Das Mathe-Kochbuch

Mathematik 4 für Ingenieure: Numerische Rezepte

Dominic Griesel

25. Juli 2011

Inhaltsverzeichnis

2.2.3. Newtonsche Interpolation

Um diese Nachteile zu vermeiden, kann man das Verfahren benutzen, dass sich Newton ausgedacht hat. Der Algorithmus liest sich zwar *deutlich komplizierter*, dafür ist er in der Anwendung um einiges einfacher. Der Ansatz lautet

$$p_n(x) = c_0 + c_1 \cdot (x - x_0) + \cdots + c_n \cdot (x - x_0) \cdot \cdots \cdot (x - x_{n-1}). \tag{2.1}$$

Er hat den Vorteil, dass man bei Hinzunahme eines weiteren Punktes nur einen zusätzlichen Summanden berechnen muss. Wie bereits erwähnt, der Algorithmus liefert das selbe Ergebnis, ist aber praktischer in der Anwendung. Die Konstanten c_i berechnen sich mit den *dividierten Differenzen*, die folgendermaßen rekursiv definiert sind:

$$\begin{aligned} f[x_i] &= f_i, \quad \text{für } i = 0, \ldots, n \\ f[x_i, \ldots, x_{i+k}] &= \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}, \quad \text{für } k = 1, \ldots, n - i \\ \text{und } c_i &= f[x_0, \ldots, x_i]. \end{aligned}$$

Alles logisch, oder? Am Besten gleich wieder vergessen, dass lässt sich nämlich mit dem Differenzenschema ganz einfach erledigen.

Das Differenzenschema

Auch dieses Schema will ich euch wieder anhand eines Beispiels zeigen. Wir haben folgende Punkte gegeben:

$\boldsymbol{x_i}$	-1	0	2	3	4
$\boldsymbol{f_i}$	-3	3	9	18	45

Wir erstellen uns wieder eine Tabelle mit genug Zeilen für alle x-Werte und viel Platz zwischen den Zeilen und genauso vielen Spalten auf der rechten Seite. Die erste Spalte nehme ich ganz gerne als Hilfsspalte, ihr könnt sie aber genauso gut weglassen. In die x_i-Spalte tragen wir logischerweise die x-Werte ein. In die erste Spalte auf der rechten Seite die f-Werte.

Die weiteren Werte berechnen sich dann in einer Art Tannenbaumstruktur. In der nächsten Spalte kommen die Werte nämlich zwischen die Zeilen der vorherigen Spalte. Wie das aussieht, seht ihr in Abbildung 2.2 auf der nächsten Seite.

Der Algorithmus an einem Beispiel erklärt

Die Berechnung erfolgt von links nach rechts. Zuerst alle Werte der 1-Spalte berechnen, dann alle Werte der 2-Spalte, etc. Auf den jeweiligen Wert kommt man mit folgender Formel:

$$f(\text{aktuell}) = \frac{f(\text{links unten}) - f(\text{links oben})}{x(\text{links unten}) - x(\text{links oben})} \tag{2.2}$$

Man kann aber nicht einfach diejenigen x-Werte nehmen, die links neben den hierfür benutzten f-Werten stehen, denn das geht in der zweiten berechneten Spalte nicht mehr. Stattdessen folgt man den Pfeilen in umgekehrter Richtung bis in die erste Spalte und nimmt die x-Werte, die hier daneben stehen.

Das wollen wir jetzt mal an diesem Beispiel anwenden. Wir berechnen den ersten Wert der 1-Spalte (also für die Stelle, wo jetzt schon die 6 steht). Also suchen wir uns unsere f-Werte:

$$f(\text{links unten}) = 3$$
$$f(\text{links oben}) = -3.$$

Für die x-Werte folgen wir den Pfeilen in umgekehrter Richtung und landen bei den f-Werten 3 und -3. Unsere x-Werte stehen links daneben:

$$x(\text{links unten}) = 0$$
$$x(\text{links oben}) = -1.$$

Setzen wir das in die Formel (2.2) auf der vorherigen Seite ein, erhalten wir

$$f(\text{aktuell}) = \frac{3-(-3)}{0-(-1)} = 6.$$

Auf die selbe Weise berechnen sich die restlichen Werte. Um z.B. den zweiten Wert der 2-Spalte zu berechnen, benutzen wir die f-Werte links daneben, also 9 und 3. Folgen wir den Pfeilen für die x-Werte, landen wir bei den f-Werten 18 und 3, somit erhalten wir die x-Werte 3 und 0.

