

Algorithmik kontinuierlicher Systeme Aufgabenblatt 3 — Gleichungssysteme am Computer lösen

Allgemeines:

- Die Abgabe der Programmieraufgaben erfolgt über das Exercise Submission Tool:
<https://est.cs.fau.de>
- Sie können während der Bearbeitungszeit Ihre Abgaben im EST beliebig oft aktualisieren. Nur die aktuellste Abgabe, die in der Bearbeitungszeit hochgeladen wurde, wird gewertet.
- Auf der Übungswebsite finden Sie zu jedem Übungsblatt eine Vorlage, sowie zu jeder Aufgabe eine Datei `*_test.py` mit der Sie ihre Lösungen jederzeit selbst überprüfen können.
- Damit Sie auf eine Teilaufgabe Punkte bekommen, muss sie mit Python 3.5 im CIP Pool funktionieren. Das bestehen der mitgelieferten Tests ist notwendig, aber nicht hinreichend dafür, dass Sie Punkte bekommen.

Im letzten Arbeitsblatt haben Sie lineare Gleichungssysteme mit Stift und Papier gelöst. Für größere Probleme ist das mühsam und fehleranfällig. In diesem Blatt werden Sie diese Operationen mit Python automatisieren. Wie im vorherigen Blatt werden alle Vektoren und Matrizen als zweidimensionale NumPy Arrays repräsentiert.

Wichtiger Hinweis: Sie müssen die Verfahren in diesem Übungsblatt selbst implementieren und dürfen nicht einfach existierende Funktionen wie `numpy.linalg.qr` verwenden.

Aufgabe 1 — LR-Zerlegung (9 Punkte)

`lu.py`

Die LR-Zerlegung ist ein Standardverfahren zur Lösung linearer Gleichungssysteme. Es entspricht einer Gauss-Elimination, bei der alle Zwischenschritte in Matrizen gespeichert werden. Im Englischen wird die LR-Zerlegung als LU-decomposition bezeichnet. Dieses Aufgabenblatt folgt der Konvention, dass Quellcode durchgehend in Englisch geschrieben wird und verwendet daher je nach Kontext LR oder LU, meint aber in beiden Fällen das gleiche.

a) Zeilen vertauschen Schreiben Sie eine Funktion `swap_rows`, die eine Matrix und zwei Zeilenindizes übergeben bekommt und die die angegebenen Zeilen dieser Matrix vertauscht.

b) Zeilen subtrahieren Schreiben Sie eine Funktion `subtract_scaled`, die eine Matrix, zwei Zeilenindizes und einen Skalierungsfaktor übergeben bekommt und die von der ersten Zeile das Vielfache der zweiten Zeile subtrahiert.

c) Pivot Element Schreiben Sie eine Funktion `pivot_index`, die eine Matrix und einen Spaltenindex übergeben bekommt und den Zeilenindex des betragsmäßig größten Element dieser Spalte zurück gibt. Elemente oberhalb der Diagonale sollen dabei ignoriert werden, so dass der zurückgegebene Zeilenindex immer größer oder gleich dem gegebenen Spaltenindex ist.

d) LR-Zerlegung Schreiben Sie eine Funktion `lu_decompose`, die eine quadratische, nichtsinguläre Matrix A übergeben bekommt und ein Tupel aus drei Matrizen P , L und R zurückgibt, so dass gilt $A = PLR$, wobei P eine Permutationsmatrix, L eine untere Dreiecksmatrix mit Einsen auf der Diagonale und R eine obere Dreiecksmatrix ist. Die Funktionen der drei vorherigen Aufgaben können Ihnen dabei helfen.

e) Vorwärtssubstitution Schreiben Sie eine Funktion `forward_substitute`, die eine untere Dreiecksmatrix L und einen Vektor b übergeben bekommt und das Gleichungssystem $Ly = b$ nach y löst. Nutzen Sie dabei die besondere Struktur der Matrix L .

f) Rückwärtssubstitution Schreiben Sie eine Funktion `backward_substitute`, die eine obere Dreiecksmatrix R und einen Vektor y übergeben bekommt und das Gleichungssystem $Rx = y$ nach x löst. Nutzen Sie dabei die besondere Struktur der Matrix R .

g) Gleichungssysteme lösen Schreiben Sie eine Funktion `linsolve`, die eine quadratische, nichtsinguläre Matrix A und beliebig viele Vektoren b_1, \dots, b_n übergeben bekommt, und ein Tupel aus Lösungen x_1, \dots, x_n zurückgibt, so dass für jede rechte Seite b_k gilt: $Ax_k = b_k$. Verwenden Sie dazu die Funktionen der vorherigen Teilaufgaben.

Aufgabe 2 — QR-Zerlegung (6 Punkte)

`qr.py`

In dieser Aufgabe sollen Sie die QR-Zerlegung einer Matrix mit Hilfe von Givens-Rotationen implementieren.

a) Givens-Rotation Schreiben Sie eine Funktion `givens_rotation`, die für eine gegebene Matrix A , sowie einem Zeilenindex i und einem Spaltenindex j , eine Givens-Rotationsmatrix J zurückgibt, so dass das Produkt aus J und A im Eintrag i, j den Wert 0 hat.

b) QR-Zerlegung Schreiben Sie eine Funktion `qr_decompose`, die für eine gegebene Matrix A ein Tupel aus den Matrizen Q und R zurückgibt, so dass gilt $A = QR$, wobei Q eine unitäre Matrix und R eine obere Dreiecksmatrix ist. Im Gegensatz zur LR-Zerlegung soll dieses Verfahren auch für nicht-quadratische Matrizen mit $m > n$ funktionieren.

c) Rückwärtssubstitution Wenn Sie die Funktion `backward_substitute` nicht schon für die LR-Zerlegung implementiert haben, holen Sie das jetzt nach. Ansonsten können Sie ihre vorherige Lösung hier wiederverwenden.

d) Gleichungssysteme lösen Schreiben Sie wie in Aufgabe 1 eine Funktion `linsolve`, die eine quadratische, nichtsinguläre Matrix A und beliebig viele Vektoren b_1, \dots, b_n übergeben bekommt, und ein Tupel aus Lösungen x_1, \dots, x_n zurückgibt, so dass für jede rechte Seite b_k gilt: $Ax_k = b_k$. Verwenden Sie dazu die zuvor implementierte QR-Zerlegung und Rückwärtssubstitution.