Winfried Auzinger
Klaus Felsenstein
Dirk Praetorius

SS 2009
(28. Mai 2009)

Übungen zur Vorlesung Computermathematik

Serie 9

Die Aufgaben mit Stern (*) sind bis zur Übung in der kommenden Woche vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung. Kopieren Sie bitte Ihre Worksheets auf Ihren Account auf `lva.student.tuwien.ac.at`. *Überprüfen Sie vor der Übung, ob Ihre Codes unter Maple auf* `lva` *einwandfrei funktionieren.*

Diese letzte Maple-Serie ist eine bunte Mischung verschiedener Aufgabenstellungen.

Aufgabe 9.1*. Verwenden Sie `plot` und `plots[animate]` dazu, um das Konvergenzverhalten der Exponentialreihe gegen die Funktion e^x zu veranschaulichen. Kreieren Sie eine möglichst 'schöne' Animation, unter Verwendung diverser zur Verfügung stehender plot-Optionen. Vergleichen Sie auch mit den Polynomen $(1+\frac{x}{n})^n$ (nicht identisch mit $\sum_{j=0}^{n}\frac{x^j}{j!}$).

Aufgabe 9.2*. Zu $n+1$ gegebenen (verschiedenen) Stellen $x_0,\ldots,x_n\in\mathbb{R}$ (als `Array` vorgegeben) und Funktionswerten $y_0,\ldots,y_n$ (ebenfalls als `Array` vorgegeben) gibt es genau ein Interpolationspolynom $p(t)$ vom Grad $\le n$ mit $p(x_i)\equiv y_i$. Die sogenannte 'Lagrange-Darstellung' von $p(t)$ lautet

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

mit den *Lagrange-Polynomen*

$$L_i(t)=\prod_{j=0,\,j\neq i}^{n}\frac{t-x_j}{x_i-x_j},\quad i=0\ldots n\qquad (L_i(x_j)=\delta_{ij})$$

Verwenden Sie diese Darstellung dazu, um zu einer gegebenen Auswertungsstelle `t` ein Array `d` zu generieren (von t abhängig, Dimension `0...n`), so dass für den Ableitungswert $p'(t)$ gilt

$$p'(t)=\sum_{j=0}^{n} y_j\,d_j.$$

Umsetzung als Prozedur `Lagrange-derivative(t,x,y)`. Generieren Sie mit Hilfe dieser Prozedur ein zweidimensionales Array `D`, das zeilenweise die Koeffizienten $d_{ij}=d_j(x_i)$ enthält (d.h. die Auswertungsstelle t durchläuft alle Knoten x_i). Speichern Sie diese 'Differentiationsmatrix' `D` in geeigneter Weise in einer externen Datei ab (so dass `D` wieder eingelesen werden kann − testen!).

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

Hinweis: Differenzieren der Lagrange-Darstellung (diesen Job kann natürlich Maple übernehmen).

Aufgabe 9.3*. Schreiben Sie eine Prozedur, die mittels `plots[pointplot](...,style=line)` eine Grafik erstellt, die zu gegebenen Wertepaaren (x_i,y_i) die stückweise lineare Funktion $y(x)$ darstellt, die durch die entsprechenden Punkte verläuft.

Es ist anzunehmen, dass die Wertepaare numerisch (als floats) zeilenweise in einer externen Datei gespeichert sind, aber nicht notwendigerweise aufsteigend nach den x_i sortiert. Für einen korrekten plot muss man daher umsortieren, denn sonst erstellt `pointplot` einen Polygonzug, der nicht der gewünschten Funktion entspricht.

Aufgabe 9.4*. Schreiben Sie unter Verwendung des `try` ⋯ `catch` ⋯ `finally` Mechanismus eine Maple-Prozedur

```
fwriteMatrix(filename::string,M::Matrix),
```

die eine gegebene Matrix `M` von `rational` oder `float`-Werten mittels `writedata` auf eine Datei mit dem Namen `filename` schreibt (`? writedata`). Die 'exception' tritt auf, wenn `M` unzulässige Einträge enthält, so dass `writedata(...,...,float)` die Operation nicht ausführen kann. In diesem Fall soll eine gleich große Matrix nur bestehend aus Nullen ausgegeben werden.

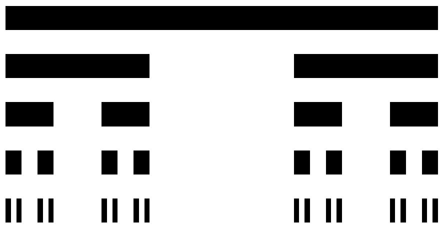
Zum Testen kann man auch `writedata(terminal,...)` verwenden (Ausgabe am Bildschirm).

