Санкт-Петербургский политехнический университет Петра Великого

Высшая школа теоретической механики, ФизМех

Отчёт по индивидуальному заданию №5

**Тема «Метод конечных элементов. Растяжение - сжатие упругих стержней»**

По дисциплине "Вычислительная механика"

Выполнила студентка гр. 5030103/00201: Е.С.Сорокопудова

Преподаватель: Е. Ю. Витохин

Санкт-Петербург

2023

# Постановка задачи

Необходимо провести расчет плоской фермы (см. рисунок 1) под действием нагрузки 𝐹. Нагрузка прикладывается на нижний пояс. Закреплены нижний левый и нижний правый узлы. Требуется определить перемещения узлов фермы и усилия в стержнях.

Расчеты требуется провести с помощью самостоятельной реализации метода конечных разностей для данной фермы (алгоритм см. в разделе «Описание метода решения») и с помощью среды Abaqus. Полученные двумя способами результаты нужно сравнить и представить в виде таблиц и полей перемещений и усилий.

Визуализацию результатов собственной реализации требуется оформить с помощью пакета ParaView.

Также даны следующие данные: площадь сечения стержня равна , модуль Юнга равен 2∙ Па, заданная нагрузка равна Н.

Дана плоская ферма заданной формы.

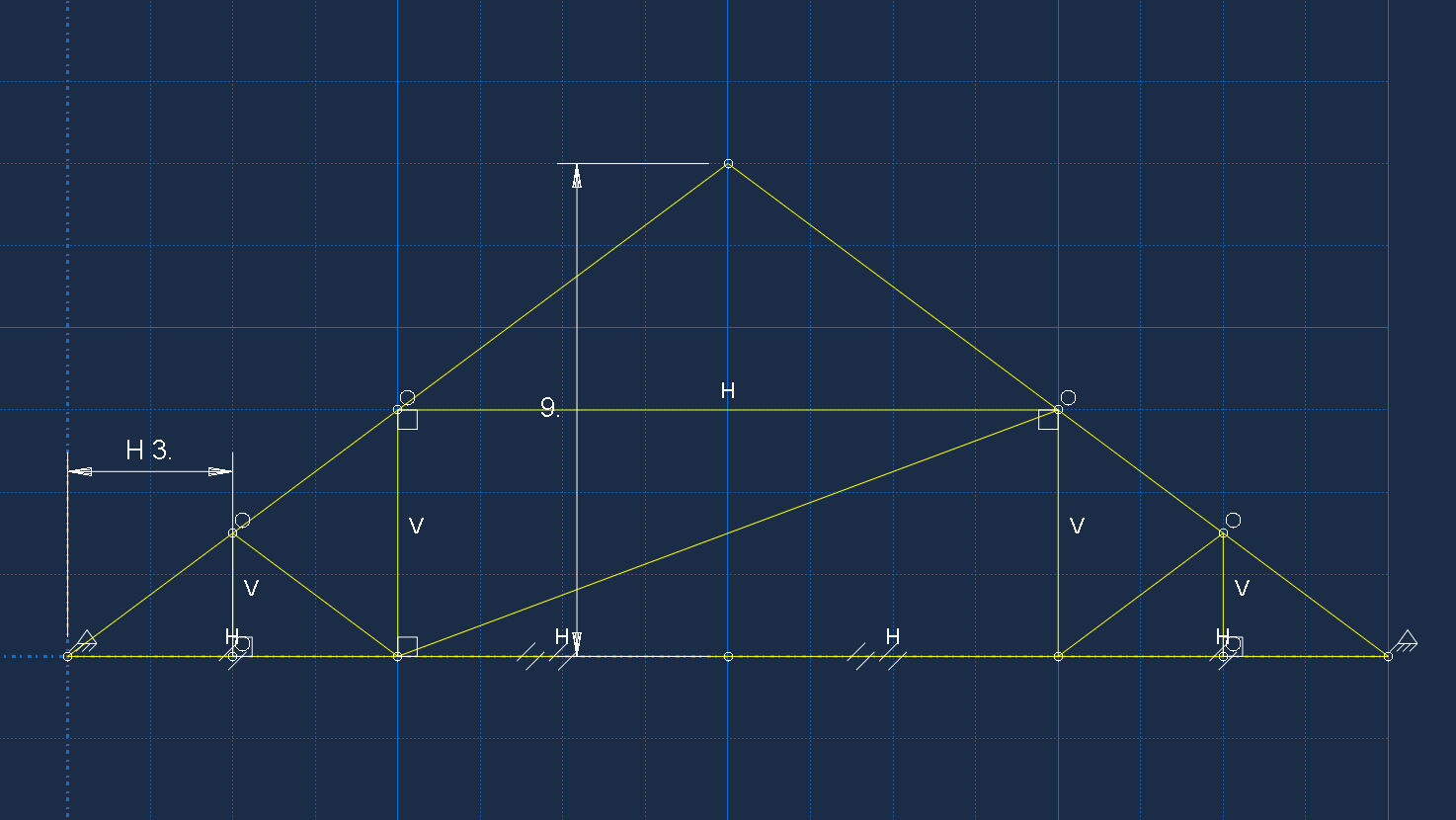


Рисунок 1. Заданная ферма

# Описание метода решения

Для решения задачи будет использоваться метод конечных элементов, который использует подход, заключающийся в нахождении минимума функционала потенциальной энергии системы. Такой подход также называют вариационным.

