Einführung in die Numerik
4. Programmierübung

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik **WS 2015/16**
Prof. Dr. Herbert Egger, Dr. Mirjam Walloth, Thomas Kugler **12.1.2015**

Programmierübung

Aufgabe P1

generateData.m: Schreiben Sie eine Funktion

```
function [x,y]=generateData(m,f)
```

welche Vektoren mit den Stützstellen $x_i = (i-1)/(m-1)$ und den Stützwerten $y_i = f(x_i)$, $i = 1,\dots,m$ liefert. Dabei sei f eine Funktion die mit `f(x)` ausgewertet werden kann.

leastSquareSystem.m Gesucht ist das Polynom $p_n = a_0 + a_1 x + \dots + a_n x^n$, welches die Werte y_i an den Stellen x_i, $i = 1,\dots,m$ bestmöglich approximiert. Schreiben Sie eine Funktion

```
function [A,b]=leastSquareSystem(x,y,n)
```

Welches das zugehörige überbestimmte Gleichungssystem aufstellt.

testp1.m: Überprüfen Sie Ihre Routine mit der Eingabe $f = @(x)1+x+x.^2$, $m = 2,5,10$ und $n = 0,1,2$.

Aufgabe P2

testp2.m: Gesucht ist das Polynom $p_n \in \Pi_n$ welches die Funktion $f(x) = \cos(2\pi x)$ an äquidistanten Stützstellen x_i, $i = 1,\dots,m$ im Intervall $[0,1]$ möglichst gut approximiert.

(a) Stellen Sie mit Hilfe der Methoden aus Aufgabe P1 das entsprechende Ausgleichsproblem auf und lösen Sie dieses über die Normalengleichungen. Berechnen Sie die Lösungen für $m = 20$ und $n = 0,\dots,6$. Erklären sie die Ergebnisse.

(b) Stellen Sie die Ausgleichspolynome und die Daten in einem Plot geeignet dar. Hierzu sollten die Polynome an einem feineren Gitter ausgewertet werden!

(c) Berechnen Sie den Least-Squares Fehler $e_n = 1/m \sum_i |f(x_i) - p_n(x_i)|^2 \approx \|f - p_n\|^2_{L^2(0,1)}$ und plotten Sie e_n vs n mit `semilogy`. Erklären Sie das Ergebnis. Wie schnelle konvergiert e_n mit $n \to \infty$; vgl Bsp. 1.17 im Skript.

Aufgabe P3

testp3.m: Gesucht ist das Polynom $p_n \in \Pi_n$ welches die Funktion $f(x) = 1 + x + \dots + x^n$ an äquidistanten Stützstellen x_i, $i = 1,\dots,m$ im Intervall $[0,1]$ möglichst gut approximiert.

(a) Wie lautet die exakte Lösung bzw der exakte Koeffizientenvektor a?

(b) Stellen Sie für $m = 100$ und $n = 0,\dots,10$ das entsprechende Ausgleichsproblem auf und bestimmen Sie die Lösung der Ausgleichsprobleme sowohl mit der Normalengleichungsmethode als auch mit der QR-Zerlegung (MATLAB Routine `qr`).

(c) Berechnen Sie die Kondition $\mathrm{cond}(A^\top A)$ und $\mathrm{cond}(R_1)$ für die Systemmatrizen der beiden Lösungsverfahren und stellen Sie diese zusammen mit dem Polynomgrad n in einer Tabelle dar. Erklären Sie die Ergebnisse.

(d) Berechnen Sie die Fehler $\|a - a^{QR}\|_\infty$ und $\|a - a^{NG}\|_\infty$ in den Koeffizientenvektoren und stellen Sie diese tabellarisch dar. Erklären Sie die Ergebnisse!

Aufgabe P4

richardson.m: Implementieren Sie eine Funktion

```
function [z,res]=richardsonNormal(A,b,tol,maxit,omega)
```

welche ausgehend von $x^0 = 0$ die Lösung der Normalengleichungen $A^\top Az = A^\top b$ mit Hilfe der Richardsoniteration mit Schrittweite ω berechnet. Dabei sei z die letzte Iterierte und res ein Vektor mit den Residuen $\text{res}^k = \|A^\top(Az^k - b)\|$, die während der Iteration erhalten wurden. Die Iteration soll abgebrochen werden, sobald $\text{res}^k < \text{tolres}^0$ gilt oder die maximale Anzahl der Iterationen erreicht wurde. Schätzen Sie die Komplexität für einen einzelnen Iterationsschritt in Abhängigkeit der Systemgröße $[m,n]$ ab.

testp4.m: Wir verwenden nun die Richardson-Iteration zum Lösen von Ausgleichsproblemen.

(a) Überzeugen Sie sich, dass die Schrittweite $\omega^k = 1/|||A^\top A|||_2$ immer zur Konvergenz führt, falls $A^\top A$ regulär ist.

(b) Wenden Sie obiges Verfahren zum Lösen der Ausgleichsprobleme aus Aufgabe P3 an.

(c) Überprüfen Sie durch Kontrolle der Residuen res^k die lineare Konvergenz des Verfahrens. Tragen Sie hierzu den Interationsindex k, das Residuum $\text{res}^k = \|r^k\|$ und den Faktor $\text{res}^k/\text{res}^{k-1}$ in einer Tabelle auf.

(d) Bestimmen Sie in Abhängigkeit es Polynomgrades n den Konvergenzfaktor $q = \max_k \|r^k\|/\|r^{k-1}\|$ und vergleichen Sie mit der Norm der Iterationsmatrix $|||M(\omega)|||_2$. Geben Sie die Konvergenzfaktoren in Abhängigkeit vom Polynomgrad n in einer Tabelle aus. Erklären Sie das Verhalten für größer werdendes n.

Organisatorische Hinweise:

a) Legen Sie ein Verzeichnis `pp4` an und speichern Sie alle Funktionen und Skripte für diese Übung darin ab. Der Inhalt des Verzeichnisses sollte hier also sein:

```
testp1.m generateData.m leastSquaresSystem.m
testp2.m
testp3.m
testp4.m richardsonNormal.m
```

b) Zippen Sie das Verzeichnis; in Linux: `zip -r pp4.zip pp4` von außerhalb des Verzeichnisses. In Windows kann WinZip oder ein ähnliches Tool verwendet werden.

c) Überprüfen Sie, dass alle Skripte und Funktionen lauffähig sind!

d) Kommentieren Sie die Funktionen geeignet. Insbesondere sollten Eingabe und Ausgabe Parameter beschrieben werden. `help <funktionsname>` sollte Information über die Routinen liefern.

e) Antworten auf die Fragen sollten in den `testp*.m` files als Kommentare hinterlegt werden.