Dr. R. Käppeli
M. Sprecher

Musterloesung 6

1. **a)** Für einen Fixpunkt x^* gilt $Ax^* = \|Ax^*\|_2 x^*$. Folglich ist x^* ein Eigenvektor mit positivem Eigenwert $\|Ax^*\|_2$. Zudem gilt $\|x^*\|_2 = \|{}^{Ax^*}/_{\|Ax^*\|_2}\|_2 = 1$. Umgekehrt ist auch jeder normierte Eigenvektor mit positivem Eigenwert ein Fixpunkt der Iteration. Die Fixpunkte sind also gerade die normierten Eigenvektoren mit positiven Eigenwerten.

b)
```
function [ew,xnew]=poweriteration(A,x)

%maximal number of iteration
itermax=10000;

%toleranz
tol=10^(-10);

%initialize xold
xold=x;

for i=1:itermax
    xnew=A*xold;
    %normieren
    xnew=xnew/norm(xnew);
    if norm(xnew-xold)<tol
        break;
    end
    xold=xnew;
end
%show number of iterations
i

%compute eigenvalue
ew=xnew'*(A*xnew);

end
```

```
%poweriterationtest.m

%size of Matrix
n=100;

%implement Matrix
A=diag(2*ones(1,n))+diag(ones(1,n-1),1)+diag(ones(1,n-1),-1);

%compute largest eigenvalue of A with power-method
ewpi=poweriteration(A,ones(n,1))

%compute eingevalues of A with Matlabfunction eig
ew=eig(A);

%choose largest eigenvalue
ewlast=ew(end)

%compare the 2 different computations of largest eigenvalue
norm(ewpi-ewlast)
```

Es werden 8101 Iteration benötigt. Der Eigenvekter ist ≈ 3.9990 und der Fehler zwischen den berechneten Eigenwerten $4.4409e-16$.

2. **a)** Die Eigenwerte λ müssen $\det(\mathbf{I}\lambda - \mathbf{S}) = \mathbf{0}$ erfüllen. Entwicklung nach der ersten Spalte und Ausnützen der Dreiecksgestalt der Untermatrizen liefert $\det(\mathbf{I}\lambda - \mathbf{S}) = \lambda^n + (-1)^{2n+1} = \lambda^n - 1 = 0 \Rightarrow |\lambda_i| = 1,\ i = 1, ..., n.$

b) Aufgrund der besonderen Struktur der Matrix gilt $\mathbf{z}^n = \mathbf{S}^n\mathbf{z}^0 = \mathbf{z}^0$, d.h. keine Konvergenz, es sei den $\mathbf{z}_0$ ist bereits Eigenvektor.

c) Eine Voraussetzung des Theorems ist, dass es einen betragsmässig grössten Eigenwert λ_1 gibt, was hier nicht der Fall ist. Gemäss Theorem ist die Konvergenzrate $\frac{\lambda_2}{\lambda_1}$, was in diesem Beispiel 1 ist $\Rightarrow$ keine Konvergenz.

3. **a)** Die (Kräfte-) Gleichgewichtsgleichungen für alle Gelenke werden mit Hilfe des Schnittprinzips berechnet. Die ässeren Kräfte verteilen sich als innere Kräfte auf die einzelnen Stäbe, die dadurch gestreckt oder gestaucht werden und somit für die Deformation des gesamten Tragwerks Verantwortlich sind. Die Berechnung der inneren Kräfte erfolgt mit Hilfe des Schnittprinzips: Ein enzelnes Gelenk wird freigeschnitten, indem alle Verbindungsstäbe zu Nachbargelenken durchtrennt werden; die inneren Kräfte kompensieren den Wegfall der festen Verbindungen an den Schnittenden. Da Stäbe Kräfte nur in ihrer Längsrichtung aufnehmen können, zeigen diese inneren Kräfte in die jeweilige Stabrichtung; sie

Siehe nächstes Blatt!

