Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального 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 (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине «Модели и методы анализа проектных решений»

Студент:	Власенков Кирилл
Группа:	РК6-74б
Тип задания:	Лабораторная работа
Название:	Метод конечных элементов
Вариант:	65

Студент ______ подпись, дата — Власенков К. (Фамилия, И.О.)

Преподаватель ______ подпись, дата — Трудоношин В. А. (Фамилия, И.О.)

Оценка: ______

Москва, 2022

Содержание

Метод конечных элементов

1 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – решение дифференциального уравнения методом конечных элементов (МКЭ), используя линейную и кубическую функции формы, и анализ точности относительной аналитического способа решения

2 Задание

Решить с помощью МКЭ уравнение 1

$$22\frac{d^2u}{dx^2} - 37\frac{du}{dx} + 12 = 0, \tag{1}$$

при следующих граничных условиях (г. у.):

$$u'(x = 3) = 5, \tag{2}$$

$$u(x = 90) = 10. \tag{3}$$

Количество конечных элементов

- для первого расчета – 20,
- для второго – 40.

Также необходимо:

1. Сравнить результаты с аналитическим решением. Оценить максимальную погрешность.
2. Определить количество линейных КЭ, обеспечивающих такую же точность как и кубические.

3 Аналитическое решение

На рисунке 1 представлено аналитическое решение поставленной задачи.

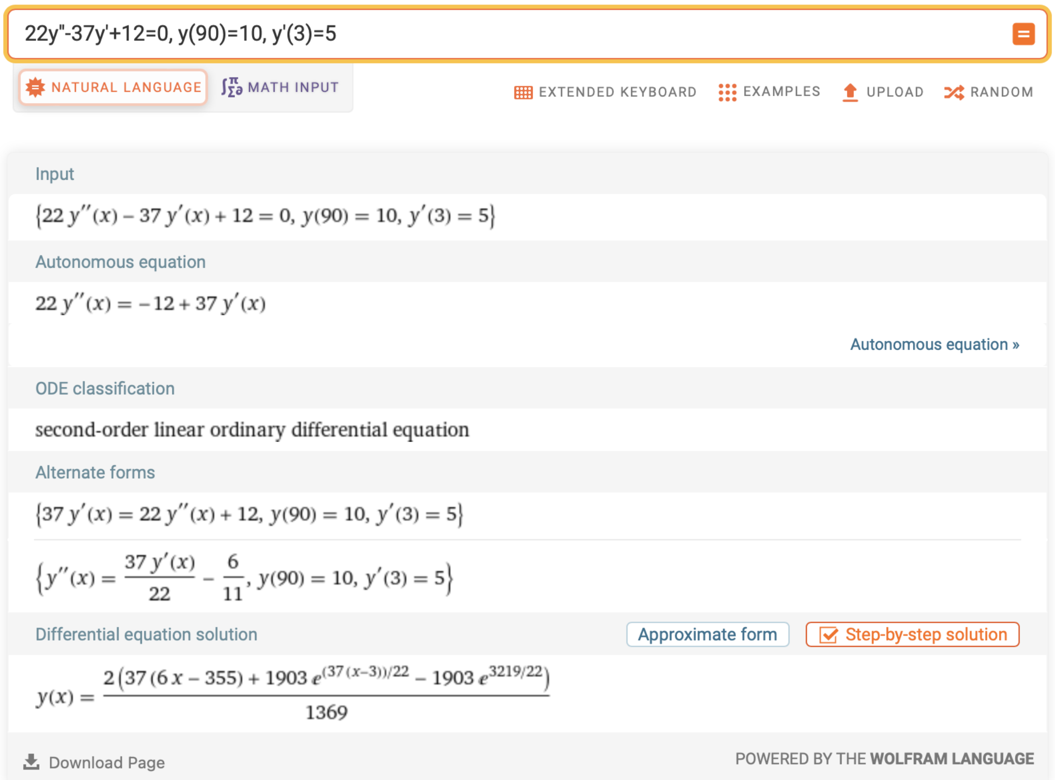

Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = \frac{2\left(37(6x - 355) + 1903e^{(37(x-3))/22} - 1903e^{3219/22}\right)}{1369}.$$

4 Получение локальных матрицы жесткости и вектора нагрузок

Составим локальные матрицу жесткости и вектор нагрузок для уравнения 1.

