

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Московский Государственный Технический Университет
имени Н.Э. Баумана»
Национальный исследовательский университет техники и
технологий
(МГТУ им. Н.Э.Баумана)

*Факультет «Робототехника и комплексная автоматизация»
Кафедра «Системы автоматизированного проектирования»
(РК-6)*

Отчет по лабораторной работе №2
По дисциплине «Прикладная механика»
На тему «Расчет статически-неопределимой балки методом конечных
элементов»

Выполнил студент группы РК6-32Б
Ершов Виталий

Москва, 2019 г.

Задача

Составить конечно-элементную программу для расчета статически неопределимой балки и проверить корректность ее работы с использованием Siemens NX 8.5.

Исходные данные: Материал балки: сталь (модуль Юнга $E = 2e11$ Па).

Сечение балки: прямоугольное (см. рисунок 1).

Геометрические параметры балки: $l = 0.1$ м, $b = 10$ мм, $h = 20$ мм

Величина нагрузки: $F = 10$ Н.

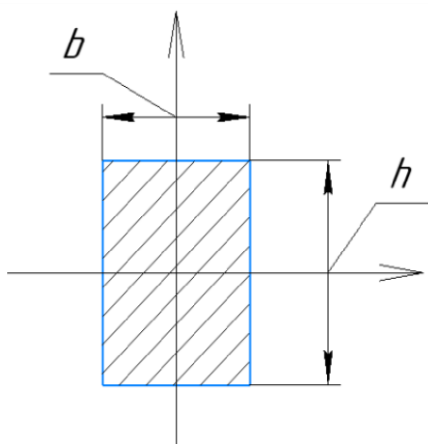
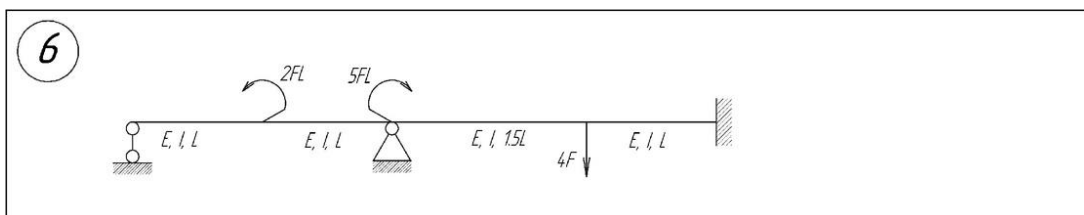


Рисунок 1. Поперечное сечение балки

Вариант задачи



Составление конечно-элементной программы в Matlab и решение исходной заданной системы

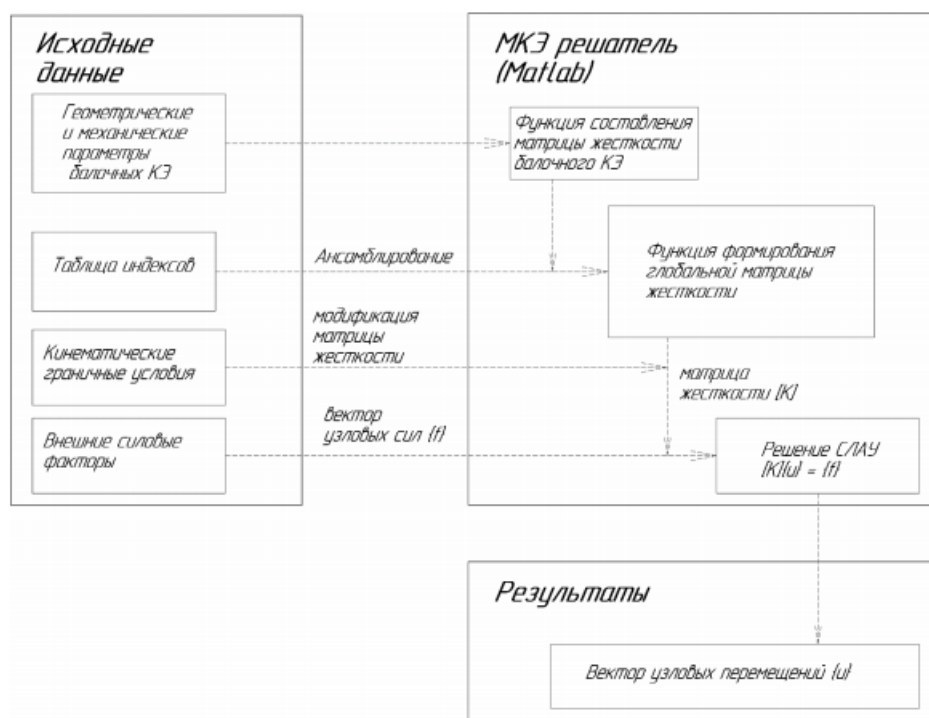


Рисунок 2. Блок-схема конечно-элементной программы в Matlab

Описание алгоритма:

