

```

% C. Petersen, H. Werkle, Dynamik der Baukonstruktionen
% 2. Auflage, Springer Vieweg, Wiesbaden, 2018
%
% ML_29_2_Mehrfreiheitsgradschwinger_7: Schwingungen eines
% Mehrfreiheitsgradschwingers infolge Fußpunkterregung (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: Beschleunigungszeitverlauf
%           Spalte 1: Zeitvektor [s],
%           Spalte 2: Fußpunktbeschleunigung [m/s2] bzw. [1/s2]
% Inputdatei_5: Vektor zur Beschreibung der von der Fußpunktbeschleunigung
%           betroffenen Freiheitsgrade [-]
% Inputdatei_6: 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]
% Outputdatei_5: Zeitverläufe der Absolutbeschleunigung [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_Dämpfungsmatrix.txt');

% Beschleunigungszeitverlauf
Beschleunigung=dlmread('Inputdatei_4_Beschleunigungszeitverlauf.txt');
t_acc=Beschleunigung(:,1); % Zeitvektor
acc=Beschleunigung(:,2); % Beschleunigungsvektor

I=dlmread('Inputdatei_5_Topologievektor.txt'); % Topologievektor

```

```

% Anfangswerte
Anfangswerte=dlmread('Inputdatei_6_Anfangswerte.txt');
y_0=Anfangswerte(:,1); % Anfangsverschiebungen
v_0=Anfangswerte(:,2); % Anfangsgeschwindigkeiten
%-----

% Eingaben im Quellcode

% Berechnungszeitschritt
dt=0.005;

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

% Gesamtzeit der Berechnung [s]
t_ber=15;

% Grafisch darzustellender Freiheitsgrad
plot_1_EF=1;
%-----

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

% 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 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;

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

% Interpolation des Beschleunigungsvektors mit Berücksichtigung des

```

```

% Berechnungszeitschrittes 'dt'
t_acc_int_0=0:dt:t_acc(length(acc))-dt;
acc_int_0 = interp1(t_acc,acc,t_acc_int_0,'linear');
% ANMERKUNG: Je nach Anwendungsziel kann die Interpolationsmethode geändert
% werden. Für weitere Interpolationsmethoden siehe Matlab-Hilfe zur
% Funktion "interp1" (Befehl 'help interp1' in Command Window)

% 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

% Erstellung der Kraftvektoren
Kraft=zeros(n,length(acc_int));
z5=-M*I;
for k=1:1:length(acc_int)
    for j=1:1:n
        Kraft(j,k)=z5(j)*acc_int(k);
    end
end

% Definition der Ergebnisvektoren
y=zeros(n,nt); % Verschiebungsmatrix
v=zeros(n,nt); % Geschwindigkeitsmatrix
a=zeros(n,nt); % Beschleunigungsmatrix
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 Singular 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(:,1));
else
    a(:,1)=-pinv(M)*(D*v(:,1)+K*y(:,1)-Kraft(:,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(:,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

% Ermittlung der absoluten Beschleunigung

```

```

a_abs=zeros(n,nt);
for i=1:1:n
    for j=1:1:nt
        a_abs(i,j)=a(i,j)+acc_int(j);
    end
end

% Extremwerte
ymax=zeros(n,1);
vmax=zeros(n,1);
amax=zeros(n,1);
a_absmax=zeros(n,1);
for i=1:1:n
    ymax(i)=max(abs(y(i,:)));
    vmax(i)=max(abs(v(i,:)));
    amax(i)=max(abs(a(i,:)));
    a_absmax(i)=max(abs(a_abs(i,:)));
end
%-----

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

subplot(2,2,1)
plot(t_acc_int,y(plot_1_EF,:), 'MarkerSize',3);
title1 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title1);
xlabel('Zeit [s]');
ylabel('Verschiebung [m]');
grid on; zoom on;

subplot(2,2,2)
plot(t_acc_int,v(plot_1_EF,:), 'MarkerSize',3);
title2 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title2);
xlabel('Zeit [s]');
ylabel('Geschwindigkeit [m/s]');
grid on; zoom on;

subplot(2,2,3)
plot(t_acc_int,a(plot_1_EF,:), 'MarkerSize',3);
title3 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title3);
xlabel('Zeit [s]');
ylabel('Relativbeschleunigung [m/s^2]');
grid on; zoom on;

subplot(2,2,4)
plot(t_acc_int,a_abs(plot_1_EF,:), 'MarkerSize',3);
title1 = [num2str(plot_1_EF), '. Freiheitsgrad'];
title(title1);
xlabel('Zeit [s]');
ylabel('Absolutbeschleunigung [m/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_2: 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','Berechnungsschritt [s]:');
fprintf(fid,'%d\n',dt);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,'%s\n','Fusspunktbeschleunigung: s. Dateiende');
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 [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', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Format=Format + '%d\n';
fprintf(fid,'%s\n','Dämpfungsmatrix [Ns/m] bzw. [Nms]:');
for jj = 1:1:n
    fprintf(fid, Format, D(jj,:));
end

```

[illegible]

```

fprintf(fid, '%s %s\n', 'Maximale Verschiebungen der Freiheitsgrade', ...
    '(Betrag) [m]:');
for jj = 1:1:n
    Name=[num2str(jj) '.ter FG'];
    fprintf(fid, '%s \t %d\n', Name, ymax(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s %s\n', 'Maximale Geschwindigkeiten der Freiheitsgrade', ...
    '(Betrag) [m/s]:');
for jj = 1:1:n
    Name=[num2str(jj) '.ter FG'];
    fprintf(fid, '%s \t %d\n', Name, vmax(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s %s\n', 'Maximale Beschleunigungen der Freiheitsgrade', ...
    '(Betrag) [m/s^2]:');
for jj = 1:1:n
    Name=[num2str(jj) '.ter FG'];
    fprintf(fid, '%s \t %d\n', Name, amax(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s %s\n', 'Maximale Absolutbeschleunigungen der', ...
    'Freiheitsgrade (Betrag) [m/s^2]:');
for jj = 1:1:n
    Name=[num2str(jj) '.ter FG'];
    fprintf(fid, '%s \t %d\n', Name, a_absmax(jj));
end
fprintf(fid, '%s\n', ' ');
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:length(acc)
    fprintf(fid, '%d \t %d\n', t_acc(jj), acc(jj));
end
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ergebnis_Verschiebungen=[t_acc_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_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

```



```

        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_acc_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_2: Geschwindigkeitszeitverlaeufe');
    fprintf(fid, '%s\n', ' ');
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    fprintf(fid,...
        '%s\n','-----');
    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], [-]');
    end
    fprintf(fid,Format_Header,Header);
    Format= "%d \t ";
    if n>1
        for ii=1:1:n-1
            Format=Format + '%d \t ';
        end
    end

```

```

        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_acc_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_2: Beschleunigungszeitverlaeufe');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
        '%s\n','-----');
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], [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);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ergebnis_Beschleunigungen_abs=[t_acc_int' a_abs'];
fid = fopen('Outputdatei_5_Beschleunigungen_abs.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_2: Beschleunigungszeitverlaeufe (abs.)');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid,...
    '%s\n','-----');
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('abs. Beschleunigung ', {' '}, num2str(iii),...
        '. FG [m/s], [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_abs)
    fprintf(fid, Format, Ergebnis_Beschleunigungen_abs(jj,:));
end
fclose(fid);
%-----

```