Санкт-Петербургский политехнический университет Петра Великого

Физико-механический институт

Высшая школа теоретической механики и математической физики

**РАБОТА №5**

**Растяжение-сжатие стержней**

по дисциплине «Вычислительная механика»

Вариант №17

Выполнил

студент гр. 5030103/10301 <*подпись*> А.Г.Фёдоров

Руководитель

Доцент, к.ф.-м.н. <*подпись*> Е.Ю. Витохин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Санкт-Петербург

2023

# **Постановка задачи**

Необходимо произвести расчет плоской фермы под действием нагрузки 1кН: определить перемещения в узлах и усилия в стержнях. Нагрузка прикладывает на верхний пояс. Левый и правый нижние углы закреплены по вертикальным и горизонтальным степеням свободы. Модуль Юнга МПа, площадь сечения S =

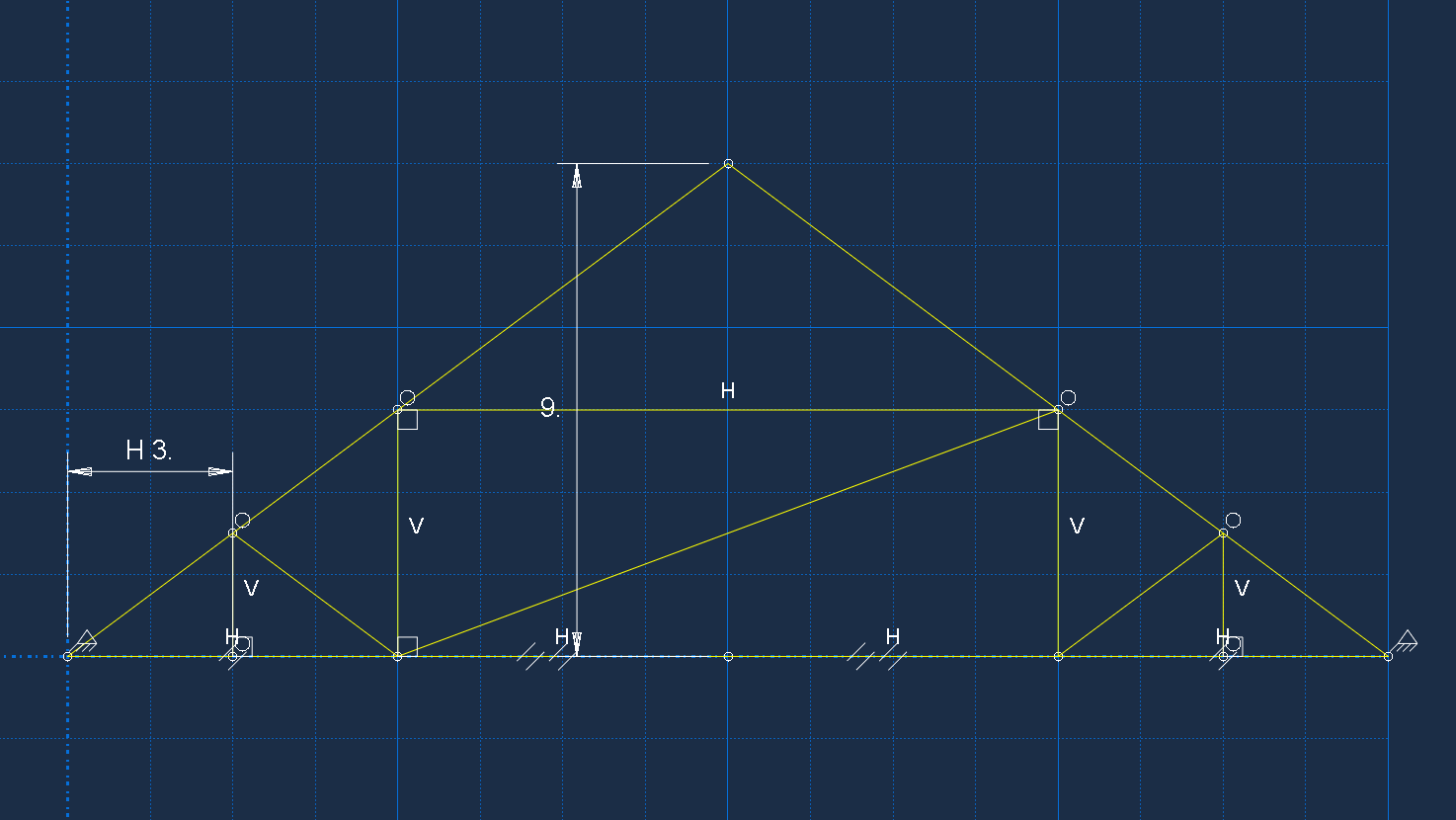


Рис.1 Плоская ферма

Координаты узлов: 1 – (0, 0), 2 – (3, 0), 3 – (6, 0), 4 – (18, 0), 5 – (21, 0),

6 – (24, 0), 7 – (21, 2.25), 8 – (18, 4.5), 9 – (12, 9), 10 – (6, 4.5), 11 – (3, 2.25).

Элемент стержень имеет два узла. Стержень 1 связан с узлами 1 и 2, стержень 2– 2 и 3, стержень 3 – 3 и 4, стержень 4 – 4 и 5, стержень 5 – 5 и 6, стержень 6 – 6 и 7, стержень 7 – 7 и 8, стержень 8 – 8 и 9, стержень 9 – 9 и 10, стержень 10 – 10 и 11, стержень 11 – 11 и 1, стержень 12 – 2 и 11, стержень 13 – 11 и 3, стержень 14 – 3 и 10, стержень 15 – 8 и 10, стержень 16 – 8 и 3, стержень 17 – 4 и 8, стержень 18 – 4 и 7, стержень 19 – 5 и 7.

# **Метод решения**

Будем применять метод конечных элементов, основанный на нахождении минимума функционала потенциальной энергии системы.

Полная потенциальная энергия системы:

– потенциальная энергия внутренних деформаций

– работа внешних сил

Потенциальная энергия бесконечно малого объема:

Потенциальная энергия внутренних сил для бесконечно малого объема:

– вектор-столбец деформаций

– вектор-столбец напряжений

Для конечного объема:

содержит одну компоненту , описывающую деформацию вдоль оси стержня и являющейся производной от перемещения вдоль этой оси

Теперь произведем замену непрерывных перемещений на узловые

Одномерный конечный элемент стержня состоит из двух степеней свободны, поэтому имеет две функции формы. Функции форм – линейные функции, потому что первая степень минимально необходимая степень для элемента с двумя степенями свободы.

Введем матрицу градиентов:

Подставим в выражение для деформации перемещения и учтем полученную матрицу градиентов:

Запишем выражение для напряжения (рассматриваем линейно-упругий изотропный стержень):

Полученные выражения для деформации и напряжения подставим в выражение для внутренней энергии:

Подставим матрицу градиентов:

Вычисляем произведение матриц и получаем:

Работа внешних сил включает в себя работу сосредоточенных, объемных и поверхностных сил:

, , – векторы-столбцы сосредоточенных, объемных и поверхностных сил.

Запишем потенциальную энергию с учетом выражений для внутренней энергии и работы внешних сил:

В задаче статики искомые перемещения деформируемого твердого тела под действием внешних нагрузок отвечают минимуму функционала потенциальной энергии (первая вариация равна нулю):

Первое слагаемое будем называть матрицей жесткости:

Остальное будем называть вектором-столбцом усилий:

Тогда выражение для первой вариации можем записать так:

Эта формула помогает найти удлинение стержня, состоящего из одного конечного элемента, расположенного вдоль оси , под действием нагрузки, направленной вдоль стержня.

Рассмотрим теперь стержень расположенный под углом *α* к оси . Такой стержень имеет 4 степени свободы.