1) Задаем исходные данные:

E – модуль Юнга; F – величина силы; b , h – ширина и высота сечения; l – длина единичного отрезка; J_y – момент инерции; N_{el} – кол-во КЭ в системе; E_{sys} – вектор упругих свойств системы; L_{sys} – вектор длин конечных элементов; N_{dofs} – кол-во степеней свободы системы; U_{node} – вектор граничных условий; F_{node} – вектор внешних сил и моментов

2) Создаем матрицу индексов $Index_M$, хранящую номера узлов в глобальной и локальной системе координат.

3) С помощью 3 циклов заполняем глобальную матрицу жесткости K_{glob} на основе матриц жесткости отдельных конечных элементов K_e , которые предварительно считаются функцией K_{loc_calc} .

4) Накладываем кинематические условия, модифицируем матрицу жесткости. Для этого задается вектор U_{node} количеством элементов N_{dofs} . В ячейке, где узел закреплен, ставим 1, в свободных узлах ставим 0. Обнуляем столбцы и строки с номерами,

соответствующими номерам элементов, равных 1, в векторе U_node. На пересечении этих столбцов и строк ставим 1.

5) Находим и выводим на экран вектор перемещений, который равен $U = \text{pinv}(K_glob) * F_node$;

6) Выводим вектор перемещений U, переводя значения в нужные единицы измерения (мм для перемещений и градусы для углов поворота). Элементы $2*n-1$ отвечают за вертикальные перемещения узлов, а $2*m$ - за угол поворота.

Текст программы:

```
function main
format long
h = 20;
b = 10;
Jy = (b*h^3)/12;
l = 100;
E = 2e5;
F = 10;
N_el = 4;
E_sys = [E, E, E, E];
L_sys = [l, l, l*3/2, l];
N_dofs = 2*(N_el + 1);
K_glob = zeros(N_dofs);
U_node = [1,0, 0,0, 1,0, 0,0, 1,1];
F_node = [0,0, 0,2*F*l, 0,-5*F*l, -4*F,0, 0,0];
K_loc = zeros(4);
Index_M = [1:4;
            3:6;
            5:8;
            7:10;];
for i = 1:N_el
    K_loc = K_loc_calc(L_sys(i), E_sys(i), Jy);
    for j = 1:4
        for k = 1:4
            K_glob(Index_M(i, j), Index_M(i, k)) = K_glob(Index_M(i, j), Index_M(i, k)) + K_loc(j,k);
        end
    end
end
for i = 1:N_dofs
    if(U_node(i) == 1)
        K_glob(i, :) = 0;
        K_glob(:, i) = 0;
        K_glob(i, i) = 1;
    end
end
U = pinv(K_glob)*F_node;
for i = 1:N_dofs
```

```

if(rem(i,2) == 1)
    disp(sprintf('%0.8f mm',U(i)))
else
    disp(sprintf('%0.8f deg',U(i)*180/pi))
end
end
end
function K = K_loc_calc(L, E, J)
K = [12*E*J/(L^3), 6*E*J/(L^2), -12*E*J/(L^3), 6*E*J/(L^2);
     6*E*J/(L^2), 4*E*J/L, -6*E*J/(L^2), 2*E*J/L;
     -12*E*J/(L^3), -6*E*J/(L^2), 12*E*J/(L^3), -6*E*J/(L^2);
     6*E*J/(L^2), 2*E*J/L, -6*E*J/(L^2), 4*E*J/L];
end

```

Вывод программы:

```

0.00000000 mm
0.00322982 deg
0.00516532 mm
0.00241890 deg
-0.00000000 mm
-0.00860823 deg
-0.00576581 mm
0.00337343 deg
0.00000000 mm
0.00000000 deg

```

Расчет в программе Siemens NX

CAD-модуль, создание геометрической модели:

- 1) Создаем новый файл (модель моделирования)
- 2) Создаем плоский эскиз (одну линию вдоль оси X) и задаем размер.
- 3) Сохраняем файл.

