

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение Высшего
образования «Московский государственный технический университет имени

Н.Э.Баумана (национальный исследовательский университет)» (МГТУ им Н.Э.
Баумана)

Факультет «Робототехника и комплексная автоматизация»
Кафедра «Системы автоматизированного проектирования» (РК6)

Отчет
По лабораторной работе №2
по дисциплине «Прикладная механика»

Выполнил: студент группы РК6-34Б, Куртбуганов И.П.

Проверил: декан факультета РК, Шашурин Г.В.

Москва

2019

Расчёт статически-неопределимой балки методом конечных элементов

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 (см. рисунок 1).

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

Величина нагрузки: $F = 10$ Н.

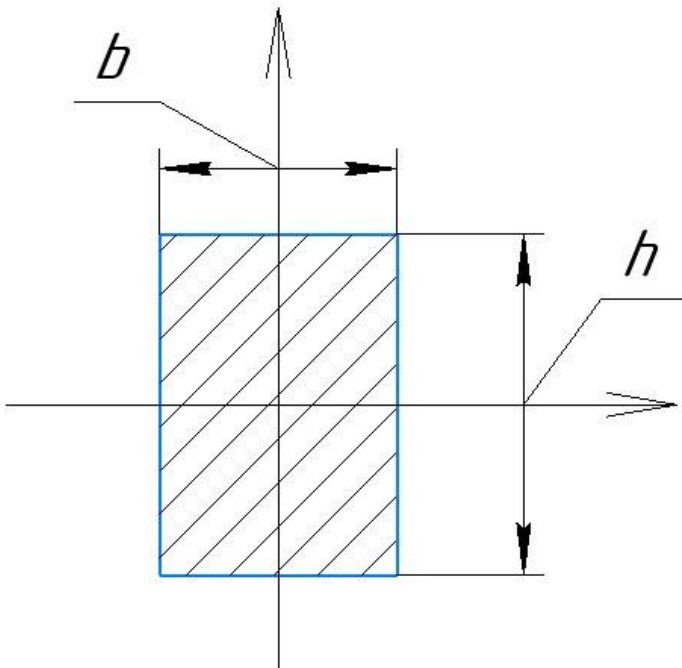
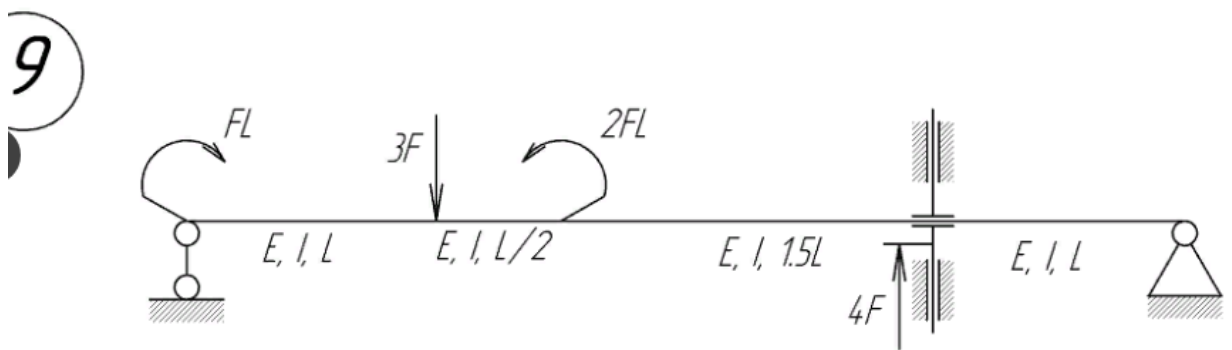


