Winfried Auzinger
Klaus Felsenstein
Dirk Praetorius

SS 2009
(14. Mai 2009)

Übungen zur Vorlesung Computermathematik

Serie 8

Die meisten der folgenden Übungsaufgaben sind dem Gebiet der Linearen Algebra zuzurechnen (`with(LinearAlgebra);`*)*

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.*

Ihre Codes sollen jeden sinnvollen Datentyp akzeptieren (symbolisch, ganzzahlig, floats) – sie werden aber nicht für beliebige Daten funktionieren. Es gibt aber auch Ausnahmen: Z.B. funktioniert die komplette Bestimmung der Singulärwertzerlegung (`?SingularValues`*, siehe unten) im Allgemeinen nur für float-Daten.* [1]

Aufgabe 8.1*. In der Maple - Online-Hilfe findet man unter **index[packages]** eine Übersicht über die in der Library vorhandenen Packages. Man gebe einen kurzen Überblick (aus individueller Sicht), mit dem einen oder anderen Beispiel. Von allgemeinem Interesse sind z.B. die Packages `CodeGeneration`, `CurveFitting`, `geometry`, `Statistics`, `Student`, `VectorCalculus` – aber es gibt noch viel mehr Interessantes.

Aufgabe 8.2*. Unter dem *dyadischen Produkt* zweier Spaltenvektoren $x \in \mathbb{R}^m$ und $y \in \mathbb{R}^n$ versteht man die $m \times n$ - Matrix $x \cdot y^\top$ (Spaltenvektor mal Zeilenvektor im Sinn des Matrixproduktes). Derartige Matrizen kann man also z.B. in einer zweispaltigen Matrix `xy` in komprimierter Form speichern, die aus den 2 Spalten `x` und `y` besteht. Das nennen wir unser (komprimiertes) ‘dyadisches Format’.

Überlegen Sie, dass das Matrixprodukt zweier kompatibel dimensionierter dyadischer Produkte wieder als dyadisches Produkt dargestellt werden kann (man beachte die Assoziativität der Matrixmultiplikation). Schreiben Sie eine Maple-Prozedur

```
muldyad(xy::Matrix,uv::Matrix),
```

die das entsprechende Matrixprodukt zweier derartiger Objekte `xy` und `uv` in effizienter Weise berechnet und wieder im dyadischen Format zurückgibt. Eine zweite Prozedur

```
dyadtimesvec(xy::Matrix,v::Vector)
```

soll die durch `xy` repräsentierte Matrix mit dem Vektor `v` multiplizieren.

Man beachte: Es ist nicht erforderlich, die Matrix $x{\cdot}y^\top$ explizit zu erzeugen. Jeweils Dimensionsüberprüfung vorsehen, ggf. mit `error` abbrechen.

[1] An dieser Stelle sei auch an die Bemerkung in der Vorlesung über die Jordan-Zerlegung einer Matrix erinnert. Dabei ist es genau umgekehrt: Falls die Matrix als float-Objekt gegeben ist, also möglicherweise bereits durch Rundung verfälscht sein könnte, ist die Bestimmung der Jordan-Zerlegung im Allgemeinen eine *sinnlose Fragestellung*, weil die Frage nach der Diagonalisierbarkeit unter der Annahme verfälschter Daten nicht generell entscheidbar ist. Sie wird daher von Maple verweigert, vgl. `?JordanForm`. Vgl. auch MATLAB: Dort gibt es überhaupt keinen Befehl zur Berechnung der Jordan-Zerlegung.

Aufgabe 8.3*. Sei $A \in \mathbb{R}^{n\times n}$ invertierbar und $U, V \in \mathbb{R}^{n\times k}$ gegeben. Dann gilt die *Sherman-Morrison-Woodbury*-Formel

$$(A+UV^\top)^{-1} = A^{-1} - A^{-1}U(I+V^\top A^{-1}U)^{-1}V^\top A^{-1}$$

in dem Sinn, dass $A+UV^\top \in \mathbb{R}^{n\times n}$ genau dann invertierbar ist wenn $I+V^\top A^{-1}U \in \mathbb{R}^{k\times k}$ invertierbar ist, und in diesem Fall gilt die behauptete Identität.

