Serie 2

(Abgabe: 8. März 2011, 17 Uhr)

Aufgabe 2.1 *

Gegeben seien die Funktion

$$f(x) = \frac{4}{3+2x}$$

und die Stützstellen $(x_0, x_1, x_2, x_3) = (-1, -0.5, 0.5, 1)$.

(a) Auf dem Intervall $-1 \leq x \leq 1$ berechnen Sie das Lagrangesche Interpolationspolynom $p(x)$ zu $f(x)$ bezüglich der gegebenen Stützstellen.

(b) Zeigen Sie, die folgende Fehlerabschätzung

$$\max_{x\in[-1,1]} |p(x) - f(x)| \leq 16\,.$$

Aufgabe 2.2 *

Sei $f(x) = \cos(\pi x/2)$ und $p(x)$ das Interpolationspolynom 2. Grades, das mit $f(x)$ an den Stellen $x = 0, 1, 2$ übereinstimmt.

(a) Schätzen Sie den maximalen Fehler $\max_{x\in[0,2]} |f(x) - p(x)|$ mit Hilfe der Fehlerschranke aus der Vorlesung ab.

(b) Vergleichen Sie diese Schätzung mit dem tatsächlichen Maximalfehler auf dem Intervall $0 \leq x \leq 2$.

Aufgabe 2.3 (P)*: Beispiel von Runge

Gegeben sei die Funktion $f(x) = \dfrac{1}{1+25x^2}$ für $x \in [-1, 1]$.

(a) Zur Berechnung der Interpolationspolynome n-ten Grades zu den Daten $(x_j^{(n)}, f(x_j^{(n)}))$ mit den äquidistanten Stützstellen

$$\left\{ x_j^{(n)} := -1 + \frac{2j}{n} \right\}_{j=0,1,\ldots,n}$$

schreiben Sie eine `MATLAB`-Routine `RungeBeispiel`. Für $n = 5, 10, 20$ stellen Sie die Interpolationspolynome zusammen mit f graphisch dar und berechnen Sie den maximalen Fehler jedes Interpolationspolynoms.

(b) Um die Funktion $f(x)$ besser zu approximieren, modifizieren Sie Ihre Routine `RungeBeispiel`, so dass Sie die Tschebyscheff-Stützstellen

$$\left\{ x_j^{(n)} := \cos\left(\frac{2j+1}{n+1} \cdot \frac{\pi}{2} \right) \right\}_{j=0,1,\ldots,n}$$

verwenden. Dann wiederholen Sie die Teilaufgabe (a).

Hinweis. Die Struktur der `MATLAB`-Routine könnte z.B. so aussehen:

```
clear all
close all

n = [5 10 20];

for k = 1 : length(n)
    j = 0 : n(k);

    X = ... ;   % aequidistante Stuetzstellen
    Y = ... ;   % Stuetzwerte

    X_T = ... ; % Tschebyscheff-Stuetzstellen
    Y_T = ... ; % Stuetzwerte

    x_plot = -1:0.001:1;
    y_plot = ... ;

    interpol_equidistant = Newton_Interpol(..., ..., ...);
    interpol_tschebyscheff = Newton_Interpol(..., ..., ...);

    max_error(k) = ... ;   % mit den aequidistanten Stuetzstellen
    max_error_T(k) = ... ; % mit den Tschebyscheff-Stuetzstellen

    subplot(length(n), 2, 2*k-1)
    plot(x_plot, interpol_equidistant, '--', x_plot, y_plot, '-')
    legend('p(x)', 'f(x)')
    n_str = ['n=' num2str(n(k)) ' und aequidistante Stuetzstellen'];
    err_str = ['maximaler Fehler: ' num2str(max_error(k))];
    title([n_str '\newline' err_str])

    subplot(length(n), 2, 2*k)
    plot(x_plot, interpol_tschebyscheff, '--', x_plot, y_plot, '-')
    legend('p(x)', 'f(x)')
    n_str = ['n=' num2str(n(k)) ' und Tschebyscheff-Stuetzstellen'];
    err_str = ['maximaler Fehler: ' num2str(max_error_T(k))];
    title([n_str '\newline' err_str])

end
```

Aufgabe 2.4 (P)

Die Funktion $f(x) = x^{1.5} + 0.1\,\mathrm{e}^{3x}\sin(11x)$ soll auf dem Intervall $[0, 1]$ stückweise interpoliert werden. Das Intervall wird dazu in fünf Teilintervalle der Länge 0.2 aufgeteilt.

