4 Funktionen

4.1 Interpolation von Funktionen durch Polynome

4.1.2 Interpolation mittels kubischer Splines

Eine Splinefunktion S(x) dritten Grades, daher auch als kubische Splinefunktion bezeichnet, ist in jedem Intervall $[x_i, x_{i+1}]$ für i=0(1)n-1 durch ein Polynom $P_i(x)$ dritten Grades bestimmt. Die Glätte wird dadurch erreicht, dass die einzelnen kubischen Parabeln sich nicht nur stetig, sondern mit stetiger zweiter Ableitung aneinander reihen. Die Berührungspunkte werden als Knoten bezeichnet. Man kann zeigen, dass kubische Splines die geringste Krümmung bei der Interpolation aufweisen.

Der Algorithmus enthält einen Teil-Algorithmus zur Lösung eines Gleichungssystems, wie zuvor im Kapitel 3 unter Gauß-Algorithmus beschrieben.

Die Methode versagt, wenn der 1. Koeffizient in der 1. Gleichung, der 2. Koeffizient in der 2. Gleichung, usw. Null sind. Dieses Problem kann durch Vertauschung behoben werden, da, wenn es eine Lösung gibt, dies auch möglich ist.

Tab. 4-1 Algorithmus zur Vertauschung von Koeffizienten

<table>
<tr><td colspan="4">Eingabe der $a_{i,j}$ für i=1,…,n; j=1,…,n+1</td></tr>
<tr><td rowspan="5"></td><td colspan="3">i=1, 1, n</td></tr>
<tr><td colspan="3">Ist $a_{i,i}$ = 0</td></tr>
<tr><td colspan="2">Ja</td><td>Nein</td></tr>
<tr><td colspan="2">j=1, 1, n</td><td rowspan="2">./.</td></tr>
<tr><td></td><td>Ist j ≠ i</td></tr>
</table>

Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

<table>
<tr><td rowspan="4"></td><td rowspan="4"></td><th colspan="2">Ja</th><th>Nein</th><td rowspan="4"></td></tr>
<tr><td colspan="2">Ist $a_{j,i} \neq 0$ and $a_{i,j} \neq 0$</td><td rowspan="3">./.</td></tr>
<tr><th>Ja</th><th>Nein</th></tr>
<tr><td>Tausche Zeile i mit j</td><td>./.</td></tr>
</table>

Code 4-1 Vertauschung

```
Sub Vertauschung()
   Dim i1, i2, n As Integer

   n = 5
   For i1 = 1 To n
      If Cells(i1 + 10, i1) = 0 Then
         For i2 = 1 To n
            If Not i2 = i1 Then
               If Not Cells(i2 + 10, i1) = 0 And _
                  Not Cells(i1 + 10, i2) = 0 Then
                  Call Tausche(i1, i2, n)
                  i2 = n
               End If
            End If
         Next i2
      End If
   Next i1
End Sub

Sub Tausche(i1, i2, n)
   Dim i As Integer
   ReDim a(n + 1), b(n + 1) As Double

   For i = 1 To n + 1
      a(i) = Cells(i1 + 10, i)
      b(i) = Cells(i2 + 10, i)
   Next i
   For i = 1 To n + 1
      Cells(i1 + 10, i) = b(i)
      Cells(i2 + 10, i) = a(i)
   Next i
End Sub
```


Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

4.1.3 Interpolation nach Lagrange

Lagrange Interpolation

Dieses Verfahren geht von dem Ansatz

$$P(x) = L_0(x) y_0 + L_1(x) y_1 + L_2(x) y_2 + \ldots + L_n(x) y_n \tag{4.1.1}$$

für ein *Näherungspolynom* aus, in dem die Koeffizienten $L_i(x)$ der Stützwerte y_i wiederum Polynome n-ten Grades in x sind.

$$L_i(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)} \tag{4.1.2}$$

Tab. 4-2 Algorithmus zur Interpolation mittels eines Lagrange Interpolationspolynoms

Eingabe der x_i und y_i für i=1,…,n				
Konstante bestimmen	i=1, 1, n			
		j = 1, 1, n+		
		j ≠ i		
		Ja	Nein	
		$q = q \cdot (x_i - x_j)$	./.	
Verlauf bestimmen	x = x_1, 1, x_n			
	y=0			
	i = 1, 1, n			
		p=1		
		j = 1, 1, n		
			j ≠ i	
			Ja	Nein
			$p = p \cdot (x - x_j)$	./.

Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

	$y = y + \frac{p_i}{q_i} y_i$

Beispiel: Stahlseilverlauf

Stahlseilverlauf

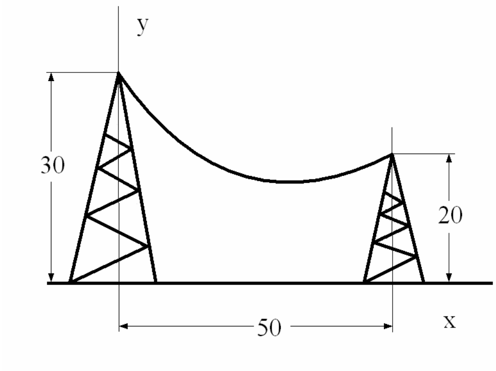

Abb. 4-1 Stahlseilverlauf

Ein Stahlseil zwischen zwei Masten hat den in der Tabelle 4-2 dargestellten Verlauf.

Tab. 4-3 Stützstellen des Seilverlaufs

X	0	10	20	30	35	40	50
Y	30	18	11.5	10	10.5	12.5	20

Ein Programm soll mittels Lagrange Interpolationspolynom den Verlauf des Seils mit einer Schrittweite von 1 Meter bestimmen.

Code 4-2 Bestimmung mittels Lagrange Interpolation

```
Option Explicit

Sub Lagrange_Leer()
   Call Verlauf_Entfernen
   ThisWorkbook.Worksheets("Lagrange").Cells.Clear
End Sub

Sub Lagrange_Testdaten()
   Call Lagrange_Leer
```


Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

```
   Cells(1, 1) = 0: Cells(1, 2) = 30
   Cells(2, 1) = 10: Cells(2, 2) = 18
   Cells(3, 1) = 20: Cells(3, 2) = 11.5
   Cells(4, 1) = 30: Cells(4, 2) = 10
   Cells(5, 1) = 35: Cells(5, 2) = 10.5
   Cells(6, 1) = 40: Cells(6, 2) = 12.5
   Cells(7, 1) = 50: Cells(7, 2) = 20
End Sub

Sub Lagrange_Koeffizienten(n)
   Dim i, j As Integer
   Dim q As Double
   Cells(Rows.Count, 1).End(xlUp).Select
   n = ActiveCell.Row

   For i = 1 To n
      q = 1
      For j = 1 To n
         If Not j = i Then
            q = q * (Cells(i, 1) - Cells(j, 1))
         End If
      Next j
      Cells(i, 3) = q
   Next i
   Exit Sub
End Sub

Private Sub Verlauf(n)
   Dim p, x, y, Li As Double
   Dim i, j, k As Integer

   k = n + 2
   For x = 0 To Cells(n, 1)
      y = 0
      For i = 1 To n
         p = 1
         For j = 1 To n
            If Not j = i Then
               p = p * (x - Cells(j, 1))
            End If
         Next j
         Li = p / Cells(i, 3)
         y = y + Li * Cells(i, 2)
      Next i
      k = k + 1
      Cells(k, 1) = x
```


Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

```
      Cells(k, 2) = y
   Next x
End Sub

Sub Auswertung()
   Dim n As Integer

   Call Lagrange_Koeffizienten(n)
   Call Verlauf(n)
End Sub

Sub Verlauf_Zeigen()
    Range("A10:B60").Select
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
    ActiveChart.SetSourceData
              Source:=Sheets("Lagrange").Range("A10:B60"),
              PlotBy _
        :=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject,
              Name:="Lagrange"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Seilverlauf"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory,
              xlPrimary).AxisTitle.Characters.Text = "Weite
              [m]"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text
              = "Höhe [m]"
    End With
    ActiveChart.Legend.Select
    Selection.Delete
End Sub

Sub Verlauf_Entfernen()
   Dim Shp As Shape
   For Each Shp In Worksheets("Lagrange").Shapes
      Shp.Delete
   Next
End Sub
```

Es ergibt sich der gleiche Verlauf wie nach dem Newton Interpolationspolynom.

Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

	A	B	C	D	E	F	G	H	I
1	0	30	420000000						
2	10	18	-60000000						
3	20	11,5	18000000						
4	30	10	-6000000						
5	35	10,5	4921875						
6	40	12,5	-12000000						
7	50	20	180000000						
8									
9									
10	0	30							
11	1	29,0965508							
12	2	28,004608							
13	3	26,785108							
14	4	25,4894034							
15	5	24,1601563							
16	6	22,832192							
17	7	21,5333155							
18	8	20,285088							
19	9	19,1035658							
20	10	18							
21	11	16,9814984							
22	12	16,051648							
23	13	15,2110993							
24	14	14,458112							
25	15	13,7890625							
26	16	13,198912							
27	17	12,6816368							
28	18	12,2306194							
29	19	11,839002							
30	20	11,5							
31	21	11,2071783							
32	22	10,954688							
33	23	10,7374655							
34	24	10,551392							
35	25	10,3934152							
36	26	10,261632							
37	27	10,155333							
38	28	10,075008							
39	29	10,0223133							
40	30	10							
41	31	10,0118045							
42	32	10,0622994							
43	33	10,1567068							
44	34	10,300672							
45	35	10,5							
46	36	10,760352							
47	37	11,0869043							
48	38	11,483968							
49	39	11,9545709							
50	40	12,5							
51	41	13,1193058							
52	42	13,808768							
53	43	14,561323							
54	44	15,365952							
55	45	16,2070313							
56	46	17,0636434							
57	47	17,9088505							
58	48	18,708928							
59	49	19,4225608							
60	50	20							

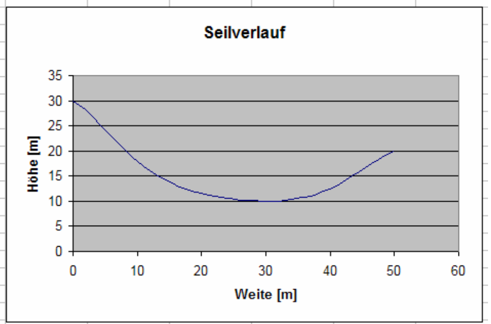

Abb. 4-2 Auswertung der Testdaten

4.3 Numerische Integration

Beispiel: Träger gleicher Zugfestigkeit

Gesucht ist das Profil eines Stabes, der nach Abbildung 4-5 einer Zugkraft unterliegt, die in jedem Querschnitt konstant sein soll.

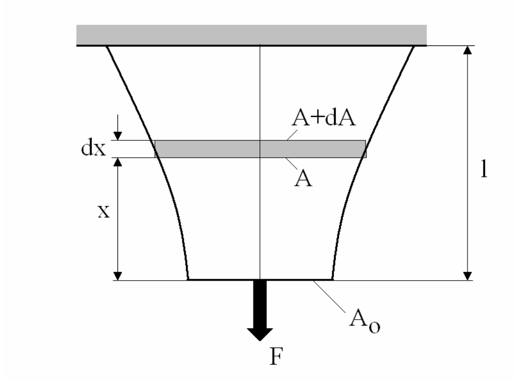

Abb. 4-3 Träger mit gleicher Zugfestigkeit

Wir gehen wieder von der Formel

$$\frac{dA}{A} = \frac{\rho \cdot g}{\sigma} \cdot dx \,. \tag{4.3.1}$$

aus. Wir setzen diesmal jedoch einen runden Querschnitt voraus, so dass sich der Radius aus der Kreisfläche

$$A = r^2 \cdot \pi \tag{4.3.2}$$

bestimmt. Diesen ermitteln wir nach der bereits beschriebenen Trapezregel.