können daher durch skalare Grössen $f_k, k = 1, ..., 18$ repräsentiert werden, deren Vorzeichen für Zugkräfte (vom Gelenk weg zeigend) positiv und Druckkäfte negativ gsesetzt wird.
Das freigeschinttene System soll sich im statischen Gleichgewicht befinden. Daraus ergibt sich für die Gelenke folgende Gleichungen:

$$
\begin{aligned}
0 &= \boldsymbol{p}_1 + f_1[1,0]^T + f_4[-c,-s]^T + f_5[0,-1]^T \\
0 &= \boldsymbol{p}_2 + f_1[-1,0]^T + f_2[1,0]^T + f_6[-c,-s]^T + f_7[0,-1]^T + f_8[c,-s]^T \\
0 &= \boldsymbol{p}_3 + f_2[-1,0]^T + f_3[1,0]^T + f_9[-c,-s]^T + f_{10}[0,-1]^T + f_{11}[c,-s]^T \\
0 &= \boldsymbol{p}_4 + f_3[-1,0]^T + f_{12}[0,-1]^T + f_{13}[c,-s]^T \\
0 &= \boldsymbol{p}_5 + f_5[0,1]^T + f_6[c,s]^T + f_{14}[-1,0]^T + f_{15}[1,0]^T \\
0 &= \boldsymbol{p}_6 + f_7[0,1]^T + f_9[c,s]^T + f_{15}[-1,0]^T + f_{16}[1,0]^T \\
0 &= \boldsymbol{p}_7 + f_8[-c,s]^T + f_{10}[0,1]^T + f_{16}[-1,0]^T + f_{17}[1,0]^T \\
0 &= \boldsymbol{p}_8 + f_{11}[-c,s]^T + f_{12}[0,1]^T + f_{17}[-1,0]^T + f_{18}[1,0]^T,
\end{aligned}
$$

wobei $s = \sin\vartheta$ und $c = \cos\vartheta$.

b) Die Längenänderungen berechnen sich aus den alten Positionen $\boldsymbol{r}_i$ und den neuen Positionen $\boldsymbol{r}_i + \boldsymbol{\Delta r}_i$ der einzelnen Gelenke. Verbindet etwa der k-te Stab die Gelenke $\boldsymbol{r}_i$ und $\boldsymbol{r}_j$ dann errechnet sich die neue Länge $l_k + \Delta l_k$ aus

$$
\begin{aligned}
(l_k + \Delta l_k)^2 &= \|(\boldsymbol{r}_i + \boldsymbol{\Delta r}_i) - (\boldsymbol{r}_j + \boldsymbol{\Delta r}_j)\| \\
&= \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 + 2(\boldsymbol{r}_i - \boldsymbol{r}_j)(\boldsymbol{\Delta r}_i - \boldsymbol{\Delta r}_j) + \|\boldsymbol{\Delta r}_i - \boldsymbol{\Delta r}_j\|^2
\end{aligned}
$$

Durch vernachlässigen der Terme Δl_k^2 und $\|\boldsymbol{\Delta r}_i - \boldsymbol{\Delta r}_j\|^2$ ergibt sich das gewünschte.

c)
```
% Anfangs-Gelenk Koordinaten
theta = 0.25*pi;
r = [1; %  0;x_1
     1; %  1;y_1
     2; %  2;x_2
     1; %  3;y_2
     3; %  4;x_3
     1; %  5;y_3
     4; %  6;x_4
     1; %  7;y_4
     1; %  8;x_5
     0; %  9;y_5
     2; % 10;x_6
     0; % 11;y_6
     3; % 12;x_7
     0; % 13;y_7
```

```
     4; % 14;x_8
     0  % 15;x_8
    ];

% Gleichgewichtsmatrix E
c = cos(theta);
s = sin(theta);
%  (  1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18)
E = [-1, 0, 0, c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;  %  1
      0, 0, 0, s, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;  %  2
      1,-1, 0, 0, 0, c, 0,-c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;  %  3
      0, 0, 0, 0, 0, s, 1, s, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;  %  4
      0, 1,-1, 0, 0, 0, 0, 0, c, 0,-c, 0, 0, 0, 0, 0, 0, 0;  %  5
      0, 0, 0, 0, 0, 0, 0, 0, s, 1, s, 0, 0, 0, 0, 0, 0, 0;  %  6
      0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,-c, 0, 0, 0, 0, 0;  %  7
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, s, 0, 0, 0, 0, 0;  %  8
      0, 0, 0, 0, 0,-c, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0, 0, 0;  %  9
      0, 0, 0, 0,-1,-s, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;  % 10
      0, 0, 0, 0, 0, 0, 0, 0,-c, 0, 0, 0, 0, 0, 1,-1, 0, 0;  % 11
      0, 0, 0, 0, 0, 0,-1, 0,-s, 0, 0, 0, 0, 0, 0, 0, 0, 0;  % 12
      0, 0, 0, 0, 0, 0, 0, c, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0;  % 13
      0, 0, 0, 0, 0, 0, 0,-s, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0;  % 14
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, c, 0, 0, 0, 0, 0, 1,-1;  % 15
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-s,-1, 0, 0, 0, 0, 0, 0;  % 16
    ];

% aeussere Kraefte p
m = 1;  % Gelenk Masse
g = 10 ;% Schwerkraft
p = [  0;   %  1
     -m*g;  %  2
       0;   %  3
     -m*g;  %  4
       0;   %  5
     -m*g;  %  6
       0;   %  7
     -m*g;  %  8
       0;   %  9
     -m*g;  % 10
       0;   % 11
     -m*g;  % 12
       0;   % 13
```

