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:	Роздорожный Илья Олегович
Группа:	РК6-74б
Тип задания:	Лабораторная работа
Название:	Метод конечных элементов
Вариант:	86

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

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

Оценка: ______

Москва, 2023

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

$$7\frac{d^2u}{dx^2} + 6\frac{du}{dx}u - 5 = 0, \tag{1}$$

при следующих граничных условиях (г. у.):

$$u(x = 0) = 10, \tag{2}$$

$$u'(x = 7) = -5. \tag{3}$$

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.

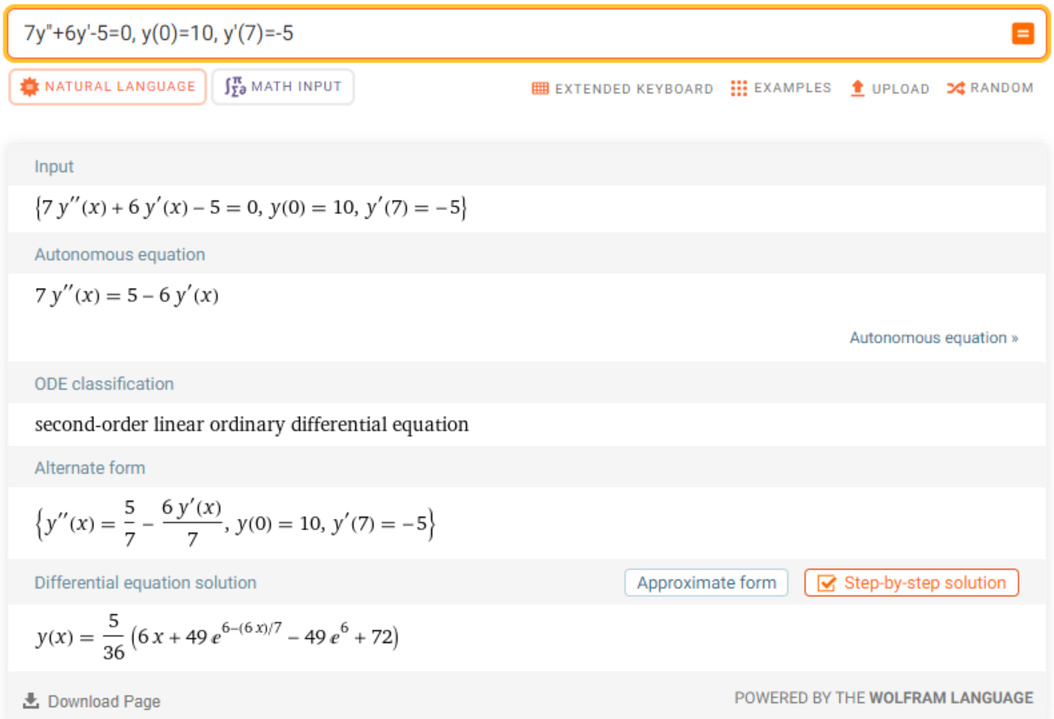

Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = \frac{5}{36}\left(6x + 49e^{6-(6x)/7} - 49e^{6} + 72\right).$$

4 Получение локальных матрицы жесткости и вектора нагрузок

Составим локальные матрицу жесткости и вектор нагрузок для уравнения 1.

Линейная функция-формы КЭ

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

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:

$$\int_0^L \mathbf{W_e}\left(7\frac{d^2\mathbf{u}}{dx^2} + 6\frac{d\mathbf{u}}{dx} - 5\right)dx = 0, \tag{4}$$

где $\mathbf{W_e} = \mathbf{N_e}^T$.

$$\int_0^L \mathbf{W_e}\left(7\frac{d^2\mathbf{u}}{dx^2} + 6\frac{du}{dx} - 5\right)dx = 7\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx + 6\int_0^L \mathbf{W_e}\frac{d\mathbf{u}}{dx}dx - 5\int_0^L \mathbf{W_e}dx = 0$$

