

```

% C. Petersen, H. Werkle, Dynamik der Baukonstruktionen
% 2. Auflage, Springer Vieweg, Wiesbaden, 2018
%
% ML_07_2_Mehrfreiheitsgradschwinger_2: Schwingungen eines
% Mehrfreiheitsgradschwingers für vorgegebene Anfangsverschiebungen und
% -geschwindigkeiten (Modalanalyse; Massenmatrix und Nachgiebigkeitsmatrix)
%
% 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. kgm²]
% Inputdatei_3: Spalte 1: Anfangsverschiebungen in [m] bzw. [-]
%                Spalte 2: Anfangsgeschwindigkeiten in [m/s] bzw. [1/s]
% Inputdatei_4: Modale Dämpfungsmaße [-]
% ANMERKUNG: Dezimaltrennzeichen '.'
%
% Ausgabedateien:
% Outputdatei_1: Eingaben- und Ergebnisübersicht
% Outputdatei_2: Verschiebungszeitverläufe [m] bzw. [-]

%----- 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');

% Anfangswerte
Anfangswerte=dlmread('Inputdatei_3_Anfangswerte.txt');
y_0=Anfangswerte(:,1); % Anfangsverschiebungen
v_0=Anfangswerte(:,2); % Anfangsauslenkungen

% Modale Dämpfungsmaße
xi_mod=dlmread('Inputdatei_4_Modale_Daempfungsmasse.txt');
%-----
% Eingaben im Quellcode
t_ber=2.5; % Berechnungszeit [s]

dt=0.0005; % Berechnungszeitschritt [s]

n_ef=5; % Anzahl der bei der Berechnung berücksichtigten
        % Eigenformen

% Es werden die Schwingungsreaktionen von nur drei ausgewählten
% Freiheitsgraden geplottet. Bitte geben Sie die gewünschten
% Freiheitsgrade an:
plot_1_EF=1;
plot_2_EF=2;
plot_3_EF=3;
%-----

```

```

%----- BERECHNUNGSBLOCK -----
n=length(H); % Matrixdimension

% 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);

% 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;

% 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

```

```

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 Steifigkeitsmatrix
K_mod=A'*K*A;

% Berechnung der modalen Massenmatrix
M_mod=A'*M*A;

% Anfangsvektoren der Normalkoordinaten
ita_y0=zeros(n,1);
ita_v0=zeros(n,1);
for j=1:1:n
    ita_y0(j)=(A(:,j))'*M*y_0/M_mod(j,j);
    ita_v0(j)=(A(:,j))'*M*v_0/M_mod(j,j);
end

% Berechnungsparameter
T_positive=T(T(:,1)>0,:); % Anzahl der von Null verschiedenen
                        % Eigenschwingzeiten (falls M singular)
nt=ceil(t_ber/dt)+1; % Anzahl Berechnungszeitschritte
t_b=0:dt:dt*(nt-1); % Zeitvektor für die Berechnung

% Modale Schwingungsreaktionen (gespeichert in Matrixform)
ita_y=zeros(n,nt);
for j=1:1:length(T_positive)
    for i=1:1:nt
        ita_y(j,i)=exp(-xi_mod(j)*Omega(j)*t_b(i))*...
            ((xi_mod(j)*Omega(j)*ita_y0(j)+ita_v0(j))/...
            ((sqrt(1-xi_mod(j)^2))*Omega(j)))*...
            sin((sqrt(1-xi_mod(j)^2))*Omega(j)*t_b(i))+...
            ita_y0(j)*cos((sqrt(1-xi_mod(j)^2))*Omega(j)*t_b(i)));
    end
end

% Überlagerung der Schwingungsreaktionen der berücksichtigten Eigenformen
% für alle Freiheitsgrade
y=A(:,1:n_ef)*ita_y(1:n_ef,:);
%-----

%----- DARSTELLUNGSBLOCK -----
% Grafische Darstellung der Ergebnisse
name_fig1 = 'Schwingungsantworten';

```

[illegible]

```

fprintf(fid, '%s\n', 'Anzahl der beruecksichtigen Eigenformen [-]:');
fprintf(fid, '%d\n', n_ef);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Anfangsbedingungen (physikalische Koordinaten):');
fprintf(fid, '%s \t %s \t %s\n', 'Freiheitsgrad', ...
    'Anfangsversch. ([m], [-])', ...
    'Anfangsgeschw. ([m/s], [1/s])');
for jj = 1:1:n
    fprintf(fid, '%d \t %d \t %d\n', jj, Anfangswerte(jj,:));
end
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 [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', 'Modale Daempfungsmasse [-]');
for jj = 1:1:n
    Name=[num2str(jj) '.te Eigenform:'];
    fprintf(fid, '%s \t %d\n', Name, xi_mod(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', 'Anzahl der Berechnungszeitschritte:');
fprintf(fid, '%d\n', nt);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Steifigkeitsmatrix in [N/m] bzw. [Nm]');
for jj = 1:1:n
    fprintf(fid, Format, K(jj,:));
end

```

```

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
    fprintf(fid, '%d \t %d \t %d\n', EF(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 Steifigkeitsmatrix in [N/m] bzw. [Nm]');
for jj = 1:1:n
    fprintf(fid, Format, K_mod(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Modale Massenmatrix in [kg] bzw. [kgm^2]');
for jj = 1:1:n
    fprintf(fid, Format, M_mod(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Anfangsbedingungen (normalisierte Koordinaten):');
fprintf(fid, '%s \t %s \t %s\n', 'Freiheitsgrad', ...
    'Anfangsversch. ([m], [-])', ...
    'Anfangsgeschw. ([m/s], [1/s])');
for jj = 1:1:n
    fprintf(fid, '%d \t %d \t %d\n', jj, ita_y0(jj), ita_v0(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, ...
    '%s\n', '-----');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ergebnis_Verschiebungen=[t_b' y'];
fid = fopen('Outputdatei_2_Verschiebungen.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_07_2: Verschiebungszeitverlaeufe');
fprintf(fid, '%s\n', ' ');

Format_Header= "%s \t ";
if n>1
    for ii=1:1:n-1

```

```

        Format_Header=Format_Header + '%s \t ';
    end
end
Format_Header=Format_Header + '%s\n ';

Header=strings(1,n+1);
Header(1)='Zeit [s]';
for iii=1:1:n
    Header(iii+1)=strcat('Verschiebung ', {' '}, num2str(iii),...
        '. FG [m], [-]');
end
fprintf(fid,Format_Header,Header);
Format= "%d \t ";
if n>1
    for ii=1:1:n-1
        Format=Format + '%d \t ';
    end
end
Format=Format + '%d\n';
for jj = 1:1:length(Ergebnis_Verschiebungen)
    fprintf(fid, Format, Ergebnis_Verschiebungen(jj,:));
end
fclose(fid);
%-----

```