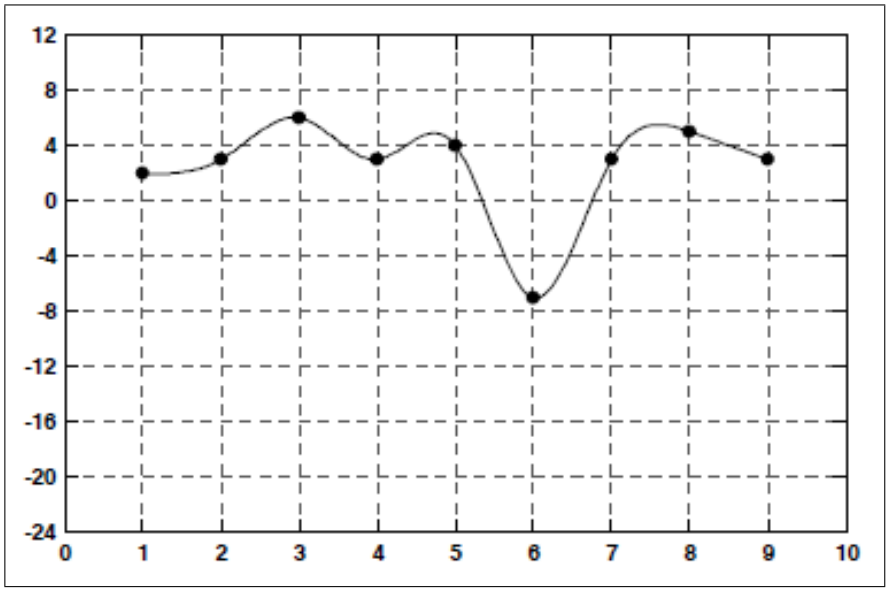

Abbildung 2.2: Darstellung des »Kubischen Splines« mit den Werten aus dem Beispiel im Bereich 2.2.2 [Ung03, 8]

2.3 Ausflug: Weitere Interpolationsverfahren

Unter der Rubrik der Stückweisen Interpolation fällt die »Kubische Splineinterpolation«. Neben der Stückweisen Interpolation existieren weiter Interpolationsverfahren, welche im folgenden Teil kurz beschrieben werden.

Lineare Interpolation
Eine der einfachsten Verfahren, welches zwischen zwei Punkten den direkten Weg berechnet. In der Praxis wird dieses Verfahren am häufigsten verwendet. [Jäh05, 296]

Höhergradige Polynome
Hier wird versucht durch die vorgegeben Stützstellen eine Polynom zu berechnen, welches durch alle Punkte verläuft.

Hermiteinterpolation
Zusätzlich zu den Stützstellen werden die Ableitungen an der Stelle der Stützstellen mit in Betracht gezogen. Die vorgeschriebenen Funktionswerte nennen sich Stutzabszissen. [Sto04, 56]

Trigonometrische Interpolation
Als Ansatz für diese Interpolationsform wird ein trigonometrisches Polynom verwendet, welche zur Analyse von periodischen Vorgängen verwendet wird. [Sto04, 79 ff.]

3 Umsetzung

Die Umsetzung der Anwendung verfolgt das Ziel, die Applikation im Unterricht verwenden zu können. Es sollen mehrere »Kubische Splines« angelegt werden, welche individuell konfiguriert werden können.

3.1 Konzept der Anwendung

Um »Kubische Splines« erfassbar und begreifbarer zu machen, soll die Applikation eine individuelle Vergabe von Punkten anbieten. Es sollen Splines angelegt und mit einer individuellen Farbe versehen werden. Zu den Splines können dann 2D-Koordinaten vergeben werden, um deren Kurve darzustellen. Als Grundlage zur Darstellung wird im Hintergrund ein kartesisches Koordinatensystem angezeigt. Eine mögliche Aufteilung der GUI-Elemente ist in Abbildung 3.1 zu sehen.