Распишем каждое слагаемое отдельно:

$$7\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 7\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx = 7\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d\mathbf{u}}{dx}\Big|_0^L -$$

$$-7\int_0^L \frac{d}{dx}\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d}{dx}\begin{bmatrix}(1-\frac{x}{L}); & \frac{x}{L}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} = \begin{bmatrix}-7\frac{d\mathbf{u}}{dx}\big|_i\\ 7\frac{d\mathbf{u}}{dx}\big|_j\end{bmatrix} - 7\begin{bmatrix}\frac{1}{L}, & -\frac{1}{L}\\ -\frac{1}{L}, & \frac{1}{L}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix}$$

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

$$= \frac{6}{L}\int_0^L \begin{bmatrix}(1-\frac{x}{L}) & (-1+\frac{x}{L})\\ -\frac{x}{L} & \frac{x}{L}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} = +6\begin{bmatrix}-\frac{1}{2}, & \frac{1}{2}\\ -\frac{1}{2}, & \frac{1}{2}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix}$$

$$-5\int_0^L \mathbf{W_e}dx = -5\begin{bmatrix}\frac{L}{2}\\ \frac{L}{2}\end{bmatrix}$$

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

$$\begin{bmatrix}7\frac{1}{L} + 6\frac{1}{2}, & -7\frac{1}{L} - 6\frac{1}{2}\\ -7\frac{1}{L} + 6\frac{1}{2}, & 7\frac{1}{L} - 6\frac{1}{2}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} = \begin{bmatrix}-7\frac{du}{dx}\big|_i - 5\frac{L}{2}\\ 7\frac{du}{dx}\big|_j - 5\frac{L}{2}\end{bmatrix}$$

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.

$$7\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 7\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix} \frac{d^2\mathbf{u}}{dx^2}dx =$$

$$= 7\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix} \frac{d\mathbf{u}}{dx}\Big|_0^L - 7\int_0^L \frac{d}{dx}\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix} \frac{d}{dx}\mathbf{u} =$$

$$= \begin{bmatrix} -7\frac{d\mathbf{u}}{dx}\big|_i \\ 0 \\ 0 \\ 7\frac{d\mathbf{u}}{dx}\big|_l \end{bmatrix} - 7\begin{bmatrix} \frac{37}{10L} & -\frac{189}{40} & \frac{27}{20L} & -\frac{13}{40L} \\ -\frac{189}{40} & \frac{54}{5L} & -\frac{297}{40L} & \frac{27}{20L} \\ \frac{27}{20L} & -\frac{297}{40L} & \frac{54}{5L} & -\frac{189}{40} \\ -\frac{13}{40L} & \frac{27}{20L} & -\frac{189}{40} & \frac{37}{10L} \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix}$$

$$+6\int_0^L \mathbf{W_e}\frac{d\mathbf{u}}{dx}dx = +6\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix} \frac{d\mathbf{u}}{dx}dx =$$

$$= +6\begin{bmatrix} -\frac{1}{2} & \frac{57}{80} & -\frac{3}{10} & \frac{7}{80} \\ -\frac{57}{80} & 0 & \frac{81}{80} & -\frac{3}{10} \\ \frac{3}{10} & -\frac{81}{80} & 0 & -\frac{3}{10} \\ -\frac{7}{80} & \frac{3}{10} & -\frac{57}{80} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix}$$

$$-5\int_0^L \mathbf{W_e}dx = -5\begin{bmatrix} \frac{L}{8} \\ \frac{3L}{8} \\ \frac{3L}{8} \\ \frac{L}{8} \end{bmatrix}$$

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

$$\begin{bmatrix} 7\frac{37}{10L} + 6\frac{1}{2} & -7\frac{189}{40L} - 6\frac{57}{80} & 7\frac{27}{20L} + 6\frac{3}{10} & -7\frac{13}{40L} - 6\frac{7}{80} \\ -7\frac{189}{40L} + 6\frac{57}{80} & 7\frac{54}{5L} + 0 & -7\frac{297}{40L} - 6\frac{81}{80} & 7\frac{27}{20L} + 6\frac{3}{10} \\ 7\frac{27}{20L} - 6\frac{3}{10} & -7\frac{297}{40L} + 6\frac{81}{80} & 7\frac{54}{5L} + 0 & -7\frac{189}{40L} - 6\frac{57}{80} \\ -7\frac{13}{40L} + 6\frac{7}{80} & 7\frac{27}{20L} - 6\frac{3}{10} & -7\frac{189}{40L} - 6\frac{57}{80} & 7\frac{37}{10L} - 6\frac{1}{2} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} -5\frac{L}{8} - 7\frac{du}{dx}\big|_i \\ -5\frac{3L}{8} \\ -5\frac{3L}{8} \\ -5\frac{L}{8} + 7\frac{du}{dx}\big|_l \end{bmatrix} \tag{5}$$

