

```

% C. Petersen, H. Werkle, Dynamik der Baukonstruktionen
% 2. Auflage, Springer Vieweg, Wiesbaden, 2018
%
% ML_29_1_Mehrfreiheitsgradschwinger_6: Schwingungen eines
% Mehrfreiheitsgradschwingers infolge Anfangsverschiebungen und
% -geschwindigkeiten sowie beliebiger Erregerkräfte (Numerische
% Integration mit Newmark-Verfahren; Massenmatrix, Dämpfungsmatrix und
% Steifigkeitsmatrix)
%
% Version 1.0, April 2018
% Softwareentwicklung:
% Andrei Firus, M.Eng (andrei.firus@gmail.com)

% Aufbau Eingabedateien:
% Inputdatei_1: Steifigkeitsmatrix [N/m bzw. Nm]
% Inputdatei_2: Massenmatrix [kg, bzw. kgm²]
% Inputdatei_3: Dämpfungsmatrix [Ns/m, bzw. Nms]
% Inputdatei_4: Erregerkraftverlauf:
%           Spalte 1 - Zeitvektor [s],
%           ab Spalte 2 bis n+1 - Kraftvektor [N] bzw. [Nm]
% Inputdatei_5: Spalte 1: Anfangsverschiebungen in [m] bzw. [-]
%           Spalte 2: Anfangsgeschwindigkeiten in [m/s] bzw. [1/s]
% HINWEIS: Alle sich bei der Berechnung ergebenden Größen sind
%           dimensionsecht. Bei der Eingabe können daher auch andere,
%           konsistente Einheiten gewählt werden, wie z.B. [t] für die Masse
%           und [kN] für die Kraft.
% ANMERKUNG: Dezimaltrennzeichen '.'

% Ausgabedateien:
% Outputdatei_1: Eingaben- und Ergebnisübersicht
% Outputdatei_2: Verschiebungszeitverläufe [m] bzw. [-]
% Outputdatei_3: Geschwindigkeitszeitverläufe [m/s] bzw. [1/s]
% Outputdatei_4: Beschleunigungszeitverläufe [m/s²] bzw. [1/s²]

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

% Steifigkeitsmatrix
K=dlmread('Inputdatei_1_Steifigkeitsmatrix.txt');

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

% Dämpfungsmatrix
D=dlmread('Inputdatei_3_Daempfungsmatrix.txt');

% Kraftverlauf
Kraftverlauf=dlmread('Inputdatei_4_Erregerkraefte.txt');
t_kraft=Kraftverlauf(:,1); % Zeitvektor der Kraft
Kraft=Kraftverlauf(1:size(Kraftverlauf,1), 2:end)'; % Matrix der Kraftwerte

% Anfangswerte
Anfangswerte=dlmread('Inputdatei_5_Anfangswerte.txt');
y_0=Anfangswerte(:,1); % Anfangsverschiebungen
v_0=Anfangswerte(:,2); % Anfangsgeschwindigkeiten
%-----

```

```

% Eingaben im Quellcode:

% Berechnungszeitschritt [s]
dt=0.0005;

% Berechnungszeit [s]
t_ber=10;

% Integrationsparameter für das Newmark-Verfahren
alpha=0.5;
beta=0.25;

% 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=2;
%-----

%----- BERECHNUNGSBLOCK -----
% Nachgiebigkeitsmatrix
H=K^-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 der Eigenkreisfrequenzen
Omega=sqrt(D_EW);

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

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

% Ermittlung der Eigenschwingzeiten
T=1./Freq;

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

% Anpassung des Kraftvektors, wenn die Berechnungszeit größer als die
% Lasteinwirkungszeit ist (Kraftvektoren werden mit Nulleinträgen ergänzt)
if t_kraft(size(Kraft,2))<t_ber
    Kraft(:,end+1)=zeros(size(Kraft,1),1);
    t_kraft(end+1)=t_kraft(end)+(t_kraft(end)-t_kraft(end-1));
    Kraft(:,end+1)=zeros(size(Kraft,1),1);
    t_kraft(end+1)=t_ber;