Линейная функция-формы КЭ

$$\mathbf{u} = \left[(1 - \tfrac{x}{L});\quad \tfrac{x}{L}\right]\begin{bmatrix} u_i \\ u_j \end{bmatrix} = \mathbf{N_e U},$$

где $\mathbf{N_e}$ – вектор функции формы конечного элемента (в данном случае линейной), его составляющие элементы - глобальные базисные функции, отличные от нуля в пределах этого элемента, L – длина КЭ.

В соответствии с методом Галеркина для уравнения 1:

$$\int_0^L \mathbf{W_e}\left(22\frac{d^2\mathbf{u}}{dx^2} - 37\frac{d\mathbf{u}}{dx} + 12\right)dx = 0, \tag{4}$$

где $\mathbf{W_e} = \mathbf{N_e}^T$.

$$\int_0^L \mathbf{W_e}\left(22\frac{d^2\mathbf{u}}{dx^2} - 37\frac{du}{dx} + 12\right)dx = 22\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx - 37\int_0^L \mathbf{W_e}\frac{d\mathbf{u}}{dx}dx + 12\int_0^L \mathbf{W_e}\mathbf{u} = 0$$

Распишем каждое слагаемое отдельно:

$$22\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 22\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx = 22\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d\mathbf{u}}{dx}\Big|_0^L -$$

$$-22\int_0^L \frac{d}{dx}\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d}{dx}\left[(1-\frac{x}{L});\quad \frac{x}{L}\right]\begin{bmatrix}u_i\\ u_j\end{bmatrix} = \begin{bmatrix}-22\frac{d\mathbf{u}}{dx}|_i\\ 22\frac{d\mathbf{u}}{dx}|_j\end{bmatrix} - 22\begin{bmatrix}\frac{1}{L}, & -\frac{1}{L}\\ -\frac{1}{L}, & \frac{1}{L}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix}$$

$$-37\int_0^L \mathbf{W_e}\frac{d\mathbf{u}}{dx}dx = -37\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d\mathbf{u}}{dx}\begin{bmatrix}u_i\\ u_j\end{bmatrix}dx =$$

$$= -\frac{37}{L}\int_0^L \begin{bmatrix}(1-\frac{x}{L}) & (-1+\frac{x}{L})\\ -\frac{x}{L} & \frac{x}{L}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} = -37\begin{bmatrix}-\frac{1}{2}, & \frac{1}{2}\\ -\frac{1}{2}, & \frac{1}{2}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix}$$

$$12\int_0^L \mathbf{W_e}dx = 12\begin{bmatrix}\frac{L}{2}\\ \frac{L}{2}\end{bmatrix}$$

Таким образом, для уравнения 4, при использовании линейной функции-формы, получаем (матмодель линейного КЭ):

$$\begin{bmatrix}22\frac{1}{L} - 37\frac{1}{2}, & -22\frac{1}{L} + 37\frac{1}{2}\\ -22\frac{1}{L} - 37\frac{1}{2}, & 22\frac{1}{L} + 37\frac{1}{2}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} = \begin{bmatrix}-22\frac{du}{dx}|_i + 12\frac{L}{2}\\ 22\frac{du}{dx}|_j + 12\frac{L}{2}\end{bmatrix}$$

Кубическая функция-формы КЭ

$$\mathbf{u} = \left[-\frac{9x^3}{2L^3} + \frac{18x^2}{2L^2} - \frac{11x}{2L} + 1;\, \frac{27x^3}{2L^3} - \frac{45x^2}{2L^2} + \frac{9x}{L};\, -\frac{27x^3}{2L^3} + \frac{36x^2}{2L^2} - \frac{9x}{2L};\, \frac{9x^3}{2L^3} - \frac{9x^2}{2L^2} - \frac{x}{L};\right]\begin{bmatrix}u_i\\ u_j\\ u_k\\ u_l\end{bmatrix} = \mathbf{N_e}\mathbf{U},$$

Как и для линейной функции-формы применим метод Галеркина (см. уравнение 4) и рассмотрим каждое слагаемое отдельно.