CAE-модуль, создание конечно-элементной модели кронштейна

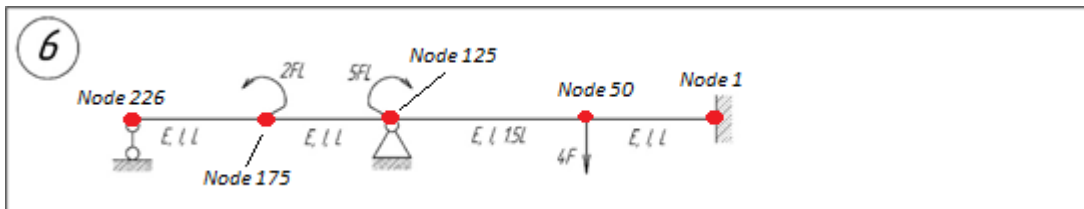
- 1) Создаем новый файл КЭ модели
- 2) Связываем файл с моделью, применив отображение геометрии прямых (В появившемся окне: «Геометрия»->«Опции геометрии» ->«прямые»)
- 3) Создаем конечно-элементную сетку. Выбираем параметр "1D сетка" и применяем к нашей модели. Размер элемента выбираем равным 2мм.
Допуск объединения узлов выбираем равным 0.01.

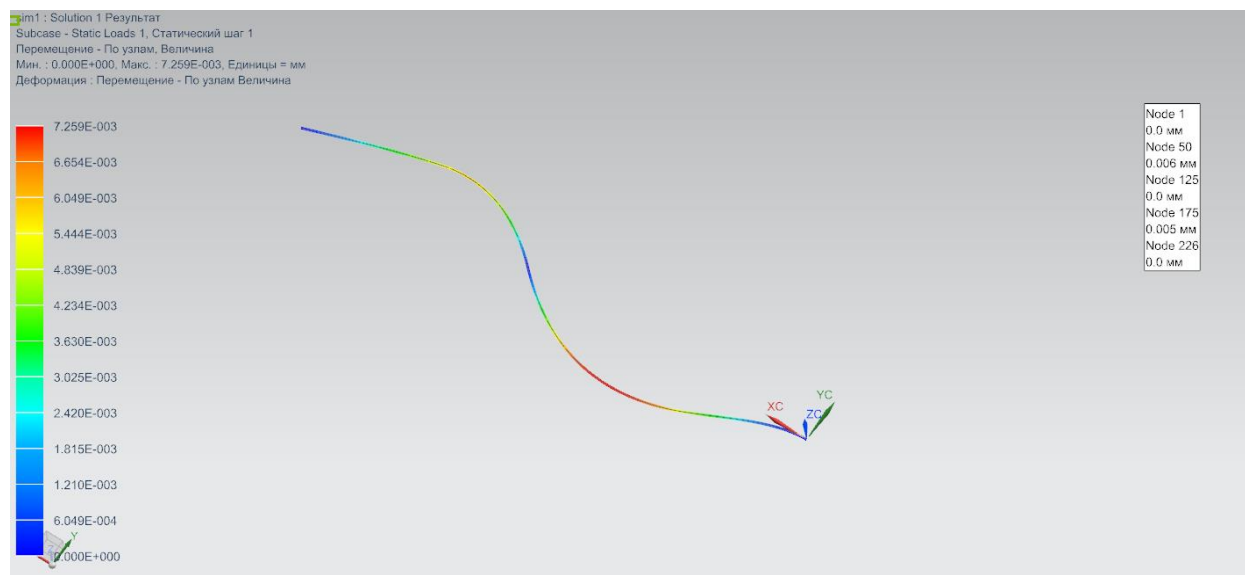
- 4) Создаем поперечное сечение прямоугольной формы с нужными шириной и высотой ($DIM1=10\text{мм}$, $DIM2=20\text{мм}$).
- 5) Выбираем тип материала "AISI_STEEL_1005".
- 6) Сохраняем файл.