Dies kann man z.B. dazu verwenden, um für bereits berechnetes A^{-1} die Inverse $(A+UV^\top)^{-1}$ zu bestimmen. Für $k < n$ benötigt man dafür nur mehr zusätzlich die ´kleinere' Inverse $(I+V^\top A^{-1}U)^{-1} \in \mathbb{R}^{k\times k}$, abgesehen von den zusätzlich erforderlichen Matrix-Multiplikationen.

Man implementiere diese Art der Berechnung von $(A+UV^\top)^{-1}$ in einer Maple-Prozedur

`WInverse(A,U,V)`

mit automatischer Dimensionsüberprüfung und Abfrage auf Singularität (`error ...`). Für U und V sollen auch Spaltenvektoren zulässig sein (dieser Spezialfall $k = 1$ soll automatisch erkannt werden!), und dieser Fall soll extra behandelt werden.

Aufgabe 8.4*. Jede symmetrische, positiv definite Matrix $A \in \mathbb{R}^{2\times 2}$ 'erzeugt' eine Ellipse E im $\mathbb{R}^2$ gemäß ihrer quadratischen Form, z.B. durch

$$E = \{x \in \mathbb{R}^2 \text{ mit } x^\top A\, x = 1\}$$

Zeichnen Sie für gegebenes A diese Ellipse (`? plot`, mit `scaling=constrained`). Zeichnen Sie auch die beiden Hauptachsen ein.

Hinweis: Jeder Vektor $x \in \mathbb{R}^2$ lässt sich schreiben als $x = \alpha u + \beta v$, wobei u, v die (auf Länge 1 normierten) Eigenvektoren von A sind (`?Eigenvectors`); u und v sind orthogonal. Dann ist $Ax = \lambda\alpha u + \mu\beta v$, wobei $Au = \lambda u$, $Av = \mu v$ ($\lambda, \mu > 0$ sind die Eigenwerte von A; `?Eigenvalues`). Damit wird aus der Gleichung $x^\top A\, x = 1$ eine quadratische Beziehung zwischen den Koeffizienten α und β (3 Terme mit α^2, $\alpha\beta$ und β^2). Man kann jetzt z.B. nach α oder auch nach β auflösen (quadratische Gleichung, je zwei beschränkte reelle Lösungszweige – Achtung auf Wurzel!) und darauf basierend den plot erzeugen. Genaue Details sind zu überlegen.

Die Hauptachsen von E entsprechen den Bildern der beiden Eigenvektoren von A (geeignet normiert). Was bedeuten die beiden Eigenwerte für das Aussehen der Ellipse?

Anmerkung: Eine 'elegantere' Lösung wäre, eine Parameterdarstellung für die Ellipse zu bestimmen (basierend auf 'Hauptachsentransformation'; nicht verlangt).

Eine *brute force* Lösung für diese Aufgabe besteht darin, einfach nur `plots[implicitplot]` zu verwenden. Probieren Sie's aus.

Aufgabe 8.5. Eine reelle *Householder-Matrix* ist eine orthogonale Matrix $H \in \mathbb{R}^{n\times n}$ der Gestalt

$$H = I - 2\,\frac{h\cdot h^\top}{h^\top\cdot h},$$

wobei $h \in \mathbb{R}^n$ ein vorgegebener Spaltenvektor $\neq 0$ ist.