$$22\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 22\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx =$$

$$= 22\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d\mathbf{u}}{dx}\Big|_0^L - 22\int_0^L \frac{d}{dx}\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d}{dx}\mathbf{u} =$$

$$= \begin{bmatrix} -22\frac{d\mathbf{u}}{dx}\big|_i \\ 0 \\ 0 \\ 22\frac{d\mathbf{u}}{dx}\big|_l \end{bmatrix} - 22\begin{bmatrix} \frac{37}{10L} & -\frac{189}{40} & \frac{27}{20L} & -\frac{13}{40L} \\ -\frac{189}{40} & \frac{54}{5L} & -\frac{297}{40L} & \frac{27}{20L} \\ \frac{27}{20L} & -\frac{297}{40L} & \frac{54}{5L} & -\frac{189}{40} \\ -\frac{13}{40L} & \frac{27}{20L} & -\frac{189}{40} & \frac{37}{10L} \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix}$$

$$-37\int_0^L \mathbf{W_e}\frac{d\mathbf{u}}{dx}dx = -37\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d\mathbf{u}}{dx}dx =$$

$$= -37\begin{bmatrix} -\frac{1}{2} & \frac{57}{80} & -\frac{3}{10} & \frac{7}{80} \\ -\frac{57}{80} & 0 & \frac{81}{80} & -\frac{3}{10} \\ \frac{3}{10} & -\frac{81}{80} & 0 & -\frac{3}{10} \\ -\frac{7}{80} & \frac{3}{10} & -\frac{57}{80} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix}$$

$$12\int_0^L \mathbf{W_e}dx = 12\begin{bmatrix} \frac{L}{8} \\ \frac{3L}{8} \\ \frac{3L}{8} \\ \frac{L}{8} \end{bmatrix}$$

Таким образом, для уравнения 4, при использовании кубической функции-формы, получаем:

$$\begin{bmatrix} 22\frac{37}{10L} - 37\frac{1}{2} & -22\frac{189}{40L} + 37\frac{57}{80} & 22\frac{27}{20L} - 37\frac{3}{10} & -22\frac{13}{40L} + 37\frac{7}{80} \\ -22\frac{189}{40L} - 37\frac{57}{80} & 22\frac{54}{5L} + 0 & -22\frac{297}{40L} + 37\frac{81}{80} & 22\frac{27}{20L} - 37\frac{3}{10} \\ 22\frac{27}{20L} + 37\frac{3}{10} & -22\frac{297}{40L} - 37\frac{81}{80} & 22\frac{54}{5L} + 0 & -22\frac{189}{40L} + 37\frac{57}{80} \\ -22\frac{13}{40L} - 37\frac{7}{80} & 22\frac{27}{20L} + 37\frac{3}{10} & -22\frac{189}{40L} + 37\frac{57}{80} & 22\frac{37}{10L} + 37\frac{1}{2} \end{bmatrix}.$$

$$\cdot \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} 12\frac{L}{8} - 22\frac{du}{dx}|_i \\ 12\frac{3L}{8} \\ 12\frac{3L}{8} \\ 12\frac{L}{8} + 22\frac{du}{dx}|_l \end{bmatrix} \tag{5}$$

Локальные матрицу жесткости и вектор нагрузок из уравнения 5 с помощью матричных преобразований приведем к следующему виду:

$$\begin{bmatrix} a_{11} & 0 & 0 & a_{14} \\ a_{21} & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 22\frac{du}{dx}|_i \\ b_2 \\ b_3 \\ b_4 + 22\frac{du}{dx}|_l \end{bmatrix}$$

Для упрощения расчетов преобразуем систему выше, исключив внутренние узлы. Таким образом СЛАУ (математическая модель кубического КЭ):

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 22\frac{du}{dx}|_i \\ b_4 + 22\frac{du}{dx}|_l \end{bmatrix}$$

5 Получение глобальных матрицы жесткости и вектора нагрузок

Проведем процедуры ансамблирования и учет граничных условий для формирования итоговой математической модели.

Ансамблирование

Пусть локальные матрица жесткости и вектор неизвестных заданы следующим образом

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} b_1 - 22\frac{du}{dx}|_i \\ b_2 + 22\frac{du}{dx}|_l \end{bmatrix},$$

тогда, при разбитие области на n КЭ, глобальная матрица жесткости будет иметь размерность $(n+1)\cdot(n+1)$:

$$
\begin{bmatrix}
a_{11}^1 & a_{12}^1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\
0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\
0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\
\vdots & \vdots & \vdots & \vdots & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\
0 & 0 & 0 & 0 & 0 & 0 & a_{21}^n & a_{22}^n
\end{bmatrix}
\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}
=
\begin{bmatrix} b_1^1 - 22\frac{du}{dx}|_0 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ b_2^n + 22\frac{du}{dx}|_L \end{bmatrix}
$$

Учет граничных условий

Применим граничные условия первого (см. 3) и второго рода (см. 2) к выведенной выше системе.

$$
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\
0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\
0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\
\vdots & \vdots & \vdots & \vdots & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}
=
\begin{bmatrix} b_1^1 - 22 \cdot 5 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ 10 \end{bmatrix}
$$

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

На рисунках 2, 3 представлены графики полученные с помощью МКЭ (линейная функция-формы).