Полная потенциальная энергия системы Π равняется разности потенциальной энергии внутренних деформаций Λ и потенциальной энергии или работы внешних сил Ω:

Π = Λ − Ω.

Потенциальная энергия Π бесконечно малого объёма 𝑑𝑉 определяется так:

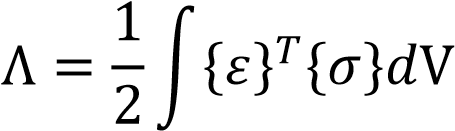
𝑑Π = 𝑑Λ − 𝑑Ω.

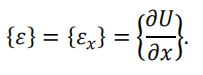
Далее определим потенциальную энергию внутренних сил, записав её для бесконечно малого объёма:

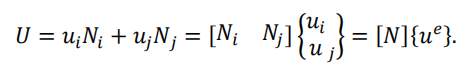
𝑑Λ =  { }𝑇{𝜎}𝑑V,

Где {Ɛ}𝑇 – вектор-столбец деформации, {𝜎} – вектор-столбец напряжений.

Для конечного объёма имеет вид

.

Вектор-столбец деформаций стержня состоит из одной компоненты x, которая описывает деформацию вдоль оси стержня, являющейся производной от перемещения вдоль этой оси:

Заменим непрерывные перемещения узловыми с помощью функции формы:

Одномерный конечный элемент стержня состоит из двух степеней свободы – перемещений вдоль оси 𝑥 с индексом 𝑖 и правого узла с индексом 𝑗. Он имеет две функции формы, по одной на каждую степень свободы.

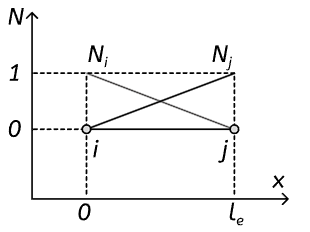
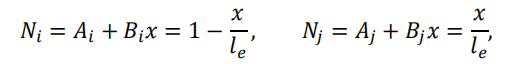
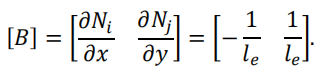


Рисунок 2. Конечный элемент линейного одномерного стержня

которые имеют линейный вид и равны:



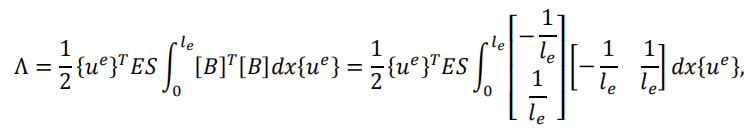
Где – длина одномерного стержня 𝑥.Введём матрицу градиентов, которая будет состоять из производных от функций форм:



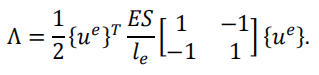
Подставим выражение для непрерывных перемещений в выражение для деформации с учётом матрицы градиентов, вектор-столбец деформации можно записать в виде произведения матрицы градиентов на вектор-столбец узловых перемещений:

Так как рассматривается линейно-упругий изотропный стержень, напряжения будут рваны произведению модуля Юнга 𝐸 на деформацию:

Подставим вектор-столбцы деформаций напряжения и напряжения в выражение потенциальной энергии внутренних деформаций, затем подставим в матрицу градиентов:



Где 𝑆 – площадь сечения стержня, которая появилась в результате вычисления интеграла по направлениям

После вычисления произведения матриц и интеграла выражение для внутренней энергии принимает вид:

Работа внешних сил Ω включает в себя работу сосредоточенных Ωс, объёмных Ω𝑉 и поверхностных Ω𝑠 сил:



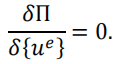
Которые можно вычислить следующим образом:



Где {𝑃𝑐}, {𝑃𝑉} и {𝑃𝑆} – векторы-столбцы сосредоточенных, объёмных и поверхностных сил соответственно.

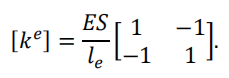
Таким образом после нахождения работы внешних сил и работы внутренних сил потенциальная энергия системы примет вид:



В статической задаче перемещения под действием внешних нагрузок отвечают минимум функционала потенциальной энергии, который достигается, когда первая вариация равна нулю:

Вычислив первую вариацию выражения для потенциальной энергии:



Первое слагаемое обозначим [𝑘𝑒] и будем называть матрицей жесткости конечного элемента:

Второе слагаемое обозначим [𝑓𝑒] и назовём вектор-столбцом усилий:

[𝑓𝑒] = {𝑃𝑐} + ∫𝑉[𝑁]𝑇{𝑃𝑉}𝑑𝑉 + ∫𝑙[𝑁]𝑇{𝑃𝑆}𝑑𝑥.