Siehe nächstes Blatt!

```
     -m*g;  % 14
       0;   % 15
     -m*g;  % 16
    ];

% Diagonal Matrix D^-1 = Di
Di = diag([1/norm([r( 1),r( 2)] - [r( 3),r( 4)]) % Stab  1
           1/norm([r( 3),r( 4)] - [r( 5),r( 6)]) % Stab  2
           1/norm([r( 5),r( 6)] - [r( 7),r( 8)]) % Stab  3
           1/norm([   0,   0] -   [r( 1),r( 2)]) % Stab  4
           1/norm([r( 9),r(10)] - [r( 1),r( 2)]) % Stab  5
           1/norm([r( 9),r(10)] - [r( 3),r( 4)]) % Stab  6
           1/norm([r(11),r(12)] - [r( 3),r( 4)]) % Stab  7
           1/norm([r( 3),r( 4)] - [r(13),r(14)]) % Stab  8
           1/norm([r(11),r(12)] - [r( 5),r( 6)]) % Stab  9
           1/norm([r(13),r(14)] - [r( 5),r( 6)]) % Stab 10
           1/norm([r( 5),r( 6)] - [r(15),r(16)]) % Stab 11
           1/norm([r(15),r(16)] - [r( 7),r( 8)]) % Stab 12
           1/norm([   5,   0] -   [r( 7),r( 8)]) % Stab 13
           1/norm([   0,   0] -   [r( 9),r(10)]) % Stab 14
           1/norm([r( 9),r(10)] - [r(11),r(12)]) % Stab 15
           1/norm([r(11),r(12)] - [r(13),r(14)]) % Stab 16
           1/norm([r(13),r(14)] - [r(15),r(16)]) % Stab 17
           1/norm([   5,   0] -   [r(15),r(16)]) % Stab 18
          ]);

% Matrix A austellen
eta = 2500; % Elastizitaetsmodule
% Stelle hier die Matrix A auf
% A
A = eta*E*Di*E';

% Loese System fuer Gleichgewichts-Verschiebungen
% verwende einen Befehl aus numpylinalg oder scipylinalg

dreq = A\p;
figure(1);
plot_bridge(r,dreq)
print('-dpdf','GleichGewicht.pdf')

% Bestimme Eigenvektoren v zum kleinsten Eigenwert von A
[~,v]=poweriteration(inv(A),rand(16,1));
```

Bitte wenden!

```
%Anzahl Zeitpunkte die dargestellt werden
n=100;
dr=dreq*ones(1,n)+v*cos(linspace(0,4*pi,n));
figure(2);
simulate_bridge(r,dr);
print('-dpdf','Eigenschwingung')
```

Durch Ausführen von $bridge.m$ wird Abbildung 1 produziert.