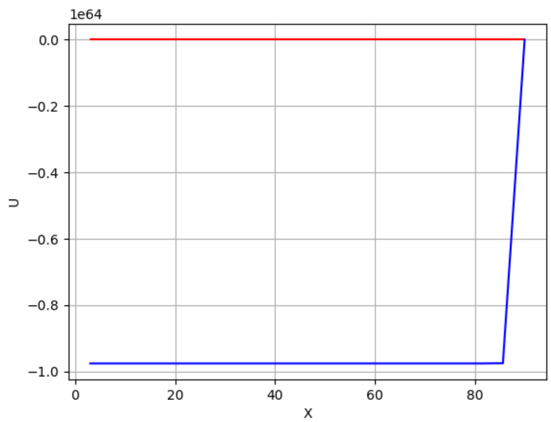

Рис. 2. Результат работы программы для 20 КЭ

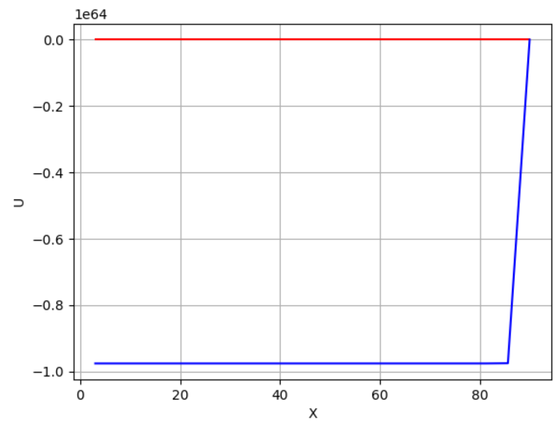

Рис. 3. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
3.000000e+00	-9.755382e+63	-2.075160e+05	9.755382e+63
7.350000e+00	-9.755382e+63	-2.075222e+05	9.755382e+63
1.170000e+01	-9.755382e+63	-2.075074e+05	9.755382e+63
1.605000e+01	-9.755382e+63	-2.075295e+05	9.755382e+63
2.040000e+01	-9.755382e+63	-2.074869e+05	9.755382e+63
2.475000e+01	-9.755382e+63	-2.075577e+05	9.755382e+63
2.910000e+01	-9.755382e+63	-2.074298e+05	9.755382e+63
3.345000e+01	-9.755382e+63	-2.076500e+05	9.755382e+63
3.780000e+01	-9.755382e+63	-2.072602e+05	9.755382e+63
4.215000e+01	-9.755382e+63	-2.079395e+05	9.755382e+63
4.650000e+01	-9.755382e+63	-2.067451e+05	9.755382e+63
5.085000e+01	-9.755382e+63	-2.088343e+05	9.755382e+63
5.520000e+01	-9.755382e+63	-2.051693e+05	9.755382e+63
5.955000e+01	-9.755382e+63	-2.115882e+05	9.755382e+63
6.390000e+01	-9.755382e+63	-2.003354e+05	9.755382e+63
6.825000e+01	-9.755382e+63	-2.200516e+05	9.755382e+63
7.260000e+01	-9.755382e+63	-1.854959e+05	9.755382e+63
7.695000e+01	-9.755382e+63	-2.460494e+05	9.755382e+63
8.130000e+01	-9.755378e+63	-1.399281e+05	9.755378e+63
8.565000e+01	-9.748896e+63	-3.258974e+05	9.748896e+63
9.000000e+01	0.000000e+00	1.000000e+01	1.000000e+01

