```
% C. Petersen, H. Werkle, Dynamik der Baukonstruktionen
% 2. Auflage, Springer Vieweg, Wiesbaden, 2018
%
% ML_13_4_Antwortspektrenverfahren_MFG: Antwortspektrenverfahren für
% beliebige Systeme
%
% Version 1.0, April 2018
% Softwareentwicklung:
% Andrei Firus, M.Eng (andrei.firus@gmail.com)

% Aufbau Eingabedateien:
% Inputdatei_1: Nachgiebigkeitsmatrix [m/N] bzw. [1/Nm]
% Inputdatei_2: Massenmatrix [kg] bzw. [kg*m^2]
% Inputdatei_3: Antwortspektrum
%               Spalte 1: Eigenschwingzeiten in [s]
%               Spalte 2: Spektralbeschleunigung in [m/s^2]
% Inputdatei_4: Vektor zur Beschreibung der an Globalverformung beteiligten
%               Freiheitsgrade [-]
% ANMERKUNG: Dezimaltrennzeichen '.'

% Ausgabedateien:
% Outputdatei_1: Eingaben- und Ergebnisübersicht

%----------------------------- EINGABEBLOCK ---------------------------------
% Einlesen von Eingabedateien und Generierung der entsprechenden Vektoren
% und Matrizen

% Nachgiebigkeitsmatrix
H=dlmread('Inputdatei_1_Nachgiebigkeitsmatrix.txt');

% Massenmatrix
M=dlmread('Inputdatei_2_Massenmatrix.txt');

% Antwortspektrum
Spektrum=dlmread('Inputdatei_3_Antwortspektrum.txt');
T_S_acc=Spektrum(:,1);  % Vektor der Eigenschwingzeiten im Spektrum
S_acc=Spektrum(:,2);    % Vektor der Beschleunigungen

% Topologievektor
I=dlmread('Inputdatei_4_Topologievektor.txt');
%---------------------------------------------------------------------------
% Eingaben im Quellcode

xi=0.05;    % Dämpfungsmaß (konstant für alle Eigenschwingungsformen)
%---------------------------------------------------------------------------

%--------------------------- BERECHNUNGSBLOCK -------------------------------
% Anzahl der Freiheitsgrade
n=length(H);

% Invertierung der Nachgiebigkeitsmatrix ergibt die Steifigkeitsmatrix
K=H^-1;

% Lösung des Eigenwertproblems: dafür steht in Matlab der vordefinierte
% Befehl "eig" zur Verfügung. Matlab wählt mit diesem Befehl das günstigste
% Lösungsverfahren, sodass das Eigenwertproblem auch in dem Fall einer
```

```
% singulären Massenmatrix gelöst werden kann.
[A,EW]=eig(K,M); % A-Eigenformmatrix, EW-Eigenwertmatrix

% Definition eines Vektors für die ermittelten Eigenwerte
D_EW=diag(EW);

% Ermittlung von Eigenkreisfrequenzen
Omega=sqrt(D_EW);

% Gesamtanzahl der Eigenformen
n_ef=length(find(Omega~=Inf));

% Sortierung der Eigenwerte in aufsteigender Reihenfolge
[Omega, index]=sortrows(Omega);
A=A(:,index);

% Ermittlung von Eigenfrequenzen
Freq=Omega/(2*pi);

% Ermittlung von Eigenschwingzeiten
T=1./Freq;

% Ermittlung der Spektralbeschleunigung für alle Eigenschwingzeiten durch
% Interpolation
T_u=zeros(n,1);
T_o=zeros(n,1);
A_u=zeros(n,1);
A_o=zeros(n,1);
for i=1:1:length(T_S_acc)
    for j=1:1:n
        if T(j)>T_S_acc(i) && T(j)<T_S_acc(i+1)
            T_u(j)=T_S_acc(i);
            T_o(j)=T_S_acc(i+1);
            A_u(j)=S_acc(i);
            A_o(j)=S_acc(i+1);
        end
    end
end

S_a=zeros(n_ef,1);
for i=1:1:n_ef
    S_a(i)=A_u(i)+(A_o(i)-A_u(i))/(T_o(i)-T_u(i))*(T(i)-T_u(i));
end

% Nachfolgend sind drei Varianten für die Normierung der Eigenvektoren
% vorbereitet. Bitte unkommentieren Sie die gewünschte Variante bzw.
% kommentieren Sie durch Anwendung des Symbols "%" am Anfang jeder Zeile
% die übrigen zwei Abschnitte.