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

$$\begin{bmatrix} a_{11} & 0 & 0 & a_{14} \\ a_{21} & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 7\frac{du}{dx}\big|_i \\ b_2 \\ b_3 \\ b_4 + 7\frac{du}{dx}\big|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 7\frac{du}{dx}\big|_i \\ b_4 + 7\frac{du}{dx}\big|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} b_1 - 7\frac{du}{dx}\big|_i \\ b_2 + 7\frac{du}{dx}\big|_l \end{bmatrix},$$

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

$$\begin{bmatrix} a_{11}^1 & a_{12}^1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{21}^n & a_{22}^n \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} b_1^1 - 7\frac{du}{dx}\big|_0 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ b_2^n + 7\frac{du}{dx}\big|_L \end{bmatrix}$$

Учет граничных условий

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

$$
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a^1_{21} & a^1_{22}+a^2_{11} & a^2_{12} & 0 & \cdots & 0 & 0 & 0 \\
0 & a^2_{21} & a^2_{22}+a^3_{11} & a^3_{12} & 0 & \cdots & 0 & 0 \\
0 & 0 & a^3_{21} & a^3_{22}+\cdots & & & & 0 \\
\vdots & \vdots & \vdots & \vdots & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots+a^n_{11} & a^n_{12} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n
\end{bmatrix}
=
\begin{bmatrix}
10 \\ b^1_2+b^2_1 \\ b^2_2+b^3_1 \\ b^3_2+b^4_1 \\ \vdots \\ b^{n-1}_2+b^n_1 \\ b^n_2+7\cdot -5
\end{bmatrix}
$$