Таблица 1. 20 линейных КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
3.000000e+00	-9.755382e+63	-5.855159e+21	9.755382e+63
5.175000e+00	-9.755382e+63	-5.855159e+21	9.755382e+63
7.350000e+00	-9.755382e+63	-5.855159e+21	9.755382e+63
9.525000e+00	-9.755382e+63	-5.855159e+21	9.755382e+63
1.170000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
1.387500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
1.605000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
1.822500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
2.040000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
2.257500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
2.475000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
2.692500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
2.910000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
3.127500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
3.345000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
3.562500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
3.780000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
3.997500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
4.215000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
4.432500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
4.650000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
4.867500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
5.085000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
5.302500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
5.520000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
5.737500e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
5.955000e+01	-9.755382e+63	-5.855159e+21	9.755382e+63
6.172500e+01	-9.755382e+63	-5.855160e+21	9.755382e+63
6.390000e+01	-9.755382e+63	-5.855157e+21	9.755382e+63
6.607500e+01	-9.755382e+63	-5.855167e+21	9.755382e+63
6.825000e+01	-9.755382e+63	-5.855132e+21	9.755382e+63
7.042500e+01	-9.755382e+63	-5.855253e+21	9.755382e+63
7.260000e+01	-9.755382e+63	-5.854841e+21	9.755382e+63
7.477500e+01	-9.755382e+63	-5.856245e+21	9.755382e+63
7.695000e+01	-9.755382e+63	-5.851452e+21	9.755382e+63
7.912500e+01	-9.755382e+63	-5.867810e+21	9.755382e+63
8.130000e+01	-9.755378e+63	-5.811988e+21	9.755378e+63
8.347500e+01	-9.755215e+63	-6.002485e+21	9.755215e+63
8.565000e+01	-9.748896e+63	-5.352394e+21	9.748896e+63
8.782500e+01	-9.503838e+63	-7.570901e+21	9.503838e+63
9.000000e+01	0.000000e+00	1.000000e+01	1.000000e+01

Таблица 2. 40 линейных КЭ

Максимальная абсолютная погрешность 9.755382e+63 и 9.755382e+63 соответственно.

Кубическая функция-формы

На рисунках 4, 5 представлены графики полученные с помощью МКЭ (кубическая функция-формы).

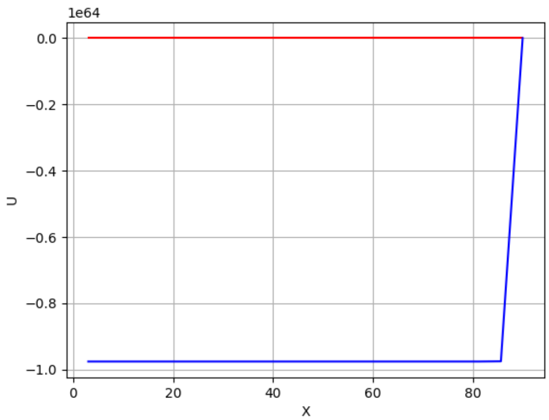

Рис. 4. Результат работы программы для 20 КЭ

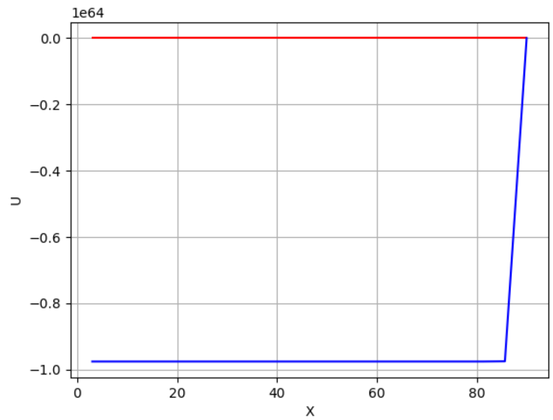

Рис. 5. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
3.000000e+00	-9.755382e+63	3.740690e+17	9.755382e+63
7.350000e+00	-9.755382e+63	3.740690e+17	9.755382e+63
1.170000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
1.605000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
2.040000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
2.475000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
2.910000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
3.345000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
3.780000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
4.215000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
4.650000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
5.085000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
5.520000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
5.955000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
6.390000e+01	-9.755382e+63	3.740690e+17	9.755382e+63
6.825000e+01	-9.755382e+63	3.740691e+17	9.755382e+63
7.260000e+01	-9.755382e+63	3.740677e+17	9.755382e+63
7.695000e+01	-9.755382e+63	3.740984e+17	9.755382e+63
8.130000e+01	-9.755378e+63	3.733822e+17	9.755378e+63
8.565000e+01	-9.748896e+63	3.900976e+17	9.748896e+63
9.000000e+01	0.000000e+00	1.000000e+01	1.000000e+01