i	x_i	0	1	2	3	4
0	-1	-3	$\frac{3-(-3)}{0-(-1)} = 6$			
1	0	3	$\frac{9-3}{2-0} = 3$	$\frac{3-6}{2-(-1)} = -1$		
2	2	9	$\frac{18-9}{3-2} = 9$	$\frac{9-3}{3-0} = 2$	$\frac{2-(-1)}{3-(-1)} = \frac{3}{4}$	
3	3	18	$\frac{45-18}{4-3} = 27$	$\frac{27-9}{4-2} = 9$	$\frac{9-2}{4-0} = \frac{7}{4}$	$\frac{\frac{7}{4}-\frac{3}{4}}{4-(-1)} = \frac{1}{5}$
4	4	45				

Abbildung 2.2.: Differenzen-Schema

Einen Tipp habe ich noch für euch: Habt ihr Brüche in den Punkten, die ihr interpolieren sollt, packt sie möglichst an die erste oder letzte Stelle. Das könnt ihr machen, da die Reihenfolge der Punkte egal ist. So erreicht ihr, dass im gesamten Schema möglichst wenige Brüche auftauchen. Wenn ihr das Schema fertig habt, könnt ihr daraus die c_i-s aus dem Ansatz (Gleichung (2.1) auf Seite 16) bestimmen. Das sind nämlich die Werte, die in jeder Spalte oben stehen. In unserem Fall:

$$c_0 = -3, \quad c_1 = 6, \quad c_2 = -1, \quad c_3 = \frac{3}{4}, \quad c_4 = \frac{1}{5}.$$

Somit ergibt sich das Interpolationspolynom

$$p_4(x) = -3 + 6(x+1) - (x+1)x + \frac{3}{4}(x+1)x(x-2) + \frac{1}{5}(x+1)x(x-2)(x-3).$$

Hinzunahme eines weiteren Punkts

Wie bereits erwähnt, lässt sich bei diesem Verfahren sehr einfach ein weiterer Punkt durch die Funktion interpolieren. Hierzu ergänzt man eine weitere Zeile, in die man den Punkt einträgt und eine weitere Spalte und berechnet die fehlenden Werte der unteren Schrägzeile. Der neue Wert an der Spitze des Tannenbaums ist dann der Koeffizient für den zusätzlichen Term im Interpolationspolynom, die restlichen Summanden bleiben unberührt.

Allgemeine Darstellung

Mit den allgmeinen Bezeichnungen aus dem Algorithmus sieht das Differenzenschema für $n = 4$ so aus:

i	x_i	0	1	2	3	4
0	x_0	$f[x_0]$				
			$f[x_0, x_1]$			
1	x_1	$f[x_1]$		$f[x_0, x_1, x_2]$		
			$f[x_1, x_2]$		$f[x_0, \ldots, x_3]$	
2	x_2	$f[x_2]$		$f[x_1, x_2, x_3]$		$f[x_0, \ldots, x_4]$
			$f[x_2, x_3]$		$f[x_1, \ldots, x_4]$	
3	x_3	$f[x_3]$		$f[x_2, x_3, x_4]$		
			$f[x_3, x_4]$			
4	x_4	$f[x_4]$				

Man kann also allgmein sagen, dass der Wert $f[x_i, \ldots, x_j]$ auf dem Schnittpunkt der Diagonalen zwischen x_i und x_j steht. Das solltet ihr euch merken, denn das brauchen wir später wieder.

Die Iteration kann man sich nun auch an der Grafik veranschaulichen. Wir beginnen mit einem Startwert, sagen wir $x_0 = 3$ und setzen diesen in die Iterationsfunktion Φ ein. Wir erhalten einen Punkt auf dem Graphen von Φ, dessen y-Koordinate gleich x_1 ist, da schließlich gilt $x_1 = \Phi(x_0)$. Wandern wir von dort zur Geraden $y = x$, erhalten wir einen Punkt mit der x-Koordinate x_1. Diesen setzen wir in die Iterationsvorschrift ein, d.h. wandern zur Kurve von Φ und gehen anschließend weiter zur Geraden, um einen Punkt mit der x-Koordinate x_2 zu erhalten, und so weiter.