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

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

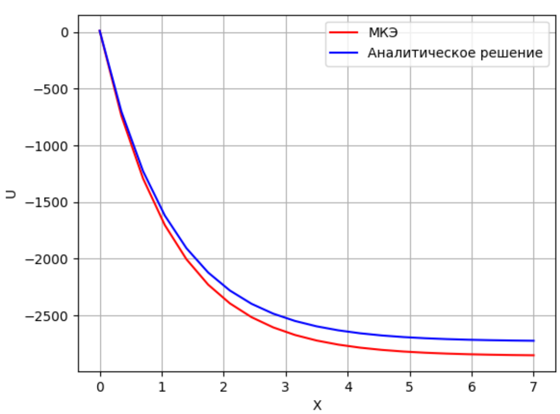

Рис. 2. Результат работы программы для 20 КЭ

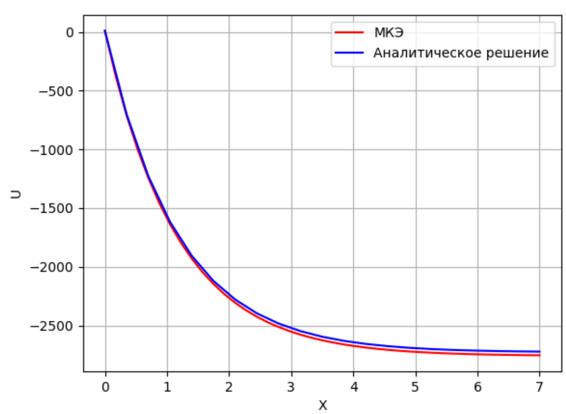

Рис. 3. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.350000	-701.306699	-739.369977	38.063278
0.700000	-1228.180068	-1293.176047	64.995979
1.050000	-1618.421864	-1702.434881	84.013017
1.400000	-1907.444503	-2004.854454	97.409951
1.750000	-2121.482145	-2228.305878	106.823732
2.100000	-2279.969536	-2393.389539	113.420003
2.450000	-2397.304288	-2515.331810	118.027522
2.800000	-2484.152416	-2605.386967	121.234551
3.150000	-2548.415496	-2671.873387	123.457891
3.500000	-2595.947163	-2720.939437	124.992274
3.850000	-2631.083893	-2757.129561	126.045668
4.200000	-2657.038228	-2783.802696	126.764468
4.550000	-2676.190077	-2803.441535	127.251458
4.900000	-2690.302522	-2817.881111	127.578590
5.250000	-2700.681683	-2828.477755	127.796072
5.600000	-2708.295160	-2836.233970	127.938810
5.950000	-2713.859768	-2841.890737	128.030969
6.300000	-2717.906536	-2845.995739	128.089203
6.650000	-2720.828861	-2848.953784	128.124923
7.000000	-2722.918178	-2851.064078	128.145900

Таблица 1. 20 линейных КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.175000	-372.288366	-377.304143	5.015777
0.350000	-701.306699	-710.545499	9.238800
0.525000	-984.475089	-997.267712	12.792623
0.700000	-1228.180068	-1243.961826	15.781758
0.875000	-1437.918573	-1456.213156	18.294583
1.050000	-1618.421864	-1638.827673	20.405808
1.225000	-1773.762173	-1795.940745	22.178572
1.400000	-1907.444503	-1931.110715	23.666212
1.575000	-2022.485637	-2047.399409	24.913772
1.750000	-2121.482145	-2147.441426	25.959280
1.925000	-2206.668916	-2233.503742	26.834826
2.100000	-2279.969536	-2307.537014	27.567478
2.275000	-2343.039651	-2371.219714	28.180063
2.450000	-2397.304288	-2425.996106	28.691818
2.625000	-2443.989981	-2473.108932	29.118951
2.800000	-2484.152416	-2513.627527	29.475111
2.975000	-2518.700230	-2548.472015	29.771785
3.150000	-2548.415496	-2578.434133	30.018636
3.325000	-2573.971350	-2604.195141	30.223792
3.500000	-2595.947163	-2626.341241	30.394079
3.675000	-2614.841607	-2645.376839	30.535232
3.850000	-2631.083893	-2661.735958	30.652066
4.025000	-2645.043444	-2675.792061	30.748617
4.200000	-2657.038228	-2687.866498	30.828270
4.375000	-2667.341920	-2698.235781	30.893861
4.550000	-2676.190077	-2707.137838	30.947761
4.725000	-2683.785443	-2714.777399	30.991956
4.900000	-2690.302522	-2721.330626	31.028105
5.075000	-2695.891510	-2726.949101	31.057591
5.250000	-2700.681683	-2731.763253	31.081570
5.425000	-2704.784310	-2735.885315	31.101005
5.600000	-2708.295160	-2739.411856	31.116696
5.775000	-2711.296663	-2742.425972	31.129309
5.950000	-2713.859768	-2744.999166	31.139398
6.125000	-2716.045539	-2747.192960	31.147422
6.300000	-2717.906536	-2749.060295	31.153759
6.475000	-2719.487998	-2750.646723	31.158725
6.650000	-2720.828861	-2751.991440	31.162578
6.825000	-2721.962639	-2753.128173	31.165533
7.000000	-2722.918178	-2754.085943	31.167765

Таблица 2. 40 линейных КЭ

Максимальная абсолютная погрешность 1.281459e+02 и 3.116776e+01 соответственно.

