Computerphysik WS 2016/2017

Prof. Dr. Petra Imhof, FU Berlin

Übungsblatt 3: Lineare Gleichungssysteme

4. November 2016

Hinweis: ~~durchgestrichener Text bedeutet eine Korrektur durch Entfernen gegenüber einer vorher online gestellten Version (und kann ignoriert werden)~~

Aufgabe 3.1: Matrixmultiplikation (3 Punkte)

a) (1 Punkt) Implementieren Sie eine Funktion welche das Matrixprodukt von zwei Matrizen beliebiger (kompatibler) Größe berechnet.

b) (1 Punkt) Bestimmen Sie die Anzahl der Punktoperationen (Multiplikationen, Divisionen, Quadratwurzel-Berechnungen) welche für das Matrixprodukt benötigt werden, um die Komplexität Ihrer Funktion abzuschätzen.

c) (1 Punkt) Implementieren Sie eine Funktion, welche die Kondition einer beliebigen Matrix bezüglich der Frobeniusnorm berechnet.

Aufgabe 3.2: Gauss-Eliminierung (12 Punkte)

Gegeben seien die linearen Gleichungssysteme $\mathbf{Kx} = \mathbf{b}$ und $\mathbf{Mx} = \mathbf{b}$ mit

$$\mathbf{K} = \begin{pmatrix} -3. & -2. & -10. & -9. & 8. & -8. \\ 0. & 4. & 5. & -10. & -4. & -7. \\ 4. & -6. & -1. & 5. & -7. & -8. \\ -9. & 7. & -8. & 8. & 2. & 7. \\ 0. & 6. & -10. & -8. & 2. & 5. \\ -3. & -6. & 9. & -8. & -5. & 5. \end{pmatrix} \text{ und } \mathbf{M} = \begin{pmatrix} 1. & -3. & -6. & 8. & 0. & 5. \\ 6. & -4. & 1. & 6. & 0. & -1. \\ 1. & 8. & 9. & -1. & -3. & -6. \\ -6. & -8. & -1. & -9. & 1. & -3. \\ 8. & -8. & -2. & -2. & 3. & 0. \\ -2. & 3. & 8. & -9. & 9. & -1. \end{pmatrix}$$

$$\text{und } \mathbf{b} = \begin{pmatrix} 1. \\ -2. \\ -3. \\ 4. \\ 0. \\ 5. \end{pmatrix}.$$

a) (3 Punkte) Schreiben sie eine Funktion, die den Gauß-Algorithmus implementiert. Eingabeparameter sollen eine (NxN) `numpy.array` (z.B. $\mathbf{M}$) und einen (Nx1) `numpy.array` (z.B. $\mathbf{b}$) sein. Die Funktion soll den Lösungsvektor x als `numpy.array` ~~, die Matrix~~ $\cancel{\mathbf{G}}$ und die Anzahl der Punktoperationen (Multiplikationen, Divisionen, Quadratwurzel-Berechnungen) zurück geben. $\cancel{\mathbf{G}}$ ~~erzeugt nach z.B.~~ $\cancel{\mathbf{G} \cdot \mathbf{M} = \mathbf{R}}$ ~~eine rechte obere Dreiecksmatrix~~ $\cancel{\mathbf{R}}$~~.~~

b) (2 Punkte) Erweitern Sie Ihre Implementierung des Gauß-Algorithmus so, dass nach Wunsch, Spaltenpivotisierung durchgeführt werden kann.

c) (2 Punkte) Lösen Sie die Gleichungssysteme mit ihrer selbst geschriebenen Funktion nach dem Gauß-Algorithmus einmal mit und einmal ohne Spaltenpivotisierung.

d) (1 Punkt) Bestimmen Sie mit der in Aufgabe 3.1 implementierten Funktion die Kondition der Matrizen $\mathbf{K}$ und $\mathbf{M}$ bezüglich der Quadratsummennorm (Frobeniusnorm).

e) (2 Punkte) Versehen Sie die Matrizen $\mathbf{K}$ und $\mathbf{M}$ mit einem Fehler von $\delta m_{ij} = 0.001$ bzw. $\delta k_{ij} = 0.001$, indem Sie diesen zu jedem Element m_{ij} bzw. k_{ij} hinzu addieren. Lösen Sie die Gleichungssysteme nun für die fehlerbehafteten Matrizen erneut, wieder je einmal mit und einmal ohne Spaltenpivotisierung.

f) (2 Punkte) Berechnen Sie den Abstand des jeweiligen Lösungsvektors des fehlerbehafteten Gleichungssystems, $\tilde{\mathbf{x}}$, mit dem Lösungsvektor des ungestörten Systems, $\mathbf{x}$ als 2-Norm (euklidische Norm). Diskutieren Sie Ihre Resultate.

Aufgabe 3.3: Cholesky-Zerlegung (5 Punkte)

Gegeben seien die linearen Gleichungssysteme $\mathbf{Ax} = \mathbf{c}$ und $\mathbf{Bx} = \mathbf{c}$ mit

$$\mathbf{A} = \begin{pmatrix} 9.0 & 3.0 & 1.0 & 4.0 \\ 3.0 & 5.0 & 2.0 & 4.0 \\ 1.0 & 2.0 & 5.0 & 6.0 \\ 4.0 & 4.0 & 6.0 & 10.0 \end{pmatrix} \text{ und } \mathbf{B} = \begin{pmatrix} 1.0 & 3.0 & 1.0 & 4.0 \\ 3.0 & 2.0 & 2.0 & 5.0 \\ 1.0 & 2.0 & 1.0 & 6.0 \\ 4.0 & 5.0 & 6.0 & 2.0 \end{pmatrix} \text{ und } \mathbf{c} = \begin{pmatrix} 34.0 \\ 35.0 \\ 44.0 \\ 70.0 \end{pmatrix}.$$

a) (2 Punkte) Schreiben Sie eine Funktion, die ein lineares Gleichungssystem mittels Cholesky-Zerlegung löst. Als Eingabeparameter soll die Funktion einen (NxN) `numpy.array` (z.B. $\mathbf{A}$) und einen (Nx1) `numpy.array` (z.B. $\mathbf{c}$) nehmen und den Lösungsvektor x als `numpy.array` sowie die Anzahl der Punktoperationen (Multiplikationen, Divisionen, Quadratwurzel-Berechnungen) zurück geben.

b) (1 Punkt) Prüfen Sie, ob $\mathbf{A}$ und $\mathbf{B}$ positiv definit sind.

c) (1 Punkt) Lösen Sie das/die geeigneten Gleichungssysteme mit ihrer selbst geschriebenen Funktion.

d) (1 Punkt) Lösen Sie das/die geeigneten Gleichungssysteme mit dem Gauss-Algorithmus und vergleichen sie die Anzahl der Punktoperationen für die Lösung über Gauss-Algorithmus und über Cholesky-Zerlegung (*Beachten Sie auch die Punktoperationen in eventuellen Matrix-multiplikationen*).

Hinweis: Die Inverse einer Matrix können Sie mit `numpy.linalg.inv` *bestimmen. Zur Kontrolle Ihrer Ergebnisse können Sie* `numpy.linalg.solve` , `numpy.linalg.norm`, `numpy.linalg.cholesky` *verwenden.*