Lösung zu Blatt 7Abgabe bis 05. Juni, 23:59

Stückweise Interpolation

1. Man skizziere die stückweise lineare Interpolation der Funktion $f(x) = x^2$ auf $I = [-1, 1]$ unterteilt in drei Teilintervalle $[-1, -\frac{1}{3}]$, $[-\frac{1}{3}, \frac{1}{3}]$ und $[\frac{1}{3}, 1]$.

2. Wie sähe die Fehlerabschätzung für die stückweise lineare Interpolation aus (entsprechend Satz 8.21)?

Hier gilt

$$\max_{x\in[a,b]} |f(x) - p(x)| \leqslant \frac{1}{6} h^3 \max_{x\in[a,b]} |f'''(x)|$$

Splines

1. Was ist der Unterschied zwischen der stückweise kubischen Lagrange-Interpolation und der natürlichen kubischen Spline?

Bei der stückweise kubischen Interpolation wird auf jeden Teilintervall, z.b. x_{i-1}, x_i die Funktion $f(x)$ durch ein kubisches Polynom, z.B. in 4 Punkten

$$x_{i-1} = x_i^0 < x_i^1 < x_i^2 < x_i^3 = x_i$$

interpoliert. Das resultierende Polynom ist global stetig und ein stückweises Polynom. Bei dem natürlichen Spline wird keine weitere Unterteilung unternommen. Stattdessen wird neben der Interpolation in $f(x_i) = p(x_i)$ die Stetigkeit erster und zweiter Ableitung gefordert. Auf jedem Intervall ergeben sich somit $2 + 2 + 2 = 6$ Bedingungen, von denen jedoch die Ableitungsbedingungen "geteilt" sind und auch für die Nachbarintervalle gelten. Global (über alle Teilintervalle) stimmt die Anzahl der Unbekannten mit der Anzahl der Bedingungen überein. Im gegensatz zur stückweisen Interpolation muss zur Bestimmung des splines ein lineares Gleichungssystem gelöst werden.

2. Was ist der Aufwand zur Berechnung eines Splines in 100 Punkten? Worin besteht der wesentliche Aufwand?

Nach entsprechender Vorbereitung muss ein Tri-Diagonalsystem gelöst werden. Der Aufwand ist $\mathcal{O}(n)$.

3. Was meint man mit dem Zusatz *natürlich?*?

An den Ränden a, b gibt es keine Übergangsbedingung für die Ableitungen. Stattdessen wird gefordert, dass hier die 2te Ableitung Null ist.

Extrapolation zum Limes Man gebe kurze Antworten:

1. Was muss für die Stützstellenfolge bei der Extrapolation gelten?

Die Schrittweitenfolge muss den zu extrapolierenden Wert schnell genug annähern. Bei Extrapolation (z.B.) gegen 0 muss gelten

$$\sup \frac{h_{i+1}}{h_i} \leqslant \rho < 1.$$

D.h. z.B. $h_i = 1/i$ genügt nicht.

2. Wir wissen, dass die Funktion $a(x) = \sin(x)/x$ eine Entwicklung in geraden Potenzen hat. Welche Konvergenzordnung zur Berechnung des Limes $\lim_{x\to 0} a(x)$ kann erreicht werden, wenn zur Extrapolation jeweils 3 Stützstellen verwendet werden?

Man sollte die Extrapolation dann auch nur mit geraden Potenzen durchführen. Bei Kombination der jeweils letzten 3 Stützstellen h_i, h_{i+1}, h_{i+2} kann so die Ordnung

$$\mathcal{O}(h_i^6)$$

erreicht werden.