Wäre die Iterationsvorschrift in diesem Bereich steiler, also besäße eine Lipschitz-Konstante $L \geq 1$, dann würde die Spirale sich nicht auf einen Punkt zusammenziehen.

5.5. Iterationsverfahren

Im Folgenden werden wir uns einige Lösungsverfahren für nichtlineare Gleichungen (also den eindimensionalen Fall) anschauen.

5.5.1. Das eindimensionale Newton-Verfahren

Es ist das wohl bekannteste und wichtigste Verfahren: Das Newton-Verfahren. Wie vieles in der numerischen Mathematik basiert es auf einer Taylor-Entwicklung der betrachteten Funktion. Stellt man zu einer Funktion $f(x)$ das lineare Taylor-Polynom auf (siehe Mathe 2), gilt in einer Umgebung um den Entwicklungspunkt x_0:

$$f(x) \approx f(x_0) + f'(x_0) \cdot (x - x_0).$$

Der Trick besteht darin, nicht die Nullstelle der Funktion f direkt zu suchen, sondern ausgehend von einem Startpunkt x_0 die Nullstelle des linearen Taylor-Polynoms (also der Tangente) zu suchen. Hat man diese gefunden, berechnen wir wieder das Taylor-Polynom zu diesem neuen Startpunkt und suchen dessen Nullstelle, etc. Wir setzen dafür die rechte Seite der obigen Formel Null und stellen nach x um:

$$f(x_0) + f'(x_0) \cdot (x - x_0) = 0$$

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Iterationsvorschrift

Als Iterationsvorschrift kann man das so schreiben:

$$x_{k+1} = \Phi(x_k) = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{5.1}$$

Damit dieses Verfahren funktionieren kann, muss die Funktion f in einer Umgebung der Nullstelle x^* stetig differenzierbar sein.

Konvergenz

- Ist x^* eine einfache Nullstelle der Funktion f (also $f(x^*) = 0$ und $f'(x^*) \neq 0$) und f dreimal stetig differenzierbar in einer Umgebung von x^*, dann konvergiert das Verfahren mindestens quadratisch mit $C = \frac{f''(x^*)}{2f'(x^*)}$.

- Ist x^* eine l-fache Nullstelle (f und alle Ableitungen bis zur $(l-1)$-ten sind dort Null) und f ist $2l$-mal stetig differenzierbar in einer Umgebung von x^*, dann konvergiert das Verfahren nur linear mit $C = \frac{l-1}{l}$. Das lässt sich verhindern, indem man es leicht modifiziert:

$$x_{k+1} = x_k - l \cdot \frac{f(x_k)}{f'(x_k)}.$$

  Das modifizierte Verfahren konvergiert in der Nähe einer l-fachen Nullstelle quadratisch.

- Ist die Berechnung von f' zu aufwendig, was in höheren Dimensionen schnell der Fall sein kann, kann man das Verfahren vereinfachen, indem man nur nach einer gewissen Anzahl an Punkten oder nur ein einziges Mal die Ableitung berechnet, etwa so:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}.$$

  Dieses vereinfachte Verfahren konvergiert nur noch linear mit dem Konvergenzfaktor $C = |\Phi'(x^*)| = \left|1 - \frac{f'(x^*)}{f'(x_0)}\right|$, der aber glücklicherweise oft sehr klein gegenüber 1 ist.

- Das Newton-Verfahren funktioniert auch für komplexe Nullstellen, allerdings muss x_0 dann auch komplex sein.

- Es ist allerdings nicht *global* konvergent, wenn man sich nicht nah genug an der Nullstelle befindet, kann es divergieren oder gegen eine andere Nullstelle konvergieren. Außerdem kann es katastrophal versagen, wenn man eine Stelle mit (nahezu) horizontaler Tangente trifft.

Geometrische Interpretation

Wie bereits erwähnt, kann man sich das Verfahren folgendermaßen vorstellen: An einem Startpunkt x_0 legt man eine Tangente an f und sucht deren Nullstelle x_1. An dieser Stelle legt man wieder eine Tangente an f und sucht deren Nullstelle x_2 etc.