Рисунок 1. Поперечное сечение балки



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

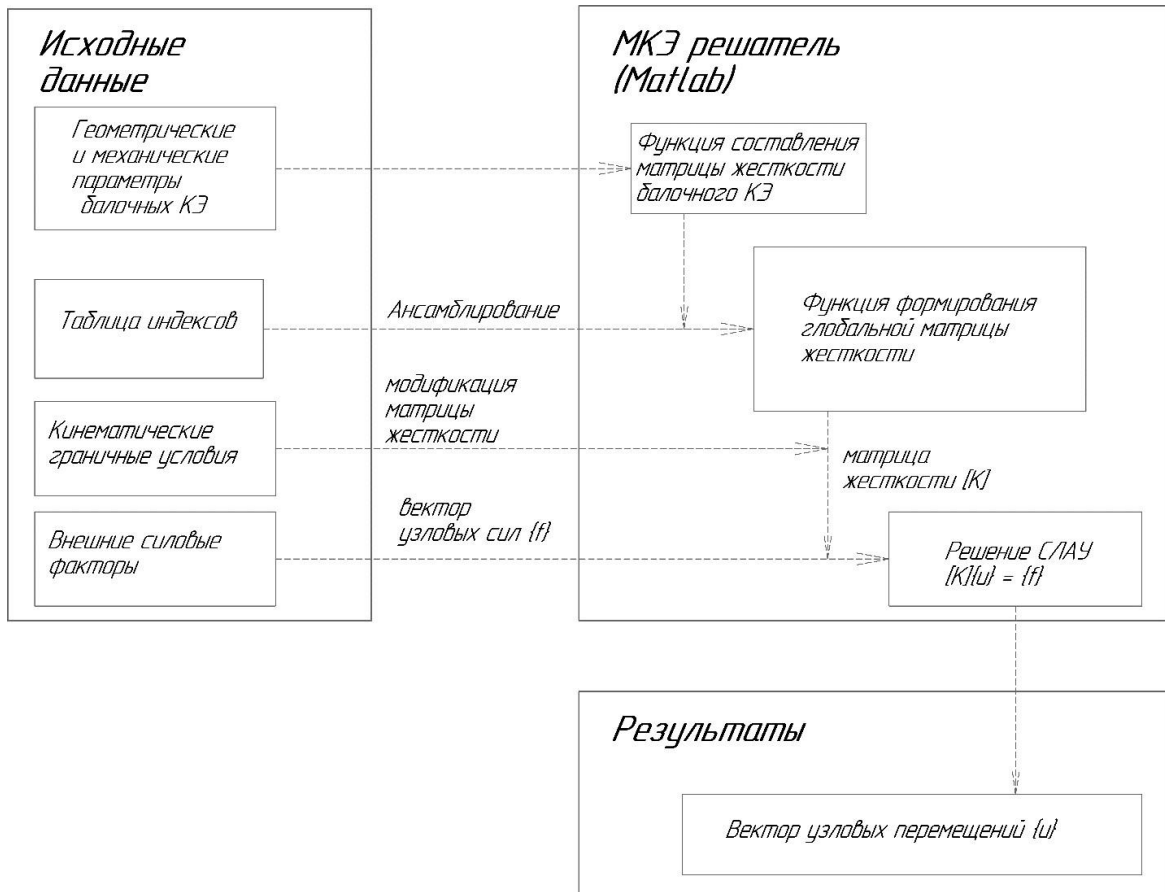


Рисунок 2. Блок-схема конечно-элементной программы в Matlab

Последовательность решения

1) Конечно-элементное разбиение системы

Выбор глобальной системы координат (с.к), назначение числа балочных конечных элементов (к.э.) системы N_{el} , определение количества узлов N_{node} , общего количества степеней свободы n .

2) Составление матриц жесткости отдельных конечных элементов

Для каждого конечного элемента балки составляется матрица жесткости в его локальной системе координат вида:

$$[K_{elem}^i] = \begin{bmatrix} \frac{12EI}{l^3} & \dots & \frac{6EI}{l^2} \\ \vdots & \ddots & \vdots \\ \frac{EI}{l^2} & \dots & \frac{4EI}{l} \end{bmatrix}_{4 \times 4}$$

где i – номер текущего к.э., $i = 1 \dots N_{el}$;

$[K_{elem}^i]$ – матрица жесткости i -го к.э. в его локальной системе координат;

E, I, l – параметры балочного к.э. (модуль Юнга, длина, геометрический момент инерции, соответственно).