Übungsaufgaben

Aufgabe 7.1

Es sei $a(h)$ ein konvergenter Prozess mit der Entwicklung

$$a(h) = a_0 + a_1 h^\alpha + o(h^{\alpha+1}).$$

Man zeige, dass aus drei aufeinander folgenden Werten zu jeweils halber Schrittweite $a(h), a(h/2), a(h/4)$ die Konvergenzordnung experimentell gemäß

$$\alpha \approx \frac{1}{\log(2)} \log\left(\frac{a(h) - a(h/2)}{a(h/2) - a(h/4)}\right)$$

bestimmt werden kann. Man nutze diese Methode zur Bestimmung der Ordnung der folgenden Prozesse

h	$a_1(h)$	$a_2(h)$
2^{-2}	0.6360915	0.6145402401
2^{-3}	0.6388248	0.6121714292
2^{-4}	0.6394830	0.6119234438
exakt	0.6397000	0.6118916834

Es ist

$$\begin{aligned} A_1 &:= a(h) \approx a_0 + a_1 h^\alpha \\ A_2 &:= a(h/2) \approx a_0 + a_1 h^\alpha 2^{-\alpha} \\ A_3 &:= a(h/4) \approx a_0 + a_1 h^\alpha 4^{-\alpha} \end{aligned}$$

Subtraktion $A_2 - A_1$ und $A_3 - A_2$ gibt (wir schreiben einfach "=" statt "≈")

$$\begin{aligned} A_2 - A_1 &= a_1 h^\alpha (2^{-\alpha} - 1) \\ A_3 - A_2 &= a_1 h^\alpha (2^{-\alpha} - 1) 2^{-\alpha} \end{aligned} \quad \Rightarrow \quad \frac{A_2 - A_1}{A_3 - A_2} = 2^\alpha,$$

also

$$\alpha = \frac{1}{\log(2)} \log\left(\frac{A_2 - A_1}{A_3 - A_2}\right).$$

Die Aufgabe könte weiter gehen und wir könnten nun α nutzen um auch a_0 und a_1 zu approximieren. Dabei gibt a_0 eine Extrapolation für das exakte Ergebnis $a(0)$ an.

Für die Tabelle liefert die Formel

$$\alpha_1 \approx 2.05,$$

d.h. der Prozess konvergiert quadratisch und

$$\alpha_2 \approx 3.26,$$

d.h., es ist vermutlich ein kubisch konvergierender Prozess.

Aufgabe 7.2

Der i*-Trick:*
Für eine Funktion $f(x)$ betrachten wir zur Approximation der ersten *reellen* Ableitung $f'(x_0)$ den Differenzenquotienten

$$\left.\frac{df(x)}{dx}\right|_{x=x_0} \approx D_i[f](x_0) := \frac{\operatorname{Im}\left(f(x_0+ih)\right)}{h},$$

dabei ist i die imaginäre Einheit und $\operatorname{Im}(z)$ der Imaginärteil von $z \in \mathbb{C}$.
Man zeige durch Taylor-Entwicklung, dass durch den i-Trick wirklich eine Approximation der ersten Ableitung gegeben ist. Mit welcher Ordnung konvergiert dieser Differenzenquotient? Welche Ordnung erhält man bei Extrapolation von $h \to 0$ mit jeweils 2 Punkten?

Hinweis: Übliche Differenzenquotienten haben immer das Problem der Auslösung, z.B. durch Berechnung von $f(x+h) - f(x)$*. Der i-Trick umgeht dieses Problem und eine Stabilitätsanalyse zeigt wirklich, dass er nicht anfällig für Rundungsfehler ist. Das Problem ist natürlich, dass er eine komplexe Arithmetik verlangt.*

Es ist

$$\begin{aligned} f(x+ih) &= f(x) + f'(x)ih + \frac{f''(x)}{2}(ih)^2 + \frac{f'''(x)}{6}(ih)^3 + O(h^4) \\ &= \left(f(x) - \frac{f''(x)}{2}h^2 + O(h^4)\right) + ih\left(f'(x) - \frac{f'''(x)}{6}h^2 + \mathcal{O}(h^4)\right). \end{aligned}$$

D.h., es gilt quadratische Konvergenz

$$\frac{\operatorname{Im}(f(x_0+ih))}{h} = f'(x) + O(h^2).$$

Bei **Extrapolation** mit quadratischen Ansätzen, also $q = 2$ erhält man mit 2 Punkten (d.h. nur ein Extrapolationsschirtt) die Ordnung $O(h^4)$. Ein quadratischer Anzatz ist angemessen, da die weitere Fehler-Entwicklung der Art

$$\frac{\operatorname{Im}(f(x_0+ih))}{h} = f'(x) + O(h^2 + h^4 + h^6 + \cdots)$$

ist, denn die ungeraden Potenzen gehören alle zu den reellen Termen.

