Serie 12

Die Kernaufgaben dieser Serie, die Sie mit Priorität bearbeiten sollten, sind die Aufgaben 2, 3 und 4. Die Aufgaben 1, 5 und 6 sind natürlich ebenfalls wichtig und sollten unbedingt zur Prüfungsvorbereitung studiert werden.

Aufgabe 5 bezieht sich auf eine wichtige Anwendung der Methode der kleinsten Quadrate und deren Implementierung in MATLAB.

Aufgabe 6 demonstriert eine überraschende und sehr wichtige Anwendung von linearen Rekursionen zur Analyse von Beziehungsnetzen.

1. Wir haben gesehen, dass uns die Tatsache, dass selbst reelle Polynome komplexe Nullstellen haben können dazu zwingt, im Zusammenhang mit der Diagonalisierung von Matrizen in $\mathbb{C}$ zu rechnen. Das ist auch der tiefere Grund für die grosse Bedeutung komplexer Zahlen in Wissenschaft und Technik. Machen Sie also nicht Ihren Mathematikprofessor dafür verantwortlich, dass Sie sich mit komplexen Zahlen herumschlagen müssen, sondern die Weigerung mancher reeller Polynome, ausschliesslich reelle Nullstellen zu haben.

Eine sehr wichtige Familie komplexer Matrizen sind die *Fouriermatrizen* $\mathbf{F}_n = (f_{kl}^n)_{k,l=1}^n \in \mathbb{C}^{n,n}$, $n \in \mathbb{N}$, definiert durch

$$f_{kl}^n := \exp\left(\frac{2\pi\imath}{n}(k-1)(l-1)\right) , \quad 1 \leq k, l \leq n . \tag{1}$$

Dabei bezeichnet $\imath \in \mathbb{C}$ die imaginäre Einheit ($\imath^2 = -1$).

a) Zeigen Sie, dass für alle $n \in \mathbb{N}$

$$\mathbf{F}_n^H \mathbf{F}_n = n \cdot \mathbf{I} , \tag{2}$$

wobei das hochgestellte H das komplex konjugierte Transponierte einer Matrix bezeichnet, die in Abschnitt 5.3 der Vorlesungsunterlagen erklärt ist.

Hinweis: Aus der Analysis sollten die folgenden Rechenregeln für die komplexe Exponentialfunktion bekannt sein, siehe Kapitel 1 im Analysis-Skript:

$$\begin{aligned} \exp(2\pi\imath k) &= 1 \quad \text{für jedes } k \in \mathbb{Z} , \\ \exp(kz) &= (\exp(z))^k \quad \text{für alle } k \in \mathbb{Z},\ z \in \mathbb{C} , \\ \overline{\exp(z)} &= \exp(\overline{z}) \quad \text{für alle } z \in \mathbb{C} . \end{aligned} \tag{3}$$

Erinnern Sie sich ausserdem an die geometrische Summenformel

$$\sum_{k=0}^{n-1} q^k = \frac{1-q^n}{1-q} \quad \text{für alle } q \in \mathbb{C} \setminus \{1\} , \quad n \in \mathbb{N} . \tag{4}$$

Bitte wenden!

b) Mit welcher komplexen Zahl muss man $\mathbf{F}_n$ multiplizieren, um eine *unitäre* Matrix zu erhalten?

c) Es bezeichne $\mathbf{e}^k \in \mathbb{R}^n$ den k. Einheitsvektor. Die *zyklische Permutationsmatrix* $\mathbf{P}_n \in \mathbb{R}^{n,n}$ is definiert durch

$$\mathbf{P}_n = [\mathbf{e}^2, \mathbf{e}^3, \ldots, \mathbf{e}^{n-1}, \mathbf{e}^n, \mathbf{e}^1] \; .$$

Zeigen Sie, dass diese Matrix durch die Fouriermatrix diagonalisiert wird, dass es also eine Diagonalmatrix $\mathbf{D} \in \mathbb{C}^{n,n}$ so gibt, dass

$$\mathbf{P}_n \mathbf{F}_n = \mathbf{F}_n \mathbf{D} \; . \tag{5}$$

Bestimmen Sie auch die Diagonaleinträge von $\mathbf{D}$.