3) Составление таблицы индексов

4) Операция ансамблирования

5) Наложение кинематических граничных условий, модификация матрицы жесткости

6) Составление вектора внешних узловых усилий $\{f\}$

7) Решение СЛАУ, определение вектора узловых перемещений

Текст программы:

```
function main()
N_el = input('Enter the number of elements: '); %count
l = 100; %length (mm)
b = 10; %selection length (mm)
h = 20; %selection width
Jy=b*h^3/12; %i-moment
f = 10; %H
E=2e5; % Pa

N_dof_el = 4;
N_dof_sys = (N_el+1)*N_dof_el/2;
U = [0, 1, 1, 1, 1, 1, 1, 0, 0, 1];
F = [0, -f*l, -3*f, 0, 0, 2*f*l, 4*f, 0, 0, 0,];

L = zeros(1, N_el);
for i = 1:N_el
    L(i) = l*input('Enter the length of i-th element: ');
end
E_sys = [E, E, E, E];

K_el = zeros(N_dof_el);
K_g = zeros(N_dof_sys);

Index_M = zeros(N_el,N_dof_el);
for i = 1:N_el
```

```

    Index_M(i, :) = [(i-1)*2+1):(i-1)*2+N_dof_el];
end

for i = 1:N_el
    K_el = K_el_calc(E_sys(i), L(i), Jy)
    j = Index_M(i, 1);
    k = Index_M(i, N_dof_el);
    K_g(j:k, j:k) = K_g(j:k, j:k) + K_el;
end

for i = 1:N_dof_sys
    if(U(i) == 0)
        K_g(:, i) = 0;
        K_g(i, :) = 0;
        K_g(i, i) = 1;
    end
end

U_node = inv(K_g)*F';

    for i = 1:N_dof_sys
        if (rem(i, 2) == 1)
            fprintf("%f mm\n", U_node(i));
        else
            fprintf("%f deg\n", U_node(i) * 180 / pi);
        end
    end

end

function K = K_el_calc(E, L, 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

K_el =

    1.0e+07 *

    0.0016    0.0800   -0.0016    0.0800
    0.0800    5.3333   -0.0800    2.6667
   -0.0016   -0.0800    0.0016   -0.0800
    0.0800    2.6667   -0.0800    5.3333

K_el =

    1.0e+08 *

    0.0013    0.0320   -0.0013    0.0320
    0.0320    1.0667   -0.0320    0.5333
   -0.0013   -0.0320    0.0013   -0.0320
    0.0320    0.5333   -0.0320    1.0667

K_el =

    1.0e+07 *

    0.0005    0.0356   -0.0005    0.0356
    0.0356    3.5556   -0.0356    1.7778

```

-0.0005	-0.0356	0.0005	-0.0356
0.0356	1.7778	-0.0356	3.5556

K_el =

1.0e+07 *

0.0016	0.0800	-0.0016	0.0800
0.0800	5.3333	-0.0800	2.6667
-0.0016	-0.0800	0.0016	-0.0800
0.0800	2.6667	-0.0800	5.3333

U_node =

0
-0.0001
-0.0067
0.0000
-0.0037
0.0001
0.0068
0
0
-0.0001

0.000000 mm
-0.007194 deg
-0.006674 mm
0.000767 deg
-0.003670 mm
0.005885 deg
0.006763 mm
0.000000 deg
0.000000 mm
-0.005813 deg

Выполнение расчета исходной системы в программе Siemens NX

Для проведения статического расчета методом конечных элементов

необходимо выполнить 3 основных этапа:

- 1) создать/импортировать геометрию рассчитываемого объекта;
- 2) создать конечно-элементную модель;
- 3) задать граничные условия (закрепления и силовые факторы).

После выполнения всех пунктов программный комплекс при помощи

встроенного решателя, обработает входные данные и предоставит

пользователю требуемые результаты расчета.

