Winfried Auzinger
Klaus Felsenstein
Dirk Praetorius

SS 2010
(23. April 2010)

Übungsaufgaben zur VL Computermathematik

Serie 5

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung fix vorzubereiten und werden dort abgeprüft (Minimalerfordernis).

Kopieren Sie Ihre Worksheets auf Ihren Account auf `lva.student.tuwien.ac.at`*. Überprüfen Sie vor der Übung, ob Ihre Codes unter Maple 12 auf* `lva` *einwandfrei funktionieren. Es wird empfohlen, für jedes Beispiel im Verzeichnis* `serie05` *ein eigenes Worksheet mit dem Namen* `aufgabey-y.mw` *anzulegen. Alle zu verfassenden Prozeduren sind zu kommentieren und – je nach Angabe – mit verschiedenen Werten für die Parameter auszutesten.*

Diese Serie ist eine bunte Mischung aus verschiedenen Aufgabenstellungen.

Aufgabe 5.1*. Schreiben Sie eine Prozedur [1]

```
GenerateMatrix(f::procedure)::Matrix;
```

die zu einer vorgegebenen linearen Abbildung `f`: $\mathbb{R}^n \to \mathbb{R}^m$ (als Prozedur gegeben) die zugehörige Koeffizientenmatrix $A \in \mathbb{R}^{m \times n}$ zurückgibt.

Anmerkung: Es darf a priori angenommen werden, dass `f` tatsächlich eine lineare Abbildung repräsentiert. Die Dimensionen `m` und `n` sollen jedoch automatisch ermittelbar sein. Dazu fordert man, dass die Prozedur `f` diese Information zu Verfügung stellt.

Vorschlag: `x` und das Ergebnis von `f` sollen Objekte vom Typ `Vector` sein, und ein Aufruf ohne Parameter soll die Dimensionen liefern: `m,n:=f()`.

Aufgabe 5.2*. Schreiben Sie eine Prozedur `checkproj(P::Matrix)`, die überprüft, ob eine gegebene $n \times n$-Matrix P eine *[Orthogonal]-Projektion* auf einen Unterraum des $\mathbb{R}^n$ repräsentiert. Dies ist genau dann der Fall wenn gilt $P^2 = P\,[= P^T]$, wobei die Symmetrieeigenschaft $P = P^T$ die Orthogonalität des Projektors charakterisiert. (Die Matrix $Q := I - P$ ist dann auch ein [Orthogonal]-Projektor; P und Q projizieren komplementär zueinander.) `checkproj` gibt die betreffende Information in geeigneter Weise zurück (z.B. 0 = kein Projektor, 1 = Projektor, 2 = Orthogonalprojektor). Falls `P` nicht quadratisch ist, Abbruch mit `error`.

Für den Fall $n = 2$ und Zutreffen der Projektoreigenschaft $P = P^2$ soll die Prozedur außerdem eine Grafik zeichnen, die die Bildräume von P und Q als Geraden darstellt, mit entsprechender Beschriftung, aber nur falls $\mathrm{Rang}(P) = 1$ (Rang 0 oder 2 sind Trivialfälle).

Alles für exakte numerische Daten und in exakter Arithmetik (rational), also ohne Rundungseffekte.[2]

Aufgabe 5.3*. Zu beliebigen $n+1$ gegebenen (paarweise verschiedenen) Stellen ('Knoten') $x_0, \ldots, x_n \in \mathbb{R}$ und Funktionswerten $y_0, \ldots, y_n$ gibt es genau ein Interpolationspolynom $p(x)$ vom Grad $\leq n$ mit $p(x_i) \equiv y_i$. Die sogenannte 'Lagrange-Darstellung' von $p(x)$ lautet

$$p(x) = \sum_{i=0}^{n} y_i\, L_i(x),$$

dabei sind die $L_i(x)$ die zu den Knoten x_i gehörigen *Lagrange-Polynome*

$$L_i(x) = \prod_{j=0,\, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}\,, \quad i = 0 \ldots n \quad \text{(mit der Eigenschaft } L_i(x_j) = \delta_{ij}\text{)}.$$

[1] Die hier verwendete Syntax legt fest, dass die Prozedur ein Objekt vom Typ `Matrix` zurückliefern soll – dies kann bei Bedarf zur Laufzeit automatisch gecheckt werden. Siehe `? proc`:
'The closing parenthesis of the `parameterSequence` may optionally be followed by `::`, a `returnType`, and a `;`. This is not a type declaration, but rather an *assertion*. If `kernelopts(assertlevel)` is set to `2`, the type of the returned value is checked as the procedure returns. If the type violates the assertion, then an exception is raised.'

[2] 'Exakte Rangbestimmung' ist für float-Daten und in Gleitpunktarithmetik im allgemeinen nicht möglich.

