

Computergestützte Mathematik zur Linearen Algebra – 10. Übungsblatt

Aufgabe 39:

Schreiben Sie eine Funktion `C=gramschmidt(B)`, die aus einer Basis B eine Orthogonalbasis C erzeugt. Hierbei ist die Eingabe B eine Matrix, deren Spalten den Basisvektoren von B entsprechen. Die Ausgabe C ist ebenfalls eine Matrix, deren Spalten die Elementen von C repräsentieren.

Hinweis: Gram-Schmidt-Orthogonalisierung. Falls Sie dieses Verfahren nicht aus der Linearen Algebra kennen, recherchieren Sie das Verfahren.

Aufgabe 40:

Versuchen Sie experimentell die Kondition der beiden folgenden Probleme abzuschätzen

- (a) Mit dem MATLAB Befehl `eig(A)` können Sie die Eigenwerte einer Matrix A berechnen.

Untersuchen Sie, wie sich kleine Störungen in den Einträgen von A auf die berechneten Eigenwerte auswirken. Sie könnten dazu etwa eine 5×5 mit einzigem Eigenwert 1 verwenden und zu den Einträgen dieser Matrix zufällige kleine Zahlen addieren.

Zeichnen Sie für verschieden kleine Störungen die Eigenwerte der gestörten Matrix in der komplexen Ebene.

- (b) Mit dem MATLAB- Befehl `A\b` können Sie ein lineares Gleichungssystem $Ax = b$ mit der Matrix A und rechter Seite b lösen.

Untersuchen Sie wie sich kleine Störungen in den Einträgen der Matrix A auf die Lösung x auswirken. Berechnen sie dazu für viele verschiedene rechte Seiten b , mit $\|b\| = 1$ und eine Matrix A , die Auswirkung von Störungen in A auf den Lösungsvektor x .

Aufgabe 41:

- (a) Laden Sie die Datei `hello.mat` von der Web-Seite herunter und lesen Sie die Matrix A mittels `load hello` nach Matlab ein. Schauen Sie sich die Besetzungsstruktur der Matrix mit `spy(A)` und mit `image(256*A); colormap gray` an
- (b) Berechnen Sie mit `svd` die Singulärwerte von A und plotten Sie diese mit `plot` und `semilogy`. Wie groß ist der mathematisch exakte Rang von A und wie erkennt man das an den numerischen Werten?
- (c) Berechnen Sie für $k = 1, \dots, \text{Rang}(A)$ die Bestapproximationen A_k an A vom Rang k , die durch

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

gegeben sind. Stellen sie die Bestapproximationen graphisch dar.

Aufgabe 42:

Für einen Parameter $k \in \mathbb{N}, k \geq 1$ ist die stückweise definierte reelle Funktion f_k definiert durch

$$f_k(x) = \begin{cases} -(-1)^k \sqrt{-x-1} & x < -1 \\ \sin(k\pi x), & \text{falls } -1 \leq x \leq 1 \\ (-1)^k \sqrt{x-1}, & \text{falls } x > 1. \end{cases}$$

- (a) Schreiben Sie eine Funktion `function y = aufgabe42a(x,k)`. Die Funktion soll zu gegebenem Eingabevektor $x \in \mathbb{R}^N$ und Parameter k die Werte von $f_k(x)$ in einen Vektor $y \in \mathbb{R}^N$ schreiben. Überprüfen Sie ob $k \geq 1$ gilt und geben Sie eine Fehlermeldung aus, wenn dies nicht der Fall ist.

- (b) Schreiben Sie ein MATLAB-Skript `aufgabe42b.m`, das die Funktion f_k in einem Fenster für $k = 2, 3$ auf dem Intervall $[-2, 2]$ graphisch darstellt. Zeichnen Sie die Kurve für $k = 2$ mit einer roten durchgezogenen Linie und die Kurve für $k = 3$ mit einer schwarzen gestrichelten Linie.

Ergänzen Sie ihre Graphik um eine Legende, die erklärt welche der Kurven zu $k = 2$ und welche zu $k = 3$ gehört und plazieren Sie die Legende oberhalb der Koordinaten-Achsen.

Aufgabe 43:

Inspizieren Sie den MATLAB-Funktion in der Datei `aufgabe43.m`. Die Funktion

```
function x = aufgabe43(A,b)
```

soll die Lösung des linearen Gleichungssystemes $Ax = b$ berechnen. Die Funktion soll einen Fehler melden, wenn die Matrix A nicht quadratisch ist.

Leider haben sich 10 syntaktische und inhaltliche Fehler in die Funktion eingeschlichen. Finden und korrigieren Sie die Fehler, indem Sie die entsprechenden fehlerbehafteten Zeilen auskommentieren und durch korrigierte Anweisungen ersetzen. Ergänzen Sie den Code zusätzlich um Kommentarzeilen, die ihre Korrekturen beschreiben.

In den letzten drei Zeilen sind zusätzliche Tests definiert, die Zwischenergebnisse überprüfen und eine Ausgabe erzeugen, wenn diese nicht erwartungsgemäß ausfallen. Diese Zeilen enthalten selbst *keine* Fehler und dienen lediglich zu Ihrer Orientierung, da sie Hinweise geben in welchem Abschnitt der Funktion noch Fehler stecken!

```
funktion x = aufgabe43(A,b)

[m,n] = size(A);

if m == n
    error('Matrix ist nicht quadratisch');
end

% Berechne pivotisierte LR-Zerlegung (P*A = L*R)
[P,L,R] = lu(A);

% Permutiere rechte Seite b
Pb = P*b;

% Loese Ly = Pb durch Vorwaertseinsetzen
y = zeros(n,1);
for i = 1:n
    y(i) = b(i);
    for j = 1:i-1
        y(i) = y(i) + L(i,j)*y(j);
    end
end

% Loese Rx = y durch Rueckwaertseinsetzen
x = zeros(n,1);
for i = n:-1:0
    x(i) = y(i);
    for j = i:n
        x(i) = x(i) - R(i,j)*x(j);
    end
    x(i) = x(i)/R(i);
end

% FOLGENDEN ZEILEN ENTHALTEN KEINE FEHLER, SIE WEISEN AUF FEHLER HIN!
if(norm(P*A-L*R,'fro') > 10^-8),fprintf('Fehler: P*A ~= L*R\n');end
if(norm(L*y-P*b) > 10^-8),fprintf('Fehler: L*y ~= P*b\n');end
if(norm(R*x-y) > 10^-8 || isnan(norm(R*x-y))),fprintf('Fehler: R*x ~= y\n');end
```