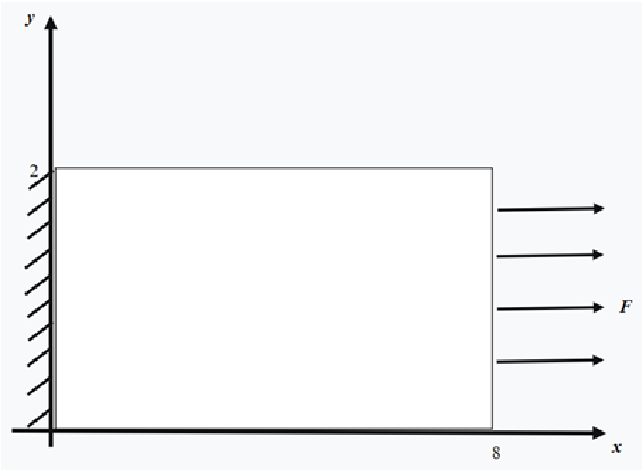
Метод конечных элементов для двумерной задачи упругости

Задание

Дана прямоугольная пластина, левый край жестко защемлен, на правом действует равномерно распределенная нагрузка вдоль оси X интенсивностью $8 \cdot 10^5$Н/см^2. Толщина тела t считается постоянной и равна 2 см. Модуль упругости $E = 2 \cdot 10^7$Н/см^2, коэффициент Пуассона $\mu = 0.25$. Тело разбито на 8 треугольных элементов. Составить локальные матрицы жесткости и правых частей. Произвести сборку в глобальную матрицу и глобальный вектор. Решить СЛАУ.



Решение

Рассмотрим задачу, соответствующую плоско-напряженному состоянию. Неизвестными функциями в данной задаче выступают перемещения $u(x, y)$, $v(x, y)$. Запишем неизвестные в векторном виде: $u = \begin{bmatrix} u \\ v \end{bmatrix}$.

Неразмерность распределения перемещений будем определять антиградиентом, называемым деформациями

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix}$$

где

$$\varepsilon_x = \frac{\partial u}{\partial x}, \varepsilon_y = \frac{\partial v}{\partial y}, \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

или

$$\varepsilon = Lu, \quad L = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}.$$

Деформации связаны с напряжением $\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}$ через физические соотношения (закон Гука):

$$\sigma = D\varepsilon = DLu$$

где

$$D = \frac{E}{1-\mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \text{ — матрица упругости.}$$

Составим вариационную постановку задачи.

Полная потенциальная энергия упругости системы равна $\Pi = \Lambda - W_p$, где Λ — энергия деформаций, W_p — потенциальная энергия внешних сил. Энергия деформации задается формулой:

$$\Lambda = \int\limits_V \frac{1}{2}\{\sigma^T \varepsilon\}\, dV,$$

$$W_p = \int\limits_S u^T p\, dS.$$

Для решения задачи методом конечных элементов выразим перемещение через узловые значения

$$u = [N]\,\{\Phi\}$$

Для треугольного симплекс элемента получаем:

$$u = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}.$$

где $N_1 = \frac{1}{2S^{(e)}}(a_i + b_i x + c_i y) = L_i, i = \overline{1,3}$. Коэффициенты были выведены ранее:

$$a_1 = x_2 y_3 - x_3 y_2 \quad b_1 = y_2 - y_3 \quad c_1 = x_3 - x_2$$

$$a_2 = x_3 y_1 - x_1 y_3 \quad b_2 = y_3 - y_1 \quad c_2 = x_1 - x_3$$

$$a_3 = x_1 y_2 - x_2 y_1 \quad b_3 = y_1 - y_2 \quad c_3 = x_2 - x_1$$

Тогда

$$\varepsilon = Lu = LN\Phi = B\Phi$$

$$B^{(e)} = \frac{1}{2S(e)} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix}.$$

Внешние нагрузки в данном случае представляются в виде работы поверхностных сил:

$$p = \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \quad W_p = \int\limits_S \Phi^T N^T \begin{bmatrix} p_x \\ p_y \end{bmatrix} dS.$$

Все полученные формулы подставим в формулу полной энергии (для симплекс элемента):

$$\Pi^{(e)} = \frac{1}{2} \int\limits_{V^{(e)}} \Phi^{(e)T} B^{(e)T} D B^{(e)} \Phi^{(e)} dV - \int\limits_S \Phi^{(e)T} N^{(e)T} \begin{bmatrix} p_x^{(e)} \\ p_y^{(e)} \end{bmatrix} dS.$$

