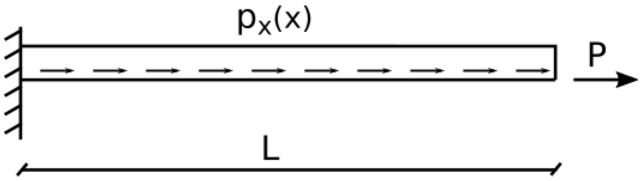
W tym ćwiczeniu omówimy zagadnienie statyki i drgań podłużnych pręta. Do wykonania obliczeń skorzystamy z oprogramowania Matlab oraz pakietu CALFEM. Przed wykonaniem ćwiczenia proszę zapoznać się z instrukcją opisującą instalację pakietu CALFEM dla oprogramowania Matlab Online.

Zadanie jest podzielone na trzy części: statyka, drgania swobodne i drgania wymuszone.

Na stronie https://www.l5.pk.edu.pl/~jpamin/dyd/MK/bar_vibrations.m znajduje się skrypt, z którego będziemy korzystać podczas ćwiczenia.

1. Statyka

Rozważamy pręt z obciążeniem rozłożonym $p_x(x)$ oraz zadaną siłą skupioną P. Przyjmujemy stały przekrój A na całej długości pręta.



Sformułowanie lokalne ma postać:

$$(EAu')' = -p_x$$

wraz z warunkami brzegowymi:

$$u(0) = 0 \quad \text{i} \quad EA(l)u'(l) = P$$

Na pierwszym wykładzie przedstawione jest wyprowadzenie sformułowania globalnego, które jest punktem wyjścia w metodzie elementów skończonych, dla przypomnienia zapiszmy wzór:

$$\int_0^l w'\, EAu' \mathrm{d}x = [w\, EAu']_0^l + \int_0^l w\, p_x \mathrm{d}x \quad \forall w \quad \text{plus podst. w.b.}$$

Naszym celem jest zbudowanie układu równań wynikających ze sformułowania metody elementów skończonych, rozwiązanie tego układu i wyznaczenie wartości przemieszczenia oraz siły podłużnej w dowolnym punkcie pręta. Układ równań MES jest następujący:

$$\boxed{\mathbf{Kd} = \mathbf{f}}$$

K – macierz sztywności, **d** – wektor stopni swobody, **f** – wektor obciążenia.

W programie Matlab Online utworzymy skrypt, w którym zapiszemy instrukcje ilustrujące wykorzystanie MES do rozwiązania powyższego zadania. Wszystkie instrukcje, które należy wprowadzić zawarte są w bloku wyróżnionym szarym kolorem. Zaczniemy od wyczyszczenia ekranu, zamknięcia wszystkich otwartych okien, usunięcia zmiennych zdefiniowanych w przestrzeni roboczej oraz zdefiniowania zmiennej symbolicznej `x`:

```
clc
close all
clear
syms x;
```

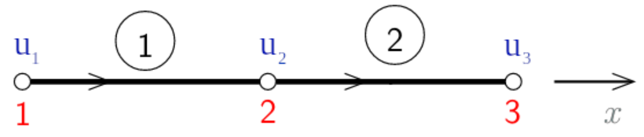
Następnie definiujemy moduł Younga, pole przekroju poprzecznego, gęstość i długość pręta:

```
E  = 210e+9;     A = 25e-4;    rho = 7800;   L=1;
```

Zakładamy obciążenie ciągłe zależne od x oraz ściskającą siłę skupioną na końcu pręta:

```
px=4.0e6*x; P=-1e6;
```

Teraz dokonujemy dyskretyzacji – dzielimy obszar na `Nel` elementów, w wyniku otrzymamy `Nel+1` węzłów. W każdym węźle mamy jeden stopień swobody – przemieszczenie w kierunku x. Poniższy rysunek przedstawia dyskretyzację obszaru na dwa elementy wraz z numeracją elementów, węzłów (kolor czerwony) i stopni swobody (kolor niebieski).