Weiteres: Die eigentliche "Stärke" des i-Tricks ist aber nicht die quadratische Konvergenz, denn diese gilt auch für den zentralen Differenzenquotienten bei rein reeller Rechnung

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2).$$

Eine Vorwärts-Stabilitätsanalyse zeigt hier jedoch große Fehleranfälligkeit. Numerisch gilt bei Störung der Eingaben

$$\frac{f(x+h)(1+\epsilon_1)-f(x-h)(1+\epsilon_2)}{2h}(1+\epsilon_3)$$
$$=\frac{f(x+h)-f(x-h)}{h}(1+\epsilon_3)+\frac{f(x+h)\epsilon_1-f(x-h)\epsilon_2}{2h}(1+\epsilon_3)$$

Mit Taylor-Entwicklung
$$f(x\pm h)=f(x)+O(h)$$

folgt

$$\frac{f(x+h)(1+\epsilon_1)-f(x-h)(1+\epsilon_2)}{2h}(1+\epsilon_3)$$
$$=\frac{f(x+h)-f(x-h)}{h}+O(eps)+\frac{f(x)(O(eps)+O(h\cdot eps))}{2h}(1+\epsilon_3)$$
$$=f'(x)+O(h^2+eps)+O(\frac{eps}{h}).$$

Für kleine $h\to 0$ steigt zwar die Approximationsgüte, der Rundungsfehler wird jedoch auch schnell groß. Die optimale Schrittweite ist erreicht, wenn beide Fehleranteile gleich sind, d.h. bei
$$h^2\approx\frac{eps}{h}\quad\Leftrightarrow\quad h\approx\sqrt[3]{eps}\approx 10^{-6}$$

bei doppelter Genauigkeit (`double`) und Maschinengenauigkeit 10^{-16}.
Beim i-Trick kann es nicht zur Auslöschung kommen. Hierdurch kann die Schrittweite prinzipiell beliebig klein gewählt werden. Dieses Ergebnis versteckt jedoch das Problem komplexer Arithmethik, bei der es selbst wieder zur Rundungsfehlerproblemen kommen kann.

Programmieraufgabe 7.3

Python bietet mit dem Paket `scipy.interpolate` verschiedene Methoden zur Interpolation. In der Programmvorlage ist die Anwendung von verschiedenen Möglichkeiten demonstriert:

- Die Lagrange-Interpolation mit einem Polynom von Grad n zu $n+1$ Stützstellen
- Die stückweise lineare Interpolation zu $n+1$ Stützstellen
- Ein natürlicher kubischer Spline zu $n+1$ stützstellen

a) Machen Sie sich zunächst mit dem diesmal vollständigen Programm vertraut. Was beobachten Sie für höhere Werte von n? Kommentieren sie eventuell die Ausgabe der Lagrange-Interpolation aus, da diese die Darstellung der anderen Interpolationen überdeckt. Erstellen Sie einen Plot für $n = 8$ sowohl im ganzen Intervall $[0, 2]$ als auch im Ausschnitt $[1, 1.5]$.

b) Erstellen Sie eine Ausgabe, die jeweils den Fehler zwischen $f(x)$ und der entsprechenden Interpolation plottet. Geben Sie für $n = 2, 4, 8, \dots$ die Fehler aus, z.B. mit Hilfe von `np.linalg.norm(...)`. Welche der drei Varianten scheint für $n \to \infty$ zu konvergieren? Welche Ordnung ermitteln Sie experimentell?

Abgabe per Mail an Henry von Wahl (henry.vonwahl@ovgu.de). Abgabe der Kurzfragen in 2er Gruppen, der Übungsaufgaben in den 5er Gruppen. Die theoretische Gruppenaufgabe ist freiwillig.