Hinweis: Erinnern Sie sich daran, dass $\mathbf{AS} = \mathbf{SD}$, $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, genau dann wenn die Spalten $\mathbf{s}^l$, $l = 1, \ldots, n$, von $\mathbf{S}$ erfüllen

$$\mathbf{A}\mathbf{s}^l = \lambda_l \mathbf{s}^l \; , \quad l = 1, \ldots, n \; .$$

2. In Abschnitt 5.2. haben wir Determinanten im $\mathbb{R}^n$ kennengelernt, die uns als "Sensoren für lineare Abhängigkeit/Invertierbarkeit" sehr gute Dienste leisten. Zur Berechnung von Determinanten quadratischer Matrizen ist es sehr nützlich zu wissen, wie sie sich bei gewissen Umformungen der Matrix ändern.

Hier betrachten wir die Zeilenumformungen aus Definition 3.3.A der Vorlesung. Zwei Typen von Zeilenumformungen sind in dieser Definition angegeben, Typ (i) (Vielfaches einer Zeile von einer anderen Zeile subtrahieren) und Typ (ii) (Multiplikation einer Zeile mit einem Skalar).

a) Zeigen Sie, dass sich die Determinante einer quadratischen Matrix *nicht ändert*, wenn man die Matrix einer Zeilenumformung vom Typ (i) unterwirft.

b) Wie ändert sich die Determinante einer Matrix bei einer Zeilenumformung vom Typ (ii)?

c) Wie ändert sich schliesslich die Determinante einer Matrix beim Vertauschen zweier Zeilen?

Hinweis. Verwenden Sie Satz 5.2.H. in Verbindung mit den Sätzen 5.2.B und 5.2.C. Natürlich brauchen Sie für Teilaufgabe b) auch noch die Linearität der Determinante in jedem Argument.

3. In Abschnitt 5.1 der Vorlesung wurde demonstriert, wie es die Diagonalisierung einer Matrix möglich macht, die allgemeine Lösung eines linearen autonomen Systems von Differentialgleichungen bzw. einer linearen Rekursion zu bestimmen. Aus diesen Formeln kann man dann sofort ablesen, ob und wann die Lösungen beliebig gross werden, ob und wann sie beschränkt bleiben, oder ob sie gar gegen 0 streben.

Wir betrachten die diagonalisierbare Matrix

$$\mathbf{A} = \mathbf{S}\mathbf{D}\mathbf{S}^{-1} \quad \text{mit} \quad \mathbf{D} = \begin{pmatrix} -1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \; , \quad \mathbf{S} = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} \; ,$$

und dazu das lineare autonomen System von Differentialgleichungen

$$\mathbf{y}'(x) = \mathbf{A}\mathbf{y}(x) \; , \tag{6}$$

bzw. die lineare Rekursion

$$\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{y}^{(k)} \; . \tag{7}$$

Siehe nächstes Blatt!

a) Gibt es Lösungen von (6), die für $x \to \infty$ gegen 0 streben? Wenn ja, charakterisieren Sie die Menge der Anfangswerte $\mathbf{y}(0) \in \mathbb{R}^2$, für die man eine solche Lösung erhält.

Hinweis: Welche Koordinaten haben die gesuchten Anfangswerte in der Basis, die von den Spalten von $\mathbf{S}$ gebildet wird?

b) Gibt es Lösungen von (7) die für $k \to \infty$ gegen 0 streben? Wenn ja, charakterisieren Sie die Menge der Startvektoren $\mathbf{y}^{(0)} \in \mathbb{R}^2$, für die man eine solche Lösung erhält.

Hinweis: Der Hinweis zur vorhergehenden Teilaufgabe ist auch hier nützlich.

c) Gibt es unbeschränkte Lösungen von (6)? Wenn ja, charakterisieren Sie die Menge der Anfangswerte $\mathbf{y}(0) \in \mathbb{R}^2$, für die man eine unbeschränkte Lösung erhält.

Hinweis: Überlegen Sie sich zuerst, welche Koordinaten in der Basis, die von den Spalten von $\mathbf{S}$ gebildet wird, diejenigen Vektoren $\mathbf{y}(0)$ haben, die zu beschränkten Lösungen führen. Alle anderen Anfangsvektoren ergeben dann unbeschränkte Lösungen.

d) Gibt es unbeschränkte Lösungen von (7)? Wenn ja, charakterisieren Sie die Menge der Startvektoren $\mathbf{y}^{(0)} \in \mathbb{R}^2$, für die man eine unbeschränkte Lösung erhält.