Man schreibe

- eine Maple-Prozedur `housemul(h,x)`, die die Matrix-Vektor-Multiplikation $H\cdot x$ ausführt (für gegebenes $x \in \mathbb{R}^n$), ohne die Matrix H explizit aufzustellen,
- eine Maple-Prozedur `DrHouse(h::list)`, die zwei Householder-Matrizen H_1 und H_2 (festgelegt durch die Vektoren `h[1]` und `h[2]`) miteinander multipliziert und das Matrixprodukt $H_1\cdot H_2$ als `Matrix` zurückliefert.

- Im Fall $n = 2$ und $h \in \mathbb{R}^2$ studiere man die Geometrie der Abbildung $x \mapsto H \cdot x$ mittels grafischer Darstellung (z.B. für konkrete Werte von h). Was für eine Art von Abbildung wird durch die Matrix H dargestellt? Welche Rolle spielt der Vektor h?

Hinweis: H ist nicht nur orthogonal ($H^{-1} = H^\top$) sondern auch involutorisch, $H^{-1} = H$. Daher gilt immer $H^2 x = x$.

Aufgabe 8.6. Jede Matrix[2] $A \in \mathbb{R}^{n \times n}$ besitzt eine sogenannte *Singulärwertzerlegung*

$$A = V \Sigma U^\top = \sum_{i=1}^{n} \sigma_i \, v_i \, u_i^\top$$

mit $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{n \times n}$ jeweils orthogonal, $\Sigma \in \mathbb{R}^{n \times n}$ eine Diagonalmatrix mit Diagonalelementen σ_i, den sogenannten *Singulärwerten* von A, siehe `?SingularValues`. Für Matrizen mit floating-point-Einträgen kann man die komplette Singulärwertzerlegung mittels `?SingularValues` numerisch berechnen; die Singulärwerte werden dabei in Σ absteigend angeordnet.

Eine Technik zur verlustbehafteten Matrixkompression (Datenkompression) besteht nun darin, nur eine gewisse Anzahl von Singulärwerten (die größeren) beizubehalten und den Rest (die kleineren) durch 0 zu ersetzen. Damit ergibt sich eine Approximation $\tilde{A}$ für A der Gestalt

$$\tilde{A} = \tilde{V} \tilde{\Sigma} \tilde{U}^\top .$$

Man schreibe eine Maple-Prozedur

`svdcompress(A::Matrix,tol::And(float,positive))`,

die zu gegebenem A die Singulärwertzerlegung berechnet und alle Singulärwerte $\sigma_i <$ `tol` durch Null ersetzt. Es verbleiben k Singulärwerte σ_i mit $\sigma_i \geq$ `tol`. Die so komprimierte Matrix[3] $\tilde{A}$ wird nun durch zwei *rechteckige* Matrizen $\tilde{U}, \tilde{V}$ (Teilmatrizen von U, V) und durch den Vektor $\tilde{\sigma}$ der verbliebenen Singulärwerte repräsentiert – diese drei Objekte sind von `svdcompress` in einer Liste zurückzuliefern. Falls nur wenige Singulärwerte verblieben sind (Anzahl $\ll$ Dimension von A), bedeutet dies eine speichereffiziente Datenkompression. Mit der komprimierten Darstellung $\tilde{A} \sim (\tilde{U}, \tilde{\sigma}, \tilde{V})$ kann man nun z.B. die Matrix-Vektor-Multiplikation $\tilde{A}x$ ausführen, ohne $\tilde{A}$ explizit zu bilden. Schreiben Sie auch eine Maple-Prozedur

`USVMatrixVectorMultiply(USV::list,x::Vector)`,

die diese Multiplikation implementiert.

Man experimentiere z.B. mit zufällig generierten Matrizen (`?RandomMatrix`); zur Überprüfung der Approximationsqualität kann man z.B. $\tilde{A}$ wieder explizit rekonstruieren (am besten mit einer dafür geeigneten Prozedur) und mit A vergleichen, siehe dazu auch `?MatrixNorm`.

Aufgabe 8.7. In MATLAB gibt es den Befehl `spy` zur grafischen Darstellung der Besetzungsstruktur schwach besetzter Matrizen (etwa so: weißes Kasterl = Nullelement, schwarzes Kasterl = ungleich Null).