Для минимизации функционала Π найдем производную и приравняем её к нулю:

$$\frac{\partial \Pi^{(e)}}{\partial \Phi^i} = \int\limits_{V^{(e)}} B^{(e)T} D^{(e)} B^{(e)} \Phi^{(e)}\, dV - \int\limits_S N^{(e)T} \begin{bmatrix} p_x^{(e)} \\ p_y^{(e)} \end{bmatrix} dS = 0.$$

$$K^{(e)} = \int\limits_{V^{(e)}} B^{(e)T} D^{(e)} B^{(e)}\, dV, \quad f^{(e)} = \int\limits_{S} N^{(e)T} \begin{bmatrix} p_x^{(e)} \\ p_y^{(e)} \end{bmatrix} dS$$

Следовательно, можно представить в виде СЛАУ:

$$K^{(e)} \Phi = f^{(e)}.$$

Выразим матрицу $K^{(e)}$ и вектор правой части $f^{(e)}$ через специальные формулы:

$$\int\limits_{S(e)} L_1^{\alpha} L_2^{\beta} L_3^{\gamma}\, dS = \frac{\alpha!\,\beta!\,\gamma!}{(\alpha+\beta+\gamma+2)!} 2S(e), \quad \int\limits_{\Gamma_{12}^{(e)}} L_1^{\alpha} L_2^{\beta}\, d\Gamma = \frac{\alpha!\,\beta!}{(\alpha+\beta+1)!} l_{ij}.$$

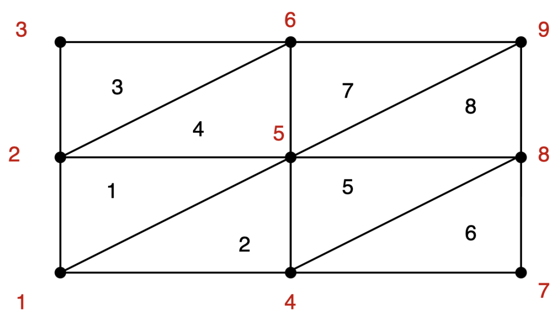
Так как толщины пластины постоянна, $dV = t\, dS$. Тогда:

$$K^{(e)} = t \int\limits_{S^{(e)}} B^{(e)T} D^{(e)} B^{(e)}\, dS = t\, B^{(e)T} D^{(e)} B^{(e)} S^{(e)}.$$

Для граничных узлов можно выразить $dS = t\, d\Gamma$, причем учитываться будут только Γ_{ij}, где i, j — граничные, причем $p_y^{(e)} = 0$, так как нагрузка действует только вдоль оси Ox, а также $N_k = 0$, так как нагрузка приходится только на два узла i, j:

$$f^{(e)} = t \int\limits_{\Gamma_{ij}^{(e)}} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_x^{(e)} \\ 0 \end{bmatrix} d\Gamma = t \int\limits_{\Gamma_{ij}^{(e)}} \begin{bmatrix} L_1 & 0 \\ 0 & L_1 \\ L_2 & 0 \\ 0 & L_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_x^{(e)} \\ 0 \end{bmatrix} d\Gamma = \frac{tl_{ij}}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_x^{(e)} \\ 0 \end{bmatrix} = \frac{tl_{ij} p_x^{(e)}}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Дискретизация области:



Код для вычисления перемещений:

```
import math
import numpy as np
import pandas as pd

px = 8 * 10**5
t = 2
E = 2 * 10**7
mu = 0.25
S = 2

node_koord = [[0,0], [0,1], [0,2], [4,0], [4,1],
          [4,2], [8,0], [8,1], [8,2]]
node_bound = [[6,7], [7,8]]
node_trial = [[0,1,4], [0,3,4], [1,2,5], [1,4,5],
          [3,4,7], [3,6,7], [4,5,8], [4,7,8]]

F = np.zeros(18)
for pair in node_bound:
        i,j = pair
        l_ij = 1
        F[2*i] += t * l_ij * px / 2
        F[2*j] += t * l_ij * px / 2
F = F/100000

K = np.zeros((18, 18))
for trial in node_trial:
        i,j,k = trial
        xi,yi = node_koord[i]
        xj,yj = node_koord[j]
        xk,yk = node_koord[k]

        ai = xj*yk - xk*yj
        aj = xk*yi - xi*yk
        ak = xi*yj - xj*yi
        bi = yj - yk
        bj = yk - yi
        bk = yi - yj
        ci = xk - xj
        cj = xi - xk
        ck = xj - xi

        B = np.array([
                [bi/(2*S), 0, bj/(2*S), 0, bk/(2*S), 0],
                [0, ci/(2*S), 0, cj/(2*S), 0, ck/(2*S)],
                [ci/(2*S), bi/(2*S), cj/(2*S),
                        bj/(2*S), ck/(2*S), bk/(2*S)]
        ])

        kof = E / (1 - mu**2)
```

```
    D = np.array([
            [kof, kof*mu, 0],
            [kof*mu, kof, 0],
            [0, 0, kof*(1-mu)/2]
    ])

    Ke = t * S * (B.T @ D @ B)
    indices = [2*i, 2*i+1, 2*j, 2*j+1, 2*k, 2*k+1]

    for m in range(6):
        for n in range(6):
            K[indices[m], indices[n]] += Ke[m, n]

K = K/100000
```