W programie zapiszemy liczbę elementów (`Nel`), liczbę stopni swobody (`Ndof`) oraz współrzędne węzłów (`coord`):

```
Nel = 20;   Ndof = Nel+1;   coord=linspace(0,L,Ndof);
```

Mamy rozwiązać poniższy układ równań:

$$\boxed{\mathbf{Kd} = \mathbf{f}}$$

W pierwszej kolejności przygotujemy globalną macierz sztywności `KG` i globalny wektor obciążenia `fG`. Wymiar macierzy sztywności i wektora obciążenia zależny jest od liczby stopni swobody w układzie:

```
KG=zeros(Ndof,Ndof);
fG=zeros(Ndof,1);
```

Następnie uwzględniamy siłę skupioną zadaną na końcu pręta – stopień swobody w ostatnim węźle będzie miał numer `Ndof`, dlatego wpisujemy instrukcję:

```
fG(Ndof)  = P;
```

Możemy teraz przejść do głównego algorytmu MES, który składa się z następujących kroków:

- w każdym elemencie wyznaczyć:
  - współrzędne węzłów elementu,
  - funkcje kształtu dla elementu,
  - macierz sztywności elementu, którą następnie należy zagregować do macierzy globalnej,
  - wektor obciążenia elementu, który następnie należy zagregować do globalnego wektora obciążenia.
- uwzględnić warunki brzegowe i rozwiązać układ równań wyznaczając wartości przemieszczeń i reakcji w węzłach,
- w każdym elemencie wyznaczyć wartości siły podłużnej.

W pierwszym kroku konieczne jest wykonanie obliczeń w każdym elemencie, dlatego zastosujemy pętlę `for` z licznikiem `iel` oznaczającym numer elementu:

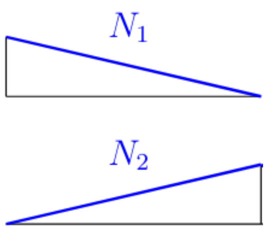
```
for iel = 1:Nel

end
```

Jasnoniebieski prostokąt oznacza instrukcje, które wpiszemy w pętli `for`.

Wyznaczamy współrzędne węzłów, element o numerze `iel` ma początek w węźle `iel`, a koniec w węźle `iel+1`, dlatego współrzędne zapisujemy:

```
x1=coord(iel); x2=coord(iel+1);
```

W każdym elemencie mamy zdefiniowane dwie liniowe funkcje kształtu:



$$N_1(x) = \frac{x - x_2}{x_1 - x_2}$$

$$N_2(x) = \frac{x - x_1}{x_2 - x_1}$$

$$\mathbf{N}^1 = [N_1, N_2]$$

Wprowadzamy odpowiednie wzory w skrypcie, zapisujemy dwie funkcje kształtu w formie wektora oraz obliczamy wektor pochodnych funkcji kształtu `B`:

```
N1=(x-x2)/(x1-x2); N2=(x-x1)/(x2-x1);
N=[N1 N2]; B=diff(N);
```

Wektor pochodnych funkcji kształtu potrzebny jest do obliczenia macierzy sztywności w danym elemencie:

$$\mathbf{K}^e = \int_{X_1}^{X_2} \mathbf{B}^{e\mathrm{T}} E A \mathbf{B}^e \mathrm{d}x^e$$

wykorzystując obliczenia symboliczne i funkcję pozwalającą na całkowanie w zadanym przedziale (`int`) obliczamy macierz sztywności elementu:

```
Kel=int(B'*E*A*B,x1,x2);
```

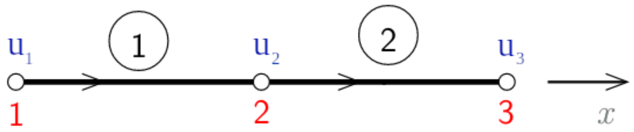
Wektor obciążenia dla elementu obliczamy korzystając ze wzoru:

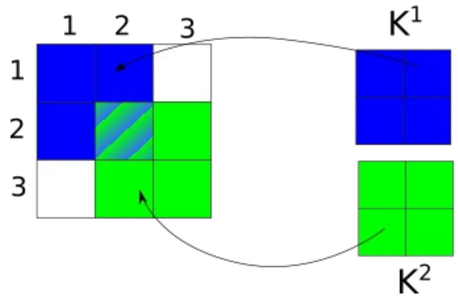
$$\mathbf{f}^e = \int_{X_1}^{X_2} \mathbf{N}^{e\mathrm{T}} p_x \mathrm{d}x^e$$

