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:	Гаспарян Грета Андраниковна
Группа:	РК6-73б
Тип задания:	Лабораторная работа
Название:	Метод конечных элементов
Вариант:	49

Студент ______________ подпись, дата — Гаспарян Г.А. 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

$$40\frac{d^2u}{dx^2} - 4u + 11 = 0, \tag{1}$$

при следующих граничных условиях (г. у.):

$$u'(x = 0) = -2, \tag{2}$$

$$u(x = 17) = 6. \tag{3}$$

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.

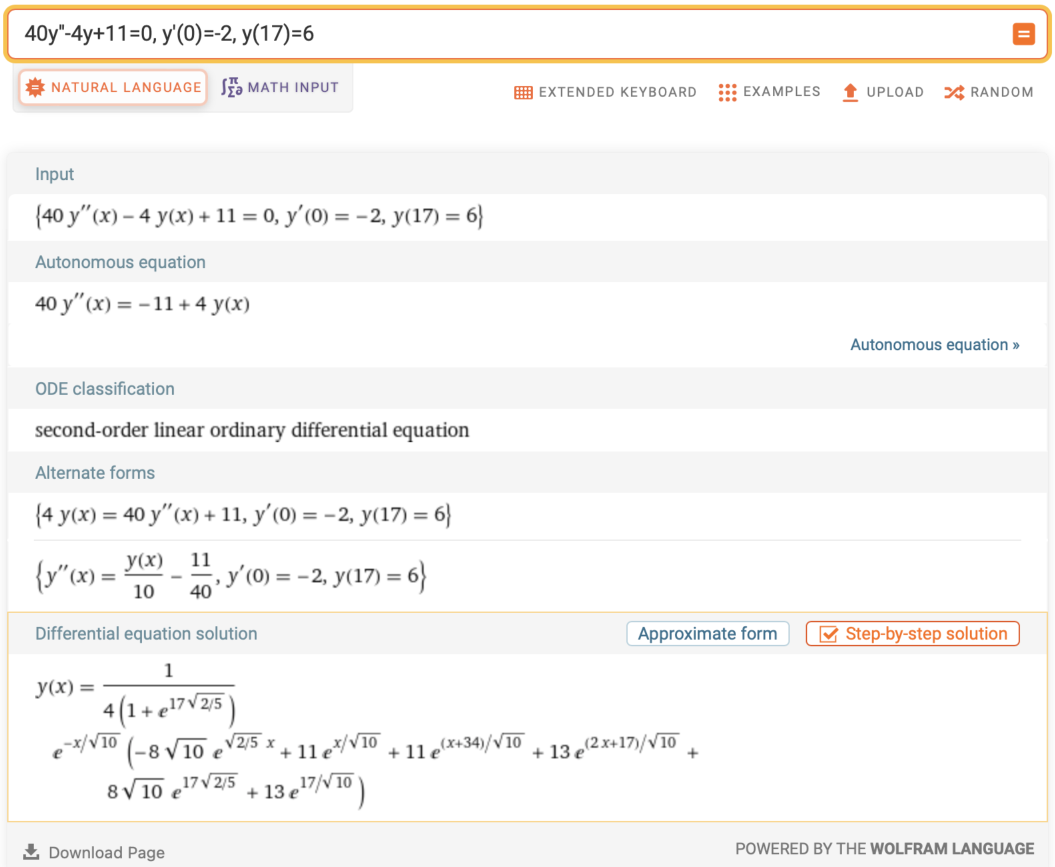

Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = \frac{e^{-x/\sqrt{10}}\left(-8\sqrt{10}e^{\sqrt{2/5}x} + 11e^{x/\sqrt{10}} + 11e^{(x+34)/\sqrt{10}} + 13e^{(2x+17)/\sqrt{10}} + 8\sqrt{10}e^{17\sqrt{2/5}} + 13e^{17/\sqrt{10}}\right)}{4\left(1+e^{17\sqrt{2/5}}\right)}.$$

4 Получение локальных матрицы жесткости и вектора нагрузок

Составим локальные матрицу жесткости и вектор нагрузок для уравнения 1.

