

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 РК6-33Б:
Ершков Алексей

Москва, 2019 г.

Задача

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

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

Сечение балки: прямоугольное (Рисунок 1).

Геометрические параметры балки: $l = 0.1$ м, $b = 10$ мм, $h = 20$ мм Величина нагрузки: $F = 10$ Н.

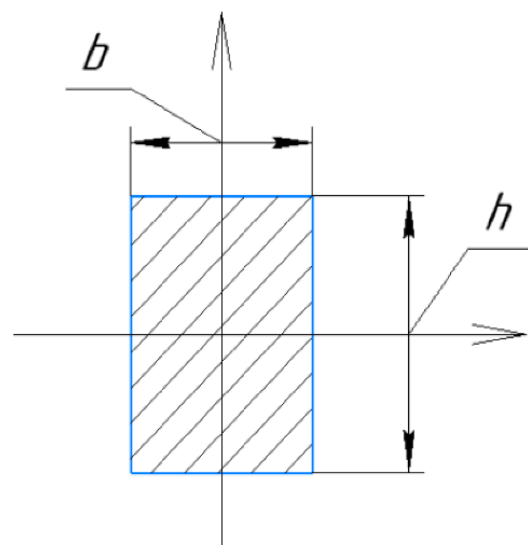
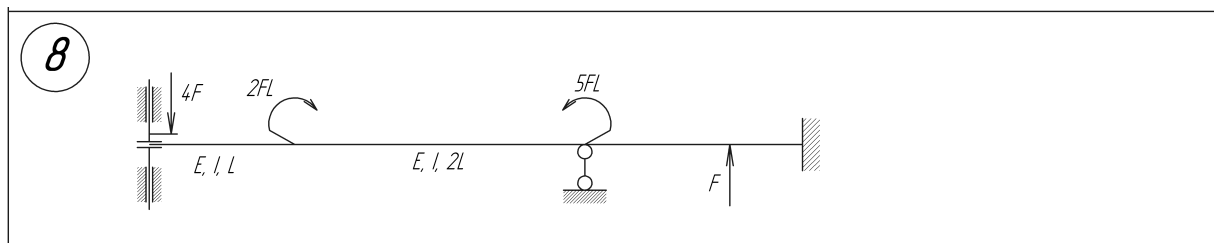