W skrypcie zapisujemy:

```
Fel=int(N'*px,x1,x2);
```

Po obliczeniu elementowej macierzy sztywności i wektora obciążenia powinniśmy dokonać agregacji, czyli do globalnej macierzy sztywności dodać w odpowiednie miejsca wartości z macierzy elementowej, podobnie postępujemy dla wektora obciążenia. Proces agregacji ilustruje poniższy rysunek:





W celu dokonania agregacji skorzystamy z pakietu CALFEM, który zawiera instrukcję `assem` odpowiedzialną za agregowanie macierzy i wektora prawej strony. Konieczne jest zdefiniowanie wektora `edof`, który opisuje element – podajemy numer elementu oraz numery stopni swobody dla tego elementu:

```
edof=[iel, iel, iel+1];
```

Następnie korzystamy z procedury `assem`:

```
[KG,fG]=assem(edof,KG,Kel,fG,Fel);
```

Na tym kończymy pierwszy etap algorytmu MES, przechodzimy do uwzględnienia warunków brzegowych i rozwiązania układu równań. W tym celu skorzystamy z funkcji `solveq` pakietu CALFEM. Zapisujemy warunek brzegowy podając numer stopnia swobody i zadaną wartość, w tym przypadku z lewej strony pręta mamy zablokowane przemieszczenie, czyli pierwszy stopień swobody ma zadaną wartość 0:

```
bc =[1,0];
```

Rozwiązujemy układ równań – wyznaczamy przemieszczenia:

```
u=solveq(KG,fG,bc);
```

Następnie rysujemy rozwiązanie:

```
figure(1) % displacement
title('displacement function')
plot(coord,u,'linewidth',2)
grid on, xlabel('x'), ylabel('u')
```

Po wyznaczeniu rozwiązania przechodzimy do ostatniego kroku algorytmu MES, czyli wyznaczenia wartości siły podłużnej w elementach. Ponownie skorzystamy z pętli `for`, która będzie odpowiedzialna za iterację po poszczególnych elementach:

```
for iel = 1:Nel

end
```

Siłę podłużną wyznaczamy na podstawie wzoru:

$$N = EAu'^e \qquad u'^e = \mathbf{B}^e\mathbf{d}^e$$

zatem w pętli musimy obliczyć pochodne funkcji kształtu oraz wyznaczyć wektor przemieszczeń dla elementu. Wartości siły podłużnej obliczymy w punktach pośrednich danego elementu. Wyznaczamy współrzędne węzłów elementu:

```
x1=coord(iel); x2=coord(iel+1);
```

Odczytujemy przemieszczenia w węzłach elementu – czyli wybieramy składowe wektora `u` o numerach `iel` i `iel+1`:

```
u1=u(iel); u2=u(iel+1);
```

Definiujemy punkty pośrednie w przedziale [x_1,x_2]:

```
xi=linspace(x1,x2,10);
```

Zapisujemy wzory na pochodne funkcji kształtu w elemencie:

```
dN1=1/(x1-x2);
dN2=1/(x2-x1);
```

i obliczamy siłę podłużną:

```
Shxi=A*E*(u1*dN1+u2*dN2);
```

następnie rysujemy wykres siły podłużnej:

```
figure(2), plot(xi,Shxi,'b.'), hold on
title('normal force function elementwise')
grid on, xlabel('x'), ylabel('N^e')
```

2. Drgania swobodne

W drugiej części ćwiczenia zmodyfikujemy wcześniej przygotowany skrypt, aby wyznaczyć wartości i postaci drgań podłużnych pręta.

Mamy do rozwiązania następujący uogólniony problem własny:

$$\hat{\mathbf{K}}\mathbf{A} = \omega^2 \hat{\mathbf{M}}\mathbf{A}$$

$\hat{\mathbf{K}}$ – globalna macierz sztywności po uwzględnieniu podstawowego w.b.

$\hat{\mathbf{M}}$ – globalna macierz mas po uwzględnieniu podstawowego w.b.

