Metoda Elementów Skończonych

Pliki do wykorzystania w poniższym ćwiczeniu można pobrać za pomocą poniższych linków:

- Plik nagłówkowy meslib.h
- Plik źródłowy meslib.cpp
- Plik nagłówkowy winbgi2.h
- Plik źródłowy winbgi2.cpp

Wprowadzenie

Niniejszą instrukcją rozpoczynamy cykl laboratoriów, z których każde kolejne będzie rozwinięciem poprzedniego i wnioski z poprzedniego będą stanowiły motywację do zastosowania kolejnych metod. Tym samym należy zadbać o to, aby zadania z każdego laboratorium były wykonywane do końca czy to na zajęciach, czy w domu. Dziś zaznajomimy się ze sformułowaniem metody elementów skończonych dla jednego elementu czworokątnego. Dla uproszczenia nie będziemy stosować transformacji geometrycznych występujących w rzeczywistym sformułowaniu metody elementów skończonych, więc faktycznie będziemy operować zawsze na jednostkowych elementach kwadratowych.

Sformułowanie algebraiczne dla jednego elementu

Zagadnienie wytrzymałości konstrukcji dla jednego elementu skończonego ma następujące sformułowanie algebraiczne: Elementowi skończonemu przypisana jest tzw. macierz sztywności $\mathbf{K}$ reprezentująca jego sztywność na poszczególne rodzaje odkształceń (przesunięcia jego wierzchołków), wektor przesunięć $\mathbf{d}$, jakim ulegną poszczególne wierzchołki elementu pod obciążeniem oraz wektor sił węzłowych $\mathbf{f}$ reprezentujący odpowiednio przeliczone siły (bądź obciążenia ciągłe) przyłożone do węzłów elementu.

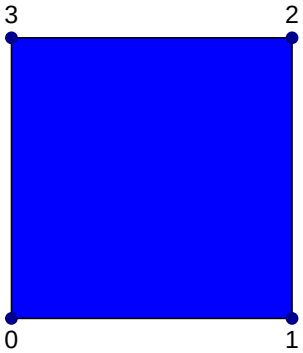
$$\mathbf{Kd} = \mathbf{f}$$

Wektor przesunięć (ang. *displacement*) zawiera odpowiednio przesunięcia względem osi x oraz y kolejnych węzłów elementu (numeracja węzłów pokazana jest na Rys. 1).

Wektor sił węzłowych $\mathbf{f}$ ma analogiczną postać. Poniżej przedstawiona jest również macierz sztywności $\mathbf{K}$.

$$\mathbf{d} = \begin{bmatrix} x_0 \\ y_0 \\ x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} f_{x_0} \\ f_{y_0} \\ f_{x_1} \\ f_{y_1} \\ f_{x_2} \\ f_{y_2} \\ f_{x_3} \\ f_{y_3} \end{bmatrix}$$

$$\mathbf{K} = \frac{E}{1-\nu^2} \begin{bmatrix} \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}+\frac{\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & -\frac{1}{8}+\frac{3}{8}\nu & -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}-\frac{\nu}{8} & \frac{\nu}{6} & \\ \frac{1}{8}+\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{3}{8}\nu & \frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}+\frac{3}{8}\nu & \\ -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}-\frac{3}{8}\nu & \frac{1}{2}-\frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} & \frac{\nu}{6} & -\frac{1}{8}+\frac{3}{8}\nu & -\frac{1}{4}+\frac{\nu}{12} & \\ -\frac{1}{8}+\frac{3}{8}\nu & \frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{3}{8}\nu & -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}+\frac{\nu}{8} & \\ -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}-\frac{\nu}{8} & \frac{\nu}{6} & \frac{1}{8}-\frac{3}{8}\nu & \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}+\frac{\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & \\ -\frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}+\frac{3}{8}\nu & -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}+\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{3}{8}\nu & \\ \frac{\nu}{6} & -\frac{1}{8}+\frac{3}{8}\nu & -\frac{1}{4}+\frac{\nu}{12} & \frac{1}{8}+\frac{\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}-\frac{3}{8}\nu & \frac{1}{2}-\frac{\nu}{6} & \\ \frac{1}{8}-\frac{3}{8}\nu & -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}+\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}+\frac{3}{8}\nu & \frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} & \end{bmatrix}$$



Rys. 1. Lokalna indeksacja w.z.ów elementu sko.czonego.