Hinweis: Der Hinweis zur vorhergehenden Teilaufgabe ist auch hier nützlich.

4. In Abschnitt 5.1 der Vorlesung wurde erklärt, wie die Diagonalisierung einer Matrix ziemlich umfassende Information über das Verhalten einer linearen Rekursion liefert. In Aufgabe 3 wurde das benutzt, um Startwerte zu identifizieren, die garantieren, dass die Iterierten einer linearen Rekursion immer beschränkt bleiben. In dieser Aufgabe werden wir einen scheinbaren Widerspruch zwischen mathematischer Theorie und numerischer Praxis erleben.

Wir betrachten die lineare Rekursion $\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{y}^{(k)}$, $k \in \mathbb{N}$, mit

$$\text{mit} \quad \mathbf{A} := \begin{pmatrix} \frac{11}{3} & \frac{10}{3} \\ -\frac{5}{3} & -\frac{4}{3} \end{pmatrix}, \text{ also } \quad \mathbf{y}^{(k+1)} = \begin{pmatrix} \frac{11}{3} & \frac{10}{3} \\ -\frac{5}{3} & -\frac{4}{3} \end{pmatrix} \mathbf{y}^{(k)}, \quad k \in \mathbb{N}_0 .$$

a) Weisen Sie nach, dass

$$\begin{pmatrix} \frac{11}{3} & \frac{10}{3} \\ -\frac{5}{3} & -\frac{4}{3} \end{pmatrix} = \mathbf{S}\, \mathrm{diag}(2, \tfrac{1}{3})\, \mathbf{S}^{-1} \quad \text{mit} \quad \mathbf{S} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} .$$

b) Wie lauten die Iterierten für den Startwert $\mathbf{y}^{(0)} = \binom{-1}{1}$? Wie verhält sich diese Folge $(\mathbf{y}^{(k)})_k$ für $k \to \infty$?

c) Wie lauten die Iterierten für den Startwert $\mathbf{y}^{(0)} = \binom{-1+2\varepsilon}{1-\varepsilon}$ mit $\varepsilon = 2^{-20}$? Wie verhält sich in diesem Fall die Folge $(\mathbf{y}^{(k)})_k$ für $k \to \infty$?

d) Listing 1 zeigt den MATLAB-Code, der die Ausgabe von Abbildung 1 produziert hat. Wie erklären Sie das Verhalten der Iterierten in der numerischen Rechnung? Liegt nicht ein Widerspruch zu Teilaufgabe b) vor?

Hinweis: Denken Sie daran, dass der Computer mit Maschinenzahlen rechnet und was das zusammen mit dem Ergebnis von Teilaufgabe c) bedeutet.

Listing 1: Aufgabe 4: Lineare Iteration

```
% MATLAB script for testing the stability of a linear iteration in a
% numerical implementation
```

Bitte wenden!

```matlab
A = [11/3 10/3; -5/3 -4/3]; % Iteration matrix
y = [-1; 1];           % Initial vector y^(k)
norms = norm(y);       % Records Euclidean norms of iterates
for k=1:100
    y=A*y;
    norms = [norms;norm(y)];
end
% Plot norms
figure('name','norms of iterates');
semilogy(0:100,norms,'r+');
xlabel('{\bf k}','fontsize',14);
ylabel('{\bf norm of y^{k}}','fontsize',14);

print -depsc2 'stabdiscdynsys.eps';
```

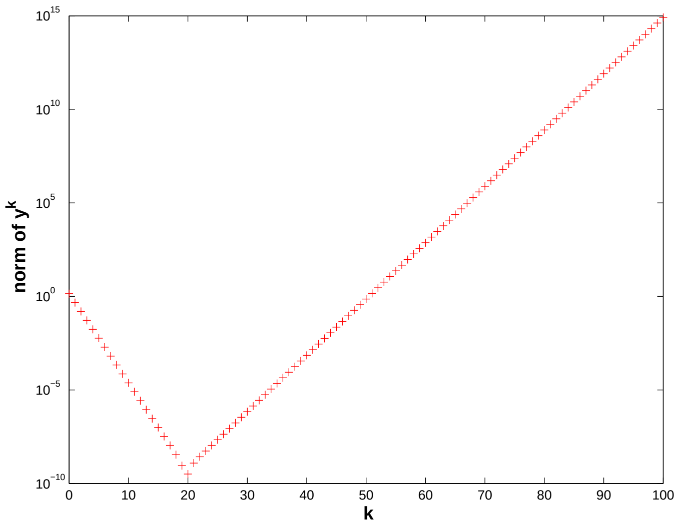

Abbildung 1: Grafische Ausgabe des MATLAB-Skripts `stabdiscdynsys.m`. Beachten Sie die logarithmische Skala.