Перемещения вдоль оси стержня двух узлов выражаются через перемещения вдоль осей и с помощью углов к осям:

Матричная форма записи перемещений:

Матрица трансформации:

Тогда связь перемещений вдоль стержня и перемещений в плоскости примет вид:

Аналогично для усилий:

Распишем и получим формулу:

Умножим слева на

Обозначим матрицей жесткости:

Чтобы решить задачу для системы стержневых конечных элементов нужно собрать глобальную матрицу жесткости (размерность N на N, количество степеней свободы). Нужно привести матрицу элемента к размерности глобальной матрицы и учесть глобальную нумерацию узлов. После матрицы элементов нужно просуммировать:

Аналогично:

Тогда основное уравнение метода конечных элементов:

– глобальный вектор-столбец перемещений

Свойства глобальной матрицы жесткости:

1. Ленточная
2. Разреженная
3. Симметричная
4. Вырожденная

Для решения системы надо добавить уравнения, описывающие граничные условия, убрав из системы уравнения для соответствующих степеней свободы.

# **Результ­аты**

В результате были получены графики перемещений и усилий.

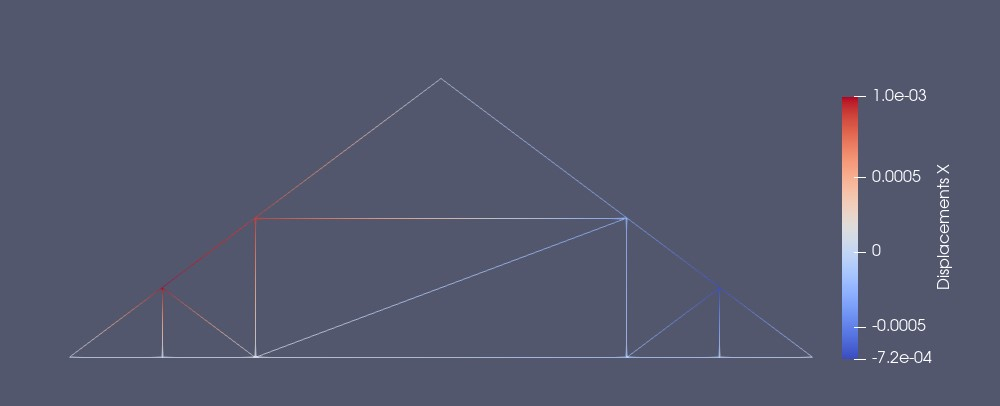
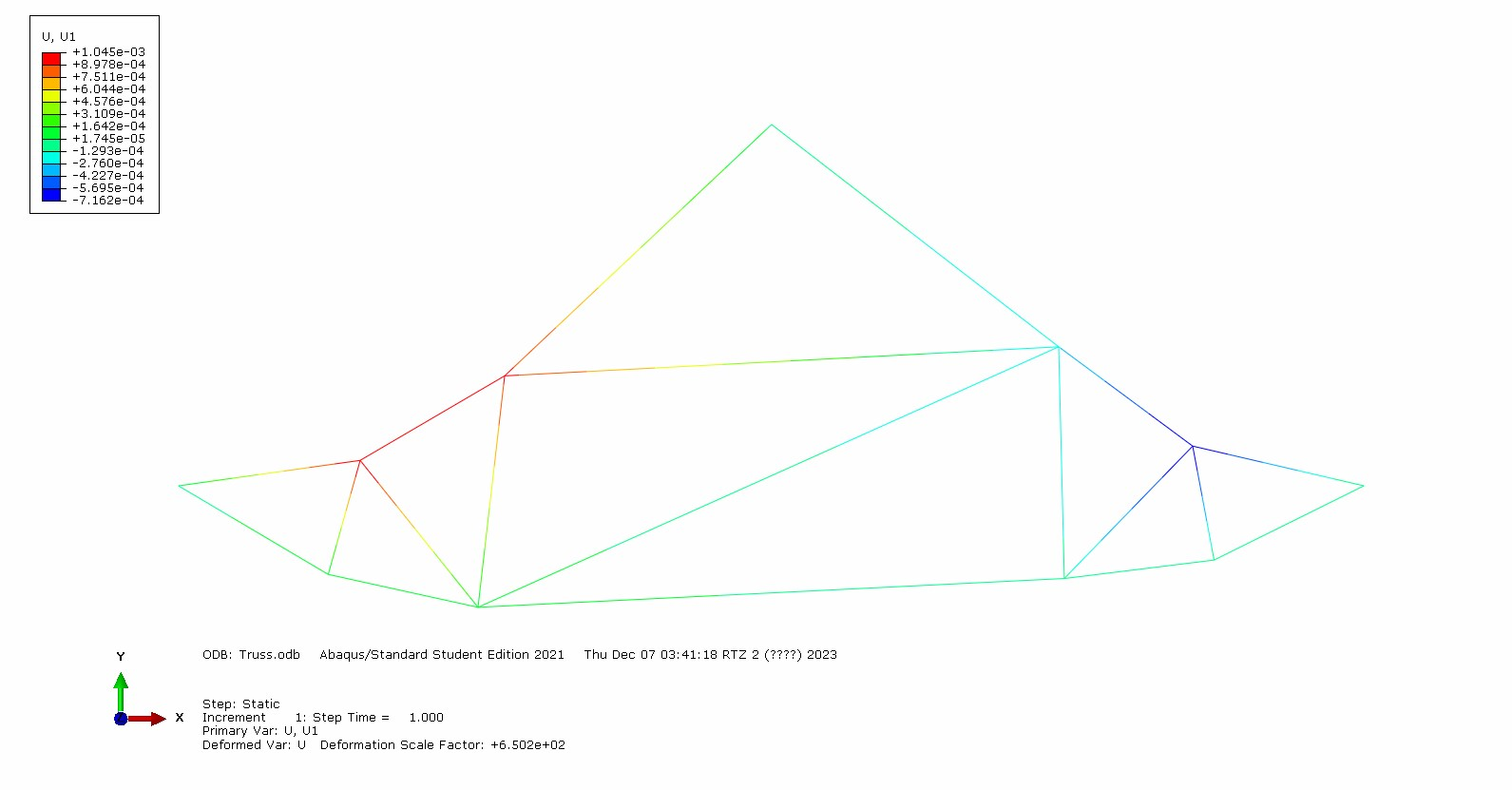