% Normierung der Eigenvektoren auf das betragsgrößte Element
% for j=1:1:n
%     if max(A(:,j))>abs(min(A(:,j)))
%        z1=max(A(:,j));
%        for i=1:1:n
%          A(i,j)=(A(i,j)/z1);
%        end
```

```
%     else
%        z1=min(A(:,j));
%        for i=1:1:n
%          A(i,j)=(A(i,j)/z1);
%        end
%     end
% end
%

% % Normierung der Eigenvektoren bezüglich der generalisierten Masse
% for j=1:1:n
%     z2=(A(:,j))'*M*(A(:,j));
%        for i=1:1:n
%          A(i,j)=(A(i,j)/sqrt(z2));
%        end
% end



% Normierung der Eigenvektoren (Wurzel-Ansatz)
z3=zeros(n,1);
z4=zeros(n,1);
for j=1:1:n
    for i=1:1:n
        z3(j)=z3(j)+(A(i,j))^2;
    end
    z4(j)=sqrt(z3(j));
end

for j=1:1:n
    for i=1:1:n
        A(i,j)=A(i,j)/z4(j);
    end
end

% Berechnung der modalen Steifigkeiten
K_mod=zeros(n_ef,1);
for i=1:1:n_ef
    K_mod(i)=A(:,i)'*K*A(:,i);
end

% Berechnung der modalen Massen
M_mod=zeros(n_ef,1);
for i=1:1:n_ef
    M_mod(i)=A(:,i)'*M*A(:,i);
end

% Berechnung der Beteiligungsfaktoren
L=zeros(n_ef,1);
for i=1:1:n_ef
    L(i)=A(:,i)'*M*I;
end

% Maximale Verschiebungen in den Eigenformen
u=zeros(n,n_ef);
for i=1:1:n_ef
    for j=1:1:n
        u(j,i)=A(j,i)*(L(i)/(M_mod(i)*Omega(i)^2))*S_a(i);
```

```
    end
end

% Erdbeben-Ersatzlasten
H_E_kj=zeros(n,n_ef);
for i=1:1:n
    for j=1:1:n_ef
        H_E_kj(i,j)=M(i,i)*A(i,j)*(L(j)/M_mod(j))*S_a(j);
    end
end

% Ermittlung der Ersatzmassen
me=zeros(n_ef,1);
for i=1:1:n_ef
    me(i)=L(i)^2/M_mod(i);
end

% Ersatzmassenfaktoren
eps=zeros(n_ef,1);
m_tot=sum(diag(M));
for i=1:1:n_ef
    eps(i)=me(i)/m_tot;
end

% Gesamtträgheitskraft
H_E_j=zeros(n_ef,1);
for i=1:1:n_ef
    H_E_j(i)=me(i)*S_a(i);
end

% Überlagerung der Verschiebungen nach der SRSS-Regel
z5=zeros(n,1);
u_srss=zeros(n,1);
for i=1:1:n
    for j=1:1:n_ef
        z5(i)=z5(i)+(u(i,j))^2;
    end
    u_srss(i)=sqrt(z5(i));
end

% Überlagerung der Verschiebungen nach der CQC-Regel

% Frequenzverhältnisse
r=zeros(n,n);
for j=1:1:n
    for k=1:1:n
        r(j,k)=Freq(k)/Freq(j);
    end
end

% Wechselwirkungsfaktor
rho=zeros(n_ef,n_ef);
for j=1:1:n_ef
    for k=1:1:n_ef
        rho(j,k)=(8*xi^2*(1+r(j,k))*r(j,k)^(3/2))/...
            ((1-r(j,k)^2)^2+4*xi^2*r(j,k)*(1+r(j,k))^2);
    end
```

```
end

% Überlagerung der Verschiebungen
z6=zeros(n,1);
u_cqc=zeros(n,1);
for i=1:1:n
    for j=1:1:n_ef
        for k=1:1:n_ef
        z6(i)=z6(i)+u(i,j)*rho(j,k)*u(i,k);
        end
    end
    u_cqc(i)=sqrt(z6(i));
end
%---------------------------------------------------------------------------

%---------------------------- AUSGABEBLOCK --------------------------------
% Ausgabe der Ergebnisse in eine Datei
EF=[Omega Freq T];
fid = fopen('Outputdatei_1_Allgemein.txt', 'w');
fprintf(fid,...
    '%s\n','C. Petersen, H. Werkle, Dynamik der Baukonstruktionen');
fprintf(fid,...
    '%s\n','2. Auflage, Springer Vieweg, Wiesbaden, 2018');
fprintf(fid,...