Таблица 3. 20 кубических КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
3.000000e+00	-9.755382e+63	1.667357e+16	9.755382e+63
5.175000e+00	-9.755382e+63	1.667357e+16	9.755382e+63
7.350000e+00	-9.755382e+63	1.667357e+16	9.755382e+63
9.525000e+00	-9.755382e+63	1.667357e+16	9.755382e+63
1.170000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
1.387500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
1.605000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
1.822500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
2.040000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
2.257500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
2.475000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
2.692500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
2.910000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
3.127500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
3.345000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
3.562500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
3.780000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
3.997500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
4.215000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
4.432500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
4.650000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
4.867500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
5.085000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
5.302500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
5.520000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
5.737500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
5.955000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
6.172500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
6.390000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
6.607500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
6.825000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
7.042500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
7.260000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
7.477500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
7.695000e+01	-9.755382e+63	1.667357e+16	9.755382e+63
7.912500e+01	-9.755382e+63	1.667357e+16	9.755382e+63
8.130000e+01	-9.755378e+63	1.667356e+16	9.755378e+63
8.347500e+01	-9.755215e+63	1.667339e+16	9.755215e+63
8.565000e+01	-9.748896e+63	1.666541e+16	9.748896e+63
8.782500e+01	-9.503838e+63	1.630472e+16	9.503838e+63
9.000000e+01	0.000000e+00	1.000000e+01	1.000000e+01

Таблица 4. 40 кубических КЭ

Максимальная абсолютная погрешность 9.755382e+63 и 9.755382e+63 соответственно.

Нахождение количества линейных КЭ, обеспечивающих ту же точность, что и 20 кубических

Так как очевидно, что при увлечении числа КЭ точность растет, найдем искомое следуя алгоритму, представленному на рисунке 6.

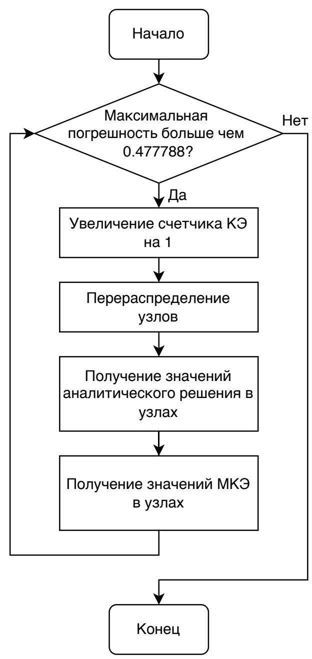

Рис. 6. Алгоритм нахождения количества КЭ, заданную точность

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 20 .