Рис.2 Перемещения вдоль оси X в ParaWiev, м



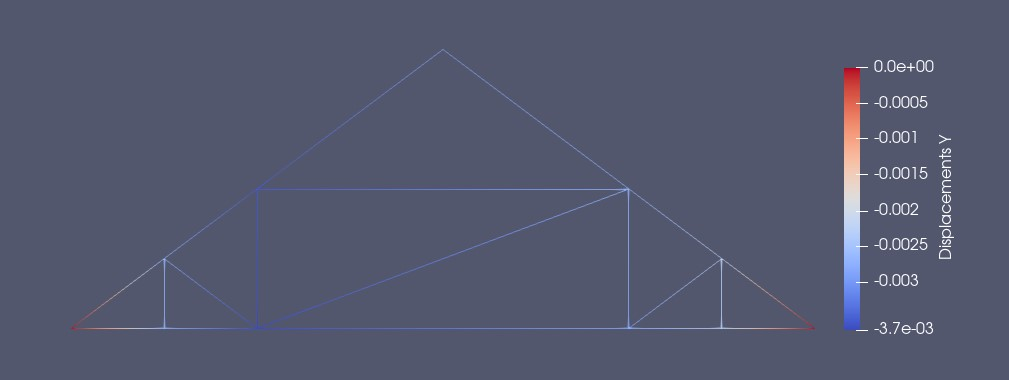
Рис.3 Перемещения вдоль оси X в Abaqus, м