(a) Schreiben Sie eine MATLAB-Funktion `function z = LinInterpol(x0, x1, x)`, die den Wert der linearen Funktion durch die Punkte $(x_0, f(x_0))$ und $(x_1, f(x_1))$ an der Stelle x ausgibt.

(b) Schreiben Sie eine MATLAB-Funktion `function z = QuadInterpol(x0, x2, x)`, die den Wert der quadratischen Funktion durch die Punkte $(x_0, f(x_0))$, $((x_0 + x_2)/2, f((x_0 + x_2)/2))$ und $(x_2, f(x_2))$ an der Stelle x ausgibt.

Laden Sie die Datei `main.m` von der Webseite herunter und zeichnen Sie damit die Funktion $f(x)$ und die Interpolationen aus (a) und (b) in zwei verschiedene Bilder.

Hinweis. Schreiben Sie eine MATLAB-Funktion `function y = f(x)`, die $f(x)$ an der Stelle x auswertet. Zur Implementation von `LinInterpol` und `QuadInterpol` könnten Sie Ihre Routine `Newton_Interpol` verwenden.

Aufgabe 2.5 (A+B)*

Sei $a > 0$. Berechnen Sie das quadratische Interpolationspolynom $p(x)$ zur Funktion $f(x) = x^4$ und den Stützstellen $x_0 = -1$, $x_1 = a$ und $x_2 = 1$. Finden Sie nun ein $\xi \in (-1, 1)$, so dass

$$f(x) - p(x) = (x - x_0)(x - x_1)(x - x_2)\frac{f^{(3)}(\xi)}{3!} \quad \forall x \in (-1, 1)\,.$$

Hinweis. Es gilt $\xi = \xi(x, a)$, d.h. ξ hängt von x und von a ab. Das Interpolationspolynom soll man so weit als möglich vereinfachen. Dies erleichtert die weitere Rechnung stark!

Aufgabe 2.6 (A+B)

Gegeben seien $n + 1$ paarweise verschiedene Stützstellen $x_0, x_1, \ldots, x_n$ und die Funktion $f_n(x) = x^n$. Zeigen Sie:

(a) $\delta^n f_n[x_0, x_1, \ldots, x_n] = 1$.

(b) $\delta^{n-1} f_n[x_1, x_2, \ldots, x_n] = x_1 + x_2 + \ldots + x_n$.
Hinweis. Sei $\omega(x) := (x - x_1) \cdot \ldots \cdot (x - x_n)$. Berechnen Sie $\delta^{n-1}\omega[x_1, \ldots, x_n]$. Überlegen Sie nun, dass

$$\omega(x) = f_n(x) - f_{n-1}(x)(x_1 + x_2 + \ldots + x_n) + p(x)\,,$$

wobei p ein Polynom von Grad kleiner als $n - 1$ ist. Dann verwenden Sie die Teilaufgabe (a) und die Linearität der dividierten Differenzen, d.h.

$$\delta^{n-1}(\alpha g + \beta h)[x_1, \ldots, x_n] = \alpha\delta^{n-1} g[x_1, \ldots, x_n] + \beta\delta^{n-1} h[x_1, \ldots, x_n]\,,$$

für $\alpha, \beta \in \mathbb{R}$ und beliebige Funktionen g und h.

Allgemeine Informationen zur Vorlesung und Übungsblätter befinden sich auf der Webseite `http://www.math.unibas.ch/~cohen`