    '%s\n','Softwareentwicklung: Andrei Firus (andrei.firus@gmail.com)');
fprintf(fid,'%s\n','Programm ML_13_4: Eingaben- und Ergebnisuebersicht');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
    '%s\n','-----------------------------------------------------------');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
    '%s\n','EINGABEDATEN:');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Daempfungsmaß (für alle Eigenformen) [-]:');
fprintf(fid,'%d\n',xi);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Format= "%d \t ";
if n>2
    for ii=1:1:n-2
        Format=Format + '%d \t ';
    end
end
Format=Format + '%d\n';
fprintf(fid,'%s\n','Nachgiebigkeitsmatrix in [m/N] bzw. [1/Nm]');
for jj = 1:1:n
     fprintf(fid, Format, H(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Massenmatrix in [kg] bzw. [kgm^2]');
for jj = 1:1:n
     fprintf(fid, Format, M(jj,:));
 end
fprintf(fid, '%s\n', ' ');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Antwortspektrum: s. Dateiende');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Topologievektor [-]:');
for jj = 1:1:n
    Name=[num2str(jj) '.ter Freiheitsgrad:'];
    fprintf(fid, '%s \t %d\n', Name, I(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
    '%s\n','-------------------------------------------------------------');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
    '%s\n','ERGEBNISSE:');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Anzahl der Freiheitsgrade:');
fprintf(fid,'%d\n',n);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Steifigkeitsmatrix [N/m] bzw. [Nm]');
for jj = 1:1:n
     fprintf(fid, Format, K(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s \t %s \t %s\n','Eigenkreisfrequenzen [1/s]',...
    'Eigenfrequenzen [Hz]','Eigenschwingzeiten [s]');
for jj = 1:1:n_ef
     fprintf(fid, '%d \t %d \t %d\n', EF(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s %s\n','Spektralbeschleunigungen bei den',...
    'Eigenschwingzeiten [m/s^2]:');
for jj = 1:1:n_ef
     Name=[num2str(jj) '.te Eigenschwingzeit:'];
     fprintf(fid, '%s \t %d\n', Name, S_a(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Normierte Eigenformmatrix [-]');
for jj = 1:1:n
     fprintf(fid, Format, A(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Modale Steifigkeiten [N/m] bzw. [Nm]:');
for jj = 1:1:n_ef
    Name=[num2str(jj) '.te Eigenform:'];
    fprintf(fid, '%s \t %d\n', Name, K_mod(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Modale Massen in [kg] bzw. [kgm^2]:');
```

```matlab
for jj = 1:1:n_ef
    Name=[num2str(jj) '.te Eigenform:'];
    fprintf(fid, '%s \t %d\n', Name, M_mod(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Beteiligungsfaktoren in [kg] bzw. [kgm^2]:');
for jj = 1:1:n_ef
    Name=[num2str(jj) '.te Eigenform:'];
    fprintf(fid, '%s \t %d\n', Name, L(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Ersatzmassen in [kg] bzw. [kgm^2]:');
for jj = 1:1:n_ef
    Name=[num2str(jj) '.te Eigenform:'];
    fprintf(fid, '%s \t %d\n', Name, me(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Ersatzmassenfaktoren [-]:');
for jj = 1:1:n_ef
    Name=[num2str(jj) '.te Eigenform:'];
    fprintf(fid, '%s \t %d\n', Name, eps(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s %s\n','Erdbebenersatzlasten (spaltenweise für',...
    'jede Eigenform) [N]:');
Format= "%s \t %d \t ";
if n_ef>2
    for ii=1:1:n_ef-2
        Format=Format + '%d \t ';
    end
end
Format=Format + '%d\n';
for jj = 1:1:n
     Name=[num2str(jj) '.ter Freiheitsgrad:'];
     fprintf(fid, Format, Name, H_E_kj(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s %s\n','Verschiebungen, spaltenweise für',...
    'jede Eigenform [m]:');
Format= "%s \t %d \t ";
if n_ef>2
    for ii=1:1:n_ef-2
        Format=Format + '%d \t ';
    end
end
Format=Format + '%d\n';
for jj = 1:1:n
     Name=[num2str(jj) '.ter Freiheitsgrad:'];
     fprintf(fid, Format, Name, u(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Verschiebungen - SRSS Überlagerung [m]:');
```

```
for jj = 1:1:n
     Name=[num2str(jj) '.ter Freiheitsgrad:'];
     fprintf(fid, '%s \t %d\n',Name, u_srss(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Verschiebungen - CQC Überlagerung [m]:');
for jj = 1:1:n
     Name=[num2str(jj) '.ter Freiheitsgrad:'];
     fprintf(fid, '%s \t %d\n',Name, u_cqc(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Antwortspektrum:');
fprintf(fid,'%s \t %s\n','Eigenschwingzeit [s]',...
    'Spektralbeschleunigung [m/s^2]');
for jj = 1:1:length(S_acc)
     fprintf(fid, '%d \t %d\n', T_S_acc(jj),S_acc(jj));
 end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fclose(fid);
%--------------------------------------------------------------------------
```