MATHEMATISCHES INSTITUT
PROF. DR. CHRISTIANE HELZEL
PAWEL BUCHMÜLLER

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

7. JANUAR 2016

Computergestützte Mathematik zur Linearen Algebra – 9. Übungsblatt

Aufgabe 33: (LU-Det)

Befehle: `prod, diag`

Die LU Zerlegung

$$LU = PA$$

kann auch zur Berechnung der Determinanten von A genutzt werden.

$$det(L)det(U) = det(P)det(A).$$

Da L eine Dreiecksmatrix mit Einsen auf der diagonalen ist, gilt $det(L) = 1$. Da U ebenfalls eine Dreiecksmatrix ist, gilt $det(U) = u_{11}u_{22}\dots u_{nn}$. Da P eine Permutationsmatrix ist, gilt $det(P) = +1$ falls die Anzahl an Permutationen gerade ist und $det(P) = -1$ falls sie ungerade ist. D.h. $det(A) = \pm u_{11}u_{22}\dots u_{nn}$.

- Ändern Sie die Funktion `lutx` so dass sie folgendes liefert

```
function [L,U,p,sig] = lutx(A)
% LUTX  Triangular factorization
% [L,U,p,sig] = lutx(A) produces a unit lower triangular matrix L,
% an upper triangular matrix U, and a permutation vector p, and a scalar sig,
% so that L*U = A(p,:) and sig = +1 or -1 if p is an even or odd permutation.
```

- Schreiben Sie eine Funktion `MyDetLU(A)`, die mithilfe der obigen Funktion die Determinante von A bestimmt.

Aufgabe 34: (LU-Inv)

Das Inverse einer Matrix A kann definiert werde als Matrix X, deren Spalten x_j die Gleichungen

$$Ax_j = e_j$$

lösen, wobei e_j der $j-$te Einheitsvektor ist.

- Schreiben Sie eine Funktion `MyInv(A)` zu Bestimmung der Inversen von A ausgehen von der Funktion `bslashtx`. Dabei soll `lutx` nur ein Mal verwendet werden. Der \ Operator sowie die Matlab eigene Funktion `inv` dürfen nicht verwendet werden.
- Testen Sie ihre Funktion an zufälligen Matrizen.

Aufgabe 35: (Variation der Ausgabe)

Befehle: `nargout`

Wird der Matlab Befehl `[L,U] = lu(A)` mit zwei Rückgabeargumenten aufgerufen, so ist `L` keine Dreiecksmatrix, sondern eine permutierte Dreiecksmatrix. Mit dieser permutierte Matrix gilt $LU = A$.

- Ändern Sie die Funktion `lutx(A)`, so dass Sie für den Aufruf `[L,U] = lutx(A)` die gleiche Ausgabe erhalten.
- Testen Sie die Änderungen an mehreren Matrizen.

Aufgabe 36: (Pivotisierung)

Befehle: `vpa, digits`

Wir wollen das Beispiel aus der Vorlesung (warum Pivotisierung wichtig ist) nachstellen. Da Matlab jedoch standardmäßig in double precision rechnet müssen wir dazu nach jeder Rechenoperation die gewünschte Arithmetik mit `vpa` 'erzwingen'. D.h. aus `a+b*c` wird `vpa(a+vpa(b*c))`. Mit `digits(n)` kann man die gewünschte Arithmetik festlegen.

- Schreiben Sie `lutx` und `bslashtx` um, so dass Sie Kontrolle über die Arithmetik erhalten. Speichern Sie die Funktionen als `lutxP` und `bslashtxP`.
- Entfernen sie nun die Pivotisierung aus `lutxP` und speichern Sie die Funktion als `lutxNP`. Ändern Sie `bslashtxP` so dass `lutxNP` verwendet wird und speichern sied ie Funktion als `bslashtxNP` .
- Lösen Sie nun das Gleichungssystem

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2.099 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 3.901 \\ 6 \end{pmatrix}$$

  mit `bslashtxP`, `bslashtxNP`. Setzen Sie zuvor `digits(n)` mit $n = 6, 5, 4, 3$.

Besprechung in den Übungen am 11. Januar 2016 - 14. Januar 2016