```
% C. Petersen, H. Werkle, Dynamik der Baukonstruktionen
% 2. Auflage, Springer Vieweg, Wiesbaden, 2018
%
% ML_07_4_Mehrfreiheitsgradschwinger_4: Schwingungen eines
% Mehrfreiheitsgradschwingers infolge Fußpunkterregung durch
% einen vorgegebenen Beschleunigungszeitverlauf (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: Beschleunigungszeitverlauf
%               Spalte 1 - Zeitvektor [s],
%               Spalte 2 - Beschleunigungsvektor [m/s^2] bzw. [1/s^2])
% Inputdatei_4: Vektor zur Beschreibung der von der Fußpunktbeschleunigung
%               betroffenen Freiheitsgrade [-]
% Inputdatei_5: 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');

% Beschleunigungszeitverlauf
Beschleunigung=dlmread('Inputdatei_3_Beschleunigungszeitverlauf.txt');
t_acc=Beschleunigung(:,1);  % Zeitvektor
acc=Beschleunigung(:,2);    % Beschleunigungsvektor

% Topologievektor
I=dlmread('Inputdatei_4_Topologievektor.txt');

% Modale Dämpfungsmaße
xi_mod=dlmread('Inputdatei_5_Modale_Daempfungsmasse.txt');
%--------------------------------------------------------------------------
% Eingaben im Quellcode
t_ber=15;          % Berechnungszeit [s]

dt=0.005;          % Berechnungszeitschritt [s]

n_ef=3;             % 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

nt=ceil(t_ber/dt)+1;      % Anzahl Berechnungszeitschritte

% 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;

% Interpolation des Beschleunigungsvektors mit Berücksichtigung des
% Berechnungszeitschrittes 'dt'
t_acc_int_0=0:dt:t_acc(length(acc));
acc_int_0 = interp1(t_acc,acc,t_acc_int_0,'linear');

% Erstellung eines Beschleunigungsvektors für die Berechnung. Sollte die
% Berechnungszeit größer als die Dauer des eingegebenen Zeitverlaufs sein,
% wird der Beschleunigungsvektor mit Nulleinträgen ergänzt.
acc_int=zeros(nt,1);
for i=1:1:nt
    if i<=length(acc_int_0)
        acc_int(i)=acc_int_0(i);
    else
        acc_int(i)=0;
    end
end
t_acc_int=0:dt:dt*(length(acc_int)-1);
% 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 Steifigkeitsmatrix
K_mod=A'*K*A;

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

% Gedämpfte Eigenkreisfrequenzen
Omega_d=zeros(n,1);
for i=1:1:n
    Omega_d(i)=sqrt(1-xi_mod(i)^2)*Omega(i);
end

% Erstellung der Kraftvektoren
```

```
Kraft=zeros(n,length(acc_int));
z5=-M*I;
for i=1:1:length(acc_int)
    for j=1:1:n
         Kraft(j,i)=z5(j)*acc_int(i);
    end
end

% Berechnung der generalisierten Kräfte
modF=A'*Kraft;

f=zeros(n,length(acc_int));
for i=1:1:length(acc_int)
    for j=1:1:n
        f(j,i)=modF(j,i)/M_mod(j,j);
    end
end

% Berechnung der Schwingreaktion mittels Übertragungsverfahren

% Breite der Zeitintervalle: Einwirkung
n_int=length(Kraft)-1;   % Anzahl der Berechnungsintervalle
tf=zeros(n,1);
for i=1:1:n_int
    tf(i)=t_acc_int(i+1)-t_acc_int(i);
end

% Steigung der Geraden in jedem Intervall
p=zeros(n,n_int);
for i=1:1:n_int
    for j=1:1:n
          p(j,i)=(f(j,i+1)-f(j,i))/(tf(i));
    end
end

% Zeitpunkte für die Berechnung
ts=zeros(n_int,1);
ts(1)=tf(1);
for i=1:1:n_int-1
    ts(i+1)=ts(i)+tf(i+1);
end

% Anfangswerte am Zeitpunkt 0 in Normalkoordinaten. Sie sind im Regelfall
% Null. Anderenfalls sind die folgenden Vektoren entsprechend zu
% modifizieren.
nu_0=zeros(n,1);
nup_0=zeros(n,1);

% Freiwerte a und b
a=zeros(n,1);
b=zeros(n,1);

nu=zeros(n,n_int);
nup=zeros(n,n_int);

% Definition der endgültigen Ergebnisvektoren (normalisierte Koordinaten)
```

```
nu_f=zeros(n,n_int+1);
nup_f=zeros(n,n_int+1);
for i=1:1:n
    nu_f(i,1)=nu_0(i);
    nup_f(i,1)=nup_0(i);
end

% Berechnung der Schwingungsantwort mittels Übertragungsverfahren
for j=1:1:n
    for i=1:1:n_int
        t=tf(i);

        a(j)=xi_mod(j)*(Omega(j)/Omega_d(j))*nu_0(j)+...
        (1/Omega_d(j))*nup_0(j)+xi_mod(j)*f(j,i)/(Omega(j)*Omega_d(j))+...
        p(j)*(1-2*xi_mod(j)^2)/(Omega(j)^2*Omega_d(j));

        b(j)=nu_0(j)+(f(j,i)/Omega(j)^2)-(2*xi_mod(j)*p(j))/Omega(j)^3;

        nu(j,i)=exp(-xi_mod(j)*Omega(j)*t)*(a(j)*sin(Omega_d(j)*t)+...
            b(j)*cos(Omega_d(j)*t))-f(j,i)/Omega(j)^2+2*xi_mod(j)*p(j)/...