5. In dieser Aufgabe geht es um Mustererkennung, ein zentrales Problem in der Bildverarbeitung, etwa von Luftbildern. Hier betrachten wir natürlich eine sehr einfache Situation.

An einem Christbaum hängen zwei Arten von Schmuck, die aus hauchdünnem Silberblech ausgestanzt wurden, nämlich fünfzackige Sterne und stilisierte Tannenbäume, siehe Abbildung 2. Die Figuren sind mit verschiedenfarbigen bunten Perlen besetzt in Abbildung 2 angedeutet durch rote Pentagramme. Die MATLAB-Funktionen, die 2×15-Matrizen mit den Koordinaten der Perlen in ihren Spalten erzeugen, sind in den Listings 2 und 3 angegeben.

Listing 2: Positionen der Perlen auf Christbaumfigur von der Gestalt eines Baumes

```matlab
function pts = Xmastree(fn)
% MATLAB function creating the points of a tree
pts = [1 0; 4 -3 ; 1 -3 ;1 -4 ; -1 -4; -1 -3; -4 -3; ...
       -1 0; -3 0; -1 2; -2 2; 0 4; 2 2; 1 2; 3 0]';
```

Siehe nächstes Blatt!

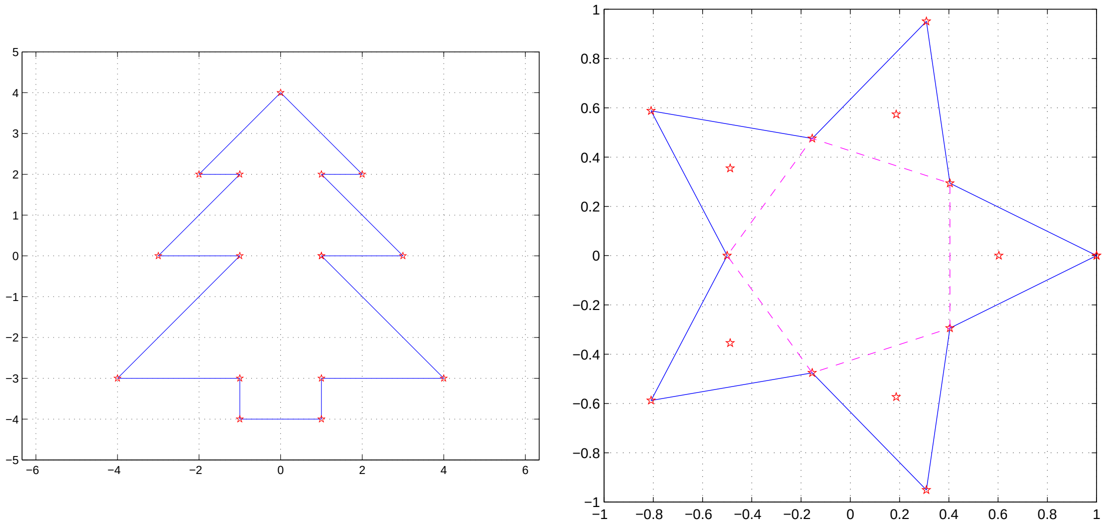

Abbildung 2: Gestalt der Christbaumfiguren aus Silberblech. Lage der Perlen ist durch rote Pentagramme markiert. Die Achsen geben die Länge in cm an.

```
% Close polygon
ppts = [pts,pts(:,1)];
% Draw tree
figure('name','tree');
plot(ppts(1,:),ppts(2,:),'b-',ppts(1,:),ppts(2,:),'rp');
grid on; axis([-5 5 -5 5]); axis equal;

% Print, if filename was specified
if (nargin > 0),  print('-depsc2',[fn , '.eps']); end
```