8.2.1. Reduktion der Ordnung

Es reicht aus, sich Methoden für das Lösen von DGLen erster Ordnung zu erarbeiten, da jede DGL n-ter Ordnung

$$y^{(n)} = f\left(x, y, y', \ldots, y^{(n-1)}\right)$$

sich auf ein System von DGLen erster Ordnung reduzieren lässt. Dazu definiert man zunächst n Funktionen

$$y_1 = y \qquad y_2 = y' \qquad \ldots \qquad y_n = y^{(n-1)}.$$

Wir sehen, dass jede Funktion die Ableitung der vorherigen ist

$$y_1' = y' = y_2 \qquad y_2' = y'' = y_3 \qquad \ldots \qquad y_{n-1}' = y^{(n-1)} = y_n,$$

sodass wir daraus $n-1$ Gleichungen erhalten. Die n-te Gleichung ergibt sich aus der DGL selbst

$$y_n' = y^{(n)} = f\left(x, y, y', \ldots, y^{(n-1)}\right).$$

Außerdem müssen wir die Randbedingungen mit den Hilfsfunktionen $y_1, \ldots, y_n$ ausdrücken. Wenn wir das System gelöst haben, steht in y_1 die gesuchte Lösung, denn schließlich ist $y_1 = y$.

Egal, ob wir die Ordnung reduzieren mussten oder nicht, am Ende erhalten wir eine DGL oder ein System von DGLen der Form

$$y' = f\left(x, y(x)\right),$$

wobei f und y für eine einzelne Funktion im eindimensionalen Fall oder einen Vektor von Funktionen im mehrdimensionalen Fall stehen.

8.2.2. Das explizite Euler-Verfahren

Die Idee

Der betrachtete Typ von Differentialgleichungen ordnet jedem Punkt $(x, y(x))$ eine Steigung y' zu. Das kann man ausnutzen, um sich Punkt für Punkt durch die DGL zu hangeln. Wir starten an einem vorgegebenen Punkt, legen durch diesen Punkt eine Gerade mit der Steigung y' und wandern diese Gerade ein kleines Stück entlang. Nun kann man wieder die Steigung an diesem Punkt ausrechnen, eine Gerade hindurchlegen, und so weiter.

Etwas mathematischer kann man das mit einer Taylor-Entwicklung beschreiben. Zunächst muss man das betrachtete Intervall diskretisieren, das heißt es muss in n Teilintervalle aufgeteilt werden, sodass wir $n+1$ Punkte $x_0, x_1, \ldots x_n$ erhalten. Wir betrachten hier nur äquidistante Stützstellen, das bedeutet der Abstand zwischen zwei benachbarten Punkten ist stets h. Entwickelt man nun die Funktion $y(x)$ an der Stelle x_{k+1}, ergibt sich

$$y(x_{k+1}) = y(x_k) + h \cdot y'(x_k) + \ldots.$$

Vernachlässigt man alle Glieder, die nach der ersten Ableitung folgen, erhält man die eben beschriebene Gerade. Nun setzen wir für y' die rechte Seite der DGL ein:

$$y(x_{k+1}) \approx y(x_k) + h \cdot f\left(x_k, y(x_k)\right).$$

Bezeichnen wir mit y_k die Näherung der Funktion an der Stelle x_k (also $y_k \approx y(x_k)$), kann man diese Formel schreiben als

$$y_{k+1} = y_k + h \cdot f\left(x_k, y_k\right).$$

Das ist das explizite Euler-Verfahren.

Der Algorithmus

Hier nochmal der Algorithmus zum Anwenden:

1. Man benötigt die Stellen x_k, an denen die DGL ausgewertet werden soll und einen Anfangspunkt (x_0, y_0).
2. Setze den letzten bekannten Punkt in das Euler-Verfahren ein und berechne damit den nächsten:
$$y_{k+1} = y_k + h \cdot f\left(x_k, y_k\right)$$
3. Wiederhole Schritt 2.

Also eigentlich ganz easy!

Geometrische Interpretation

In der folgenden Skizze kann man gut die Arbeitsweise des expliziten Euler-Verfahrens erkennen. Man beginnt links und baut sich die Lösung Schritt für Schritt mit den Geradenstücken zusammen. Im letzten Schritt ist ein Fehler zu sehen, der auftreten kann, wenn die Näherungslösung zu weit von der exakten entfernt ist.