Рисунок 1. Поперечное сечение балки

Вариант



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

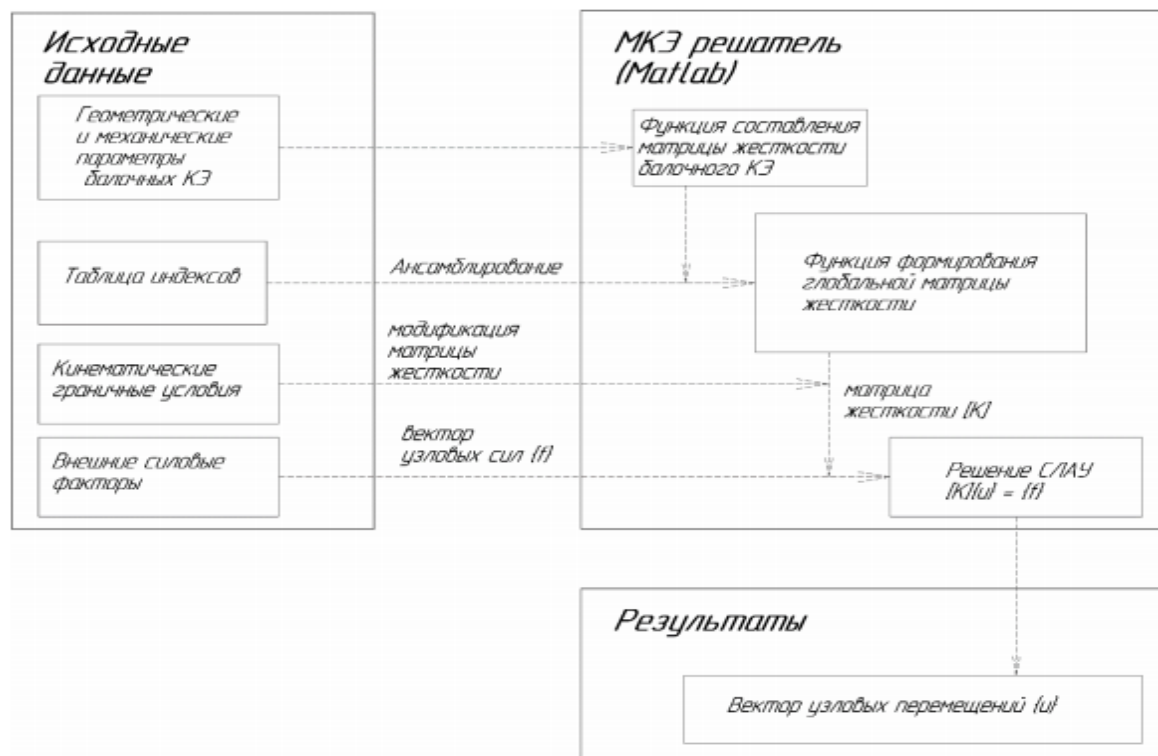


Рисунок 2. Блок-схема конечно-элементной программы в Matlab

Описание алгоритма решения

- 1) Задаем исходные данные:
 - b, h – ширина и высота сечения;
 - J_y – момент инерции;
 - l – длина единичного отрезка;
 - E – модуль Юнга;
 - F – величина силы;
 - 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;

$K_GlobalMatrix$ – глобальная матрица жёсткости;

$Vector_U$ – вектор граничных условий;

$Vector_F$ – вектор внешних сил и моментов;

$K_LocalMatrix$ – локальная матрица жесткости;

$IndexMatrix$ – матрица индексов.

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

3) С помощью вложенных циклов заполняем глобальную матрицу жесткости $K_GlobalMatrix$ на основе матриц жесткости конечных элементов, которые считаются функцией $K_LocalMatrixCalc$.

4) Накладываем кинематические граничные условия, модифицируем матрицу жесткости. Для этого задается вектор $Vector_U$ количеством элементов $DegreeOfFreedomNumbers$. 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) Вычисляем вектор перемещений $U = pinv(K_GlobalMatrix)*Vector_F$;

6) Выводим на экран вектор перемещений U , переводя значения в нужные единицы измерения (мм для перемещений и градусы для углов поворота). Элементы, стоящие на нечетных позициях, отвечают за вертикальные перемещения узлов, а на четных - за угол поворота.

Текст программы

```
function prik_meich_lab2
    format long
    h = 20; % Высота сечения, мм
    b = 10; % Ширина сечения, мм
    Jy = (b*h^3)/12; % Момент инерции сечения
    l = 100; % Длина, мм
    E = 2e5; % Модуль Юнга
    F = 10; % Сила, Н
    NumberOfElements = 4; % Кол-во КЭ в системе
