

```

% C. Petersen, H. Werkle, Dynamik der Baukonstruktionen
% 2. Auflage, Springer Vieweg, Wiesbaden, 2018
%
% ML_19_1_Einfeldtraeger_personenerregte_Schwingungen: Personenerregte
% Schwingungen eines Einfeldträgers
%
% Version 1.0, April 2018
% Softwareentwicklung:
% Andrei Firus, M.Eng (andrei.firus@gmail.com)

% Aufbau Eingabedateien:
% Inputdatei_1: Lastdefinition durch Fourier-Reihe
%               - Spalte 1: Fourierkoeffizienten [-]
%               - Spalte 2: Phasenverschiebungen [-] (Zeile "i" entspricht
%                 der Phase der "i"-ten Harmonischen gegenüber der ersten
%                 Harmonischen)
% ANMERKUNG: Dezimaltrennzeichen '.'

% Ausgabedateien:
% Outputdatei_1: Eingaben- und Ergebnisübersicht
% Outputdatei_2: Zeitverläufe der Bewegungsgrößen an dem Ausgabepunkt:
%               Verschiebung [m], Geschwindigkeit [m/s] und
%               Beschleunigung [m/s^2]

%----- EINGABEBLOCK -----
% Einlesen von Eingabedateien und Generierung der entsprechenden Vektoren
% und Matrizen
Kraftdefinition=dlmread('Inputdatei_1.txt');
% Fourierkoeffizienten
alpha_f=Kraftdefinition(:,1);
% Phasenwinkel
phi=Kraftdefinition(:,2);
%-----

% Eingaben im Quellcode

% Strukturdefinition
l=28.4; % Trägerlänge [m]
mue=3100; % Masse pro Längeneinheit [kg/m]
xi=0.01; % Dämpfungsmaß [-]
I=0.02340523; % Flächenträgheitsmoment [m^4]
E=2.1*10^11; % Elastizitätsmodul [N/m^2]

% Lastdefinition
G=700; % Personengewicht [N]
fs=2.45; % Schrittfrequenz [Hz]
ls=0.9; % Schrittlänge [m]
n_p=2; % Anzahl Personen [-]

% Berechnungsparameter
a_tmax=1.5; % Faktor, der die Gesamtzeit der Berechnung
            % steuert. Für a_tmax=1 ist die
            % Berechnungsdauer gleich der
            % Überquerungszeit des Trägers.

ndt=10; % Anzahl der Unterteilungen der kleinsten
        % Eigenschwingzeit zwecks Definition eines
        % geeigneten Berechnungszeitschritts

```

```

x0=14.2; % Stelle der Ausgabe [m]

n=5; % Anzahl der berücksichtigten Eigenformen
%-----

%----- BERECHNUNGSBLOCK -----
% Berechnung weiterer System- und Berechnungsparameter
Freq=zeros(n,1); % Eigenfrequenzen
Omega=zeros(n,1); % Eigenkreisfrequenzen
T=zeros(n,1); % Eigenschwingzeiten

for i=1:1:n
    Freq(i)=((i*pi)^2/(2*pi*l^2))*sqrt(E*I/mue);
    Omega(i)=2*pi*Freq(i);
    T(i)=1/Freq(i);
end

m=zeros(n,1); % Vektor der modalen Massen
k=zeros(n,1); % Vektor der modalen Steifigkeiten
d=zeros(n,1); % Vektor der modalen Dämpferkonstanten

for i=1:1:n
    m(i)=0.5*mue*l;
    k(i)=Omega(i)^2*m(i);
    d(i)=2*xi*sqrt(k(i)*m(i));
end

dt=min(T)/ndt; % Berechnungszeitschritt
vs=fs*ls; % Schrittgeschwindigkeit
T_u=l/vs; % Zeit zur Überquerung des Trägers
t_ber=T_u*a_tmax; % Dauer der Berechnung
nt=ceil(t_ber/dt)+1; % Anzahl der Zeitschritte

% Erstellung des Zeitvektors
t=0:dt:dt*(nt-1);

% Erstellung der Matrix der Eigenformen
x=0:vs*dt:l;
eig_f=zeros(n,length(x));
for j=1:1:n
    for i=1:1:length(x)
        eig_f(j,i)=sin(j*x(i)*pi/l);
    end
end

% Erstellung einer Matrix der Eigenformen für die Berechnung (Ergänzung der
% oben ermittelten Eigenformmatrix mit Nullelementen unter Berücksichtigung
% des Faktors a_tmax für die Gesamtzeit der Berechnung).
x_ber=0:vs*dt:l*a_tmax+vs*dt;
ef=zeros(n,length(x_ber));
for j=1:1:n
    for i=1:1:length(x_ber)
        if i<=length(x)
            ef(j,i)=eig_f(j,i);
        else

