AKNUM Computernumerik UE 106.054
Computernumerik VU 101.484
Numerische Mathematik f. LA UE 106.986

4. Übungsblatt (Dienstag 19.05.2015)

Kreuzen Sie in der Übung diejenigen Beispiele an, die Sie durchgerechnet haben und an der Tafel präsentieren können. Bereiten Sie alle numerische Beispiele (*) mit MATLAB vor und bringen Sie den MATLAB Code auf USB Stick oder Laptop, sodass Sie diesen ebenfalls in der Übung vorführen können. Auch bei Beispielen ohne (*) ist die Verwendung von MATLAB manchmal nützlich. Wenden Sie sich bei Unklarheiten an Stefan Wurm (`stefan.wurm@tuwien.ac.at`) *oder an Caroline Geiersbach* (`caroline.geiersbach@tuwien.ac.at`).

1. Betrachten Sie die Funktion
$$f(x) = \sinh x$$
auf dem Intervall $[0, \pi]$. Führen Sie verschiedene Polynominterpolationen $p(x)$ durch und geben Sie jeweils den Fehler $f(x) - p(x)$ an der Stelle $\frac{\pi}{3}$ an:
   (a) Polynom vom Grad 1: $x_0 = 0, x_1 = \pi$
   (b) Polynom vom Grad 2: x_0, x_1 und $x_2 = \frac{\pi}{2}$
   (c) Polynom vom Grad 3: $x_0, x_1, x_2, x_3 = \frac{\pi}{4}$ und $x_0, x_1, x_2, x_3 = \frac{3\pi}{4}$
   (d) Theoretische Abschätzung des Interpolationsfehlers für (c) an der Stelle $\frac{\pi}{3}$

2. Betrachten Sie die Funktion
$$f(x) = \sinh x$$
auf dem Intervall $[0, \pi]$. Führen Sie verschiedene Polynominterpolationen $p(x)$ durch und geben Sie jeweils den Fehler $f(x) - p(x)$ an der Stelle $\frac{\pi}{3}$ an:
   (e) gerades Polynom 4. Grades: x_0, x_1, x_2
   (f) ungerades Polynom 3. Grades: x_0, x_1
   (g) Taylorpolynom vom Grad $n = 4$ an der Entwicklungsstelle $x_0 = 0$
   (h) Interpolation mit Lagrange Polynomen an x_0, x_1, x_2

3. (*) *Nevilleschema.* Programmieren Sie eine MATLAB Funktion mit der Signatur `function y0 = neville(x,y,x0)`, wobei `x` ein Vektor mit Stützstellen, `y` der zugehörige Vektor mit Funktionswerten, `x0` die gewünschte Auswertungsstelle und `y0` der Wert an dieser Stelle sein sollen. Testen Sie die Funktion anhand des folgenden Problems: Die Stützpunkte $(0, 1)$, $(1, 3)$ und $(3, 2)$ sind gegeben. Berechnen Sie den Wert des Interpolationspolynoms an der Stelle $x = 2$.

4. *Hermiteinterpolation.* Berechnen Sie händisch das Hermiteinterpolationspolynom zu den Stützstellen
$$f(1) = 2, \quad f'(1) = 1, \quad f(4) = 3, \quad f'(4) = 2 \quad f''(4) = 1.$$
Verwenden Sie dazu das Nevilleschema.

5. Berechnen Sie für die Stützpunkte $\{(x_k, y_k)\} = \{(0, 0), (1, 3), (2, 2), (3, 1)\}$ die Dividierten Differenzen
$$[x_0, x_1] \qquad [x_0, x_1, x_2] \qquad [x_0, x_1, x_2, x_3]$$
und geben Sie das entsprechende Newtonsche Interpolationspolynom an.

6. (*) Interpolieren Sie mithilfe von MATLAB die Funktion von Runge
$$f(x) = \frac{1}{25x^2 + 1}$$
auf $[-1, 1]$ auf zwei Arten für $n = 20$:
   (a) an den äquidistanten Punkten $x_j = -1 + \frac{2j}{n} \quad j = 0, 1, \ldots, n$
   (b) an den Nullstellen des $n + 1$-ten Tschebyscheff Polynoms T_{n+1}

   Plotten Sie den Fehlerverlauf von beiden Varianten und interpretieren Sie. *Hinweis:* Alternantensatz.

7. (*) Um die Koeffizienten eine Polynoms in der Monom-Basis zu erhalten, kann folgendes Vorgehen über die Lagrange-Interpolation benutzt werden.

   (a) Bestimmen Sie die Koeffizienten der Lagrange-Basispolynome in der Monombasis.
   Anleitung: Das i-te Lagrange-Basispolynom φ_i ist gegeben durch

$$\varphi_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_i}{x_j - x_i} = \prod_{j=0, j\neq i}^{n} \left(\frac{1}{x_j - x_i} \cdot x - \frac{x_i}{x_j - x_i} \right).$$

   Es lässt sich daher dadurch berechnen, dass das konstante Polynom 1 mit den Polynomen $\frac{1}{x_j - x_i} \cdot x - \frac{x_i}{x_j - x_i}$ iterativ zum gewünschten φ_i aufmultipliziert wird.
   Das lässt sich technisch am einfachsten realisieren, indem nur die Koeffizienten zwischengespeichert werden und jede der Multiplikationen mittels der MATLAB-Funktion `conv` durchgeführt wird. Dadurch erhält man abschließend die Darstellung in der Monom-Basis

$$\varphi_i(x) = \sum_{j=0}^{n} c_{ij} x^j.$$

   (b) Die Koeffizienten lassen sich anschließend leicht bestimmen. Da das gewünschte Interpolationspolynom die Darstellung

$$p(x) = \sum_{i=0}^{n} f(x_i)\varphi_i(x) = \sum_{i=0}^{n} a_i x^i$$

   und die φ_i die oben berechneten Darstellungen besitzen, ergibt sich

$$a_i = f(x_i) \sum_{j=0}^{n} c_{ij},$$

   was genau den gesuchten Koeffizienten in der Monombasis entspricht.

Implementieren Sie dieses Vorgehen, indem Sie eine MATLAB-Funktion mit der Signatur `function [c] = lagrange(x,y)` programmieren, wobei `x` der Vektor der Stützstellen, `y` der Vektor der Stützwerte und `c` der resultierende Koeffizientenvektor in der Monombasis seien.

Vergleichen Sie Ihre Funktion mit dem Berechnen der Koeffizienten durch Lösen des Gleichungssystems $Ax = b$. Dabei bezeichnet A die Vandermondematrix (*Hinweis:* MATLAB-Funktion `vander`), wobei die Funktion $\ln x$ an den Stützstellen $0.1, 0.5, 0.6, 1, 2$ interpoliert werden soll.