Линейная функция-формы КЭ

$$\mathbf{u} = \begin{bmatrix}(1-\frac{x}{L}); & \frac{x}{L}\end{bmatrix}\begin{bmatrix}u_i\\u_j\end{bmatrix} = \mathbf{N_e}\mathbf{U},$$

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:

$$\int_0^L \mathbf{W_e}\left(40\frac{d^2\mathbf{u}}{dx^2}\frac{d\mathbf{u}}{dx} - 4u + 11\right)dx = 0, \tag{4}$$

где $\mathbf{W_e} = \mathbf{N_e}^T$.

$$\int_0^L \mathbf{W_e}\left(40\frac{d^2\mathbf{u}}{dx^2}\frac{du}{dx} - 4u + 11\right)dx = 40\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx - 4\int_0^L \mathbf{W_e}\mathbf{u}dx + 11\int_0^L \mathbf{W_e}\mathbf{u} = 0$$

Распишем каждое слагаемое отдельно:

$$40\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 40\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx = 40\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d\mathbf{u}}{dx}\Big|_0^L -$$

$$-40\int_0^L \frac{d}{dx}\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d}{dx}\begin{bmatrix}(1-\frac{x}{L}); & \frac{x}{L}\end{bmatrix}\begin{bmatrix}u_i\\u_j\end{bmatrix} = \begin{bmatrix}-40\frac{d\mathbf{u}}{dx}\big|_i\\ 40\frac{d\mathbf{u}}{dx}\big|_j\end{bmatrix} - 40\begin{bmatrix}\frac{1}{L}, & -\frac{1}{L}\\ -\frac{1}{L}, & \frac{1}{L}\end{bmatrix}\begin{bmatrix}u_i\\u_j\end{bmatrix}$$

$$-4\int_0^L \mathbf{W_e}\mathbf{u}dx = -4\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\mathbf{u}\begin{bmatrix}u_i\\u_j\end{bmatrix}dx = -4\begin{bmatrix}\frac{L}{3}, & \frac{L}{6}\\ \frac{L}{6}, & \frac{L}{3}\end{bmatrix}\begin{bmatrix}u_i\\u_j\end{bmatrix}$$

$$11\int_0^L \mathbf{W_e}dx = 11\begin{bmatrix}\frac{L}{2}\\ \frac{L}{2}\end{bmatrix}$$

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

$$\begin{bmatrix}40\frac{1}{L} - 4\frac{L}{3}, & -40\frac{1}{L} - 4\frac{L}{6}\\ -40\frac{1}{L} - 4\frac{L}{6}, & 40\frac{1}{L} - 4\frac{L}{3}\end{bmatrix}\begin{bmatrix}u_i\\u_j\end{bmatrix} = \begin{bmatrix}-40\frac{du}{dx}\big|_i + 11\frac{L}{2}\\ 40\frac{du}{dx}\big|_j + 11\frac{L}{2}\end{bmatrix}$$

Кубическая функция-формы КЭ

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

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

$$40\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 40\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx =$$

$$= 40\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d\mathbf{u}}{dx}\Big|_0^L - 40\int_0^L \frac{d}{dx}\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d}{dx}\mathbf{u} =$$

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

$$-4\int_0^L \mathbf{W_e}\mathbf{u}dx = -4\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\mathbf{u}dx =$$

$$= -4\begin{bmatrix} \frac{8L}{105} & \frac{33L}{560} & -\frac{3L}{140} & \frac{119L}{1680} \\ \frac{33L}{560} & \frac{27L}{170} & -\frac{27L}{560} & -\frac{3L}{140} \\ -\frac{3L}{140} & -\frac{27L}{560} & \frac{27L}{170} & \frac{33L}{560} \\ \frac{19L}{1680} & -\frac{3L}{140} & \frac{33L}{560} & \frac{8L}{105} \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix}$$

$$11\int_0^L \mathbf{W_e}dx = 11\begin{bmatrix} \frac{L}{8} \\ \frac{3L}{8} \\ \frac{3L}{8} \\ \frac{L}{8} \end{bmatrix}$$