```

```

end

% Interpolation des Kraftvektors mit Berücksichtigung des
% Berechnungszeitschrittes 'dt'
n=size(K,1); % Matrixdimension
Kraft_int=zeros(n,nt);
for j=1:1:n
    t_kraft_int=0:dt:dt*(nt-1);
    Kraft_int_h = interp1(t_kraft,Kraft(j,:),t_kraft_int,'linear');
    for i=1:1:nt
        Kraft_int(j,i)=Kraft_int_h(i);
    end
end
% ANMERKUNG: Je nach Anwendungsziel kann die Interpolationsmethode geändert
% werden. Für weitere Interpolationsmethoden siehe Matlab-Hilfe zur
% Funktion "interp1" (Befehl 'help interp1' in Command Window)

% ANMERKUNG: Bei Berechnung freier Schwingungen mit Anfangsverschiebung
% bzw. -geschwindigkeit soll die nächste Zeile unkommentiert werden.
% Dadurch werden alle Einträge der Kraftvektoren zu Null gesetzt.
% Alternativ kann die Kraft in der Eingabedatei zu Null gesetzt werden.

Kraft_int=zeros(n,nt);

% 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 mit dem 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

% Definition der Ergebnisvektoren
y=zeros(n,nt); % Verschiebungsmatrix
v=zeros(n,nt); % Geschwindigkeitsmatrix
a=zeros(n,nt); % Beschleunigungsmatrix

% Berücksichtigung der Anfangsbedingungen
for i=1:1:n
    y(i,1)=y_0(i);
    v(i,1)=v_0(i);
end
% In der Regel sind alle Freiheitsgrade mit einer Masse belegt. Sollten
% jedoch Nulleinträge auf der Diagonale der Massenmatrix auftreten, wird
% diese Singulär und somit nicht invertierbar. In diesem Fall kann die
% Pseudoinverse (Matlab-Befehl: "pinv") genutzt oder die Nulleinträge
% durch sehr kleine Zahlen, z.B. 10^-20 ersetzt werden.
count_m0=0; % Zähler der Nulleinträge auf der Diagonale der Massenmatrix
for ii=1:1:size(M,1)
    if M(ii,ii)==0
        count_m0=count_m0+1;
    end
end
if count_m0==0
    a(:,1)=(-M^-1)*(D*v(:,1)+K*y(:,1)-Kraft_int(:,1));
else
    a(:,1)=-pinv(M)*(D*v(:,1)+K*y(:,1)-Kraft_int(:,1));
end

% Berechnung der Schwingreaktion mittels Newmark-Verfahren
for i=2:1:nt
    a_h=((1/beta)*M)+(alpha/beta)*D*dt+K*dt^2;
    b_h=((1/beta)*M+(alpha/beta)*D*dt)*y(:,i-1)+((1/beta)*M+...
        (alpha/beta-1)*D*dt)*dt*v(:,i-1)+((1/(2*beta)-1)*M+...
        (alpha/(2*beta)-1)*D*dt)*dt^2*a(:,i-1)+Kraft_int(:,i)*dt^2;
    y(:,i)=a_h^-1*b_h;
    v(:,i)=(alpha/(beta*dt))*(y(:,i)-y(:,i-1))-((alpha/beta)-1)*...
        v(:,i-1)-(alpha/(2*beta)-1)*dt*a(:,i-1);
    a(:,i)=(1/(beta*dt^2))*(y(:,i)-y(:,i-1))-1/(beta*dt)*v(:,i-1)-...
        (1/(2*beta)-1)*a(:,i-1);
end
%-----

%----- DARSTELLUNGSBLOCK -----
% Grafische Darstellung der Ergebnisse
name_fig1 = 'Schwingungsantworten';
fig1=figure('Name',name_fig1,'NumberTitle','off');
set(fig1,'Position',[200 300 800 500]);

```

```

subplot(3,3,1)
plot(t_kraft_int,y(plot_1_EF,:), 'MarkerSize',3);
title1 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title1);
xlabel('Zeit [s]');
ylabel('Verschiebung [m] bzw. [-]');
grid on; zoom on;