    E_System = [E, E, E, E]; % Вектор упругих свойств системы
    LenghtSystem = [1, 2*1, 1, 1]; % Вектор длин конечных элеметов
    DegreesOfFreedomNumber = 2*(NumberOfElements + 1); % Кол-во степеней свободы системы
    K_GlobalMatrix = zeros(DegreesOfFreedomNumber); % Глобальная матрица жёсткости
    Vector_U = [0,1, 0,0, 1,0, 0,0, 1,1]; % Вектор граничных условий
    Vector_F = [-4*F,0, 0,-2*F*1, 0,5*F*1, F,0, 0,0]; % Вектор внешних сил и мементов
    K_LocalMatrix = zeros(4); % Локальная матрица жёсткости
    IndexMatrix = [1:4; % Матрица индексов
        3:6;
        5:8;
        7:10;];

    for i = 1:NumberOfElements % Вычисление матрицы жёсткости для каждого КЭ и ансамблирование
        K_LocalMatrix = K_LocalMatirxCalc(LenghtSystem(i), E_System(i), Jy);
        for j = 1:4
            for k = 1:4
                K_GlobalMatrix(IndexMatrix(i, j),IndexMatrix(i, k)) = K_GlobalMatrix(IndexMatrix(i, j),IndexMatrix(i, k))+K_LocalMatrix(j,k);
            end
        end
    end

    for i = 1:DegreesOfFreedomNumber % Наложение кинематических граничных условий
        if(Vector_U(i) == 1)
            K_GlobalMatrix(i, :) = 0;
            K_GlobalMatrix(:, i) = 0;
            K_GlobalMatrix(i, i) = 1;
        end
    end

    U = pinv(K_GlobalMatrix)*Vector_F';

    for i = 1:DegreesOfFreedomNumber % Вывод вектора перемещений на экран
        if(rem(i,2) == 1)
            fprintf('%f mm\n',U(i)) % Вывод перемещений в мм
        else
            fprintf('%f deg\n',U(i)*180/pi) % Вывод угла поворота в градусах
        end
    end
end

function K = K_LocalMatirxCalc(L, E, J) % Вычисление матрицы жесткости для плоского балочного элемента
    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
```