ω – częstość drgań własnych

$\mathbf{A}$ – wektor amplitud drgań

W skrypcie przygotowujemy zmienną, która będzie przechowywała macierz mas:

```
KG=zeros(Ndof,Ndof);
MG = zeros(Ndof, Ndof);
fG=zeros(Ndof,1);
```

W pętli po elementach dodajemy instrukcje, które obliczą elementową macierz mas i zagregują tę macierz do globalnej macierzy mas. Aby obliczyć elementową macierz mas korzystamy ze wzoru:

$$\mathbf{M}^e = \int_{X_1}^{X_2} \mathbf{N}^{\mathrm{T}} \rho A \mathbf{N} \, \mathrm{d}x^e$$

W pierwszej pętli iterującej po elementach dodajemy instrukcje:

```
for i = 1:Nel
  x1=coord(iel);
  x2=coord(iel+1);
  N1=(x-x2)/(x1-x2); N2=(x-x1)/(x2-x1);
  N=[N1 N2]; B=diff(N);
  Kel=int(B'*E*A*B,x1,x2);
  Fel=int(N'*px,x1,x2);

  edof=[iel, iel, iel+1];
  [KG,fG]=assem(edof,KG,Kel,fG,Fel);

  Mel=int(A*rho*N'*N,x1,x2);
  MG=assem(edof,MG,Mel);

end
```

Po obliczeniu globalnej macierzy mas, rozwiązujemy powyższy uogólniony problem własny, korzystamy z instrukcji `eigen` przybornika CALFEM:

```
[La,Ev]=eigen(KG,MG,bc(:,1));
```

Należy pamiętać, żeby uwzględnić warunki brzegowe, dlatego jako ostatni argument funkcji `eigen` podajemy numery stopni swobody, gdzie nałożone są podstawowe warunki brzegowe. W wyniku otrzymamy wektor wartości własnych `La` oraz macierz postaci drgań własnych `Ev`. Wektor `La` zawiera częstości drgań podniesione do kwadratu, natomiast kolumny macierzy `Ev` reprezentują formy deformacji odpowiadające danej częstości drgań (pierwsza kolumna odpowiada pierwszej częstości drgań, druga kolumna drugiej częstości itd.).

Wprowadzamy instrukcję, która wyznaczy częstotliwości drgań:

```
Freq=sqrt(La)/(2*pi)
```

Po wyznaczeniu częstotliwości i postaci drgań, rysujemy rozwiązanie – narysujemy maksymalnie cztery formy drgań własnych:

```
figure(3)
title('basic eigen modes')
color=['r','g','b','m'];
modes=min([4,Nel]);
for iev=1:modes
    subplot(2,2,iev)
    plot(coord,Ev(:,iev),color(iev),'linewidth',2),
    grid on, axis([0,L,-1,1]), hold on
    nm=num2str(iev); forma=strcat('Eigenform ',nm);
    xlabel('x'), ylabel('Ev')
    title(forma)
    plot([0,L],[0,0],'k')
end
```

3. Drgania wymuszone

W ostatniej części ćwiczenia zajmiemy się drganiami wymuszonymi, do zadania wprowadzamy siłę wymuszającą zależną od czasu

$$\hat{\mathbf{M}}\hat{\mathbf{a}} + \hat{\mathbf{K}}\hat{\mathbf{d}} = \hat{\mathbf{f}}$$

$\hat{\mathbf{a}}$– wektor przyspieszeń węzłów

$\hat{\mathbf{f}}$– funkcja obciążenia, zależna od czasu

W tym zadaniu konieczne jest wprowadzenie całkowania po czasie. Do tego celu wykorzystamy metodę Newmarka, której algorytm zdefiniowany jest następująco:

- przyjąć dane początkowe (dla t = 0)

  $$\mathbf{d}_0, \quad \dot{\mathbf{d}}_0 = \mathbf{v}_0$$

  wyznaczyć początkową wartość przyspieszenia ze wzoru:

  $$\hat{\mathbf{M}}\mathbf{a}_0 = \hat{\mathbf{f}}_0 - \hat{\mathbf{K}}\mathbf{d}_0$$

- w kolejnych chwilach czasowych wyznaczyć wartości:
  - przyspieszenia

    $$\left[\hat{\mathbf{M}} + \tfrac{1}{2}(\Delta t)^2\hat{\mathbf{K}}\right]\mathbf{a}_{k+1} = \hat{\mathbf{f}}_{k+1} - \hat{\mathbf{K}}(\mathbf{d}_k + \Delta t\mathbf{v}_k)$$