7 Код

Листинг 1. *Реализация МКЭ*

```

 #include <iostream>
#include <vector>
#include <cmath>

double EPS = 1e-16;
double X_BEGIN = 3.0;
double X_END = 90.0;
size_t ELEMS_NUM = 20;
```

```
double L = (X_END - X_BEGIN) / ELEMS_NUM;

double a = 22.0, B = -37.0, C = 0.0, D = 12.0, usl_left = 5.0, usl_right = 10.0; // au''+Bu'+Cu+D=0

std::vector<double> solve_with_gauss(std::vector<std::vector<double>>& A, std::vector<double>& b){
    size_t row_size = A.size();
    size_t col_size = A.back().size();

    // Прямой ход Гаусса
    double pivot = 0.;
    for (size_t i = 0; i < row_size; i++) {
        for (size_t j = i + 1; j < col_size; j++) {
            if (std::abs(A.at(j).at(i)) < EPS) {
                continue;
            }
            pivot = A.at(j).at(i) / A.at(i).at(i);
            b.at(j) -= pivot * b.at(i);
            for (size_t k = 0; k < row_size; k++) {
                A.at(j).at(k) -= pivot * A.at(i).at(k) ;
            }
        }
    }

    // Обратный ход Гаусса
    std::vector<double> x(row_size);
    for (int i = row_size - 1.; i >= 0; i--) {
        x.at(i) = b.at(i);
        for (size_t j = i + 1; j < row_size; j++) {
            x.at(i) -= x.at(j) * A.at(i).at(j);
        }
        x.at(i) /= A.at(i).at(i);
    }

    return x;
}

double analytical_solution(double x) {
    return 2. * (37. * (6. * x - 355) + 1903. * exp(37. * (x - 3.) / 22) - 1903. * exp(3219./22.)) / 1369.;
}

std::vector<double> build_analytical_solution(std::vector<double>& x_vec) {
    size_t x_vec_size = x_vec.size();
    std::vector<double> y_vec = std::vector<double>(x_vec_size);
```

```cpp
    for (size_t i = 0; i < x_vec_size; i++) {
        y_vec.at(i) = analytical_solution(x_vec.at(i));
    }
    return y_vec;
}

std::vector<double> build_linear_solution(size_t elems_num) {
    double L = (X_END - X_BEGIN) / elems_num;
    size_t size = elems_num + 1;
    std::vector< std::vector<double> > A(size, std::vector<double>(size));
    std::vector<double> b(size);

    // Локальная матрица жесткости для линейного КЭ
    std::vector< std::vector<double> > local_matrix = {
        { (a / L) + (B / 2.) - C * L / 3., -(a / L) - (B / 2.) - C * L / 6.},
        { -(a / L) + (B / 2.) - C * L / 6., (a / L) - (B / 2.) - C * L / 3.},
    };

    // Ансамблирование и получение глобальной матрицы жесткости для линейного КЭ
    for (size_t i = 0; i < elems_num; i++) {
        for (size_t j = 0; j < 2; j++) {
            for (size_t k = 0; k < 2; k++) {
                A.at(i + j).at(i + k) += local_matrix.at(j).at(k);
            }
        }
    }

    for (size_t i = 0; i < size ; i++) {
        b.at(i) = D * L;
    }

    // Учет ГУ
    if ( 1 == 1 ) {
        b.at(0) = D * L /2. - a*usl_left;
    } else {
        b.at(0) = usl_left;
        A.at(0).at(0) = 1;
        A.at(0).at(1) = 0;
    }

    if ( 0 == 1 ) {
        b.at(size - 1) = D * L /2. + a*usl_right;
    } else {
        b.at(size - 1) = usl_right;
        A.at(size - 1).at(size - 1) = 1;
        A.at(size - 1).at(size - 2) = 0;
```

```cpp
    }

    // Решение полученной СЛАУ методом Гаусса
    std::vector<double> res = solve_with_gauss(A, b);
    return res;
}

std::vector<double> build_cube_solution(size_t elems_num) {
    double L = (X_END - X_BEGIN) / elems_num;
    size_t size = elems_num + 1;
    std::vector< std::vector<double> > A(size,std::vector<double>(size));
    std::vector<double> b(size);

    // Локальная матрица жесткости для кубического КЭ
    std::vector< std::vector<double> > local_matrix = {
        { a * 37./(10.*L) + B / 2. + C * 8. / 105. * L, -a * 189./(40.*L) - B * 57./80. + C * 33. / 560. * L, a * 27./(20.*L) + B * 3./10. - C * 3. / 140. * L, -a * 13./(40.*L) - B * 7./80. + C * 19. / 1680. * L},
        { -a * 189./(40.*L) + B * 57./80. + C * 33./ 560. * L, a * 54./(5.*L) + 0. + C * 27. / 70. * L, -a * 297./(40.*L) - B * 81./80. - C * 27. / 560. * L, a * 27./(20.*L) + B * 3./10. - C * 3. / 140. * L},
        { a * 27./(20.*L) - B * 3./10. - C * 3. / 140. * L, -a * 297./(40.*L) + B * 81./80. - C * 27. / 560. * L, a * 54./(5.*L) - 0. + C * 27. / 70. * L, -a * 189./(40.*L) - B * 57./80. + C * 33. / 560. * L},
        { -a * 13./(40.*L) + B * 7./80. + C * 19./ 1680.* L, a * 27./(20.*L) - B * 3./10. + C * 3. / 140. * L, -a * 189./(40.*L) + B * 57./80. + C * 33. / 560. * L, a * 37./(10.*L) - B * 1./2. + C * 8. / 105. * L}
    };

    // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего элементов учитываются далее)
    std::vector<double> local_b = { D * L / 8.0,
                                    D * 3.0 * L / 8.0,
                                    D * 3.0 * L / 8.0,
                                    D * L / 8.0 };


    // Производим матричные преобразования для обнуления элементов локальной матрицы жесткости, относящихся к внутренним узлам