Кубическая функция-формы

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

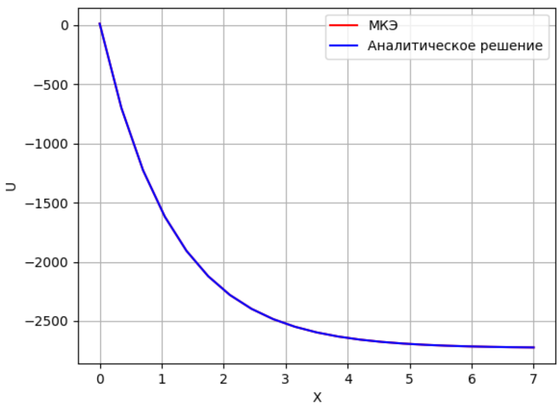

Рис. 4. Результат работы программы для 20 КЭ

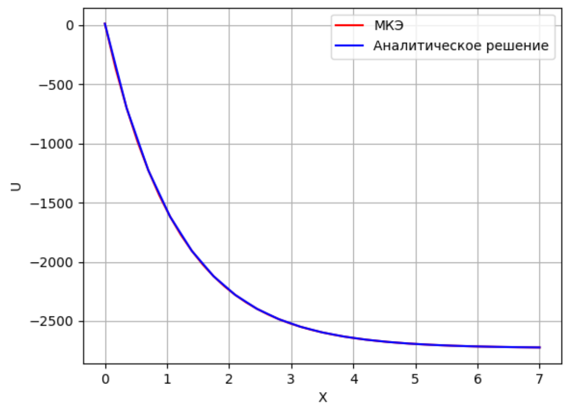

Рис. 5. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.350000	-701.306699	-701.306735	0.000035
0.700000	-1228.180068	-1228.180128	0.000061
1.050000	-1618.421864	-1618.421943	0.000078
1.400000	-1907.444503	-1907.444594	0.000091
1.750000	-2121.482145	-2121.482245	0.000100
2.100000	-2279.969536	-2279.969642	0.000106
2.450000	-2397.304288	-2397.304398	0.000110
2.800000	-2484.152416	-2484.152529	0.000113
3.150000	-2548.415496	-2548.415612	0.000115
3.500000	-2595.947163	-2595.947279	0.000117
3.850000	-2631.083893	-2631.084010	0.000118
4.200000	-2657.038228	-2657.038346	0.000118
4.550000	-2676.190077	-2676.190196	0.000119
4.900000	-2690.302522	-2690.302641	0.000119
5.250000	-2700.681683	-2700.681802	0.000119
5.600000	-2708.295160	-2708.295279	0.000119
5.950000	-2713.859768	-2713.859887	0.000119
6.300000	-2717.906536	-2717.906656	0.000119
6.650000	-2720.828861	-2720.828981	0.000120
7.000000	-2722.918178	-2722.918297	0.000120

Таблица 3. 20 кубических КЭ

X	Аналитическое решение	МКЭ-решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.175000	-372.288366	-372.288367	0.000000
0.350000	-701.306699	-701.306700	0.000001
0.525000	-984.475089	-984.475090	0.000001
0.700000	-1228.180068	-1228.180069	0.000001
0.875000	-1437.918573	-1437.918574	0.000001
1.050000	-1618.421864	-1618.421866	0.000001
1.225000	-1773.762173	-1773.762175	0.000001
1.400000	-1907.444503	-1907.444505	0.000001
1.575000	-2022.485637	-2022.485639	0.000001
1.750000	-2121.482145	-2121.482147	0.000002
1.925000	-2206.668916	-2206.668918	0.000002
2.100000	-2279.969536	-2279.969538	0.000002
2.275000	-2343.039651	-2343.039652	0.000002
2.450000	-2397.304288	-2397.304290	0.000002
2.625000	-2443.989981	-2443.989983	0.000002
2.800000	-2484.152416	-2484.152417	0.000002
2.975000	-2518.700230	-2518.700232	0.000002
3.150000	-2548.415496	-2548.415498	0.000002
3.325000	-2573.971350	-2573.971351	0.000002
3.500000	-2595.947163	-2595.947165	0.000002
3.675000	-2614.841607	-2614.841609	0.000002
3.850000	-2631.083893	-2631.083894	0.000002
4.025000	-2645.043444	-2645.043446	0.000002
4.200000	-2657.038228	-2657.038229	0.000002
4.375000	-2667.341920	-2667.341922	0.000002
4.550000	-2676.190077	-2676.190079	0.000002
4.725000	-2683.785443	-2683.785445	0.000002
4.900000	-2690.302522	-2690.302523	0.000002
5.075000	-2695.891510	-2695.891511	0.000002
5.250000	-2700.681683	-2700.681685	0.000002
5.425000	-2704.784310	-2704.784312	0.000002
5.600000	-2708.295160	-2708.295162	0.000002
5.775000	-2711.296663	-2711.296665	0.000002
5.950000	-2713.859768	-2713.859770	0.000002
6.125000	-2716.045539	-2716.045541	0.000002
6.300000	-2717.906536	-2717.906538	0.000002
6.475000	-2719.487998	-2719.488000	0.000002
6.650000	-2720.828861	-2720.828863	0.000002
6.825000	-2721.962639	-2721.962641	0.000002
7.000000	-2722.918178	-2722.918180	0.000002