Schreiben Sie eine Maple-Prozedur

`spy(A::Matrix,...)`

die für Objekte vom Typ `Matrix` etwas Ähnliches macht. Design und Werkzeuge, allfällige optionale Parameter etc. seien Ihrer persönlichen Kreativität überlassen.

[2] Eine ähnliche Darstellung gilt auch für beliebige rechteckige Matrizen.

[3] Beachte: $\tilde{A} = \sum_{i=1}^{k} \sigma_i \, v_i \, u_i^\top$

Aufgabe 8.8. Man schreibe eine Maple-Prozedur

`Hessqform(f::procedure,x::Vector[column],u::Vector[column])`,

die zu einer gegebenen (zweimal stetig differenzierbaren) reellwertigen Funktion $y = f(x_1, \ldots, x_n)$ intern die Hesse-Matrix $H = \left(h_{ij}\right) = \left(\frac{\partial^2 f}{\partial x_i\, \partial x_j}\right)$ an der gegebenen Stelle $x \in \mathbb{R}^n$ berechnet und den Wert der quadratischen Form $u^\top H u$ zurückliefert.

Aufgabe 8.9. Man denke sich eine Matrix $A \in \mathbb{R}^{n\times n}$ in 4 Blöcke unterteilt,

$$A = \begin{pmatrix} U & V \\ Y & X \end{pmatrix}$$

und zwar so dass $U \in \mathbb{R}^{k\times k}$ und $X \in \mathbb{R}^{(n-k)\times(n-k)}$ quadratisch ($1 \leq k < n$). Angenommen, A ist regulär. Dann kann man die Inversion von A auf die Inversion einer $k\times k$ - Matrix und einer $(n-k) \times (n-k)$ - Matrix (plus einige weitere Additionen und Multiplikationen) zurückführen. Dazu macht man für A^{-1} einen analogen Block-Ansatz und berechnet die zu bestimmenden Blöcke aus der Forderung $A \cdot A^{-1} = I$ (Schur-Komplement-Methode, siehe `http://de.wikipedia.org/wiki/Schur-Komplement`). Die dabei auftretenden kleinen Inversen der Dimension $k\times k$ bzw. $(n-k) \times (n-k)$ müssen dabei invertierbar sein (was nicht immer der Fall ist) – sonst funktioniert das nicht in der beschriebenen Weise.

Schreiben Sie eine Maple-Prozedur

`SchurInv(A::Matrix,k::posint)`,

die diese Methode zur Matrixinversion implementiert.

Aufgabe 8.10. Im Paket `LinearAlgebra` gibt es eine Funktion, die das *charakteristische Polynom* $p(z) := \det(zI - A)$ einer quadratischen Matrix $A \in \mathbb{R}^{n\times n}$ erzeugt (I = Einheitsmatrix). Verwenden Sie dies, um anhand einiger Beispiele den *Satz von Cayley-Hamilton* experimentell zu 'verifizieren': $p(A) = 0$ (Nullmatrix). (Was mit $p(A)$ gemeint ist, sollte klar sein – ein 'Matrixpolynom'.)

Schreiben Sie auch eine Prozedur `CHsolve(A::Matrix,b::Vector)`, die die Lösung $x = A^{-1}b$ eines linearen Gleichungssystems $A\,x = b$ unter der Ausnutzung der Identität $p(A) = 0$ berechnet und zurückgibt (A wird als invertierbar angenommen). `CHsolve` soll, unter Zuhilfenahme des charakteristischen Polynoms von A, intern nur Matrix-Vektor-Multiplikationen $y \mapsto A\,y$ und Vektoradditionen ausführen.

Anmerkung: Dies hat eher 'theoretische' Bedeutung; in der hier beschriebenen Form handelt es sich nicht um einen praktisch sinnvoll einsetzbaren Algorithmus.