МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ФН
КАФЕДРА «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ ФИЗИКА»

Дисциплина:
Основы методов конечных элементов

Отчет по выполнению лабораторной работы №5

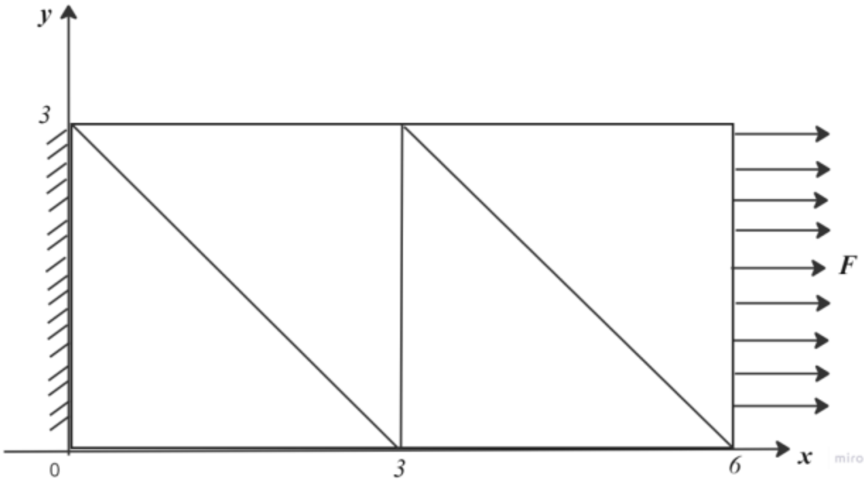
Группа: ФН11-72Б

Студент: Ладыгина Л.В.
Преподаватель:Захарова Ю.В.

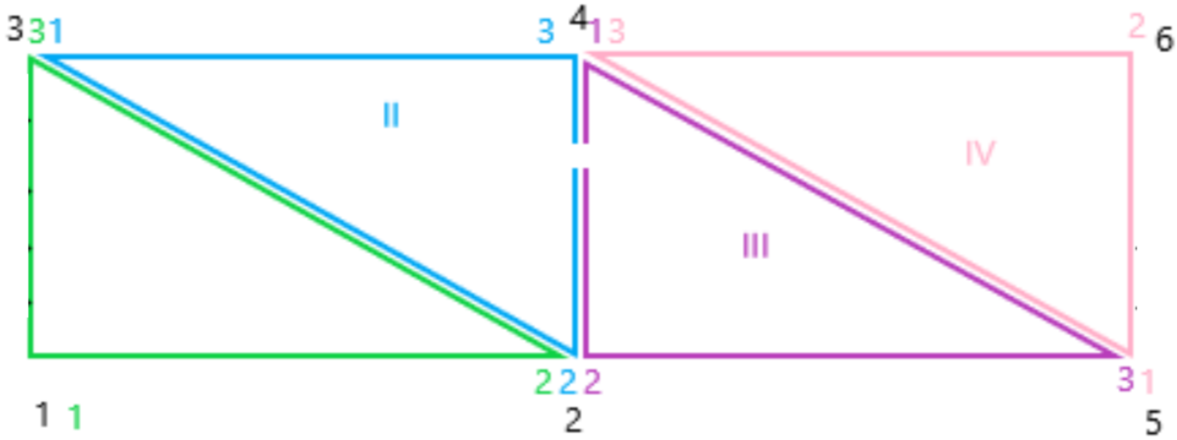
Москва, 2022

Метод конечных элементов для двумерной задачи упругости

Задача. Дана прямоугольная пластина левый край жестко защемлен, на правом действует равномерно распределенная нагрузка вдоль оси x интенсивностью 20 Н/см^2. Толщина тела t считается постоянной и равна 2 см. Модуль упругости $E = 6 \cdot 10^6 \, H / см^2$, коэффициент Пуассона $\mu = 0.25$. Тело разбито на 4 треугольных элемента. Составить локальные матрицы жесткости и правых частей. Произвести сборку в глобальную матрицу и глобальный вектор. Решить СЛАУ.