Таблица 4. 40 кубических КЭ

Максимальная абсолютная погрешность 1.195516e-04 и 1.830221e-06 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.

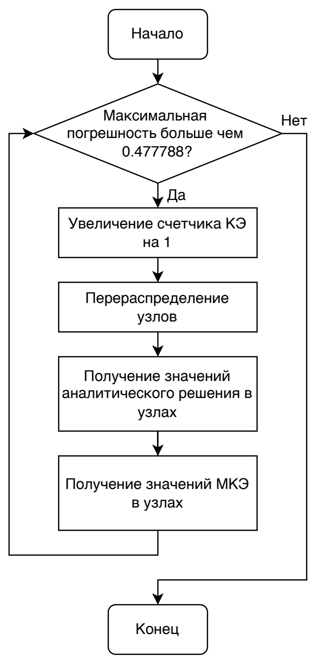

Рис. 6. Алгоритм нахождения количества КЭ, заданную точность

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 19354 .

7 Код

Листинг 1. *Реализация МКЭ*

```

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

double EPS = 1e-16;
double X_BEGIN = 0.0;
double X_END = 7.0;
size_t ELEMS_NUM = 20;
```

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

double a = 7.0, B = 6.0, C = 0.0, D = -5.0, usl_left = 10.0, usl_right = -5.0; // 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 5. / 36. * (6. * x + 49. * exp(6. - (6. * x) / 7.) - 49. * exp(6.) + 72.);
}

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 - C * L/3.0 + B*1.0/2.0, -a/L - C * L/6.0 - B*1.0/2.0},
        { -a/L - C * L/6.0 + B*1.0/2.0, a/L - C*L/3.0 - B*1.0/2.0},
    };

    // 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 ( 0 == 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 ( 1 == 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;
    }
```

```

    // 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*37.0/(10.0*L) - C*8*L/105.0 +B*1.0/2.0, -a*189.0/(40.0*L) - C*33*L/560.0
            - B*57/80.0, a*27.0/(20.0*L) + C*3*L/140.0 + B*3.0/10.0,
            -a*13.0/(40.0*L) - C*19.0*L/1680.0 - B*7/80.0},
        { -a*189.0/(40.0*L) - C*33*L/560.0 + B*57/80.0, a*54.0/(5.0*L)-C*27*L/70.0,
            -a*297.0/(40*L) + C*27*L/560.0 - B*81.0/80.0, a*27.0/(20.0*L) +
            C*3*L/140.0 + B*3.0/10.0},
        { a*27.0/(20.0*L) + C*3*L/140.0 - B*3.0/10.0, -a*297.0/(40.0*L) +
            C*27*L/560.0 + B*81.0/80.0, a*54.0/(5.0*L) - C*27*L/70.0,
            -a*189.0/(40.0*L) - C*33*L/560.0 - B*57/80.0},
        { -a*13.0/(40.0*L) - C*19.0*L/1680.0 + B*7/80.0, a*27.0/(20.0*L) +
            C*3*L/140.0 - B*3.0/10.0 , -a*189.0/(40.0*L) - C*33*L/560.0 +
            B*57/80.0, a*37.0/(10.0*L) - C*8*L/105.0 - B*1.0/2.0}

    };


    // 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 (0 == 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 (1 == 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, "%lf & %lf & %lf & %lf \\\\\n", x.at(i), y_real.at(i), y.at(i),
            std::fabs(y_real.at(i) - y.at(i)));
    }
    fprintf(tab, "%lf & %lf & %lf & %lf", 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, Роздорожный И. О.*

2023, осенний семестр