```

```

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

```

```

subplot(3,3,3)
plot(t_kraft_int,y(plot_3_EF,:), 'MarkerSize',3);
title3 = [num2str(plot_3_EF), '. Freiheitsgrad'];
title(title3);
xlabel('Zeit [s]');
ylabel('Verschiebung [m] bzw. [-]');
grid on; zoom on;

```

```

subplot(3,3,4)
plot(t_kraft_int,v(plot_1_EF,:), 'MarkerSize',3);
title1 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title1);
xlabel('Zeit [s]');
ylabel('Geschwindigkeit [m/s] bzw. [1/s]');
grid on; zoom on;

```

```

subplot(3,3,5)
plot(t_kraft_int,v(plot_2_EF,:), 'MarkerSize',3);
title2 = [num2str(plot_2_EF), '. Freiheitsgrad'];
title(title2);
xlabel('Zeit [s]');
ylabel('Geschwindigkeit [m/s] bzw. [1/s]');
grid on; zoom on;

```

```

subplot(3,3,6)
plot(t_kraft_int,v(plot_3_EF,:), 'MarkerSize',3);
title3 = [num2str(plot_3_EF), '. Freiheitsgrad'];
title(title3);
xlabel('Zeit [s]');
ylabel('Geschwindigkeit [m/s] bzw. [1/s]');
grid on; zoom on;

```

```

subplot(3,3,7)
plot(t_kraft_int,a(plot_1_EF,:), 'MarkerSize',3);
title1 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title1);
xlabel('Zeit [s]');
ylabel('Beschleunigung [m/s^2] bzw. [1/s^2]');
grid on; zoom on;

```

```

subplot(3,3,8)
plot(t_kraft_int,a(plot_2_EF,:), 'MarkerSize',3);
title2 = [num2str(plot_2_EF), '. Freiheitsgrad'];
title(title2);
xlabel('Zeit [s]');
ylabel('Beschleunigung [m/s^2] bzw. [1/s^2]');
grid on; zoom on;

subplot(3,3,9)
plot(t_kraft_int,a(plot_3_EF,:), 'MarkerSize',3);
title3 = [num2str(plot_3_EF), '. Freiheitsgrad'];
title(title3);
xlabel('Zeit [s]');
ylabel('Beschleunigung [m/s^2] bzw. [1/s^2]');
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_29_1: Eingaben- und Ergebnisseuebersicht');
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','Anzahl der Freiheitsgrade:');
fprintf(fid,'%d\n',n);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Berechnungszeitschritt [s]:');
fprintf(fid,'%d\n',dt);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Berechnungszeit [s]:');
fprintf(fid,'%0.2f\n',t_ber);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Integrationsparameter fuer das Newmark-Verfahren:');
fprintf(fid,'%s \t %0.4f\n','alpha=',alpha);
fprintf(fid,'%s \t %0.4f\n','beta=',beta);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Anfangsbedingungen:');
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', 'Steifigkeitsmatrix in [N/m] bzw. [Nm]');
for jj = 1:1:n
    fprintf(fid, Format, K(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', 'Daempfungsmatrix in [Ns/m] bzw. [Nms]');
for jj = 1:1:n
    fprintf(fid, Format, D(jj,:));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Kraftzeitverlaeuft: siehe Dateiende');
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 Berechnungsschritte:');
fprintf(fid, '%d\n', nt);
fprintf(fid, '%s\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 \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', ' ');

```

[illegible]


```

Ergebnis_Verschiebungen=[t_kraft_int' 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_29_1: 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=[t_kraft_int' v'];
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_29_1: 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);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ergebnis_Beschleunigungen=[t_kraft_int' a'];
fid = fopen('Outputdatei_4_Beschleunigungen.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_29_1: Beschleunigungszeitverlaeufe');
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('Beschleunigung ', {' '}, num2str(iii),...
        '. FG [m/s^2], [1/s^2]');
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_Beschleunigungen)
    fprintf(fid, Format, Ergebnis_Beschleunigungen(jj,:));
end
fclose(fid);
%-----

```