Technische Universität Berlin Sommersemester 2025
Fakultät II; Institut für Mathematik
M. Karow

Numerische Mathematik I für Ingenieurwissenschaften

4. Übungsblatt zur Vorlesung

Aufgabe 1 **3 Punkte**

Führe mit Hilfe des Hornerschemas eine Polynomdivision von $p(x)$ mit dem Linearfaktor $x - x_0$ durch. Lies aus dem Hornerschema $p(x_0)$ ab.

Üa) $p(x) = 5x^4 + 3x^3 - 30x^2 + 7x + 8, \quad x_0 = 2$

Ha) $p(x) = 4x^4 + 7x^3 + 6x^2 + 71x + 24, \quad x_0 = -3$ Hb) $p(x) = 11x^3 - 38x^2 + 1, \quad x_0 = 4$

Hc) $p(x) = x^4 + x^3 + x^2 + x + 1, \quad x_0 = 1$

Aufgabe 2 **2 Punkte**

Gegeben seien die folgenden Interpolationsdaten (x_j, f_j).

H)

x_j	4	2	0	-1
f_j	7	29	27	-73

Sei $p(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ das zugehörige Interpolationspolynom.

1) Stelle ein lineares Gleichungssystem zur Berechnung der Koeffizienten a_k auf. Das Gleichungssystem muss nicht gelöst werden.

2) Schreibe das Interpolationspolynom p in Lagrange-Darstellung (d.h. als Linearkombination der Lagrange-Basispolynome) hin. Die dabei entstehenden Terme brauchen nicht ausgerechnet zu werden.

Aufgabe 3 **4 Punkte**

Leite durch Ableiten einer Interpolationsparabel die folgenden finiten Differenzenformeln her.

Üa) $f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$.

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

H) $f'(x) \approx \frac{1}{h}\left(-\frac{3}{2} f(x) + 2 f(x+h) - \frac{1}{2} f(x+2h)\right)$.

Ü) Gib für a) und b) eine Fehlerabschätzung an.

Aufgabe 4 **4 Punkte**

Ü) Gegeben sei eine Quadraturformel der Gestalt

$$\int_a^b f(x)\,dx \;\approx\; (b-a)\left(\gamma_1 f(a) + \gamma_2 f(\frac{a+b}{2}) + \gamma_3 f(b)\right). \qquad (*)$$

Bestimme die Gewichte $\gamma_1, \gamma_2, \gamma_3$ so, dass die Quadraturformel für alle Polynome vom Grad ≤ 2 exakt ist.

Ha) Rechne nach, dass die Formel $(*)$ mit den im Tutorium berechneten Gewichten sogar exakt für alle Polynome vom Grad ≤ 3 ist.

Hb) Gegeben sei eine Quadraturformel der Gestalt

$$\int_0^1 f(x)\,dx \;\approx\; \gamma_1 f(0) + \gamma_2 f(\frac{1}{3}).$$

Bestimme die Gewichte γ_1, γ_2 so, dass die Quadraturformel für alle Polynome vom Grad ≤ 1 exakt ist. Ist sie dann auch exakt für Polynome vom Grad 2?

Hinweis zu den Programmieraufgaben: In allen MATLAB-Befehlen, die Polynome betreffen, sind die Polynomkoeffizienten folgendermaßen nummeriert:

$$p(x) = a_1\, x^n + a_2\, x^{n-1} + \ldots + a_n\, x + a_{n+1}.$$

Der Koeffizient vor der höchsten Potenz hat also den Index 1 usw..

Programmieraufgabe 1

Schreibe eine Funktion `interpoly(x,f)`, welche zu gegebenen Stützpunkten (x_j, f_j) mit `x` $= [x_1, ..., x_n]$, `f` $= [f_1, ..., f_n]$ das zugehörige Interpolationspolynom berechnet und im Intervall $x \in [\min x_j,\ \max x_j]$ plottet. Zur Berechnung des Interpolationspolynoms darf eine beliebige Methode benutzt werden. Der Befehl `polyfit` darf aber nicht verwendet werden. Teste das Programm mit Werten f_j der Funktionen $f(x) = \cos(x)$ und $f(x) = 1/(1 + x^2)$; jeweils im Intervall $[-6, 6]$. Verwende sowohl äquidistante Stützstellen als auch Tschebyscheff-Stützstellen (siehe Vorlesung).

Hinweise:
1) n äquidistante Stützstellen im Intervall $[a, b]$ kann man mit `linspace(a,b,n)` erzeugen.
2) Beim Plotten des Polynoms sollen nicht nur die Stützpunkte verbunden werden. Beispiel: Bei 3 Stützpunkten (x_j, f_j), $j = 1, 2, 3$ ist der Graph des Interpolationspolynoms eine Parabel. Um die Parabel zu zeichnen braucht man man aber nicht nur 3 Punkte (das gibt nur eine Zickzack-Linie), sondern ca. 100 Punkte. Es werden also zwei `x`-Vektoren gebraucht. Einer zum Berechnen des Polynoms, und einer zum Plotten.

Programmieraufgabe 2

Sei $f(x) = p(x)/q(x)$, wobei $p, q : \mathbb{R} \to \mathbb{R}$ zwei beliebige differenzierbare Funktionen sind.