С учётом обозначений выражение можно записать в следующем виде:

[𝑘𝑒]{𝑢𝑒} = [𝑓𝑒].

С помощью этого выражения можно найти удлинение стержня, состоящего из одного конечного элемента, разложенного вдоль оси 𝑥, под действием нагрузки, направленной вдоль оси стержня. Далее рассмотрим стержень, который расположен в плоскости 𝑥𝑦 и наклонен под углом 𝛼 к оси 𝑥 так, как показано на Рис.2.

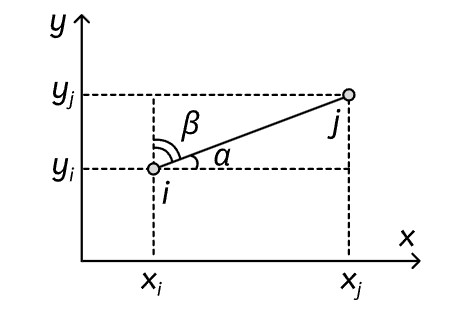


Рисунок 3. Конечный элемент стержня в плоскости

В таком стержне будет четыре степени свободы – по две в каждом узле:



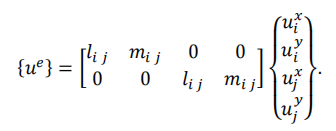
Перемещения вдоль оси стержня можно получить с помощью следующих формул:



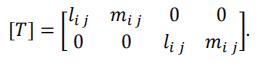
Где косинусы углов можно получить с помощью геометрической интерпретации рисунка:



С учётом обозначений матрицу перемещений можно записать следующим образом:



Левый множитель назовём матрицей трансформации:



Тогда выражение, связывающее перемещение вдоль оси стержня с перемещениями в плоскости, кратко можно записать так:

{𝑢𝑒} = [𝑇]{}.

Аналогичным образом связаны усилия в плоскости и вдоль оси стержня:

{𝑓𝑒} = [𝑇]{}.

Принимая во внимание полученные выражения, запишем уравнение

[𝑘𝑒]{𝑢𝑒} = [𝑓𝑒], описывающее деформацию вдоль оси стержня, для случая деформации в плоскости 𝑥𝑦:

[𝑘𝑒][𝑇]{} = [𝑇]{}.

Умножим слева на транспонированную матрицу трансформации [𝑇]𝑇:

[𝑇]𝑇[𝑘𝑒][𝑇]{} = {}.

Обозначим произведение матриц трансформации на одномерную матрицу жёсткости плоской матрицей жёсткости:

[] = [𝑇]𝑇[𝑘𝑒][𝑇].

Для того, чтобы решить задачу деформирования стержневых конечных элементов нужно из плоских матриц жёсткости [] составить глобальную матрицу жёсткости [𝐾]. Для этого размерность матрицы жёсткости элемента приводится в соответствие с размерностью глобальной матрицы жёсткости. Размерность глобальной матрицы жёсткости 𝑁 × 𝑁, где 𝑁 – общее

Изображение выглядит как Шрифт, часы, число, линия

Автоматически созданное описаниеколичество степеней свободы, без учёта граничных условий (закрепления в узлах). Строки и столбцы заполняются с учётом глобальной нумерации узлов. После этого матрицы жёсткости элементов суммируются, в результате чего получается глобальная матрица жёсткости:

Изображение выглядит как Шрифт, линия, диаграмма, белый

Автоматически созданное описаниеАналогичным образом суммируются векторы-столбцы нагрузок:

В итоге уравнение (основное уравнение метода конечных элементов), описывающее деформацию стержней для системы конечных элементов, примет вид:

[𝐾]{𝑈} = {𝐹},

Где {𝑈} – глобальный вектор-столбец перемещений.

Глобальная матрица жёсткости [𝐾] обладает четырьмя свойствами:

* Ленточная
* Растяжная
* Симметричная
* Вырожденная

Для того, чтобы решить СЛАУ, нужно добавить уравнения для граничных условий, убрать при этом из системы уравнения, которые соответствуют степеням свободы граничных условий.

# Результаты

Приведём номера узлов и номера элементов для нашей фермы (см. рисунок 2).