CAE-модуль, решение прочностной задачи

- 1) Создаем новый файл симуляции
- 2) Связываем с файлом конечно-элементной модели, для этого в появившемся окне нажимаем «ОК».
- 3) Задаем граничные условия. Выбираем в верхнем меню «Тип закрепления» -> «Ограничение задаваемое пользователем». В левом верхнем углу в фильтре выбора ставим «Узел». Выбираем нужный узел и ставим «фиксировано» в необходимом месте.
- 4) Задаем внешний силовой фактор. Выбираем в верхнем меню «Тип нагрузки» -> «Сила». Выбираем нужный узел и прикладываем к нему силу. Задаем вектор направления силы.
- 5) Задаем момент. Выбираем в верхнем меню «Тип нагрузки» -> «Момент». Выбираем нужный узел и прикладываем к нему момент.
- Задаем вектор направления момента.
- 6) Сохраняем файл.
- 7) Запускаем решение.
- 8) Знакомимся с результатами.

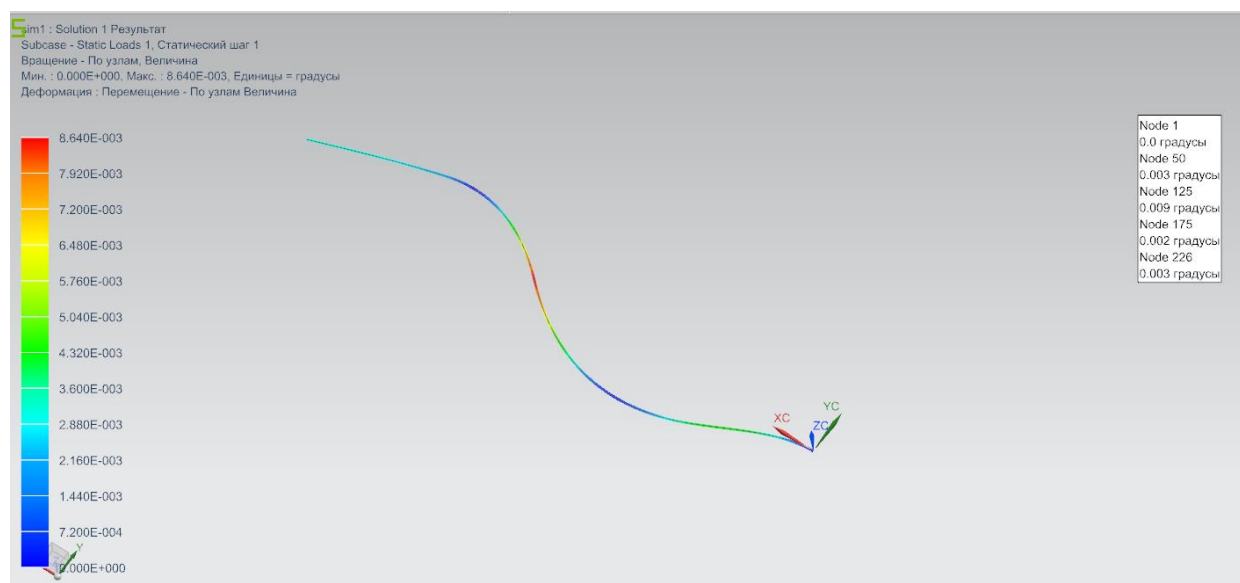
Для получения точных результатов в NX воспользуемся инструментом "аннотация" в режиме просмотра результатов расчета для явного отображения значений в узлах.





(Рис.1 – перемещение по узлам)

Node 1
0.0 mm
Node 50
0.006 mm
Node 125
0.0 mm
Node 175
0.005 mm
Node 226
0.0 mm



(Рис.2 – вращение по узлам)

Node 1
0.0 градусы
Node 50
0.003 градусы
Node 125
0.009 градусы
Node 175
0.002 градусы
Node 226
0.003 градусы

Таблица №1 (Перемещения)

	W_1, mm	W_2, mm	W_3, mm	W_4, mm	W_5, mm
<i>Matlab</i>	0	0.00517	0	-0.00577	0
<i>Nx</i>	0	0.005	0	-0.006	0
%		0,034		0,0398	

Вращение

	θ_1, deg	θ_2, deg	θ_3, deg	θ_4, deg	θ_5, deg
<i>Matlab</i>	0.00323	0.00242	-0.00861	0.00337	0
<i>Nx</i>	0.003	0.002	-0.009	0.003	0
%	0,07	0,21	0,045	0,123	

Вывод:

Из приведенных выше данных можно сделать вывод о том, что при решении методом конечных элементов (в NX и в MatLab) были получены примерно одинаковые результаты. Так как при решении использовался один и тот же метод (МКЭ), небольшая разница в полученных значениях связана с компьютерной погрешностью.