Aufgabe 9.5. Schreiben Sie eine Maple-Prozedur `filedotprod(file1,file2)`, die zu zwei numerisch auf externen Dateien gespeicherten Vektoren das innere Produkt berechnet. `file1,file2`: Je ein Eintrag pro Zeile.

Laut Dokumentation (`?fscanf`) wird beim Einlesen 0 zurückgeliefert, falls das Ende der Datei erreicht wurde. Dieses Verhalten kann man dazu ausnützen, den kürzeren der beiden Vektoren mit Nullen 'aufzufüllen'. Wie aber erkennt man, wenn man am Ende beider Dateien angelangt ist? Überlegen Sie dafür eine Lösung.

Aufgabe 9.6. Eine rekursive Grafik-Prozedur:

Die sogenannte *Cantormenge* entsteht dadurch, dass man aus dem Intervall $[0,1]$ den mittleren Teil $(1/3,2/3)$ herausnimmt. Dann betrachtet man die beiden verbleibenden Intervalle $[0,1/3]$ und $[2/3,1]$, nimmt wiederum jeweils das mittlere (offene) Drittel heraus, usw. (rekursiv bis ins Unendliche fortgesetzt, siehe Abbildung).



Schreiben Sie eine rekursive Maple-Prozedur `PlotCantorSet(n)`, die den n-ten Zustand dieses fraktalen Prozesses grafisch mittels waagrechter Balken darstellt ($n=0$ entspricht dem vollen Intervall $[0,1]$, usw.).

(Freiwillige Übung: Reproduktion der obigen Abbildung.)

Aufgabe 9.7. Schreiben Sie eine Maple-Prozedur

`CreateMatrix(f::procedure)::Matrix`,

die zu einer vorgegebenen linearen Abbildung `f`: $\mathbb{C}^n \to \mathbb{C}^m$ (als Prozedur gegeben) die zugehörige Koeffizientenmatrix $A \in \mathbb{C}^{m\times n}$ zurückliefert. Es darf angenommen werden, dass `f` tatsächlich eine lineare Abbildung repräsentiert.

Anmerkung: Sinnvollerweise wird man über das Verhalten von `f` gewisse Konventionen treffen, etwa (Vorschlag): `x` und das Ergebnis von `f` sollen Objekte vom Typ `Vector` sein, und `f` kann auch danach abgefragt werden kann, was die Dimensionen `m` und `n` sind, etwa durch den Aufruf ohne Parameter, `m,n:=f()` .

Aufgabe 9.8. Schreiben Sie eine Prozedur, 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.) Die Prozedur gibt die betreffende Information in geeigneter Weise zurück (z.B. 0 = kein Projektor, 1 = Projektor, 2 = Orthogonalprojektor).

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 Beschrifung, 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.[1]

Aufgabe 9.9. Schreiben Sie zwei Prozeduren (egal was) und basteln Sie daraus Ihr eigenenes package.

Aufgabe 9.10. Extra-Aufgabe für Interessierte:

Maplets sind graphical user interfaces (GUIs, basierend auf dem Java [TM] Runtime Environment), mittels derer man Maple-Berechnungen in einem eigenen Fenster ablaufen lassen kann, mit Menüs, Steuerelementen etc. Damit kann man Applikationen entwerfen, die intern auf Maple-Code beruhen aber in einfacher Weise interaktiv zu verwenden sind, etwa für Demonstrationszwecke. Man kann auch stand-alone Applikationen generieren, die im *Maplet Application Viewer* angezeigt werden können.

Sehen Sie sich die mit Maple mitgelieferten Beispiele an und modifizieren Sie das eine oder andere davon, oder verpacken Sie eine der von Ihnen gelösten Aufgaben in ein Maplet. Es geht hier nur darum, dass man mal gesehen hat, wie das etwa funktioniert. Maplets sind nicht so mächtig und flexibel wie Java-Applets, aber wesentlich einfacher zu realisieren.

Hilfe: Programming … Maplets … Examples …

---

[1] 'Exakte Rangbestimmung' ist für float-Daten und in Gleitpunktarithmetik im Allgemeinen nicht möglich. Man denke z.B. an die Bestimmung der Schnittmenge zweier fast paralleler Geraden ('parallel' oder 'fast' parallel ist bei gerundeten Daten nicht immer entscheidbar).