Рис.4 Перемещения вдоль оси Y в ParaWiev, м

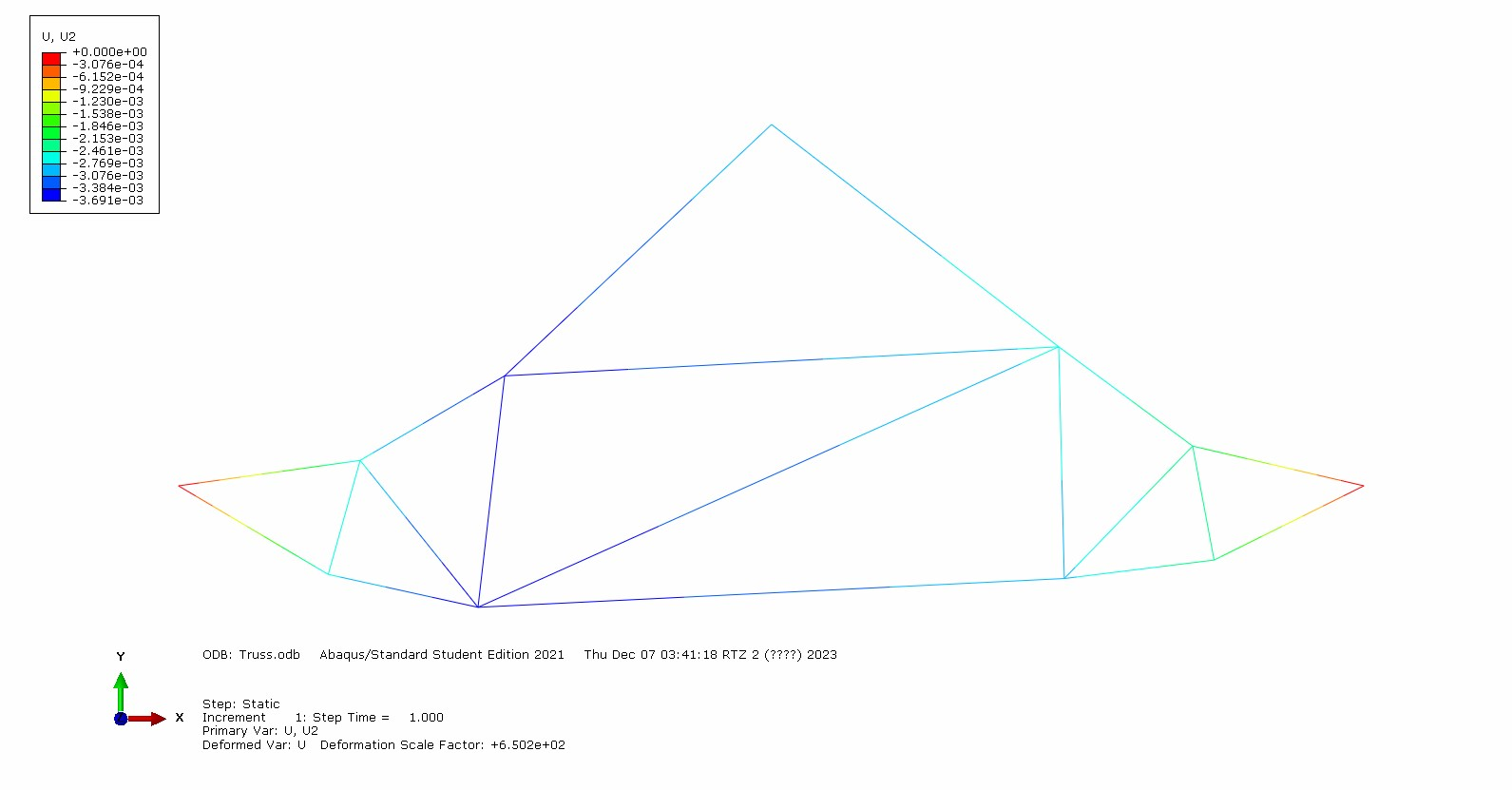
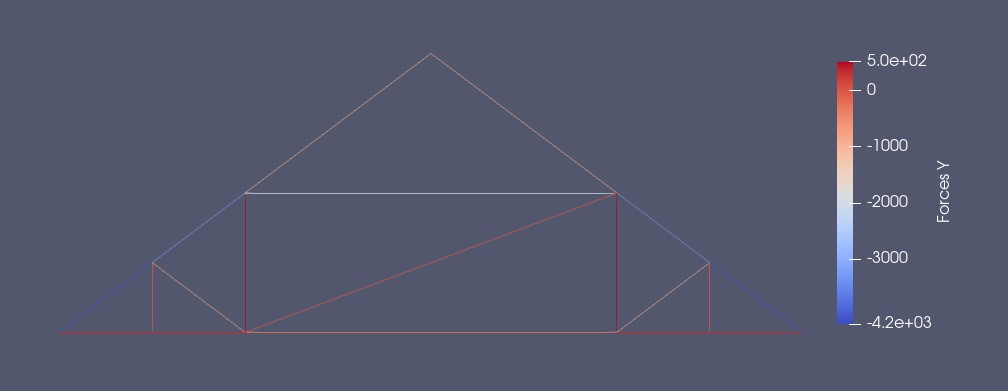