Вектор правых частей:

$$F = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 8 \\ 0 \\ 16 \\ 0 \\ 8 \\ 0 \end{bmatrix}$$

Полученная матрица K:

$$
K = \begin{bmatrix}
373.33 & 0 & -320 & 80 & 0 & 0 & -53.33 & 53.33 & 0 & -133.33 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
0 & 873.33 & 53.33 & -853.33 & 0 & 0 & 80 & -20 & -133.33 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
-320 & 53.33 & 746.67 & -133.33 & -320 & 80 & 0 & 0 & -106.67 & 133.33 \\
0 & -133.33 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
80 & -853.33 & -133.33 & 1746.67 & 53.33 & -853.33 & 0 & 0 & 133.33 & -40 \\
-133.33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
0 & 0 & -320 & 53.33 & 373.33 & -133.33 & 0 & 0 & 0 & 0 \\
-53.33 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
0 & 0 & 80 & -853.33 & -133.33 & 873.33 & 0 & 0 & 0 & 0 \\
53.33 & -20 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
-53.33 & 80 & 0 & 0 & 0 & 0 & 746.67 & -133.33 & -640 & 133.33 \\
0 & 0 & -53.33 & 53.33 & 0 & -133.33 & 0 & 0 & & \\
53.33 & -20 & 0 & 0 & 0 & 0 & -133.33 & 1746.67 & 133.33 & -1706.67 \\
0 & 0 & 80 & -20 & -133.33 & 0 & 0 & 0 & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -133.33 & 0 \\
80 & -20 & 0 & 0 & 53.33 & -853.33 & 0 & 873.33 & &
\end{bmatrix}
$$

Редуцируем матрицу, так как нам известны значения для первых трех узлов:

```
known_indices = [0, 1, 2, 3, 4, 5]
known_values = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
K_reduced = np.delete(K, known_indices, axis=0)
K_reduced = np.delete(K_reduced, known_indices, axis=1)
F_reduced = np.delete(F, known_indices, axis=0)
```

Вектор правых частей:

$$F = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 8 \\ 0 \\ 16 \\ 0 \\ 8 \\ 0 \end{bmatrix}$$

Полученная матрица K:

$$K = \begin{bmatrix}
746.67 & -133.33 & -640 & 133.33 & 0 & 0 & -53.33 & 53.33 & 0 & -133.33 & 0 & 0 \\
-133.33 & 1746.67 & 133.33 & -1706.67 & 0 & 0 & 80 & -20 & -133.33 & 0 & 0 & 0 \\
-640 & 133.33 & 1493.33 & -266.67 & -640 & 133.33 & 0 & 0 & -106.67 & 133.33 & 0 & -133.33 \\
133.33 & -1706.67 & -266.67 & 3493.33 & 133.33 & -1706.67 & 0 & 0 & 133.33 & -40 & -133.33 & 0 \\
0 & 0 & -640 & 133.33 & 746.67 & -133.33 & 0 & 0 & 0 & 0 & -53.33 & 80 \\
0 & 0 & 133.33 & -1706.67 & -133.33 & 1746.67 & 0 & 0 & 0 & 0 & 53.33 & -20 \\
-53.33 & 80 & 0 & 0 & 0 & 0 & 373.33 & -133.33 & -320 & 53.33 & 0 & 0 \\
53.33 & -20 & 0 & 0 & 0 & 0 & -133.33 & 873.33 & 80 & -853.33 & 0 & 0 \\
0 & -133.33 & -106.67 & 133.33 & 0 & 0 & -320 & 80 & 746.67 & -133.33 & -320 & 53.33 \\
-133.33 & 0 & 133.33 & -40 & 0 & 0 & 53.33 & -853.33 & -133.33 & 1746.67 & 80 & -853.33 \\
0 & 0 & 0 & -133.33 & -53.33 & 53.33 & 0 & 0 & -320 & 80 & 373.33 & 0 \\
0 & 0 & -133.33 & 0 & 80 & -20 & 0 & 0 & 53.33 & -853.33 & 0 & 873.33
\end{bmatrix}$$

Решая СЛАУ KX=F для редуцированных K и F, получаем вектор перемещений:

$$[0., 0., 0., 0., 0., 0., 0.16, 0.02, 0.15, 0.01, 0.15, 0.01, 0.32, 0.03, 0.31, 0.02, 0.31, 0.01]$$