Listing 3: Position der Perlen auf der sternförmigen Christbaumfigur

```
function pts = star(fn)
% MATLAB function creating the points of a star
% points of rays
outpts = [cos((0:4)*2*pi/5);sin((0:4)*2*pi/5)];
% inner corners
inpts = 0.5*[cos((0:4)*2*pi/5+pi/5);sin((0:4)*2*pi/5+pi/5)];
% merge both point sets
pts = zeros(2,10); pts(:,1:2:9) = outpts; pts(:,2:2:10) = inpts;
% centers of rays are stored in mp
inpts = [inpts(:,end),inpts];
mp = (outpts+inpts(:,1:end-1)+inpts(:,2:end))/3;

% Draw star
figure('name','star');
plot([pts(1,:),pts(1,1)],[pts(2,:),pts(2,1)],'b-',...
     [pts(1,:),pts(1,1)],[pts(2,:),pts(2,1)],'rp',...
     inpts(1,:),inpts(2,:),'m--',...
     mp(1,:),mp(2,:),'rp');
```

Bitte wenden!

```
grid on; axis equal; axis([-1 1 -1 1]);

% Print, if filename was specified
if (nargin > 0),  print('-depsc2',[fn , '.eps']); end
pts = [pts mp];
```

Der Christbaum wird nun fotografiert und eine Bilderkennungssoftware kann dann die Lage der Perlen zu einzelnen Christbaumfiguren in der Bildebene bestimmen. Sie kann diese Positionen auch so normalisieren, dass das Zentrum einer Figur (der Punkt $\mathbf{0}$) wieder die Koordinaten $\binom{0}{0}$ hat.

Allerdings ist das Bild einer Christbaumfigur auf dem Foto durch schiefe Ansicht und unterschiedliche Entfernungen nicht kongruent zur ursprünglichen Figur, sondern einer ziemlich allgemeinen *linearen Abbildung* unterworfen worden. Was also an Daten zur Verfügung steht sind 15 Koordinatenvektoren $\mathbf{p}^i$, $i = 1, \dots, 15$, (Positionen der Perlen auf dem Foto) und das Wissen, dass es eine 2×2-Matrix so gibt, dass

$$\mathbf{p}^i = \mathbf{A}\mathbf{x}^i \;, \quad i = 1, \dots, 15 \;, \tag{8}$$

wobei $\mathbf{x}^i \in \mathbb{R}^2$ die Koordinaten der Perlen im "Figursystem" sind, also die Koordinaten, die von den Funktionen `Xmastree` und `star` aus den Listings 2 und 3 zurückgegeben werden. Man weiss allerdings nicht, ob es sich bei der fotografierten Figur um einen Baum oder einen Stern handelt. Erschwerend kommt hinzu, dass das Foto nicht ganz scharf ist, so dass die Positionen $\mathbf{p}^i$ nur ungefähr bekannt sind (Messfehler!).

In dieser Aufgabe soll untersucht werden, wie mit der Methode der kleinsten Quadrate die Verzerrungsmatrix $\mathbf{A}$ aus (8) gefunden werden kann und darüber hinaus auch noch bestimmt werden kann, um welche Figur es sich handelt. Wir lösen also auch ein Problem der *Mustererkennung*.

a) Schreiben Sie (8) explizit als überbestimmtes Gleichungssystem. Was ist dessen Grösse und was sind die Unbekannten?

b) Bestimmen Sie, wann die Koeffizientenmatrix dieses überbestimmten Gleichungssystems vollen Rang hat und geben Sie eine geometrische Interpretation dieser Bedingung.

c) Implementieren Sie eine MATLAB-Funktion

`function B = shapeidentmat(X)` ,

die die Matrix $\mathbf{B}$ des überbestimmten linearen Gleichungssystems aus der ersten Teilaufgabe berechnet, wobei in den Spalten der 2×15-Matrix `X` die Koordinatenvektoren $\mathbf{x}^i \in \mathbb{R}^2$ übergeben werden.

d) Schreiben Sie die eine MATLAB-Funktion

`function A = complinmap(X,P)` ,

die die Kleinste-Quadrate-Lösung des überbestimmten linearen Gleichungssystems aus der ersten Teilaufgabe berechnet, wobei in den Spalten der 2×15-Matrizen `X` und `P`, die Positionen der Perlen im "Figurkoordinatensystem" und "Fotokoordinatensystem" übergeben werden. Der Rückgabewert soll eine Näherung für die Matrix $\mathbf{A} \in \mathbb{R}^{2,2}$ aus (8) sein.

e) Erweitern Sie nun Ihre Implementierung von `complinmap` so, dass neben $\mathbf{A}$ auch noch die Euklidische Norm des Residuums der Kleinste-Quadrate-Lösung des überbestimmten linearen Gleichungssystems zurückgegeben wird:

Siehe nächstes Blatt!

```
function [A,resnorm] = complinmap(X,P).
```

f) Erklären Sie, warum die Norm des Residuums der Kleinste-Quadrate-Lösung des überbestimmten linearen Gleichungssystems benutzt werden kann, um den Typ der Figur, die man fotografiert hat, zu bestimmen.

g) In der Datei `fotodaten.dat` (die Datei kann in MATLAB mithilfe des Befehls `load 'fotodaten.dat'` geladen werden) finden Sie 20 Matrizen der Grösse 2×15, gespeichert in Form einer 40×15-Matrix $\mathbf{D}$, wobei die i. Matrix der 20 Matrizen gegeben ist durch $\mathbf{D}(2i : 2i+1, :)$. Die Matrizen $\mathbf{D}(2i : 2i+1, :)$ enthalten die "Fotokoordinaten" der Perlen der oben beschriebenen Christbaumfiguren, wie sie aus einer Aufnahme des gesamten Christbaums extrahiert worden sind.

Schreiben Sie ein MATLAB-Skript `countXmasdecoration`, das zählt, wieviele Sterne und wieviele Bäume auf dem Foto zu sehen sind.

Hinweis: Bestimmen Sie einfach die Euklidische Norm der Residuen für die Kleinste-Quadrate-Lösungen für die "Figurkoordinaten" sowohl des Sterns als auch des Baums und vergleichen Sie diese.

6. Mit Hilfe von (numerischer) linearer Algebra kann man tatsächlich Milliardär werden, wie es die Gründer von Google, Sergey Brin und Larry Page, vorgemacht haben. Die erste Version des überlegenen PageRank Priorisierungsalgorithmus von Google stützte sich tatsächlich auf eine stationäre Markov-Kette, also eine spezielle lineare Rekursion, wie in Abschnitt 5.1 der Vorlesung diskutiert.

Das Problem ist, eine sinnvolle Rangordnung einer grossen Anzahl $N \in \mathbb{N}$ von untereinander verlinkten Webseiten aus der blossen Information zu bestimmen, welche Seite mit welcher anderen verlinkt ist. Das geschieht mit Hilfe einer stationären Markov-Kette, die das Verhalten eines "Zufallswebsurfers modelliert".

Die Webseiten seien durchnummeriert von 1 bis N. Die Markov-Kette hat N Zustände, wobei der Zustand i bedeutet, dass sich der Websurfer auf der Webseite mit der Nummer i befindet. In festen Zeitabständen springt der Surfer zu einer anderen Webseite und zwar nach folgenden Regeln:

1. Enthält die aktuelle Seite keinen Link, dann besucht er mit der gleichen Wahrscheinlichkeit $\frac{1}{N}$ irgendeine der N Seiten im nächsten Schritt.
2. Andernfalls, wenn es auf der Seite i, auf der er sich aktuell befindet, $n_i \in \mathbb{N}$ Links gibt, so klickt er einen davon mit der Wahrscheinlichkeit $\frac{1-\delta}{n_i}$ an.
3. Gibt es auf der aktuellen Seite Links, so springt er trotzdem mit der Wahrscheinlichkeit $\frac{\delta}{N}$ zu irgendeiner Seite (, die auch die sein kann, auf der er sich gerade befindet).

Dabei ist $\delta \in]0, 1[$ fest vorgegeben.

Dieses Verhalten lässt durch folgende *Übergangswahrscheinlichkeiten* beschreiben:

$$p_{ij} = \begin{cases} \frac{1}{N} & \text{, falls die Seite } j \text{ keine Links aufweist, also } n_j = 0 \\ \frac{1-\delta}{n_j} + \frac{\delta}{N} & \text{, falls Link auf Seite } i \text{ auf Seite } j \\ \frac{\delta}{N} & \text{, sonst.} \end{cases} \tag{9}$$

Dabei ist p_{ij} zu interpretieren als die Wahrscheinlichkeit, dass der Websurfer im nächsten Schritt auf Seite i landet, wenn er sich aktuell auf Seite j befindet.