Изображение выглядит как легкий, темный

Автоматически созданное описание

Рисунок 4. Нумерация узлов и стержней

Полученные поля перемещений по оси в Abaqus и Matlab:

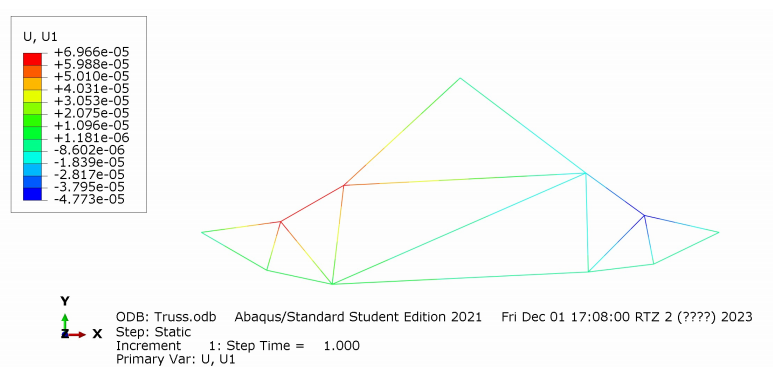
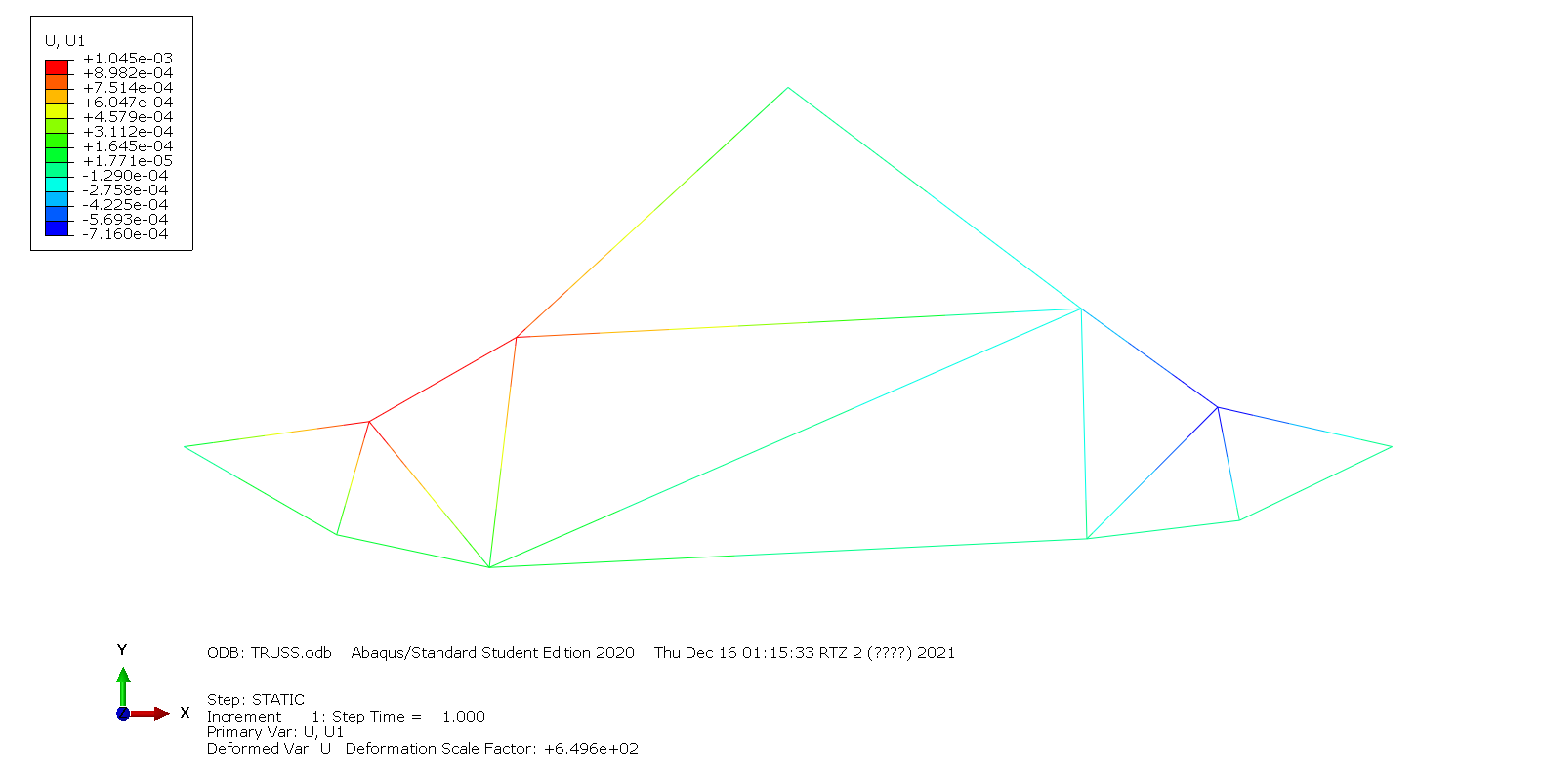