проведем дискретизацию области



Ищем вектора перемещения по x и y в виде вектора Ф по формулам:

$$k^{(e)}\Phi = f^{(e)}$$

где $\left[k^{(e)}\right] = \int\limits_{V^{(e)}} \left[B^{(e)}\right]^T \left[D^{(e)}\right]\left[B^{(e)}\right] dV$

$$f^{(e)} = \int\limits_{S_1} \left[N^{(e)}\right]^T \begin{bmatrix} p^{(e)}{}_x \\ p^{(e)}{}_y \end{bmatrix} dS$$

Найдем матрицу жесткости D:

$$D = \frac{E}{1-\mu^2}\begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & (1-\mu)/2 \end{bmatrix}$$

```
D=[[0 for i in range(3)]for j in range(3)]

D[0][0]=float(E)/(1-mu**2)
D[0][1]=float(E)/(1-mu**2)*mu
D[1][0]=D[0][1]
D[1][1]=D[0][0]
D[2][2]=float(E)/(2*(1-mu**2))*(1-mu)

for i in range(3):
  for j in range(3):
    print(D[i][j], end=' ')
  print("\n")
```

```
6400000.0 1600000.0 0

1600000.0 6400000.0 0

0 0 2400000.0
```

Матрица N имеет вид

$$\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}$$

$$N_i = \frac{1}{2S^{(e)}}\left(a_i + b_i x + c_i y\right)$$

Функции B:

$$\left[B^{(e)}\right] = \begin{bmatrix} \dfrac{\partial[N]}{\partial x} \\ \dfrac{\partial[N]}{\partial y} \end{bmatrix} = \frac{1}{2S^{(e)}}\begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix}.$$

функции a(x,y), b(x,y), c(x,y) и функция определителя det(x,y):

```
def det(x,y):
  delta=(x[1]*y[2]-y[1]*x[2])+(x[2]*y[0]-x[0]*y[2])+(x[0]*y[1]-x[1]*y[0])
  return delta
```

```
def a(x,y):
  A=[]
  A.append(x[1]*y[2]-x[2]*y[1])
  A.append(x[2]*y[0]-x[0]*y[2])
  A.append(x[0]*y[1]-x[1]*y[0])
  return A
```

```
def b(x,y):
  B=[]
  B.append(y[1]-y[2])
  B.append(y[2]-y[0])
  B.append(y[0]-y[1])
  return B
```

```
def c(x,y):
  C=[]
  C.append(x[2]-x[1])
  C.append(x[0]-x[2])
  C.append(x[1]-x[0])
  return C
```

1) Рассмотрим первый элемент

```
x=[0,3,0]
y=[0,0,3]
```

Расчетные формулы:

```
A1=a(x,y)
B1=b(x,y)
C1=c(x,y)

for i in range(3):
  print("N",i+1,"= ", float(1)/(2*S)*A1[i], "+", float(1)/(2*S)*B1[i], "x +", float(1)/(2*S)*C1[i], "y\n")

print("\n\n")

B=[[0 for i in range(6)]for j in range(3)]

B[0][0]=float(1)/(2*S)*B1[0]
B[0][2]=float(1)/(2*S)*B1[1]
B[0][4]=float(1)/(2*S)*B1[2]

B[1][1]=float(1)/(2*S)*C1[0]
B[1][3]=float(1)/(2*S)*C1[1]
B[1][5]=float(1)/(2*S)*C1[2]

B[2][0]=float(1)/(2*S)*C1[0]
B[2][1]=float(1)/(2*S)*B1[0]
B[2][2]=float(1)/(2*S)*C1[1]
B[2][3]=float(1)/(2*S)*B1[1]
B[2][4]=float(1)/(2*S)*C1[2]
B[2][5]=float(1)/(2*S)*B1[2]


print("B\n")
for i in range(3):
  for j in range(6):
    print(round(B[i][j],2), end=' ')
  print("\n")

K1=t*S*np.dot(np.transpose(B),np.dot(D,B))
print("\n\nK:")
for i in range(6):
  for j in range(6):
    print(round(K1[i][j],7), end=' ')
  print("\n")


f=[0,0,0,0,0,0]
```

Т.к площадь и толщина представляют из себя постоянные величины, то для матрицы жесткости формула приобретает вид:

$$K^{(e)} = tS \cdot B^T DB$$

Для первого элемента отсутствуют нагрузки, поэтому $f^{(1)} \equiv 0$

Функции формы и матрица жесткости:

```
N 1 =  1.0 + -0.3333333333333333 x + -0.3333333333333333 y

N 2 =  0.0 + 0.3333333333333333 x + 0.0 y

N 3 =  0.0 + 0.0 x + 0.3333333333333333 y



B

-0.33 0 0.33 0 0.0 0

0 -0.33 0 0.0 0 0.33

-0.33 -0.33 0.0 0.33 0.33 0.0


K:
8800000.0 4000000.0 -6400000.0 -2400000.0 -2400000.0 -1600000.0

4000000.0 8800000.0 -1600000.0 -2400000.0 -2400000.0 -6400000.0

-6400000.0 -1600000.0 6400000.0 0.0 0.0 1600000.0

-2400000.0 -2400000.0 0.0 2400000.0 2400000.0 0.0

-2400000.0 -2400000.0 0.0 2400000.0 2400000.0 0.0

-1600000.0 -6400000.0 1600000.0 0.0 0.0 6400000.0
```