Führt man also die Vektoren $\mathbf{y}^{(k)} \in \mathbb{R}^N$ ein, für die die Komponente $y_j^{(k)}$ angibt, mit welcher Wahrscheinlichkeit sich der Websurfer nach dem k. Schritt auf der Webseite j befindet, so findet man dafür die lineare Rekursion

$$\mathbf{y}^{(k+1)} = \mathbf{P}\mathbf{y}^{(k)} \quad , \quad \mathbf{P} = (p_{ij})_{i,j=1}^N \ . \tag{10}$$

Bezeichnet $\mathbf{y}^{(\infty)} = \lim_{k\to\infty} \mathbf{y}^{(k)}$ (falls dieser Grenzwert existiert), dann lassen sich die Seiten nach den ihnen in $\mathbf{y}^{(\infty)}$ zugewiesenen Wahrscheinlichkeiten anordnen. Dies liefert dann ihren Rang.

Wir wollen PageRank nun in MATLAB implementieren und benötigen dazu eine Datenstruktur, die einen Teil des Internets mit N Seiten und ihren Links (untereinander) beschreiben kann. Wir wählen dafür eine (dünnbesetzte) $N \times N$-Matrix $\mathbf{G} = (g_{ij})$ und folgende Konvention:

$g_{ij} = 1$, falls die Seite j einen Link auf die Seite i enthält, andernfalls $g_{ij} = 0$.

a) Weisen Sie nach, dass $\mathbf{P}$ tatsächlich eine *stochatische Matrix* ist, also eine Matrix mit Spaltensummen $= 1$.

b) Schreiben Sie eine MATLAB-Funktion

`function nl = countlinks(G)` ,

die einen Spaltenvektor $(n_j)_{j=1}^N \in \mathbb{N}_0^N$ zurückgibt, in dessen Komponenten die Anzahlen der Links auf den einzelnen Webseiten stehen.

Hinweis: Machen Sie sich kundig, was das MATLAB-Kommando `sum` bewirkt, wenn man es auf eine Matrix anwendet.

c) Implementieren Sie nun eine Funktion

`function P = buildP(G,delta)` ,

welche die Matrix $\mathbf{P}$ der Übergangswahrscheinlichkeiten aus $\mathbf{G}$ aufbaut. Das Argument `delta` teilt der Funktion den Wert von δ mit.

d) In dem File `harvard500.dat` befindet sich die Matrix $\mathbf{G}$ für ein "Modellinternet". Lassen Sie nun, für $\delta = 0.15$ und beginnend mit irgendeinem Einheitsvektor, die lineare Iteration (10) laufen, um $\mathbf{y}^{(\infty)}$ zu bestimmen. Dabei machen wir die (mathematisch wohlbegründete) Annahme, dass $\mathbf{y}^{(\infty)}$ existiert.

Wir sind allerdings alle endliche Wesen, so dass Sie nicht abwarten können bis $\mathbf{y}^{(\infty)}$ tatsächlich berechnet ist. Daher brechen Sie die Rechnung ab, wenn

$$\frac{\left\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\right\|}{\left\|\mathbf{y}^{(k+1)}\right\|} \leq 10^{-4} \ . \tag{11}$$

Hierbei steht $\|\cdot\|$ für die Euklidische Vektornorm, verfügbar durch die MATLAB-Funktion `norm`. Plotten Sie dann Ihre Näherung für $\mathbf{y}^{(\infty)}$ mit den Befehlen

```
plot((1:N)',y,'r+'); xlabel('Seitenindex');
ylabel('Komponente von y^{\infty}').
```

Hinweis: Ihr Plot sollte so ungefähr aussehen wie der aus Abbildung 3

e) Wir nehmen nun an, dass die Gesamtzahl aller Links $\leq 5N$ ist. Wie gross ist der asymptotische Rechenaufwand für $N \to \infty$ eines Schrittes der Iteration (10) in Ihrer Implementierung aus der vorhergehenden Teilaufgabe?

Hinweis: Repetieren Sie das Konzept des *Rechenaufwandes* und seines asymptotischen Verhaltens aus Abschnitt 2.2.2. der Vorlesung.

Siehe nächstes Blatt!

f) Bestimmen Sie zwei *Diagonalmatrizen* $\mathbf{D}_1$ und $\mathbf{D}_2$ so, dass sich schreiben lässt:

$$\mathbf{P} = \mathbf{1}\mathbf{1}^T\mathbf{D}_1 + (1-\delta)\mathbf{G}\mathbf{D}_2 \ . \tag{12}$$

Das Symbol $\mathbf{1}$ bezeichnet einen Spaltenvektor der Länge N ein allen Einträgen $= 1$.

g) Wie ist nun die Aysmptotik des Rechenaufwandes für ein Matrix-Vektor Produkt mit $\mathbf{P}$ in der Darstellung (12) unter derselben Annahme wie in Teilaufgabe **e)** (Gesamtzahl aller Links $\leq 5N$)?