Tab. 4-4 Lösungsalgorithmus: Träger gleicher Zugspannung

Eingabe der erforderlichen Parameter A_0, l, ρ, σ, n
$\Delta x = \frac{l}{n}$

x = Δx, (Δx), l	
	$\Delta A_x = \frac{\rho \cdot g}{\sigma} \cdot A_{x-1} \cdot \Delta x$
	$A_x = A_{x-1} + \Delta A_x$
	$r_x = \sqrt{\frac{A_{x-1} + A_x}{2 \cdot \pi}}$

Code 4-3 Träger gleicher Zugspannung

```
Option Explicit

Sub Zugspannung_Leer()
   Dim Shp As Shape
   For Each Shp In Worksheets("Konstante
              Zugspannung").Shapes
      Shp.Delete
   Next
   ThisWorkbook.Worksheets("Konstante
              Zugspannung").Cells.Clear

   Range("A1:E1").Select
   Selection.MergeCells = True
   Selection.Font.Bold = True
   Selection.Font.Italic = True
   Selection.Value = "Träger mit konstanter Zugspannung"

   Range("A2:A16").Select
   Selection.Font.Bold = True
   Selection.Font.Italic = True
   Range("A2") = "ro [m]"
   Range("A3") = "l [m]"
   Range("A4") = ChrW(961) & " [kg/m" & ChrW(179) & " ]"
   Range("A5") = ChrW(963) & " [N/m" & ChrW(178) & " ]"
   Range("A6") = "n"
   Range("B:B").ColumnWidth = "15"

   Range("C:C").ColumnWidth = "2"
   Range("D2") = "x [m]"
   Range("E2") = "r [m]"
   Range("D2:E2").Select
   Selection.Font.Bold = True
```


Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

```
   Selection.Font.Italic = True


   Range("B2").Select
End Sub

Sub Zugspannung_Testdaten()
   Cells(2, 2) = 0.02
   Cells(3, 2) = 0.5
   Cells(4, 2) = 0.00785
   Cells(5, 2) = 0.01
   Cells(6, 2) = 50
End Sub

Sub Zugspannung_Auswertung()
   Dim r0, l, r, S, n As Double
   Dim dx, x, rx, dA, A1, A2 As Double
   Dim i As Integer

   r0 = Cells(2, 2)
   l = Cells(3, 2)
   r = Cells(4, 2)
   S = Cells(5, 2)
   n = Cells(6, 2)

   dx = l / n
   A1 = r0 * r0 * 4 * Atn(1)
   i = 2
   For x = dx To l + dx Step dx
      dA = r * 9.81 / S * A1 * dx
      A2 = A1 + dA
      i = i + 1
      Cells(i, 4) = x
      rx = Sqr((A1 + A2) / 8 / Atn(1))
      Cells(i, 5) = rx
      A1 = A2
   Next x
End Sub

Sub Zeige_Querschnittsverlauf()
   Range("D3:E52").Select
   Charts.Add
   ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
   ActiveChart.SetSourceData Source:=Sheets("Konstante
              Zugspannung").Range( _
       "D3:E52"), PlotBy:=xlColumns
```


Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

```
   ActiveChart.SeriesCollection(1).Name =
              "=""Kreisradius"""
   ActiveChart.Location Where:=xlLocationAsObject, Name:= _
       "Konstante Zugspannung"
   With ActiveChart
       .HasTitle = True
       .ChartTitle.Characters.Text = "Träger mit konstanter
              Spannung"
       .Axes(xlCategory, xlPrimary).HasTitle = True
       .Axes(xlCategory,
              xlPrimary).AxisTitle.Characters.Text = "x
              [m]"
       .Axes(xlValue, xlPrimary).HasTitle = True
       .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text
              = "r [m]"
   End With
   ActiveChart.Legend.Select
   Selection.Left = 229
   Selection.Top = 274
   ActiveChart.Axes(xlValue).MajorGridlines.Select
   ActiveChart.PlotArea.Select
   Selection.Width = 314
   ActiveWindow.Visible = False
End Sub

Sub Lösche_Querschnittsverlauf()
   Dim Shp As Shape
   For Each Shp In Worksheets("Konstante
              Zugspannung").Shapes
      Shp.Delete
   Next
End Sub
```

Nachfolgend sehen Sie die Auswertung der Testdaten. Der Radius des Stabes nimmt nach oben exponentiell zu. Nur so ist eine konstante Spannung in jedem Querschnitt gewährleistet.

Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

	A	B	C	D	E	F
1	*Träger mit konstanter Zugspannung*					
2	*ro [m]*	0,02		*x [m]*	*r [m]*	
3	*l [m]*	0,5		0,01	0,02038141	
4	*ρ [kg/m³]*	0,00785		0,02	0,02115162	
5	*σ [N/m²]*	0,01		0,03	0,02195095	
6	*n*	50		0,04	0,02278048	
7				0,05	0,02364136	
8				0,06	[illegible]	
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23				0,21	[illegible]	
24				0,22	0,04441555	
25				0,23	0,04609402	
26				0,24	0,04783592	
27				0,25	0,04964365	
28				0,26	0,05151969	
29				0,27	0,05346663	
30				0,28	0,05548715	
31				0,29	0,05758402	
32				0,3	0,05976013	
33				0,31	0,06201848	
34				0,32	0,06436217	
35				0,33	0,06679443	
36				0,34	0,0693186	
37				0,35	0,07193817	
38				0,36	0,07465672	
39				0,37	0,07747802	
40				0,38	0,08040593	
41				0,39	0,08344448	
42				0,4	0,08659787	
43				0,41	0,08987042	
44				0,42	0,09326664	
45				0,43	0,0967912	
46				0,44	0,10044896	
47				0,45	0,10424495	
48				0,46	0,10818438	
49				0,47	0,11227269	
50				0,48	0,1165155	
51				0,49	0,12091865	
52				0,5	0,12548819	

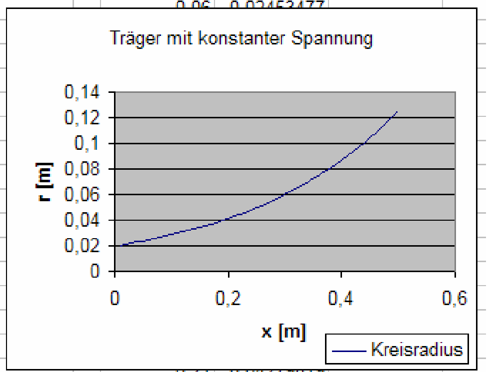

Abb. 4-4 Auswertung der Testdaten

Beispiel: Ausflusszeit von Flüssigkeiten

Wir betrachteten einen Behälter, in dem sich Flüssigkeit mit der Höhe h befindet.

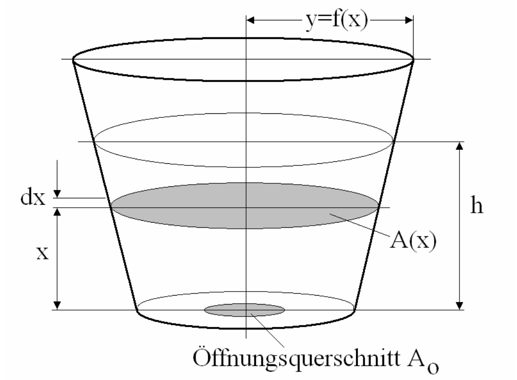

Abb. 4-5 Ausfluss aus einem Behälter

Die Lösung lautete

$$t = \frac{1}{A_O \sqrt{2 \cdot g}} \int_h^x \frac{A(x)}{\sqrt{x}} dx . \tag{4.3.3}$$

Für eine Zylinderform ist der Querschnitt konstant und somit

$$A = r^2 \cdot \pi . \tag{4.3.4}$$

Damit folgt

$$t = \frac{r^2 \cdot \pi}{A_O \sqrt{2 \cdot g}} \int_h^x \frac{dx}{\sqrt{x}} . \tag{4.3.5}$$

Für die völlige Entleerung wird x=0 und damit

$$t = \frac{r^2 \cdot \pi}{A_O \sqrt{2 \cdot g}} \int_h^0 \frac{dx}{\sqrt{x}} . \tag{4.3.6}$$

bzw.

$$t = \frac{r^2 \cdot \pi \cdot \sqrt{2 \cdot g \cdot h}}{g \cdot A_O} . \tag{4.3.7}$$

Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.

Harald Nahrstedt, Algorithmen für Ingenieure, ISBN 3-8348-0015-5.