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

$$\begin{bmatrix} 40\frac{37}{10L} - 4\frac{8L}{105} & -40\frac{189}{40L} - 4\frac{33L}{560} & 40\frac{27}{20L} + 4\frac{3L}{140} & -40\frac{13}{40L} - 4\frac{119L}{1680} \\ -40\frac{189}{40L} - 4\frac{33L}{560} & 40\frac{54}{5L} + 0 - 4\frac{27L}{170} & -40\frac{297}{40L} + 4\frac{27L}{560} & 40\frac{27}{20L} + 4\frac{27L}{560} \\ 40\frac{27}{20L} + 4\frac{27L}{560} & -40\frac{297}{40L} + 4\frac{27L}{560} & 40\frac{54}{5L} + 0 - 4\frac{27L}{170} & -40\frac{189}{40L} - 4\frac{33L}{560} \\ -40\frac{13}{40L} - 4\frac{19L}{1680} & 40\frac{27}{20L}\frac{3}{10} - 4\frac{3L}{140} & -40\frac{189}{40L} - 4\frac{33L}{560} & 40\frac{37}{10L} - 4\frac{8L}{105} \end{bmatrix}.$$

$$\cdot \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} 11\frac{L}{8} - 40\frac{du}{dx}|_i \\ 11\frac{3L}{8} \\ 11\frac{3L}{8} \\ 11\frac{L}{8} + 40\frac{du}{dx}|_l \end{bmatrix} \tag{5}$$

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

$$\begin{bmatrix} a_{11} & 0 & 0 & a_{14} \\ a_{21} & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 40\frac{du}{dx}|_i \\ b_2 \\ b_3 \\ b_4 + 40\frac{du}{dx}|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 40\frac{du}{dx}|_i \\ b_4 + 40\frac{du}{dx}|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} b_1 - 40\frac{du}{dx}|_i \\ b_2 + 40\frac{du}{dx}|_l \end{bmatrix},$$

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

$$\begin{bmatrix} a_{11}^1 & a_{12}^1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{21}^n & a_{22}^n \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} b_1^1 - 40\frac{du}{dx}|_0 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ b_2^n + 40\frac{du}{dx}|_L \end{bmatrix}$$

Учет граничных условий

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

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} b_1^1 - 40 \cdot -2 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ 6 \end{bmatrix}$$

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

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

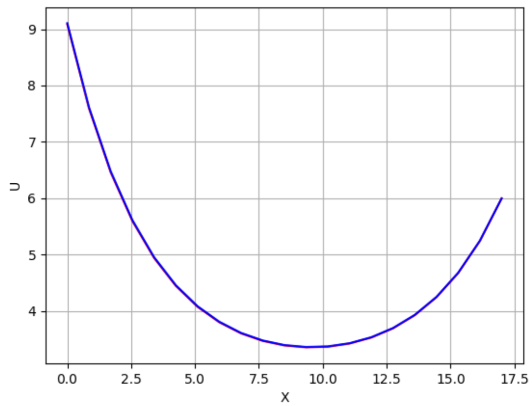

Рис. 2. Результат работы программы для 20 КЭ

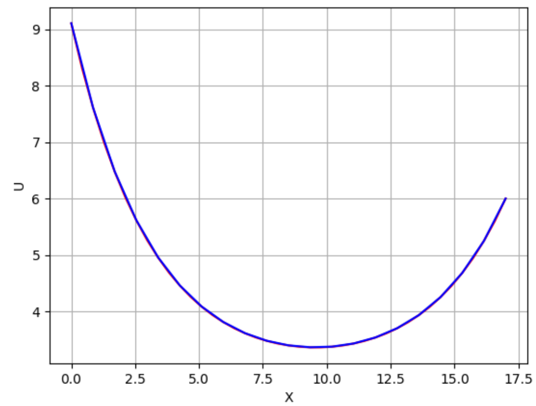

Рис. 3. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000e+00	9.104359e+00	9.084927e+00	1.943122e-02
8.500000e-01	7.614750e+00	7.595846e+00	1.890471e-02
1.700000e+00	6.478742e+00	6.461144e+00	1.759803e-02
2.550000e+00	5.613760e+00	5.597840e+00	1.592067e-02
3.400000e+00	4.956934e+00	4.942800e+00	1.413447e-02
4.250000e+00	4.460522e+00	4.448121e+00	1.240083e-02
5.100000e+00	4.088441e+00	4.077627e+00	1.081363e-02
5.950000e+00	3.813645e+00	3.804223e+00	9.422222e-03
6.800000e+00	3.616162e+00	3.607915e+00	8.247014e-03
7.650000e+00	3.481637e+00	3.474347e+00	7.289854e-03
8.500000e+00	3.400291e+00	3.393751e+00	6.540548e-03
9.350000e+00	3.366213e+00	3.360232e+00	5.980464e-03
1.020000e+01	3.376924e+00	3.371340e+00	5.583791e-03
1.105000e+01	3.433204e+00	3.427888e+00	5.316758e-03
1.190000e+01	3.539144e+00	3.534009e+00	5.134817e-03
1.275000e+01	3.702444e+00	3.697466e+00	4.977566e-03
1.360000e+01	3.934972e+00	3.930211e+00	4.760910e-03
1.445000e+01	4.253632e+00	4.249267e+00	4.365630e-03
1.530000e+01	4.681585e+00	4.677964e+00	3.621156e-03
1.615000e+01	5.249937e+00	5.247654e+00	2.282793e-03
1.700000e+01	6.000000e+00	6.000000e+00	3.552714e-15