            Omega(j)^3-p(j)*t/(Omega(j)^2);

        nup(j,i)=exp(-xi_mod(j)*Omega(j)*t)*(Omega_d(j)*(a(j)*...
            cos(Omega_d(j)*t)-b(j)*sin(Omega_d(j)*t))-xi_mod(j)*...
            Omega(j)*(a(j)*sin(Omega_d(j)*t)+...
            b(j)*cos(Omega_d(j)*t)))-p(j)/Omega(j)^2;

        nu_0(j)=nu(j,i);
        nup_0(j)=nup(j,i);
    end
end
for i=1:1:n_int
    for j=1:1:n
    nu_f(j,i+1)=nu(j,i);
    nup_f(j,i+1)=nup(j,i);
    end
end

% Überlagerung der Schwingungsreaktionen der berücksichtigten Eigenformen
% für alle Freiheitsgrade
u=A(:,1:n_ef)*nu_f(1:n_ef,:);
up=A(:,1:n_ef)*nup_f(1:n_ef,:);

% Extremwerte
umax=zeros(n,1);
umin=zeros(n,1);
upmax=zeros(n,1);
upmin=zeros(n,1);
for i=1:1:n
    umax(i)=max(u(i,:));
    umin(i)=min(u(i,:));
    upmax(i)=max(up(i,:));
    upmin(i)=min(up(i,:));
end
%-------------------------------------------------------------------------
```

```matlab
%------------------------- DARSTELLUNGSBLOCK ------------------------------
% Grafische Darstellung der Ergebnisse
Zeit_plot=zeros(length(nu_f),1);
Zeit_plot(1)=0;
for i=2:1:length(nu_f)
    Zeit_plot(i)=ts(i-1);
end

name_fig1 = 'Beschleunigungsverlauf und Schwingungsantworten';
fig1=figure('Name',name_fig1,'NumberTitle','off');
set(fig1,'Position',[200 300 800 500]);

subplot(2,2,1)
plot(t_acc_int,acc_int,'MarkerSize',3);
title1 = 'Fußpunktbeschleunigung';
title(title1);
xlabel('Zeit [s]');
ylabel('Beschleunigung [m/s^2]');
grid on; zoom on;

subplot(2,2,2)
plot(Zeit_plot,u(plot_1_EF,:),'MarkerSize',3);
title1 = [num2str(plot_1_EF),'. Freiheitsgrad'];
title(title1);
xlabel('Zeit [s]');
ylabel('Verschiebung [m]');
grid on; zoom on;

subplot(2,2,3)
plot(Zeit_plot,u(plot_2_EF,:),'MarkerSize',3);
title2 = [num2str(plot_2_EF),'. Freiheitsgrad'];
title(title2);
xlabel('Zeit [s]');
ylabel('Verschiebung [m]');
grid on; zoom on;

subplot(2,2,4)
plot(Zeit_plot,u(plot_3_EF,:),'MarkerSize',3);
title3 = [num2str(plot_3_EF),'. Freiheitsgrad'];
title(title3);
xlabel('Zeit [s]');
ylabel('Verschiebung [m]');
grid on; zoom on;
%--------------------------------------------------------------------------

%---------------------------- 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_07_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','Berechnungszeit [s]:');
fprintf(fid,'%.2f\n',t_ber);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Berechnungszeitschritt [s]:');
fprintf(fid,'%d\n',dt);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Anzahl der beruecksichtigen Eigenformen [-]:');
fprintf(fid,'%d\n',n_ef);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Fusspunktbeschleunigung:');
fprintf(fid,'%s \t %s\n','Zeit [s]','Beschleunigung [m/s^2]');
for jj=1:1:n
    fprintf(fid,'%d \t %d\n',t_acc(jj),acc(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','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','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
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','Maximale Verschiebungen der Freiheitsgrade [m]:');
for jj = 1:1:n
     Name=[num2str(jj) '.ter FG'];
     fprintf(fid, '%s \t %d\n', Name, umax(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Minimale Verschiebungen der Freiheitsgrade [m]:');
```

```
for jj = 1:1:n
     Name=[num2str(jj) '.ter FG'];
     fprintf(fid, '%s \t %d\n', Name, umin(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Minimale Geschwindigkeiten der Freiheitsgrade [m/s]:');
for jj = 1:1:n
     Name=[num2str(jj) '.ter FG'];
     fprintf(fid, '%s \t %d\n', Name, upmax(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Minimale Geschwindigkeiten der Freiheitsgrade [m/s]:');
for jj = 1:1:n
     Name=[num2str(jj) '.ter FG'];
     fprintf(fid, '%s \t %d\n', Name, upmin(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
    '%s\n','------------------------------------------------------------');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ergebnis_Verschiebungen=[Zeit_plot u'];
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_4: 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);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ergebnis_Geschwindigkeiten=[Zeit_plot up'];
fid = fopen('Outputdatei_3_Geschwindigkeiten.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_4: Geschwindigkeitszeitverlaeufe');
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('Geschwindigkeit ', {' '}, num2str(iii),...
        '. FG [m/s], [1/s]');
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_Geschwindigkeiten)
    fprintf(fid, Format, Ergebnis_Geschwindigkeiten(jj,:));
end
fclose(fid);
%----------------------------------------------------------------------------
```