```

```

        ef(j,i)=0;
    end
end
end

% Eigenformen am Ausgabepunkt
phi_a=zeros(n,1);
for i=1:1:n
    phi_a(i)=sin(i*pi*x0/l);
end

% Erstellung des Kraftvektors
h=zeros(length(alpha_f),length(t)); % Matrix für alle Kraftharmonischen
kraft_harm=zeros(1,length(t)); % Überlagerung aller Kraftharmonischen
Kraft=zeros(1,length(t)); % Gesamtkraft nach dem Fourier-Ansatz

for i=1:1:length(alpha_f)
    for j=1:1:length(t)
        if t(j)<T_u
            h(i,j)=alpha_f(i)*sin(2*pi*fs*i*t(j)-phi(i));
            kraft_harm(1,j)=kraft_harm(1,j)+h(i,j);
        else
            kraft_harm(1,j)=0;
        end
    end
end

for j=1:1:length(t)
    Kraft(j)=G*(1+kraft_harm(j));
end

% Berechnung der Schwingungsantwort mittels Newmark-Verfahren

% Integrationsparameter
alpha=0.5;
beta=0.25;

% Berechnungsvorschrift
ita_y=zeros(n,length(t));
ita_v=zeros(n,length(t));
ita_a=zeros(n,length(t));
for j=1:1:n
    for i=2:1:nt
        a_h=((1/beta)*m(j))+(alpha/beta)*d(j)*dt+k(j)*dt^2;
        b_h=((1/beta)*m(j)+(alpha/beta)*d(j)*dt)*ita_y(j,i-1)+((1/beta)*...
            m(j)+(alpha/beta-1)*d(j)*dt)*dt*ita_v(j,i-1)+...
            ((1/(2*beta)-1)*m(j)+(alpha/(2*beta)-1)*d(j)*dt)*dt^2*...
            ita_a(j,i-1)+Kraft(i)*ef(j,i)*dt^2;
        ita_y(j,i)=a_h^-1*b_h;
        ita_v(j,i)=(alpha/(beta*dt))*(ita_y(j,i)-ita_y(j,i-1))-...
            ((alpha/beta)-1)*ita_v(j,i-1)-(alpha/(2*beta)-1)*dt*...
            ita_a(j,i-1);
        ita_a(j,i)=(1/(beta*dt^2))*(ita_y(j,i)-ita_y(j,i-1))-...
            1/(beta*dt)*ita_v(j,i-1)-(1/(2*beta)-1)*ita_a(j,i-1);
    end
end
end