Результат работы программы

```
>> prik_meich_lab2  
-0.092946428571430 mm  
-0.000000000578523 deg  
-0.069285714285716 mm  
0.423214285714293 deg  
0.0000000000000703 mm  
0.069642857142861 deg  
0.002366071428571 mm  
-0.017410714285715 deg  
0.000000000000000 mm  
0.000000000000000 deg
```

Выполнение расчета исходной системы в программе Siemens NX

CAD-модуль, создание геометрической модели:

- 1) Создаем новый файл (модель моделирования);
- 2) Создаем плоский эскиз (строим 4 соединённых линии вдоль оси x) и задаем размер;
- 3) Сохраняем файл.

CAE-модуль, создание конечно-элементной модели балочного элемента:

- 1) Создаем новый файл КЭ модели;

- 2) Связываем файл с моделью, применив отображение геометрии прямых (Геометрия->Опции геометрии ->Прямые);
- 3) Создаем конечно-элементную сетку. Выбираем параметр "1D сетка" и применяем к модели. Число элементов выбираем равным 1. Допуск объединения узлов выбираем равным 0.001.
- 4) Создаем поперечное сечение прямоугольной формы с нужными шириной и высотой;
- 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;
- 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) Получаем результат.

Результат расчёта в Siemens NX

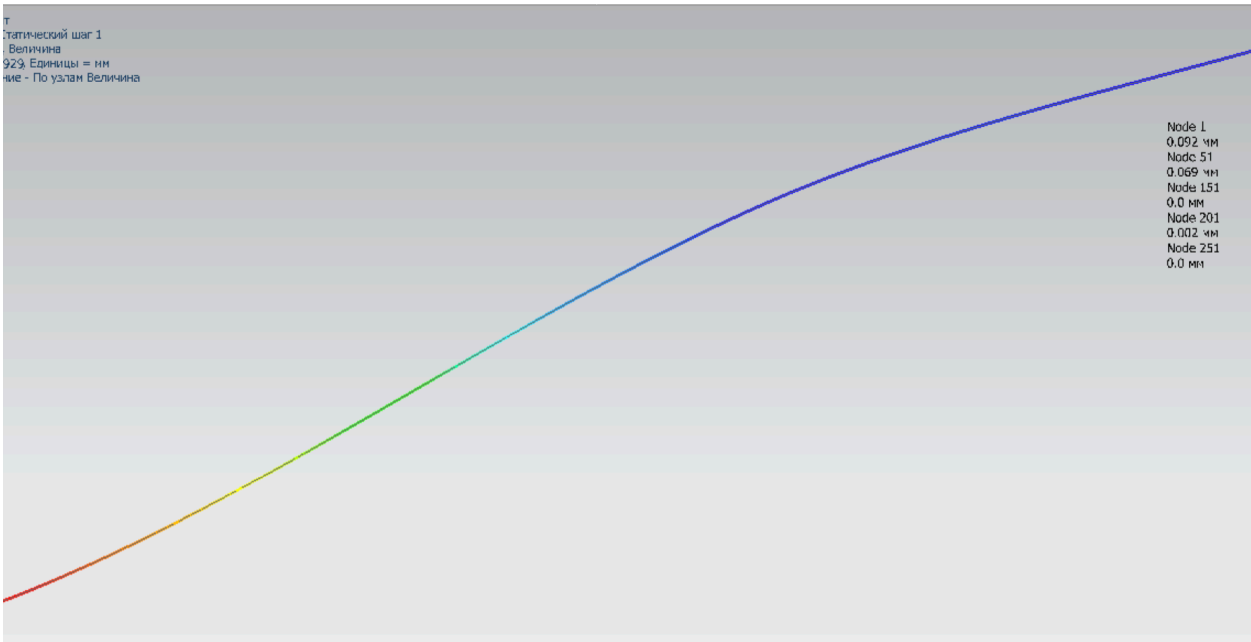


Рис. 3 - Перемещения в узлах

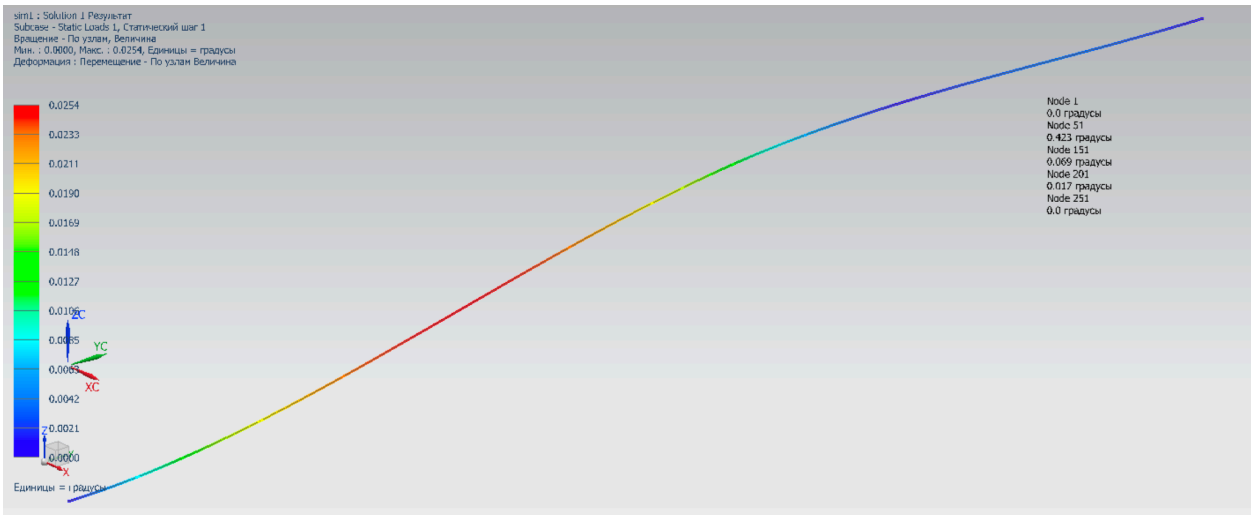


Рис. 4 - Вращения в узлах

	W_1 , mm	θ_1 , °	W_2 , mm	θ_2 , °	W_3 , mm	θ_3 , °	W_4 , mm	θ_4 , °	W_5 , mm	θ_5 , °
MatLab	0.0929 464	0	0.0692 857	0.4232 142	0	0.0696 428	0.0023 660	0.0174 107	0	0
Siemens Nx	0.092	0	0.069	0.423	0	0.069	0.002	0.017	0	0

Вывод

Сравнив полученные в NX и MatLab значения, можно сделать вывод, что полученные в обеих программах результаты приблизительно равны, так как в обоих случаях использовался один и тот же метод (метод конечных элементов).