Рис.5 Перемещения вдоль оси Y в Abaqus, м

Таблица 1. Перемещения вдоль оси X и Y

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Python | | Abaqus | |
| Узел | Ux , м | Uy, м | Ux, м | Uy, м |
| 1 | 0,00000 | 0,00000 | 0.00000 | 0.00000 |
| 2 | 0,00005 | -0,00270 | 0.0000497 | -0.002691 |
| 3 | 0,00010 | -0,00369 | 0.000943 | -0.003691 |
| 4 | -0,00010 | -0,00282 | -0.00014 | -0.0028164 |
| 5 | -0,00005 | -0,00226 | -0,00005 | -0.0022558 |
| 6 | 0,00000 | 0,00000 | 0.0000 | 0.0000 |
| 7 | -0,00072 | -0,00226 | -0,000716 | -0,002256 |
| 8 | -0,00027 | -0,00271 | -0,00027 | -0,002703 |
| 9 | 0,00000 | -0,00286 | 0.000001 | -0,002862 |
| 10 | 0,00093 | -0,00358 | 0,00094 | -0,00361 |
| 11 | 0,00104 | -0,00270 | 0,001044 | -0,00269 |



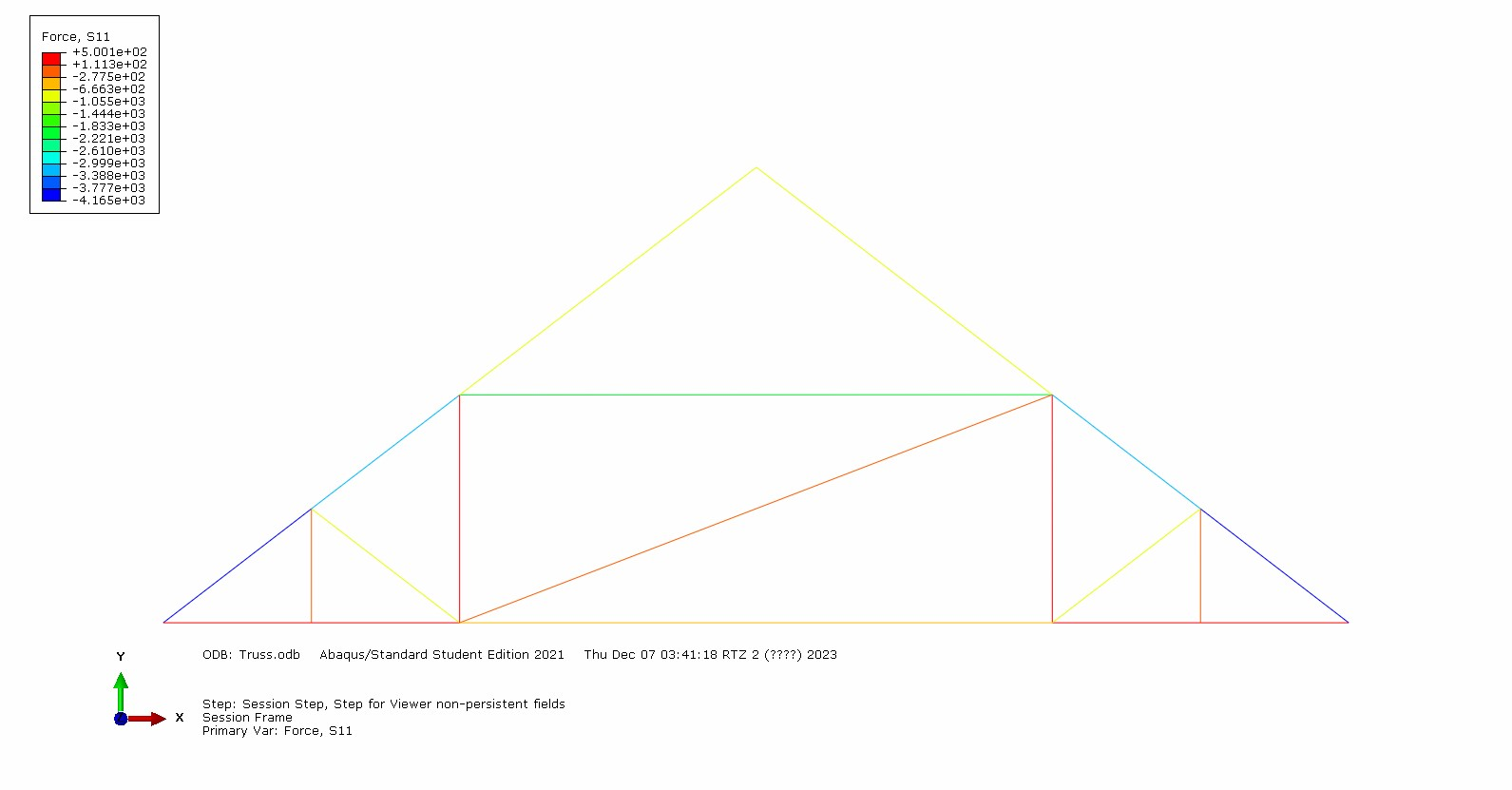
Рис.6 Усилия в ParaWiev, Н

Рис.7 Усилия в Abaqus, Н

Таблица 2. Усилия