Таблица 1. 20 линейных КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000e+00	9.104359e+00	9.099485e+00	4.873804e-03
4.250000e-01	8.309272e+00	8.304435e+00	4.836518e-03
8.500000e-01	7.614750e+00	7.610015e+00	4.734889e-03
1.275000e+00	7.008231e+00	7.003645e+00	4.586061e-03
1.700000e+00	6.478742e+00	6.474338e+00	4.403917e-03
2.125000e+00	6.016704e+00	6.012504e+00	4.199632e-03
2.550000e+00	5.613760e+00	5.609778e+00	3.982134e-03
2.975000e+00	5.262621e+00	5.258863e+00	3.758498e-03
3.400000e+00	4.956934e+00	4.953400e+00	3.534271e-03
3.825000e+00	4.691171e+00	4.687857e+00	3.313746e-03
4.250000e+00	4.460522e+00	4.457422e+00	3.100194e-03
4.675000e+00	4.260816e+00	4.257920e+00	2.896049e-03
5.100000e+00	4.088441e+00	4.085738e+00	2.703073e-03
5.525000e+00	3.940277e+00	3.937755e+00	2.522482e-03
5.950000e+00	3.813645e+00	3.811290e+00	2.355054e-03
6.375000e+00	3.706254e+00	3.704053e+00	2.201217e-03
6.800000e+00	3.616162e+00	3.614101e+00	2.061120e-03
7.225000e+00	3.541738e+00	3.539803e+00	1.934687e-03
7.650000e+00	3.481637e+00	3.479815e+00	1.821662e-03
8.075000e+00	3.434770e+00	3.433049e+00	1.721640e-03
8.500000e+00	3.400291e+00	3.398657e+00	1.634091e-03
8.925000e+00	3.377576e+00	3.376017e+00	1.558371e-03
9.350000e+00	3.366213e+00	3.364719e+00	1.493728e-03
9.775000e+00	3.365997e+00	3.364558e+00	1.439303e-03
1.020000e+01	3.376924e+00	3.375530e+00	1.394115e-03
1.062500e+01	3.399192e+00	3.397835e+00	1.357047e-03
1.105000e+01	3.433204e+00	3.431878e+00	1.326818e-03
1.147500e+01	3.479575e+00	3.478273e+00	1.301952e-03
1.190000e+01	3.539144e+00	3.537863e+00	1.280730e-03
1.232500e+01	3.612988e+00	3.611727e+00	1.261139e-03
1.275000e+01	3.702444e+00	3.701203e+00	1.240806e-03
1.317500e+01	3.809128e+00	3.807911e+00	1.216917e-03
1.360000e+01	3.934972e+00	3.933786e+00	1.186123e-03
1.402500e+01	4.082252e+00	4.081108e+00	1.144426e-03
1.445000e+01	4.253632e+00	4.252545e+00	1.087045e-03
1.487500e+01	4.452212e+00	4.451204e+00	1.008255e-03
1.530000e+01	4.681585e+00	4.680684e+00	9.012001e-04
1.572500e+01	4.945900e+00	4.945142e+00	7.576716e-04
1.615000e+01	5.249937e+00	5.249369e+00	5.678435e-04
1.657500e+01	5.599198e+00	5.598878e+00	3.199657e-04
1.700000e+01	6.000000e+00	6.000000e+00	3.552714e-15

Таблица 2. 40 линейных КЭ

Максимальная абсолютная погрешность 1.943122e-02 и 4.873804e-03 соответственно.