Рисунок 5. Поле перемещений по x в Abaqus

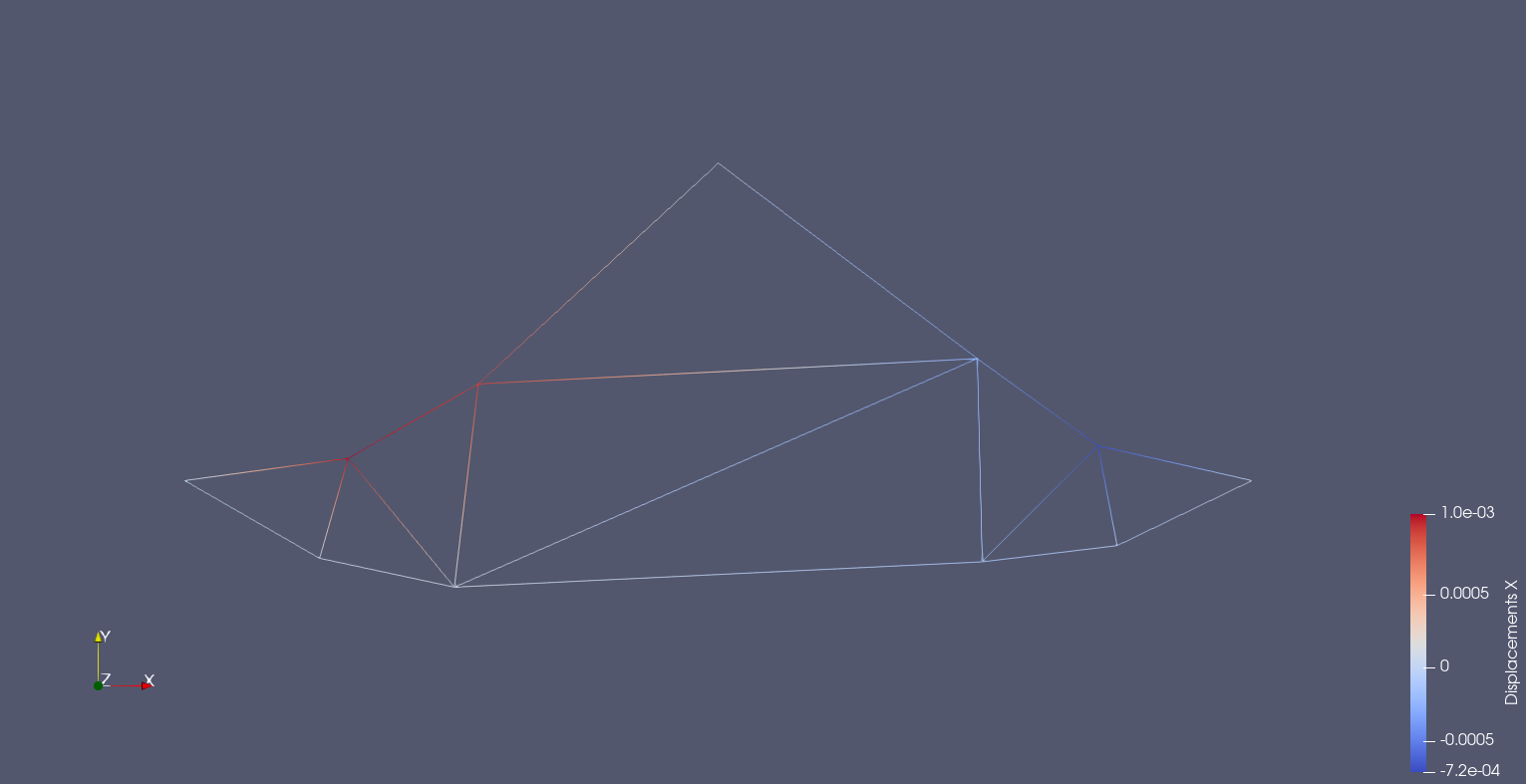


Рисунок 6. Поле перемещений по x в ParaView

Полученные поля перемещений по оси в Abaqus и Matlab:

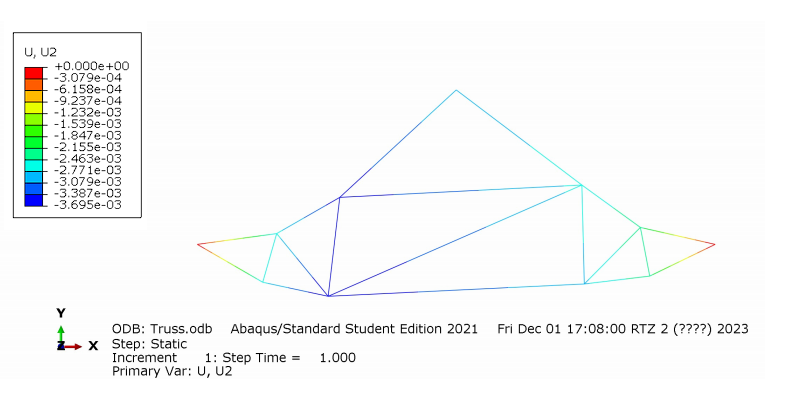


Рисунок 7. Поле перемещений по y в Abaqus

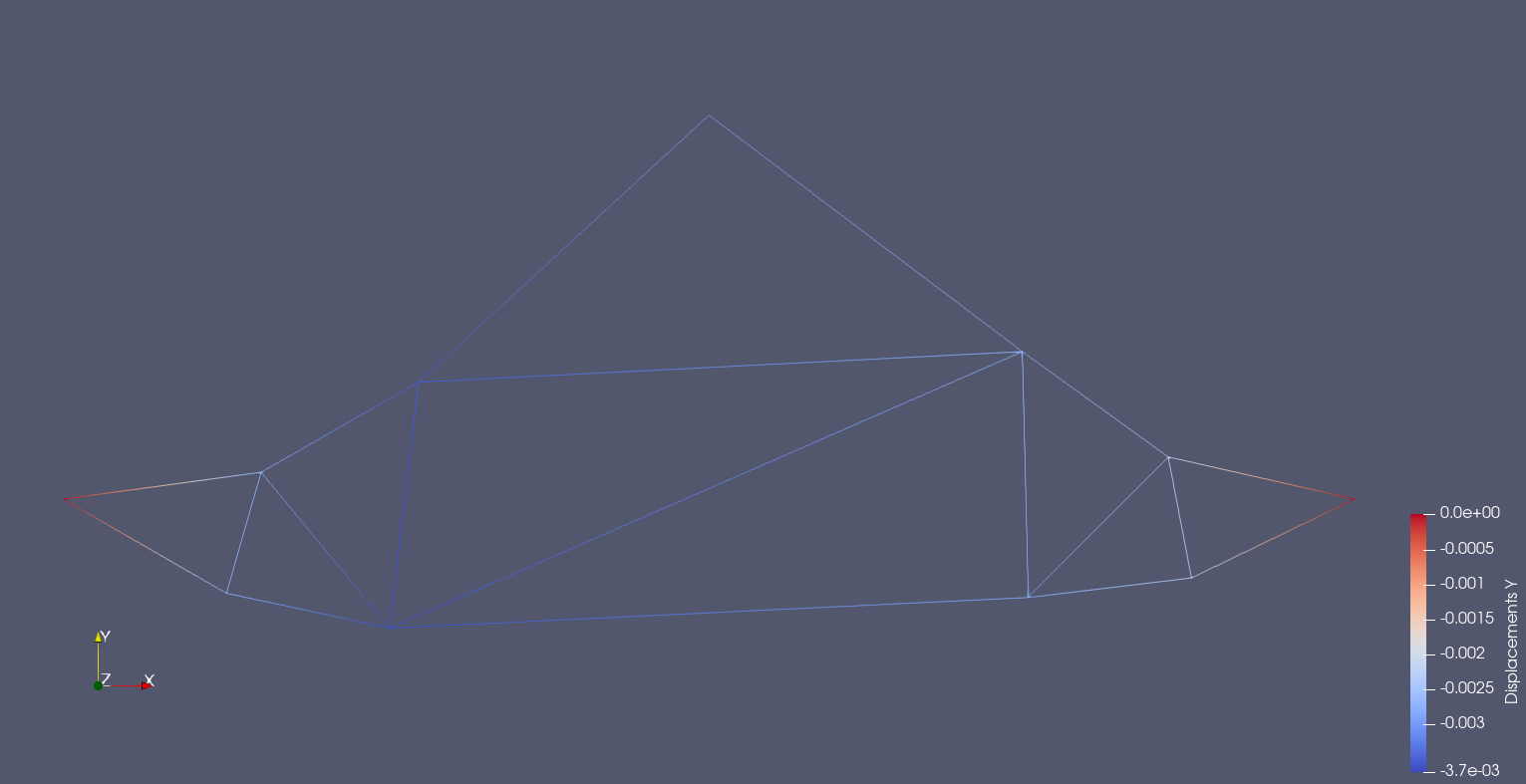


Рисунок 8. Поле перемещений по y в ParaView

Полученные поля усилий в Abaqus и Matlab:

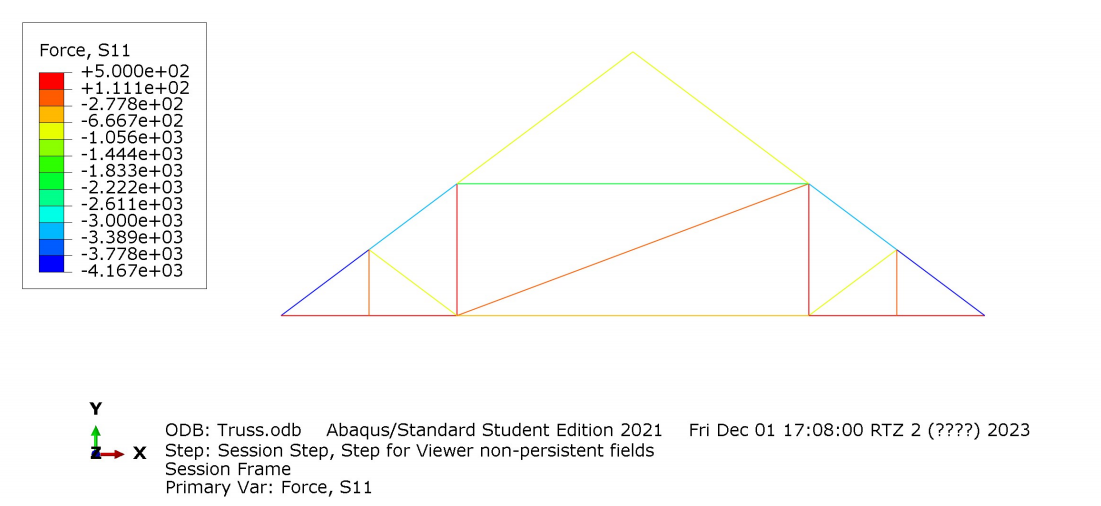


Рисунок 9. Поле усилий в Abaqus

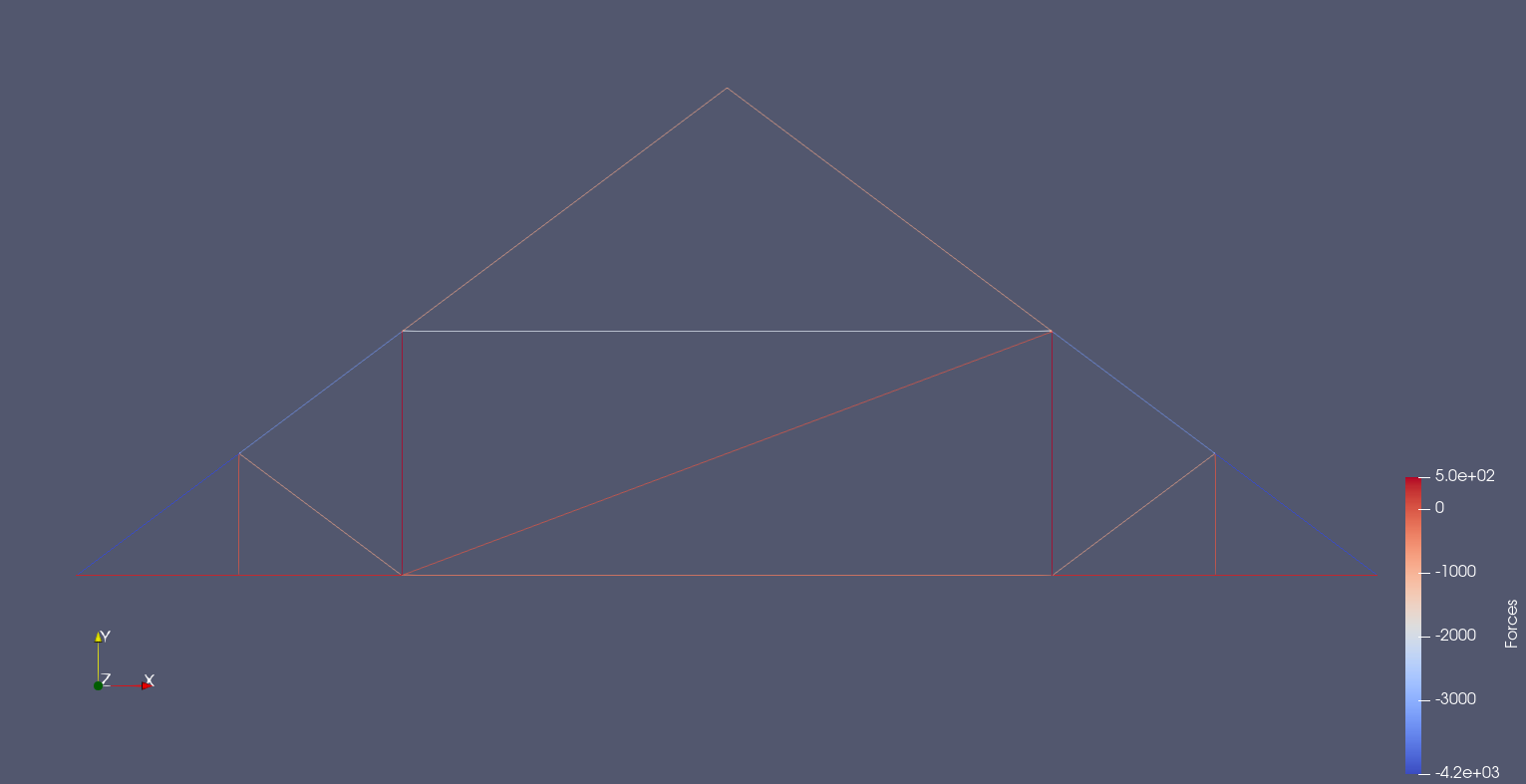


Рисунок 10. Поле усилий в ParaView

Таблицы со сравнением значений перемещений, полученных с помощью Abaqus и Matlab как по оси , так и по оси .