|  |  |  |
| --- | --- | --- |
|  | Python | Abaqus |
| Стержень | Усилия, Н | Усилия, Н |
| 1 | 341,4051 | 342,3325 |
| 2 | 334,4433 | 333,833 |
| 3 | -333,2799 | -332,638 |
| 4 | 333,6829 | 331,444 |
| 5 | 338,9921 | 338,3325 |
| 6 | -4163,1137 | -4163,2661 |
| 7 | -3333,3273 | -3331,6504 |
| 8 | -833,3331 | -833,225 |
| 9 | -833,1057 | -833,833 |
| 10 | -3333,0420 | -3332,476 |
| 11 | -4161,1571 | -4165,3755 |
| 12 | 1,9553 | 1,4134 |
| 13 | -832,0053 | -832,985 |
| 14 | 500,3393 | 500,0727 |
| 15 | -1999,9466 | -1999,584 |
| 16 | 0,0680 | 0.06793 |
| 17 | 500,0145 | 500,0727 |
| 18 | -832,8571 | -832,958 |
| 19 | 0,8762 | 0.8763 |

# **Код программы**

from calendar import leapdays

import numpy as np

import math

import meshio

import matplotlib.pyplot as plt

from module import save\_data

S = 1 \* 10 \*\* (-4) #cross-sectional area (m^2)