Кубическая функция-формы

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

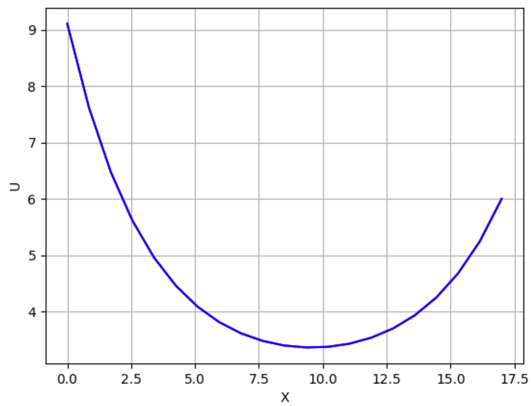

Рис. 4. Результат работы программы для 20 КЭ

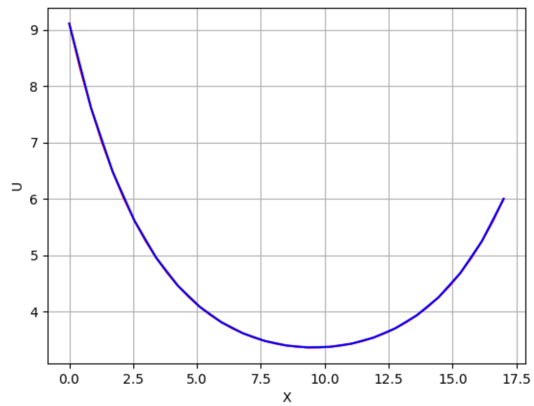

Рис. 5. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000e+00	9.104359e+00	9.104359e+00	1.202399e-08
8.500000e-01	7.614750e+00	7.614750e+00	1.170358e-08
1.700000e+00	6.478742e+00	6.478742e+00	1.089952e-08
2.550000e+00	5.613760e+00	5.613760e+00	9.864972e-09
3.400000e+00	4.956934e+00	4.956934e+00	8.761987e-09
4.250000e+00	4.460522e+00	4.460522e+00	7.690551e-09
5.100000e+00	4.088441e+00	4.088441e+00	6.708945e-09
5.950000e+00	3.813645e+00	3.813645e+00	5.847870e-09
6.800000e+00	3.616162e+00	3.616162e+00	5.120112e-09
7.650000e+00	3.481637e+00	3.481637e+00	4.526946e-09
8.500000e+00	3.400291e+00	3.400291e+00	4.062154e-09
9.350000e+00	3.366213e+00	3.366213e+00	3.714261e-09
1.020000e+01	3.376924e+00	3.376924e+00	3.467327e-09
1.105000e+01	3.433204e+00	3.433204e+00	3.300460e-09
1.190000e+01	3.539144e+00	3.539144e+00	3.186104e-09
1.275000e+01	3.702444e+00	3.702444e+00	3.086887e-09
1.360000e+01	3.934972e+00	3.934972e+00	2.950809e-09
1.445000e+01	4.253632e+00	4.253632e+00	2.704194e-09
1.530000e+01	4.681585e+00	4.681585e+00	2.241705e-09
1.615000e+01	5.249937e+00	5.249937e+00	1.412366e-09
1.700000e+01	6.000000e+00	6.000000e+00	3.552714e-15