Macierz sztywności elementu **K**, dopóki nie zostaną na układ narzucone więzy unieruchamiające układ w przestrzeni jako ciało sztywne, jest osobliwa. Aby móc obliczyć przemieszczenia węzłów elementu pod działaniem określonych sił, nakładamy najpierw więzy. Załóżmy, że w naszym przypadku będą to więzy zamurowania lewego brzegu elementu, tzn. $x_0 = y_0 = x_3 = y_3 = 0$. W postaci macierzowej powyższe cztery równania są równoznaczne z zastąpieniem wierszy o indeksach 0, 1, 6 i 7 jedynkami na głównej diagonali macierzy i przypisaniu zer na tych indeksach w wektorze prawej strony. Po narzuceniu więzów macierz nie jest już osobliwa i można rozwiązać układ choćby procedurą Gaussa.

Zadania

- Napisz program, w którym zaalokujesz pamięć na:
  - globalną macierz sztywności **Kg** (uzupełniona macierz sztywności elementu **K** jest już w bibliotece `meslib`),
  - wektor przemieszczeń węzłowych **d**,
  - wektor prawych stron **f**,
  - wektor więzów **fix** do narzucenia warunków brzegowych.
- Uzupełnij zaalokowane tablice.
- Odbierz wszystkie stopnie swobody po lewej stronie elementu.
- Jako wymuszenie przyłóż siłę ciągnącą w dół prawy dolny węzeł.
- Rozwiąż zagadnienie metodą Gaussa i narysuj odkształcony element.
- Zmodyfikuj w programie obciążenie: przyłóż siłę skierowaną pionowo w dół do obu węzłów na prawej krawędzi. Rozwiąż zadanie.
- Przyłóż siły, które rozciągną element i rowiąż zadanie.
- Sprawdź wpływ współczynnika Poissona na rozwiązanie.
- Sprawdź różne modyfikacje programu, aby zaznajomić się z zadawaniem obciążeń i ruchami poszczególnych stopni swobody.
- Zmień sposób zamurowania na inny.
- Sprawdź, co się stanie z rozwiązaniem, gdy w ogóle nie narzucisz warunków brzegowych lub nie w pełni odbierzesz sztywne stopnie swobody układu.
- Spróbuj stworzyć teraz belkę składającą się z dwóch elementów skończonych.

Wskazówki implementacyjne

- Elementy w belce mają być indeksowane od 0 do $MX * MY - 1$, poczynając od elementu w lewym dolnym rogu i indeksując je po kolei w prawo. Po dojściu do końca belki, zaczynamy od lewego brzegu analogicznie indeksować kolejnymi liczbami elementy w pasie położonym o jeden element wyżej niż dopiero co indeksowany pas. W identyczny sposób indeksowane są węzły. Takiego indeksowania wymaga uproszczenie implementacji (brak transformacji geometrycznej macierzy sztywności) oraz prostota obecnej funkcji do rysowania całego układu.
- `MX` i `MY` oznaczają odpowiednio liczbę elementów w poziomie i w pionie w prostokątnej belce. Muszą to być zmienne globalne.
- Funkcja do rysowania `draw` wymaga podania wskaźnika `int *` do tablicy `fix` o długości równej liczbie stopni swobody w siatce $N = 2(MX + 1)(MY + 1)$. Jeśli na danej pozycji w tablicy stoi zero, przyjmujemy, że ten stopień swobody ulega przemieszczeniom. Wpisanie na danym miejscu wartości 1, oznacza, że stopień swobody o tym indeksie jest odebrany (tak układ zostanie narysowany przez procedurę rysującą `draw`). Tablicę `fix` należy także wykorzystać do narzucenia więzów na globalną macierz sztywności.
- Początek głównego pliku z kodem programu powinien mieć następującą postać. Poniższy przykład pokazuje też sposób użycia biblioteki `meslib` w celu narysowania rozwiązanego układu.

```c
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "meslib.h"
#include "winbgi2.h"

int MX = 1;                       // liczba elementow w poziomie
int MY = 1;                       // liczba elementow w pionie
int N = 2 * (MX + 1) * (MY + 1); // calkowita liczba stopni swobody

int main()
{
  // deklaracja wskaznika i alokacja tablicy wiezow
  int *fix;

  fix = (int*) calloc(N, sizeof(int));
  if (fix == NULL) {
    perror("main (fix)");
    exit(1);
  }

  // pozostale deklaracje i alokacje pamieci

  // uzupelnienie globalnej macierzy sztywnosci oraz wektorow wiezow i s

  // modyfikacja ukladu r-n ze wzgledu na wiezy

  // rozwiazanie ukladu za pomoca funkcji gauss

  // rysowanie rozwiazania
  graphics(700, 700);
  scale(0, 0.5 * (MY - MX - 3), MX + 3, 0.5 * (MY + MX + 3));
  title("X", "Y", "MES");
  draw(d, f, fix);
  wait();

  // zwolnienie pamieci
  free(fix);

  return 0;
}
```