Аналогичным образом найдем матрицы жесткости для элементов 2, 3, 4:

```
K2:
6400000.0 0.0 0.0 1600000.0 -6400000.0 -1600000.0

0.0 2400000.0 2400000.0 0.0 -2400000.0 -2400000.0

0.0 2400000.0 2400000.0 0.0 -2400000.0 -2400000.0

1600000.0 0.0 0.0 6400000.0 -1600000.0 -6400000.0

-6400000.0 -2400000.0 -2400000.0 -1600000.0 8800000.0 4000000.0

-1600000.0 -2400000.0 -2400000.0 -6400000.0 4000000.0 8800000.0
```

```
K3:
2400000.0 0.0 -2400000.0 -2400000.0 0.0 2400000.0

0.0 6400000.0 -1600000.0 -6400000.0 1600000.0 0.0

-2400000.0 -1600000.0 8800000.0 4000000.0 -6400000.0 -2400000.0

-2400000.0 -6400000.0 4000000.0 8800000.0 -1600000.0 -2400000.0

0.0 1600000.0 -6400000.0 -1600000.0 6400000.0 0.0

2400000.0 0.0 -2400000.0 -2400000.0 0.0 2400000.0
```

```
K4:
2400000.0 0.0 -2400000.0 -2400000.0 0.0 2400000.0

0.0 6400000.0 -1600000.0 -6400000.0 1600000.0 0.0

-2400000.0 -1600000.0 8800000.0 4000000.0 -6400000.0 -2400000.0

-2400000.0 -6400000.0 4000000.0 8800000.0 -1600000.0 -2400000.0

0.0 1600000.0 -6400000.0 -1600000.0 6400000.0 0.0

2400000.0 0.0 -2400000.0 -2400000.0 0.0 2400000.0
```

Найдем вектор правой части для 4-го элемента:

```
N 1 =  1.0 + 0.0 x + -0.3333333333333333 y

N 2 =  -2.0 + 0.3333333333333333 x + 0.3333333333333333 y

N 3 =  2.0 + -0.3333333333333333 x + 0.0 y
```

$$f1 = F \cdot \int_{3}^{6} \int_{6-x}^{3} N1 \, \mathrm{d}y \mathrm{d}x$$

$$f2 = F \cdot \int_{3}^{6} \int_{6-x}^{3} N2 \, \mathrm{d}y \mathrm{d}x$$

$$f3 = F \cdot \int_{3}^{6} \int_{6-x}^{3} N3 \, \mathrm{d}y \mathrm{d}x$$

$$f = \begin{bmatrix} f_x^{(5)} \\ \mathrm{f}_x^{(6)} \\ \mathrm{f}_x^{(5-6)} \end{bmatrix} = \begin{bmatrix} f1 \\ f2 \\ f3 \end{bmatrix} = \begin{bmatrix} 3.006000000\ 10^7 \\ 3.006000000\ 10^7 \\ 3.006000000\ 10^7 \end{bmatrix}$$

$$F^{(5)} = f_x^{(5)} + \mathrm{f}_x^{(5-6)} = 6 \cdot 10^7$$

$$F^{(6)} = f_x^{(6)} + \mathrm{f}_x^{(5-6)} = 6 \cdot 10^7$$

Найдем глобальный вектор нагрузки F и глобальную матрицу жесткости:

Глобальный вектор нагрузки имеет ненулевые компоненты только в узлах 5 и 6 по оси OX , следовательно, представлен в виде :

```
F=[0,0,0,0,0,0,0,0,6000000,0,6000000,0]
```

Индексы вершин для каждого тела:

```
index_I=[0,1,2]
index_II=[2,1,3]
index_III=[3,1,4]
index_IV=[4,5,3]
```