Hinweis: Bitte studieren Sie nochmals die Lösung von Aufgabe 5 von Serie 4.

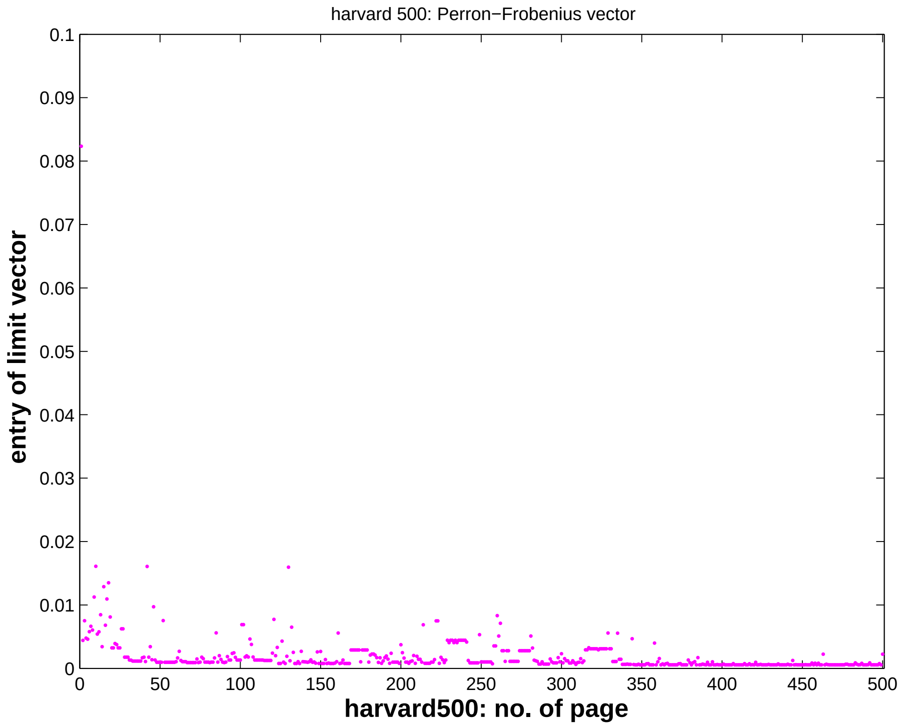

Abbildung 3: Stationäre Besuchswahrscheinlichkeiten $\mathbf{y}^{(\infty)}$ aus dem PageRank-Algorithmus

- **Abgabe der Serien:** Donnerstag, 19.12.2013 in der Übungsgruppe oder bis 16:00 Uhr in den Fächern im Vorraum zum HG G 53. Die Serien müssen sauber und ordentlich geschrieben und zusammengeheftet abgegeben werden, sofern eine Korrektur gewünscht wird.
- **Semesterpräsenz:** Montag, 15:15 - 17:00 Uhr, ETH Zentrum, LFW E 13. Falls keine grosse Nachfrage besteht, warten die Assistenten maximal eine halbe Stunde. Wir bitten Sie deshalb, bei Fragen so früh als möglich zu erscheinen.
- **Ferienpräsenz:**
  - Montag, 13.01.2014, 15:15-17:00 Uhr, HG E 41
  - Montag, 27.01.2014, 15.15-17:00 Uhr, HG D 3.2
  - Montag, 10.02.2014, 15.15–17:00 Uhr, HG D 3.2

Bitte wenden!

Falls keine grosse Nachfrage besteht, warten die Assistenten auch hier maximal eine halbe Stunde. Wir bitten Sie deshalb, bei Fragen so früh als möglich zu erscheinen. Weitere allgemeine Präsenzen (nicht nur für diese Vorlesung) werden bald bekannt gegeben unter
http://www.sam.math.ethz.ch/teaching/ferienpraesenz .

- **Homepage:** Hier werden zusätzliche Informationen zur Vorlesung und die Serien und Musterlösungen als PDF verfügbar sein.
  www.math.ethz.ch/education/bachelor/lectures/hs2013/other/linalgnum_BAUG