Таблица 3. 20 кубических КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000e+00	9.104359e+00	9.104359e+00	1.889155e-10
4.250000e-01	8.309272e+00	8.309272e+00	1.874785e-10
8.500000e-01	7.614750e+00	7.614750e+00	1.835501e-10
1.275000e+00	7.008231e+00	7.008231e+00	1.777849e-10
1.700000e+00	6.478742e+00	6.478742e+00	1.707336e-10
2.125000e+00	6.016704e+00	6.016704e+00	1.628164e-10
2.550000e+00	5.613760e+00	5.613760e+00	1.543858e-10
2.975000e+00	5.262621e+00	5.262621e+00	1.457208e-10
3.400000e+00	4.956934e+00	4.956934e+00	1.370282e-10
3.825000e+00	4.691171e+00	4.691171e+00	1.284830e-10
4.250000e+00	4.460522e+00	4.460522e+00	1.202016e-10
4.675000e+00	4.260816e+00	4.260816e+00	1.122844e-10
5.100000e+00	4.088441e+00	4.088441e+00	1.048059e-10
5.525000e+00	3.940277e+00	3.940277e+00	9.780088e-11
5.950000e+00	3.813645e+00	3.813645e+00	9.130963e-11
6.375000e+00	3.706254e+00	3.706254e+00	8.534018e-11
6.800000e+00	3.616162e+00	3.616162e+00	7.990630e-11
7.225000e+00	3.541738e+00	3.541738e+00	7.500534e-11
7.650000e+00	3.481637e+00	3.481637e+00	7.061729e-11
8.075000e+00	3.434770e+00	3.434770e+00	6.674217e-11
8.500000e+00	3.400291e+00	3.400291e+00	6.334178e-11
8.925000e+00	3.377576e+00	3.377576e+00	6.040324e-11
9.350000e+00	3.366213e+00	3.366213e+00	5.789635e-11
9.775000e+00	3.365997e+00	3.365997e+00	5.578071e-11
1.020000e+01	3.376924e+00	3.376924e+00	5.403056e-11
1.062500e+01	3.399192e+00	3.399192e+00	5.258771e-11
1.105000e+01	3.433204e+00	3.433204e+00	5.141176e-11
1.147500e+01	3.479575e+00	3.479575e+00	5.044809e-11
1.190000e+01	3.539144e+00	3.539144e+00	4.961986e-11
1.232500e+01	3.612988e+00	3.612988e+00	4.886314e-11
1.275000e+01	3.702444e+00	3.702444e+00	4.806999e-11
1.317500e+01	3.809128e+00	3.809128e+00	4.714229e-11
1.360000e+01	3.934972e+00	3.934972e+00	4.595302e-11
1.402500e+01	4.082252e+00	4.082252e+00	4.433254e-11
1.445000e+01	4.253632e+00	4.253632e+00	4.211387e-11
1.487500e+01	4.452212e+00	4.452212e+00	3.905765e-11
1.530000e+01	4.681585e+00	4.681585e+00	3.490452e-11
1.572500e+01	4.945900e+00	4.945900e+00	2.934897e-11
1.615000e+01	5.249937e+00	5.249937e+00	2.199130e-11
1.657500e+01	5.599198e+00	5.599198e+00	1.239275e-11
1.700000e+01	6.000000e+00	6.000000e+00	3.552714e-15

Таблица 4. 40 кубических КЭ

Максимальная абсолютная погрешность 1.202399e-08 и 1.889155e-10 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.

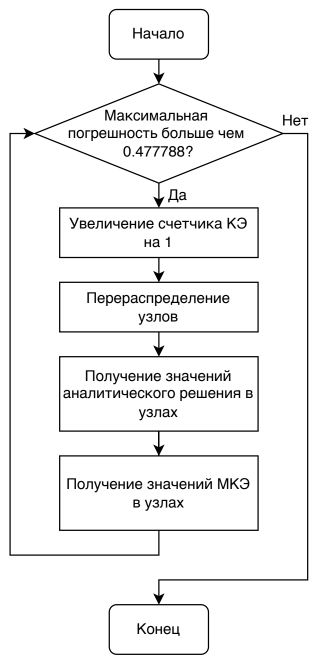

Рис. 6. Алгоритм нахождения количества КЭ, заданную точность

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 25526 КЭ.

7 Код

Листинг 1. *Реализация МКЭ*

```

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

double EPS = 1e-16;
double X_BEGIN = 0.0;
double X_END = 17.0;
size_t ELEMS_NUM = 20;
```

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

double a = 40.0, B = 0.0, C = -4.0, D = 11.0, usl_left = -2.0, usl_right = 6.0; // 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 (exp(-x/sqrt(10)) * (-8 * sqrt(10) * exp(sqrt(0.4) * x) + 11 * exp(x/sqrt(10))
        + 11 * exp((x + 34.)/sqrt(10)) + 13 * exp((2. * x + 17)/sqrt(10)) + 8 * sqrt(10) *
        exp(17. * sqrt(0.4)) + 13. * exp(17./sqrt(10))))/(4. * (1. + exp(17. * sqrt(0.4))));
}

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);
    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;
```

```cpp
        A.at(size - 1).at(size - 2) = 0;
    }

    // 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 РК6-73б, Гаспарян Г.А.*

2022, осенний семестр