составление глобальной матрицы:

```
K=[[0 for i in range(12)]for j in range(12)]

for i in range(0,6):
  for j in range(0,6):
    for k in range(3):
      for t in range(3):

        if i==index_I[k] and j==index_I[t]:
          if (i==j):
            if (i%2==0):
             K[2*i][2*j]+=K1[2*k][2*t]
            else:
             K[2*i+1][2*j+1]+=K1[2*k+1][2*t+1]
          else:
            K[2*i][2*j]+=K1[2*k][2*t]
```

```
          K[2*i+1][2*j]+=K1[2*k+1][2*t]
          K[2*i+1][2*j+1]+=K1[2*k+1][2*t+1]
          K[2*j][2*i+1]=K[2*i+1][2*j]

        if i==index_II[k] and j==index_II[t]:
           if (i==j):
            if (i%2==0):
             K[2*i][2*j]+=K2[2*k][2*t]
            else:
             K[2*i+1][2*j+1]+=K2[2*k+1][2*t+1]
           else:
            K[2*i][2*j]+=K2[2*k][2*t]
           K[2*i+1][2*j]+=K2[2*k+1][2*t]
           K[2*i+1][2*j+1]+=K2[2*k+1][2*t+1]
           K[2*j][2*i+1]=K[2*i+1][2*j]

        if i==index_III[k] and j==index_III[t]:
          if (i==j):
            if (i%2==0):
             K[2*i][2*j]+=K3[2*k][2*t]
            else:
             K[2*i+1][2*j+1]+=K3[2*k+1][2*t+1]
          else:
            K[2*i][2*j]+=K3[2*k][2*t]
          K[2*i+1][2*j]+=K3[2*k+1][2*t]
          K[2*i+1][2*j+1]+=K3[2*k+1][2*t+1]

          K[2*j][2*i+1]=K[2*i+1][2*j]




        if i==index_IV[k] and j==index_IV[t]:
         if (i==j):
            if (i%2==0):
             K[2*i][2*j]+=K4[2*k][2*t]
            else:
             K[2*i+1][2*j+1]+=K4[2*k+1][2*t+1]
         else:
            K[2*i][2*j]+=K4[2*k][2*t]
        K[2*i+1][2*j]+=K4[2*k+1][2*t]
        K[2*i+1][2*j+1]+=K4[2*k+1][2*t+1]
```

```
        K[2*j][2*i+1]=K[2*i+1][2*j]

for i in range(12):
  for j in range(12):
    print(round(K[i][j],7), end=' ')
  print("\n")
```

Глобальная матрица К с учетом закреплений в 1-ом и 3-ем узле:

```
1000000 0 0 0 0 0 0 0 0 0 0 0

0 1000000 0 0 0 0 0 0 0 0 0 0

0 0 17600000.0 4000000 0 4000000 -4800000.0 -4000000 -6400000.0 -2400000.0 0 0

0 0 4000000 17600000.0 4000000 0 -4000000 -12800000.0 -1600000.0 -2400000.0 0 0

0 0 0 0 1000000 0 0 0 0 0 0 0

0 0 0 0 0 1000000 0 0 0 0 0 0

0 0 -4800000.0 -4000000 0 0 17600000.0 4000000 0 4000000 -6400000.0 -1600000.0

0 0 -4000000 -12800000.0 0 0 4000000 17600000.0 4000000 0 -2400000.0 -2400000.0

0 0 -6400000.0 -1600000.0 0 0 0 4000000 8800000.0 0 -2400000.0 -2400000.0

0 0 -2400000.0 -2400000.0 0 0 4000000 0 0 8800000.0 -1600000.0 -6400000.0

0 0 0 0 0 0 -6400000.0 -2400000.0 -2400000.0 -1600000.0 8800000.0 4000000

0 0 0 0 0 0 -1600000.0 -2400000.0 -2400000.0 -6400000.0 4000000 8800000.0
```

Найдем вектор Ф из матричного уравнения $\Phi = K^{-1} f$:

```
np.dot(np.linalg.inv(K), F)
```

полученные значения:

```
[ 0.00000000e+00,  0.00000000e+00,  2.47934096e-06,  5.63485622e-08,
        1.09187843e-15,  0.00000000e+00,  3.65497819e-06,
-8.76780297e-07,
        6.81818583e+00, -9.08330837e-07,  8.67769025e+00,
-2.08489967e+00)
```

№ узла, *i*	u_i(перемещения по ОХ)	v_i (перемещения по ОУ)
1	0	0
2	2.47934096e-06	5.63485622e-08
3	0	0
4	3.65497819e-06	-8.76780297e-07
5	6.81818583e+00	-9.08330837e-07
6	8.67769025e+00	-2.08489967e+00