d) Die Bewegungsgleichung ergibt sich aus dem Newtonschen Gesetz zu

$$m\ddot{\boldsymbol{\Delta r}} = p - A\boldsymbol{\Delta r}$$

e) Durch einfaches einsetzen ergibt sich

$$m\ddot{\boldsymbol{\Delta r}} = -\lambda\cos(\sqrt{\lambda}t)\boldsymbol{v} \tag{1}$$

und

$$p - A\boldsymbol{\Delta r} = p - A\boldsymbol{\Delta r_0} - A\cos(\sqrt{\lambda}t)\boldsymbol{v} = -\lambda\cos(\sqrt{\lambda}t)\boldsymbol{v} \tag{2}$$

also $m\ddot{\boldsymbol{\Delta r}} = p - A\boldsymbol{\Delta r}$.

f) Durch Ausführen von $bridge.m$ wird Abbildung 2 und eine kleine Simulation der Eigenschwingung produziert.

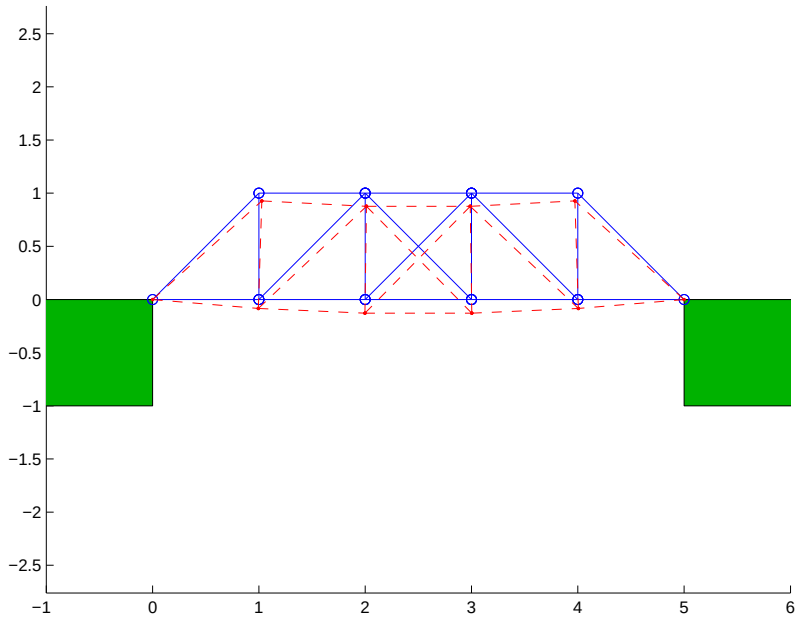

Abbildung 1

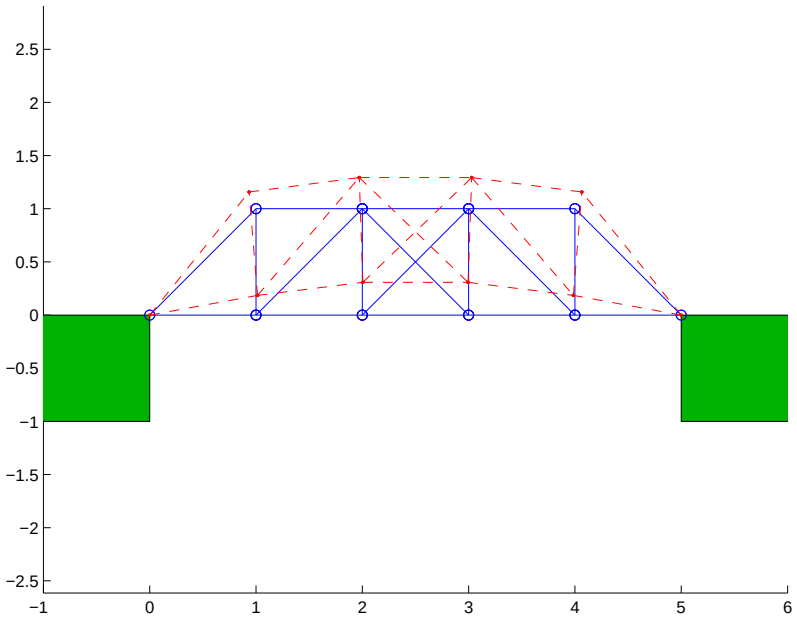

Abbildung 2