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 11. Сравнение значений перемещений по x, м

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 12. Сравнение значений перемещений по y, м

Таблица со сравнением полученных значений усилий:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 13. Сравнение значений для усилий, Н

# Выводы

Таким образом, была решена задача растяжения-сжатия упругого стержня с использованием вариационного подхода.

Значения перемещений узлов фермы и усилий, возникающих в ней, полученные с помощью алгоритма на Matlab и с помощью среды Abaqus, оказались достаточно близки друг к другу.

Для сравнения результатов были приведены таблицы и изображения полей перемещений узлов и усилий, возникающих в стержнях.

# Код программы

for i = 1:n\_elements

element = Elements(i, 1:3);

element\_1 = element(2);

element\_2 = element(3);

node\_1 = Nodes(element\_1,1:3);

node\_2 = Nodes(element\_2,1:3);

node\_1\_x = node\_1(2);

node\_1\_y = node\_1(3);

node\_2\_x = node\_2(2);

node\_2\_y = node\_2(3);

l = el\_length(node\_1\_x, node\_1\_y, node\_2\_x, node\_2\_y);

ke = ((E\*A)/(l)) \* [1,-1;-1,1];

l\_ij = (node\_2\_x - node\_1\_x)/l;

m\_ij = (node\_2\_y - node\_1\_y)/l;

T = [l\_ij, m\_ij, 0, 0; 0, 0,l\_ij, m\_ij];

ke\_gl = (T' \* ke \* T);

q = node\_1(1);

j = node\_2(1);

N = [q, j];

K(2\*q-1,2\*q-1) = K(2\*q-1,2\*q-1)+ke\_gl(1,1);

K(2\*q,2\*q) = K(2\*q,2\*q)+ke\_gl(2,2);

K(2\*j-1,2\*j-1) = K(2\*j-1,2\*j-1)+ke\_gl(3,3);

K(2\*j,2\*j) = K(2\*j,2\*j)+ke\_gl(4,4);

K(2\*q-1,2\*q) = K(2\*q-1,2\*q)+ke\_gl(1,2);

K(2\*q-1,2\*j-1) = K(2\*q-1,2\*j-1)+ke\_gl(1,3);

K(2\*q-1,2\*j) = K(2\*q-1,2\*j)+ke\_gl(1,4);

K(2\*q,2\*q-1) = K(2\*q,2\*q-1)+ke\_gl(2,1);

K(2\*q,2\*j-1) = K(2\*q,2\*j-1)+ke\_gl(2,3);

K(2\*q,2\*j) = K(2\*q,2\*j)+ke\_gl(2,4);

K(2\*j-1,2\*q-1) = K(2\*j-1,2\*q-1)+ke\_gl(3,1);

K(2\*j-1,2\*q) = K(2\*j-1,2\*q)+ke\_gl(3,2);

K(2\*j-1,2\*j) = K(2\*j-1,2\*j)+ke\_gl(3,4);

K(2\*j,2\*q-1) = K(2\*j,2\*q-1)+ke\_gl(4,1);

K(2\*j,2\*q) = K(2\*j,2\*q)+ke\_gl(4,2);

K(2\*j,2\*j-1) = K(2\*j,2\*j-1)+ke\_gl(4,3);

end

for r = 1 : length(fixed\_nodes)

q = fixed\_nodes(r);

K(:,2\*q) = zeros(2\*n\_nodes,1);

K(2\*q,:) = zeros(1,2\*n\_nodes);

K(:,2\*q-1) = zeros(2\*n\_nodes,1);

K(2\*q-1,:) = zeros(1,2\*n\_nodes);

K(2\*q,2\*q) = 1;

K(2\*q-1,2\*q-1) = 1;

end

F\_in = zeros(1,n\_elements);

for i = 1:n\_elements

element = Elements(i, 1:3);

element\_1 = element(2);

element\_2 = element(3);

node\_1 = Nodes(element\_1,1:3);

node\_2 = Nodes(element\_2,1:3);

node\_1\_x = node\_1(2);

node\_1\_y = node\_1(3);

node\_2\_x = node\_2(2);

node\_2\_y = node\_2(3);

new\_node1\_x = node\_1\_x + u(2\*node\_1(1)-1);

new\_node1\_y = node\_1\_y + u(2\*node\_1(1));

new\_node2\_x = node\_2\_x + u(2\*node\_2(1)-1);

new\_node2\_y = node\_2\_y + u(2\*node\_2(1));

l = el\_length(node\_1\_x, node\_1\_y, node\_2\_x, node\_2\_y);

l\_new = el\_length(new\_node1\_x, new\_node1\_y,new\_node2\_x,new\_node2\_y);

eps = (l\_new - l)/l;

F = E\*eps\*A;

F\_in(i) = F;

end