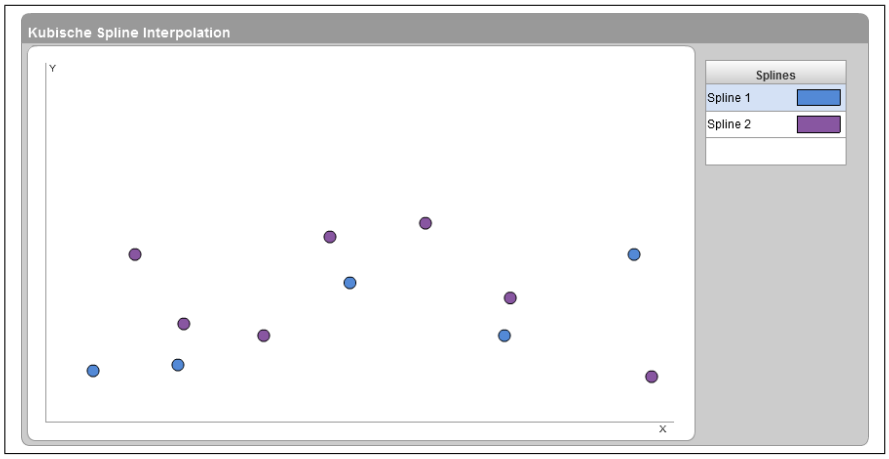

Abbildung 3.1: Entwurf eines GUI für die Darstellung von »Kubischen Splines«

3.2 Entwicklung

Die Entwicklung der Applikation erfolgt in zwei unterschiedlichen Teilen. Ein Teil beschäftigt sich mit der reinen Umsetzung des Algorithmus, während ein zweiter Teil zur grafischen Darstellung dient.

3.2.1 Bibliothek

Als Basis für die Bibliothek dient eine statische Bibliothek, welche den »Kubischen Spline« berechnet.

```
1  class CCubicSpline
2  {
3  public:
4
5      // -------------------------------------------------------
6      // Add specific points to list for interpolation
7      // -------------------------------------------------------
8      void AddPoint(Float2& _rNewPoint);
9      // -------------------------------------------------------
10     // Interpolate to given x-value and gives y-value back
11     // -------------------------------------------------------
12     float Interpolate(float _XPos);
13
14     // -------------------------------------------------------
15     // Returns min and max x-value
16     // -------------------------------------------------------
17     float MinX();
18     float MaxX();
19
20     // -------------------------------------------------------
21     // Returns amount of all points in this spline
22     // -------------------------------------------------------
23     int SizeOfPoints();
24 };
```

Listing 3.1: Header Definition für die kubischen Splines in C++

Als Grundlage für den Aufbau der Methoden wurde auf einen Pseudocode zurückgegriffen [CK07, S. 392].

```
1  procedure Spline3 Coef (n, (ti ), (yi), (ui))
2  integer i, n; real array (ti )0:n, (yi )0:n, (zi )0:n
3  allocate real array (hi )0:n-1, (bi )0:n-1, (ui )1:n-1, (vi )1:
       n-1
4  for i = 0 to n - 1 do
5      hi + ti+1 - ti
6      bi + (yi+1 - yi)/hi
7  end for
8  u1 + 2(h0 + h1)
9  v1 + 6(b1 -b0)
10 for i = 2 to n - 1 do
11     ui + 2(hi + hi+1) - h2
12     i-1/ui-1
13     vi + 6(bi - bi-1) - hi-1vi-1/ui-1
14 end for
15 zn + 0
16 for i = n - 1 to 1  step -1 do
17     zi + (vi - hi zi+1)/ui
18 end for
19 z0 + 0
20 deallocate array (hi ), (bi ), (ui ), (vi )
21 end procedure Spline3 Coef
22
23 real function Spline3 Eval(n, (ti ), (yi ), (zi ), x)
24 integer i ; real h, tmp
25 real array (ti )0:n, (yi )0:n, (zi )0:n
26 for i = n - 1 to 0 step -1 do
27     if x - ti = 0 then exit loop
28 end for
29 tmp + (zi/2) + (x - ti)(zi+1 - zi)/(6h)
30 tmp+ -(h/6)(zi+1 + 2zi) + (yi+1 - yi)/h + (x - ti)(tmp)
31 Spline3 Eval + yi + (x - ti)(tmp)
32 end function Spline3 Eval
```

Listing 3.2: Pseudocode zur Implementierung des Algorithmus. [CK07]

Der komplette Code ist in C++ geschrieben und bietet dadurch eine solide Grundlage in verschiedenste Systeme eingebunden zu werden.

3.2.2 GUI

Die Programmierung des GUI erfolgt zum einen auf dem Mac und auf Windows. Für die jeweilige Plattform werden unterschiedliche Programmiersprachen verwendet. Die Tabelle 3.1 zeigt die Auswahl der jeweiligen Programmiersprache.

Windows	Visual C++
Mac OSX	Objective-C

Tabelle 3.1: Auswahl der jeweiligen Programmiersprache auf dem Zielsystem

Das endgültige Resultat unter Windows ist in der Abbildung 3.2 zu sehen. Nach dem Start des Programms können beliebig Punkte zu den Splines hinzugefügt und entfernt werden.