  - przemieszczenia

    $$\mathbf{d}_{k+1} = \mathbf{d}_k + \Delta t\mathbf{v}_k + \tfrac{1}{2}\Delta t^2\mathbf{a}_{k+1}$$

  - prędkości

    $$\mathbf{v}_{k+1} = \mathbf{v}_k + \Delta t\mathbf{a}_{k+1}$$

Naszymi danymi są globalna macierz mas i macierz sztywności oraz początkowa wartość siły wymuszającej obliczone na podstawie statyki oraz początkowe przemieszczenia i prędkość.

Zacznijmy od zdefiniowania powyższych danych. Najpierw uwzględniamy podstawowy warunek brzegowy – usuwamy z macierzy mas i macierzy sztywności pierwszy wiersz i pierwszą kolumnę, natomiast z wektora prawej strony usuwamy pierwszy element. Zmniejszamy liczbę stopni swobody o 1 – znamy rozwiązanie w punkcie x=0 (początek pręta z zadaną blokadą przemieszczenia):

```
MG(1,:)=[]; MG(:,1)=[];
KG(1,:)=[]; KG(:,1)=[];
fG(1)=[];
Ndof_1=Ndof-1;
```

W zadaniu przyjmujemy funkcję obciążenia wg wzoru:

$f(t) = F \cdot \sin(\omega \cdot t)$

gdzie ω to 1.5 najmniejszej częstości drgań własnych, a F to wektor obciążenia statycznego.

Określamy częstotliwość siły wymuszającej i na tej podstawie obliczamy częstość:

```
freq_imp=Freq(1)*1.5; om=2*pi*freq_imp;
```

Podajemy czas początkowy i końcowy – w tym przedziale czasowym będziemy śledzić odpowiedź dynamiczną pręta:

```
time=0; t_end=0.002;
```

Zapisujemy wzór funkcji wymuszającej:

```
f=inline('sin(o*t)');
```

Zapisujemy warunki początkowe zadania – liczbę kroków czasowych, wartość kroku czasowego, zerujemy wektor przemieszczeń i prędkości:

```
N=200;  % time steps
dt=(t_end-time)/N;

uu=zeros(Ndof_1,1); vv=zeros(Ndof_1,1);
```

Wprowadzamy dwa wektory `thist` i `uhist`, w których będziemy przechowywać odpowiednio historię zmian czasu oraz historię przemieszczeń ostatniego węzła:

```
thist(1)=time;
uhist(1)=uu(Ndof_1);
```

W metodzie Newmarka posługujemy się następującym wzorem, aby wyznaczyć wartości przyspieszenia w węzłach:

$$\left[\hat{\mathbf{M}} + \frac{1}{2}(\Delta t)^2\hat{\mathbf{K}}\right]\mathbf{a}_{k+1} = \hat{\mathbf{f}}_{k+1} - \hat{\mathbf{K}}(\mathbf{d}_k + \Delta t\mathbf{v}_k)$$

aby rozwiązać ten układ równań lewostronnie mnożymy przez odwrotność członu:

$$\left[\hat{\mathbf{M}} + \frac{1}{2}(\Delta t)^2\hat{\mathbf{K}}\right]$$

dlatego użyjemy zmiennej pomocniczej `Amat`:

```
Amat=MG+1/2*dt^2*KG;
Amat=inv(Amat);
```

Następnie zapisujemy pętlę po kolejnych krokach czasowych:

```
for itm=2:N+1
    time=time+dt;

    aa=Amat*(fG*f(om,time)-KG*(uu+dt*vv));
    uu=uu+dt*vv+1/2*dt^2*aa;
    vv=vv+dt*aa;    thist(itm)=time;
    uhist(itm)=uu(Ndof_1);
end
```

Zgodnie z algorytmem metody Newmarka, wyznaczymy wartości przyspieszenia, przemieszczenia i prędkości w kolejnych chwilach czasowych.

Na końcu rysujemy wykres historii zmian przemieszczenia ostatniego węzła:

```
figure(4)
plot(thist,uhist,'r','linewidth',2), grid on, hold on
xlabel('t'); ylabel('u(1,t)');
```