```

```

% Überlagerung aller Eigenformen
y=zeros(1,length(t));      % Vektor der Gesamtverschiebung
v=zeros(1,length(t));      % Vektor der Gesamtgeschwindigkeit
a=zeros(1,length(t));      % Vektor der Gesamtbeschleunigung

for j=1:1:n
    for i=1:1:length(t)
        y_tot_h=phi_a(j)*ita_y(j,i);
        y(i)=y(i)+y_tot_h;
        v_tot_h=phi_a(j)*ita_v(j,i);
        v(i)=v(i)+v_tot_h;
        a_tot_h=phi_a(j)*ita_a(j,i);
        a(i)=a(i)+a_tot_h;
    end
end

% Extremwerte
ymax=max(y);    ymin=min(y);
vmax=max(v);    vmin=min(v);
amax=max(a);    amin=min(a);

% Maximale Beschleunigung bei n_p Personen
amax_n_p=sqrt(n_p)*amax;
%-----

%----- DARSTELLUNGSBLOCK -----
% Grafische Darstellung der Ergebnisse
name_fig1 = 'Zeitverläufe';
fig1=figure('Name',name_fig1,'NumberTitle','off');
set(fig1,'Position',[1000 50 700 900]);

subplot(4,1,1)
plot(t,Kraft,'r','LineWidth', 1);
title('Kraftzeitverlauf');
xlabel('Zeit [s]');
ylabel('Kraft [N]');
grid on;

subplot(4,1,2)
plot(t,y,'b','LineWidth', 1);
title('Verschiebungszeitverlauf');
xlabel('Zeit [s]');
ylabel('Verschiebung [m]');
grid on;

subplot(4,1,3)
plot(t,v,'b','LineWidth', 1);
title('Geschwindigkeitszeitverlauf');
xlabel('Zeit [s]');
ylabel('Geschwindigkeit [m/s]');
grid on;

subplot(4,1,4)
plot(t,a,'b','LineWidth', 1);
title('Beschleunigungszeitverlauf');
xlabel('Zeit [s]');
ylabel('Beschleunigung [m/s^2]');

```

[illegible]

```

fprintf(fid, '%s\n', 'Schrittlaenge [m]:');
fprintf(fid, '%.3f\n', ls);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Anzahl der Personen [-]:');
fprintf(fid, '%.3f\n', n_p);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Faktor fuer die Berechnungsdauer [-]:');
fprintf(fid, '%.3f\n', a_tmax);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Anzahl Unterteilungen der kleinsten Eigenschwingzeit');
fprintf(fid, '%.3f\n', ndt);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Anzahl der beruecksichtigten Eigenformen');
fprintf(fid, '%.3f\n', n);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Stelle der Ausgabe [m]:');
fprintf(fid, '%.3f\n', x0);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, ...
        '%s\n', '-----');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, ...
        '%s\n', 'ERGEBNISSE:');
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
EF=[Omega Freq T];
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', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Modal=[m k d];
fprintf(fid, '%s \t %s \t %s \t %s\n', 'EF', 'Modale Masse [kg]', ...
        'Modale Steifigkeit [N/m]', 'Modale Daempfung [Ns/m]');
for jj = 1:1:n
    fprintf(fid, '%d \t %d \t %d \t %d\n', jj, Modal(jj,:));
end
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 Berechnungszeitschritte [-]:');
fprintf(fid, '%d\n', nt);
fprintf(fid, '%s\n', ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fid, '%s\n', 'Gesamtdauer der Berechnung [s]:');
fprintf(fid, '%d\n', t_ber);

```

[illegible]

```

Ergebnis_Bewegungsgroessen=[t' Kraft' y' v' a'];
fid = fopen('Outputdatei_2_Bewegungsgroessen.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 %s\n', 'Programm ML_19_1: Zeitverlaeufe der', ...
    'Bewegungsgroessen am Ausgabepunkt');
fprintf(fid, '%s\n', ' ');
fprintf(fid,...
    '%s\n', '-----');
fprintf(fid, '%s\n', ' ');
fprintf(fid, '%s %s\n', 'Zeitverlaeufe der Bewegungsgroessen und', ...
    'der Erregerkraft:');
fprintf(fid, '%s \t %s \t %s \t %s \t %s\n', 'Zeit [s]:', ...
    'Kraft [N]', 'Verschiebung [m]', 'Geschwindigkeit [m/s]', ...
    'Beschleunigung [m/s^2]');
for ii=1:1:length(y)
    fprintf(fid, '%d \t %d \t %d \t %d \t %d\n', ...
        Ergebnis_Bewegungsgroessen(ii,:));
end
fclose(fid);
%-----

```