    for (size_t i = 1; i < 3; i++) {
        for (size_t j = 0; j < 4; j++) {
            if (std::fabs(local_matrix.at(j).at(i)) > EPS && i != j) {
                double val = local_matrix.at(j).at(i) /local_matrix.at(i).at(i);
                local_b.at(j) -= val * local_b.at(i);
                for (size_t k = 0; k < 4; k++) {
                    local_matrix.at(j).at(k) -= val *local_matrix.at(i).at(k);
```

```
                }
            }
            continue;
        }
    }


    // Исключаем внутренние узлы из рассмотрения
    std::vector< std::vector<double> > local_matrix_mod = { { local_matrix.at(0).at(0), local_matrix.at(0).at(3) },
                                                            { local_matrix.at(3).at(0), local_matrix.at(3).at(3) } };
    std::vector<double> local_b_mod = { local_b.at(0),
                                        local_b.at(3)
                                        };

    // Ансамблирование и получение глобальной матрицы жесткости для кубического КЭ
    for (size_t i = 0; i < elems_num; i++) {
        for (size_t j = 0; j < 2; j++) {
            for (size_t k = 0; k < 2; k++) {
                A.at(i + j).at(i + k) += local_matrix_mod.at(j).at(k);
            }
        }
    }

    for (size_t i = 0; i < elems_num; i++) {
        b.at(i) += local_b_mod.at(0);
        b.at(i+1) += local_b_mod.at(1);
    }

    // Учет ГУ
    if (1 == 1 ) {
        b.at(0) = local_b_mod.at(0) - a * usl_left;
    } else {
        b.at(0) = usl_left;
        A.at(0).at(0) = 1.;
        A.at(0).at(1) = 0.;
    }

    if (0 == 1 ) {
        b.at(size - 1) = local_b_mod.at(1) + a * usl_right;
    } else {
        b.at(size - 1) = usl_right;
        A.at(size - 1).at(size - 1) = 1.;
        A.at(size - 1).at(size - 2) = 0.;
```

```
    }

    // Решение полученной СЛАУ методом Гаусса
    std::vector<double> res = solve_with_gauss(A, b);
    return res;
}

double calc_abs_error(const std::vector<double>& y_real, const std::vector<double>&
    y) {
    double max_err = 0.0;
    for (size_t i = 0; i < y_real.size(); i++) {
        double err = std::fabs(y_real.at(i) - y.at(i));
        if (err > max_err) {
            max_err = err;
        }
    }
    return max_err;
}

int main() {

     std::vector<double> x(ELEMS_NUM + 1);
     for (size_t i = 0; i < x.size(); i++) {
         x.at(i) = X_BEGIN + i * L;
     }
     size_t x_size = x.size();

    std::vector<double> y;
    if (true) {
        y = build_linear_solution(ELEMS_NUM);
    } else {
        y = build_cube_solution(ELEMS_NUM);
    }
     std::vector<double> y_real = build_analytical_solution(x);


     FILE* gp;
     FILE* ab;
     FILE* pgr;
     FILE* tab;
     if (true) {
        if(ELEMS_NUM == 20) {
            gp = fopen("res/labs/text/graph/lin_20.txt", "w");
            ab = fopen("res/labs/text/graph/abs.txt", "w");
            for (size_t i = 0; i < x_size; i++) {
                fprintf(ab, "%lf %lf\n", x.at(i), y_real.at(i));
```

```
            }
            pgr = fopen("res/labs/text/pgr/lin_20.txt", "w");
            tab = fopen("res/labs/text/tab/lin_20.txt", "w");
        }
        if(ELEMS_NUM == 40) {
            gp = fopen("res/labs/text/graph/lin_40.txt", "w");
            pgr = fopen("res/labs/text/pgr/lin_40.txt", "w");
            tab = fopen("res/labs/text/tab/lin_40.txt", "w");
        }
    } else {
        if(ELEMS_NUM == 20) {
            gp = fopen("res/labs/text/graph/cub_20.txt", "w");
            pgr = fopen("res/labs/text/pgr/cub_20.txt", "w");
            tab = fopen("res/labs/text/tab/cub_20.txt", "w");
        }
        if(ELEMS_NUM == 40) {
            gp = fopen("res/labs/text/graph/cub_40.txt", "w");
            pgr = fopen("res/labs/text/pgr/cub_40.txt", "w");
            tab = fopen("res/labs/text/tab/cub_40.txt", "w");
        }
    }

    for (size_t i = 0; i < x.size()-1; i++) {
        fprintf(tab, "%le & %le & %le & %le \\\\\n", x.at(i), y_real.at(i), y.at(i),
            std::fabs(y_real.at(i) - y.at(i)));
    }
    fprintf(tab, "%le & %le & %le & %le", x.at(x.size()-1), y_real.at(x.size()-1),
        y.at(x.size()-1), std::fabs(y_real.at(x.size()-1) - y.at(x.size()-1)));

    for (size_t i = 0; i < x_size; i++) {
        fprintf(gp, "%lf %lf\n", x.at(i), y.at(i));
    }

    fprintf(pgr, "%e", calc_abs_error(y_real, y));
    fclose(gp);
    fclose(ab);
    fclose(pgr);
    fclose(tab);

    return 0;
}
```

8 Вывод

В ходе выполнения лабораторной работы был реализован МКЭ для различных функций форм, а также найдено количество линейных КЭ обеспечивающих точность 20ти кубических КЭ.

Постановка: © *доцент кафедры РК-6, кандидат технических наук, доцент, Трудоношин В.А.*

Решение и вёрстка: © *студент группы РК6-74б, Власенков К.*

2022, осенний семестр