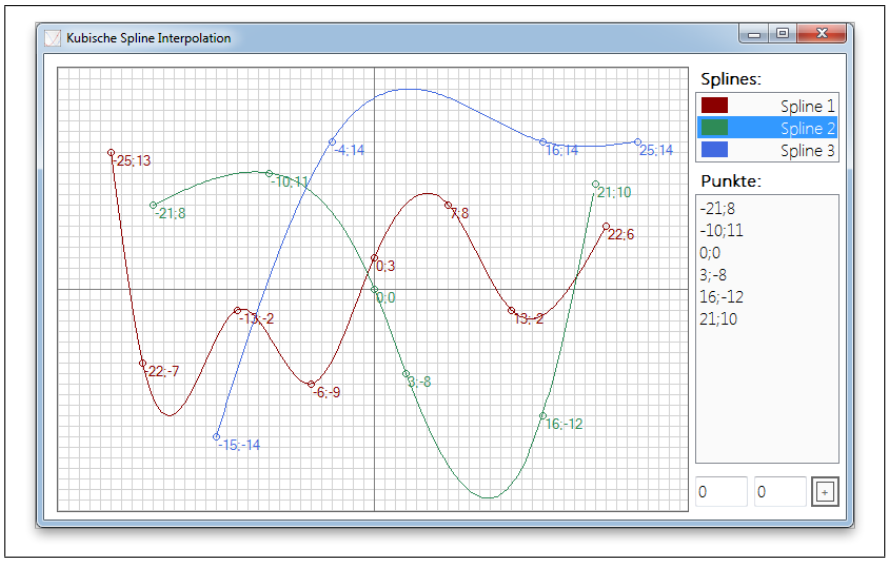

Abbildung 3.2: GUI für die Windows-Anwendung entwickelt in Visual C++

4 Zusammenfassung und Ausblick

Mithilfe der Grundlage über »Kubische Splines« konnte ein Einblick in ein Teilgebiet der Polynominterpolation gegeben werden. In der Zukunft wäre es erstrebenswert, weitere Ausarbeitungen rund um das Thema Interpolation durchzuführen und das entstandene Programm dementsprechend weiterzuentwickeln.

Das Programm kann im Lehrbetrieb eingesetzt werden, um die Splineinterpolation zu veranschaulichen. Durch das manuelle Setzen von Punkten innerhalb eines Koordinatensystems lassen sich konkrete Beispiele darstellen. Die Auswahl der verwendeten Interpolationsform wäre ein nächster Schritt bei der Entwicklung des Programms.

Literaturverzeichnis

[BB05] BENDER, Michael ; BRILL, Manfred: *Computergrafik*. Carl Hanser Verlag GmbH & CO. KG, 2005. – 528 S.

[CK07] CHENEY, Ward ; KINCAID, David: *Numerical Mathematics and Computing*. 6. Brooks Cole, 2007. – 784 S.

[Her11] HERRMANN, Norbert: *Mathematik für Naturwissenschaftler*. Spektrum Akademischer Verlag, 2011. – 290 S.

[Hoc08] HOCHBRUCK, Prof. Dr. M.: *INTERPOLATION UND APPROXIMATION*. UNI Düsseldorf, 2008

[Jäh05] JÄHNE, Bernd: *Digitale Bildverarbeitung*. Springer, 2005

[Kup02] KUPKE, Susann: *Deterministische und stochastische Interpolationsverfahren*. GRIN Verlag, 2002

[Loc93] LOCHER, Franz: *Numerische Mathematik für Informatiker*. Springer, 1993. – 401 S.

[Sto04] STOER, Josef: *Numerische Mathematik 1*. 9. Berlin Heidelberg : Springer, 2004. – 383 S.

[Ung03] UNGER, Dr. R.: *Numerik I SS 2002*. Webseite (TU Chemnitz). http://www-user.tu-chemnitz.de/~uro/teaching/SS2002-numerik/misc/Splines.pdf. Version: 03 2003. – Von http://www-user.tu-chemnitz.de/~uro/teaching/SS2002-numerik/ Zuletzt besucht am 27. Juli 2012

Selbstständigkeitserklärung

Ich, Tobias Schwandt , versichere hiermit, dass die vorliegende Belegarbeit mit dem Thema

Interpolationsverfahren: Das kubische Spline-Verfahren

selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt wurde.

Erfurt, 29.07.2012

Tobias Schwandt