E = 2 \* 10 \*\* 11 # Young's module (Pa)

absF = 1 \* 10 \*\*(3) # Forse N

forsed\_hinges = [11,10,9,8,7] #hinges where force impact

sealed\_hinges = [1, 6] # hinges wich are sealed

####coordinates of nodes in the truss (x, y)

nodes = [

(0,0),(3,0),(6,0),

(18,0),(21,0),

(24,0),(21,2.25),(18,4.5),

(12,9),(6,4.5),(3,2.25)

]

###connection between rods and hinges (number of rod, number of first hinge, number of the second hinge)

elements = [

(1,1,2),(2,2,3),(3,3,4),(4,4,5),

(5,5,6),(6,6,7),(7,7,8),

(8,8,9),(9,9,10),(10,10,11),(11,11,1),

(12,2,11),(13,11,3),(14,3,10),(15,8,10),

(16,8,3),(17,4,8),(18,4,7),(19, 5, 7)

]

#number degrees of freedom

number = len(nodes) \* 2

#array for forses

F = np.zeros(number )

#system stiffness matrix

K = np.zeros((number , number ))

F\_in = np.zeros((len(elements)))

for node in forsed\_hinges:

F[2 \* (node-1)+1] = -absF

for element in elements:

element\_number, hinge1, hinge2 = element

x1, y1 = nodes[hinge1-1]

x2, y2 = nodes[hinge2-1]

le = np.sqrt((x2 - x1) \*\* 2 +(y2 - y1) \*\* 2)

l = ((x2 - x1))/le

m = ((y2 - y1))/le

transform\_matrix = np.array([[l, m, 0, 0],

[0, 0, l, m]])

B = np.array([[1, -1], [-1, 1]])

k = (transform\_matrix.T @ B @ transform\_matrix) \* E \* S / le

place = [2 \* hinge1-2 , 2 \* hinge1 - 1, 2 \* hinge2 -2 , 2 \* hinge2 - 1]

K[np.ix\_(place, place)] +=k

#hecking the fulfillment of conditions for K

print(np.linalg.det(K), "determinant K")

print(np.amax(abs(K-K.T)))

for node in sealed\_hinges:

F[2 \* (node-1)] = 0

F[2 \* (node-1) + 1] = 0

K[2 \* (node-1), :] = 0

K[2 \* (node-1)+1, :] = 0

K[:, 2 \* (node-1)] = 0

K[:, 2 \* (node-1)+1] = 0

K[2 \* (node-1), 2 \* (node-1)] = 1

K[2 \* (node-1) + 1, 2 \* (node-1) + 1] = 1

print(np.linalg.det(K), "determinant K")

print(np.amax(abs(K-K.T)))

U = np.zeros(number, float )

U = np.linalg.solve(K, F)

print(U)

print("-------------------------------------------------------------")

elementnum = np.zeros(len(elements))

nodenum = np.linspace(0,len(nodes)-1, len(nodes))

nodenum += 1

nodenum = nodenum.astype(int)

for i in range(len(elements)):

elementnum[i] = i+1

for element in elements:

element\_number, hinge1, hinge2 = element

x1, y1 = nodes[hinge1-1]

x2, y2 = nodes[hinge2-1]

le = np.sqrt((x2 - x1) \*\* 2 +(y2 - y1) \*\* 2)

Ux1 = U[2\*(hinge1-1)]

Uy1 = U[2\*(hinge1-1)+1]

Ux2 = U[2\*(hinge2-1)]

Uy2 = U[2\*(hinge2-1)+1]

x1 += Ux1

x2 += Ux2

y1 += Uy1

y2 += Uy2

L = np.sqrt((x2-x1)\*\*2 + (y2-y1) \*\* 2)

F\_in[element\_number-1] = E \* S \* (L/le-1)

print(F\_in)

save\_data(U, F\_in, elementnum, nodenum)