Verwenden Sie diese Darstellung, um zu vorgegebenen Knoten x_i (übergeben als Array, `xnodes(0..n)`) und zu einer gegebenen Auswertungsstelle `x` ein Array `d` zu generieren (von t abhängig, Dimension `0...n`), so dass für den *Ableitungswert* $p'(x)$ gilt

$$p'(x) = \sum_{j=0}^{n} d_j \, y_j.$$

Man beachte: Die $d_j = d(x)$ sind von x und natürlich von den Knoten x_i abhängig, aber nicht von den y_i. Die obigen Darstellungen für $p(x)$ und $p'(x)$ gelten für beliebige Funktionsdaten y_i.

Umsetzung als Prozedur `Dp_coefficients(x,xnodes)`. (Hinweis: Differenzieren der Lagrange-Darstellung für $p(x)$ – diesen Job kann natürlich Maple übernehmen.)

Anmerkung: Man kann das entweder mit floats realisieren oder in exakter rationaler Rechnung (soferne die Daten als rationale Zahlen gegeben sind) – das bleibt Ihnen überlassen.

Aufgabe 5.4*. Generieren Sie mit Hilfe der Prozedur `Dp_coefficients` aus Aufgabe 5.3 für einen konkreten Satz von Knoten x_i (z.B. $n = 4$, $x_i = i/n$) eine 'Differentiationsmatrix' in Form eines ein zweidimensionalen Arrays `(0..n,0..n)`, das zeilenweise die Koeffizienten $d_{ij} = d_j(x_i)$ enthält (d.h. die Auswertungsstelle x durchläuft alle Knoten x_i). Speichern Sie diese Differentiationsmatrix in geeigneter Weise in einer externen Datei ab. Schreiben Sie auch eine Prozedur `Dapply(Dfile,y)`, die die Differentiationsmatrix von der Datei einliest und zu gegebenen Funktionswerten y_i die Werte $p'(x_i)$ in einem Array zurückliefert.

Aufgabe 5.5. Schreiben Sie eine Prozedur `dataplot(dfile)`, die mittels `plots[pointplot](..., style=point)` und `plots[pointplot](...,style=line)` eine Grafik erstellt, die zusammen mit gegebenen Wertepaaren (x_i, y_i) auch die stückweise lineare Funktion $y(x)$ darstellt, die durch die entsprechenden Punkte verläuft (Polygonzug). Zusätzlich ist auch das Interpolationspolynom $p(x)$ gemäß Aufgabe 5.3 darzustellen. (Verwende `? plots[display]`.)

Es ist anzunehmen, dass die Wertepaare numerisch (als floats) zeilenweise in einer externen Datei `dfile` nach den x_i aufsteigend sortiert gespeichert sind. Verwenden Sie einen `try ... catch`-Block dazu, um den Fall der Nicht-Existenz der Datei `dfile` in kontrollierter Weise abzufangen, und generieren Sie für diesen Fall Ihre eigene Fehlermeldung (`error ...`).

Aufgabe 5.6. Recherchieren Sie, wie man ebene bzw. räumliche Kurven grafisch darstellt, die in Parameterdarstellung gegeben sind: $\{(x(t), y(t)),\ t \in [a, b]\}$ bzw. $\{(x(t), y(t), z(t)),\ t \in [a, b]\}$, und implementieren Sie einige Beispiele. Lösen Sie z.B. ein einfaches System von linearen Differentialgleichungen, wie etwa

```
dsolve([D(x)(t)=-2*y(t),D(y)(t)=x(t)/2,x(0)=1,y(0)=1],[x(t),y(t)]);
```

und plotten Sie die Lösung. Analog für ein System von 3 linearen Differentialgleichungen für $x(t), y(t), z(t)$.

Hinweis: `? plots/parametric`, `? plots[spacecurve]`; siehe auch `? plots[tubeplot]` .

Aufgabe 5.7. Verwenden Sie `plot` und `plots[animate]` dazu, um das Konvergenzverhalten der Partialsummen der Taylorreihe (`taylor(ln(1+x),x=0,...)`)

$$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} + \ldots$$

gegen die Funktion $\ln(1+x)$ im Intervall $x \in (-1, 1)$ zu veranschaulichen. Kreieren Sie eine möglichst 'schöne' Animation, unter Verwendung diverser zur Verfügung stehender plot-Optionen.

Anmerkung: Die Reihe hat den Konvergenzradius 1. Versuchen Sie Sie insbesondere das Konvergenzverhalten in der Nähe der Intervallenden zu verdeutlichen. Um welche speziellen Reihen handelt es sich für $x = -1$, $x = +1$?

Aufgabe 5.8. Die Maple-Befehle `kernelopts(...)` und `interface(...)` dienen der individuellen Konfiguration des Verhaltens zwischen dem Benutzer und dem Kernel bzw. dem Worksheet-Interface. Studieren Sie die Online-Hilfe (`? kernelopts`, `? interface`) und erstellen Sie ein Worksheet, in dem einige wichtige Einstellungen dokumentiert sind.