Das Newton-Verfahren zur Berechnung einer Nullstelle von f kann in der Form

$$x_{k+1} = x_k - \frac{1}{\frac{p'(x_k)}{p(x_k)} - \frac{q'(x_k)}{q(x_k)}} \qquad (*)$$

geschrieben werden.[1]

Diese Tatsache kann man zur Berechnung aller Nullstellen eines Polynoms p verwenden: Sei $p(x) = a_1\, x^n + a_2\, x^{n-1} + \ldots + a_n\, x + a_{n+1}$ ein Polynom mit reellen oder komplexen Koeffizienten a_k. Seien $z_1, \ldots, z_n \in \mathbb{C}$ die Nullstellen von p. Diese sollen mit dem Newton-Verfahren berechnet werden.
Eine der Nullstellen bekommt man durch direkte Anwendung des Newton-Verfahrens für p:

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

Um die weiteren Nullstellen zu bekommen, stellt man folgende Überlegung an. Für das Polynom p gilt (Zerlegung in Linearfaktoren)

$$p(x) = a_1\,(x - z_1)(x - z_2)\ldots(x - z_n)$$

Angenommen, man hat bereits die Nullstellen $z_1, z_2, \ldots, z_\ell$ berechnet. Sei

$$q(x) = (x - z_1)(x - z_2)\ldots(x - z_\ell). \qquad (**)$$

Dann ist

$$f(x) = p(x)/q(x) = a_1\,(x - z_{\ell+1})\ldots(x - z_n)$$

[1] Grund: Anwendung der Quotientenregel ergibt

$$\frac{f}{f'} = \frac{p/q}{\frac{p'q - q'p}{q^2}} = \frac{1}{\frac{q}{p}\,\frac{p'q-q'p}{q^2}} = \frac{1}{\frac{p'}{p} - \frac{q'}{q}}.$$

das Polynom, welches die übrigen Nullstellen von p als Nullstellen hat. Das Polynom f könnte durch Polynomdivision bestimmt werden. Dies soll hier jedoch nicht getan werden. Stattdessen soll die Formel $(*)$ angewendet werden: Indem man $(**)$ nach der Produktregel ableitet, bekommt man

$$\frac{q'(x)}{q(x)} = \frac{1}{x - z_1} + \frac{1}{x - z_2} + \ldots + \frac{1}{x - z_\ell}.$$

Diesen Ausdruck setzt man in $(*)$ ein. Die Iteration zur Berechnung der 2. Nullstelle z_2 ist demnach

$$x_{k+1} = x_k - \frac{1}{\frac{p'(x_k)}{p(x_k)} - \frac{1}{x_k - z_1}}.$$

Die Iteration zur Berechnung der 3. Nullstelle z_3 ist

$$x_{k+1} = x_k - \frac{1}{\frac{p'(x_k)}{p(x_k)} - \frac{1}{x_k - z_1} - \frac{1}{x_k - z_2}},$$

usw.

Schreibe eine Funktion `z=polyzeros(a)`, welche alle Nullstellen eines gegebenen Polynoms p mit Koeffizienten $[a_1 \;\; a_2 \;\; \ldots \;\; a_{n+1}]$ berechnet. Hinweise:

- Um auch die komplexen Nullstellen zu erwischen, braucht man komplexe Startwerte. Am besten zufällige Startwerte benutzen. Der Befehl dafür ist `rand` (Für komplexe Werte `rand+i*rand`).
- Zur Berechnung von $p(x)$ benutze die MATLAB-Anweisung `polyval`. Zur Berechnung von $p'(x)$ berechne zunächst die Koeffizienten der Ableitung mit `polyder` und benutze dann `polyval`.
- Um abzuschätzen, wie nahe man bereits an einer Nullstelle ist, kann man $|p(x)|$ verwenden. Die MATLAB-Anweisung für den Absolutbetrag einer komplexen Zahl ist `abs`.
- Zum Testen des Programms gibt es mindestens 2 Möglichkeiten:
  1. Wähle $z_1, \ldots, z_n \in \mathbb{C}$ aus, und berechne die Koeffienten a_k des Polynoms $p(x) = (x - z_1)(x - z_2) \ldots (x - z_n)$. Wende das Programm auf diese Koeffizienten an. Dann sollten die z_k als Ergebnisse herauskommen.

  2. Vergleiche die berechneten Nullstellen mit dem Ergebnis, welches die MATLAB-Anweisung `roots` liefert.
- Konkretes Testbeispiel: Das Polynom $x^4 - 1$ (mit Koeffizientenvektor `a=[1 0 0 0 -1]`) hat die vier Nullstellen ± 1 und $\pm i$. Diese vier Zahlen(+kleine Fehler) stehen bei korrekter Programmierung im Vektor `z`.

Programmieraufgabe 3. Schreibe eine Funktion `ableitungsplot(f,a,b,n,h)`, welches die Graphen von f, f', f'' im Intervall $[a, b]$ berechnet und plottet. Dabei ist n die Anzahl der Stützpunkte. Zur Berechnung der Ableitungen sollen die Differenzenformeln

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}, \qquad f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}$$

verwendet werden. Teste mit den Funktionen $f(x) = \sin(x)$ und $f(x) = x^3$.

Programmieraufgabe 4. Schreibe eine Funktion `[T,S]=integral(f,a,b,n)`, welche das Integral

$\int_a^b f(x)\,dx$ mit Hilfe der summierten Trapezregel (Ergebnis `T`) und mit Hilfe der summierten Simpsonregel (Ergebnis `S`) berechnet. Dabei ist `n` die Anzahl der Stützstellen. Die Intervalllänge ist also `h=(b-a)/(n-1)`.