Krótka dokumentacja biblioteki `meslib`

Biblioteka `meslib` została stworzona na potrzeby laboratorium, aby uprościć i przyspieszyć implementację metody elementów skończonych. Zawiera szereg prostych i przydatnych funkcji oraz macierz sztywności i macierz masową pojedynczego elementu. Poniżej krótko opiszemy poszczególne funkcje:

- `int P(int id_x, int id_y, int dir)` - funkcja, która zwraca globalny indeks stopnia swobody przy założeniu, że `id_x` to jego indeks w kierunku x (liczony od 0 na lewej krawędzi), `id_y` to jego indeks w kierunku osi y (liczony od 0 na dolnej powierzchni belki), a `dir` to 0 lub 1 zależnie od tego czy interesuje nas stopień swobody w kierunku poziomym, czy pionowym.
- `int Q(int id_x, int id_y)` - funkcja zwracająca globalny indeks elementu przy założeniu, że jest to element o indeksie `id_x` w kierunku poziomym i o indeksie `id_y` w kierunku pionowym.
- `int DOF(int elid_x, int elid_y, int locdofid)` - funkcja zwracająca globalny indeks stopnia swobody przy założeniu, że `elid_x` oznacza indeks elementu w kierunku poziomym, `elid_y` indeks elementu w kierunku pionowym, a `locdofid` to liczba od 0 do 7 oznaczająca lokalny indeks danego stopnia swobody.
  Lokalna indeksacja stopni swobody jest następująca: w węźle o indeksie 0 przesunięcia w kierunku poziomym i pionowym to odpowiednio 0-wy i 1-szy stopień swobody, w węźle o indeksie 1 występują 2. i 3. stopień swobody itd.
- Lokalna macierz sztywności jest zdefiniowana w dwuwymiarowej tablicy `K[8][8]`. Przy składaniu macierzy sztywności powinna być w programie przemnożona przez czynnik `Md` zawierający wpływ modułu Younga oraz współczynnika Poissona oraz przez grubość elementu.
- Lokalna macierz masowa jest zdefiniowana w dwuwymiarowej tablicy `M[8][8]`. Przy składaniu globalnej macierzy masowej powinna być w programie przemnożona przez czynnik `Mm` zawierający wpływ gęstości materiału oraz dodatkowo należy ją przemnożyć przez grubość elementu.
- `void gauss(int n, double **M, double *f, double *x)` - procedura eliminacji Gaussa rozwiązująca układ równań o macierzy zapisanej w dynam-

icznie zaalokowanej dwuwymiarowej tablicy `M` o wymiarze `n`, i wektorze prawej strony `f`. Wynik zostanie wpisany do miejsc w pamięci wskazywanych przez wskaźnik `*x`.

- `void gauss_psp(int tabsize, double **A_ext, double *b_ext, double *x)` - procedura eliminacji Gaussa z wyborem częściowym elementu głównego (ang. partial scaled pivoting). Metoda zmniejsza propagację błądów numerycznych i pozwala odwracać większą klasę macierzy.
- `void draw(double *p, double *f, int *fix)` - funkcja rysująca cały układ odkształconych elementów. `* p` to wskaźnik na tablicę przesunięć poszczególnych stopni swobody, `* f` to wskaźnik na tablicę sił węzłowych w tych stopniach swobody a `* fix` to wskaźnik na tablice więzów.