9 Übung

Aufgabe 9.1: Kondition von Hilbert-Matrizen 7 Punkte

Die Hilbertmatrix $H_n \in \mathbb{R}^{n\times n}$ wird bekanntlich durch $(H_n)_{i,j} := \frac{1}{i+j-1}$ definiert.

a) Berechnen Sie (von Hand) die Konditionen $\kappa_1(H_3)$, $\kappa_\infty(H_3)$.

$$\text{Hinweis:} \quad H_3^{-1} = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix} .$$

b) Für $b \in \mathbb{R}^n$, $b_i = \sum_{j=1}^n (H_n)_{i,j}$ ist $x = \begin{pmatrix} 1 & \dots & 1 \end{pmatrix}^T$ Lösung von $H_n x = b$. Berechnen Sie für $n \in \{4, 5, \dots, 15\}$ die numerischen Lösungen (`MatLab`, `Modula-3`, …) für $H_n x_n = b$, die relativen Fehler $\frac{\|x-x_n\|}{\|x\|}$ und die Norm der Residuen $r = b - H_n x_n$.

c) Berechnen Sie, wieder für $n \in \{4, 5, \dots, 15\}$, die Konditionen $\kappa_1(H_n)$, $\kappa_2(H_n)$, $\kappa_\infty(H_n)$.

Aufgabe 9.2: Matrixnormen 6 Punkte

Zeigen Sie für $A \in \mathbb{R}^{m\times n}$:

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^m |a_{ij}| .$$

Aufgabe 9.3: Horner-Schema 7 Punkte

Ein Polynom

$$p(x) = \sum_{k=0}^n a_k x^k, \quad a_k \in \mathbb{R}, \tag{1}$$

sollte an einer Stelle $x \in \mathbb{R}$ aus Geschwindigkeitsgründen und zur Vermeidung von Überläufen mit dem *Horner-Schema* ausgewertet werden. Dabei wird $p(x)$ wie folgt umgeschrieben:

$$p(x) = \underbrace{(\dots((}_{n-1} a_n x + a_{n-1})x + a_{n-2})x + \dots + a_1)x + a_0 \tag{2}$$

a) Formulieren Sie einen Algorithmus, der (2) realisiert und ausgehend von $y_n := a_n$ rekursiv $y_0 := p(x)$ für einen Wert $x \in \mathbb{R}$ berechnet.
Geben Sie die Anzahl der benötigten elementaren Rechenoperationen zur Berechnung von $y = p(x)$ mit der naiven Formel (1) und dem Horner-Schema an.
Angenommen, Sie wollen den Graphen eines Polynoms vom Grad 20 im Intervall $[-1, 1]$ zeichnen. Dazu werten Sie das Polynom zwischen den Intervallenden an Stützstellen im Abstand $h = 0.01$ aus und verbinden die berechneten Funktionswerte zum Graphen des Polynoms. Wie lange würde die Berechnung bei Benutzung von (1) bzw. des Horner-Schemas dauern, falls die Durchführung einer elementaren Rechenoperation 1/100 Sekunde benötigt?

b) Implementieren Sie den Algorithmus aus a). Die Argumente der Funktion seien der reelle Wert x und ein Vektor p, der die Koeffizienten des Polynoms enthält.

In `MatLab` verwenden Polynomfunktionen wie `poly`, `polyval` und `roots` diese Reihenfolge:
$$\mathtt{p} = [a_n, a_{n-1}, \ldots, a_0] \,.$$

Routinen für `Modula-3` wie die in `LongRealPolynomial` verwenden dagegen die Reihenfolge
$$\mathtt{p} = \mathtt{P.TBody}\{a_0, \ldots, a_{n-1}, a_n\} \,.$$

Der Funktionskopf soll also folgendes Aussehen haben:

```
function y = horner(p,x)
```

bzw.

```
PROCEDURE Horner (READONLY p: ARRAY OF LONGREAL;
                           x: LONGREAL): LONGREAL;
```

Programmieren Sie eine zweite Funktion mit dem gleichen Funktionskopf aber anderem Namen, die die Formel (1) auf naive Weise auswertet.

Testen Sie diese Funktionen anhand von

$$\begin{aligned} p_1(x) &= 16x^4 - 96x^3 + 216x^2 - 216x + 81, & x &\in \{1.49, 1.499, 1.501, 1.51\}\,; \\ p_2(x) &= x^3 - 10^{103}x^2 + 1, & x &= 10^{103}\,. \end{aligned}$$

Vergleichen Sie die mit Ihren Funktionen berechneten Werte mit den von `polyval` (`MatLab`-Funktion) bzw. `LongRealPolynomial.Eval` (`Modula-3`-Funktion) berechneten. Was fällt auf?