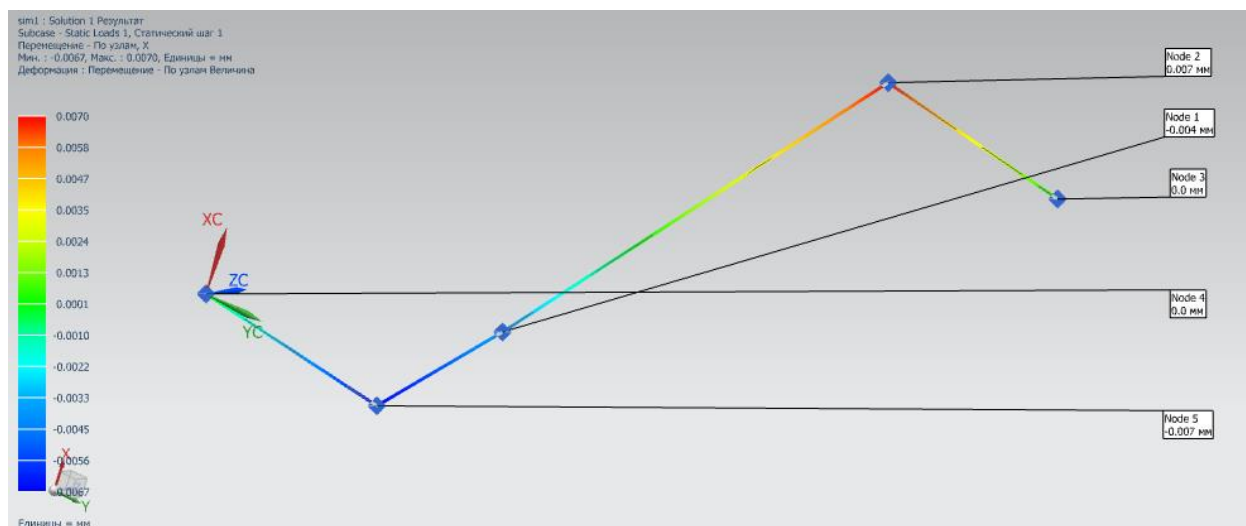


Рисунок 3 – перемещение по узлам

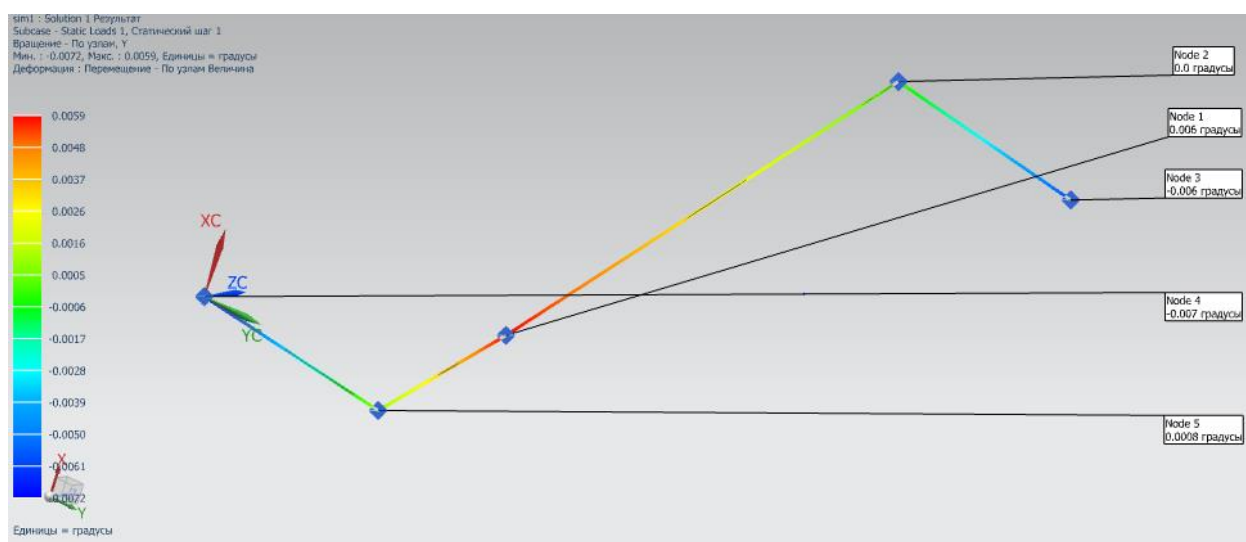


Рисунок 4 – вращение по узлам

Вращение по узлам (градусы)	MatLab	0	0.006	-0.006	-0.007	0.0008
	NX	0	0.006	-0.006	-0.007	0.0008
	Погрешн.	0	0	0	0	0
Перемещение по узлам (мм)	MatLab	0.007	-0.04	0	0	-0.007
	NX	0.07	-0.04	0	0	-0.007
	Погрешн.	0	0	0	0	0

Таблица 1 – соотношение полученных данных в NX и MatLab

Вывод: Значения, что были получены в программе и в NX сходятся