

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:	Коротина Ксения Викторовна
Группа:	PK6-716
Тип задания:	Лабораторная работа
Название:	Метод конечных элементов
Вариант:	11

Студент	подпись, дата	<u>Коротина К. В.</u> Фамилия, И.О.
Преподаватель	подпись, дата	<u>Трудоношин В. А</u> Фамилия, и.о.
Оценка:		

Содержание

Метод	конечных элементов	3
1	Цель выполнения лабораторной работы	3
2	Задание	3
3	Аналитическое решение	4
4	Получение локальных матрицы жесткости и вектора нагрузок	4
	Линейная функция-формы КЭ	5
	Кубическая функция-формы КЭ	5
5	Получение глобальных матрицы жесткости и вектора нагрузок	8
	Ансамблирование	8
	Учет граничных условий	8
6	Анализ результатов	8
	Линейная функция-формы	9
	Кубическая функция-формы	12
	Нахождение количества линейных КЭ, обеспечивающих ту же точность,	
	что и 20 кубических	15
7	Код	15
8	Вывод	22

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

$$7\frac{d^2u}{dx^2} - 1u + 12 = 0, (1)$$

при следующих граничных условиях (г. у.):

$$u(x=3)=0, (2)$$

$$u(x=10) = 20. (3)$$

Количество конечных элементов

- \bullet 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.

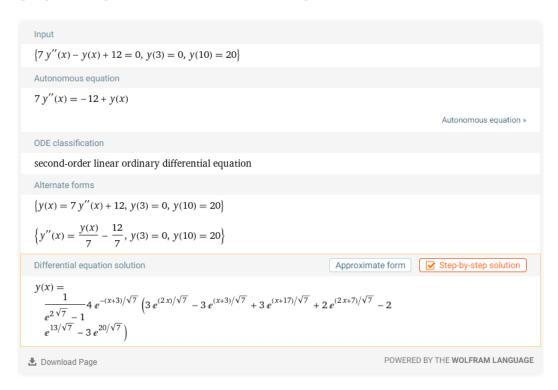


Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = \frac{1}{e^{2\sqrt{7}} - 1} - 4e^{-(x+3)/\sqrt{7}} \left(3e^{(2x)/\sqrt{7}} - 3e^{(x+3)/\sqrt{7}} + 3e^{(x+17)/\sqrt{7}} + 2e^{(2x+7)/\sqrt{7}} - 2ight. \right)$$
$$e^{13/\sqrt{7}} - 3e^{20/\sqrt{7}}$$

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 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 $K\Theta$.

В соответствии с методом Галеркина для уравнения 1:

$$\int_0^L \mathbf{W_e} \left(7 \frac{d^2 \mathbf{u}}{dx^2} - 1 \mathbf{u} + 12 \right) dx = 0, \tag{4}$$

где $\mathbf{W_e} = \mathbf{N_e}^T$.

$$\int_{0}^{L} \mathbf{W_{e}} \left(7 \frac{d^{2} \mathbf{u}}{dx^{2}} - 1 \mathbf{u} + 12 \right) dx = 7 \int_{0}^{L} \mathbf{W_{e}} \frac{d^{2} \mathbf{u}}{dx^{2}} dx - 1 \int_{0}^{L} \mathbf{W_{e}} \mathbf{u} dx + 12 \int_{0}^{L} \mathbf{W_{e}} dx = 0$$

Распишем каждое слагаемое отдельно:

$$7 \int_{0}^{L} \mathbf{W_{e}} \frac{d^{2}\mathbf{u}}{dx^{2}} dx = 7 \int_{0}^{L} \left[\begin{pmatrix} 1 - \frac{x}{L} \end{pmatrix} \right] \frac{d^{2}\mathbf{u}}{dx^{2}} dx = 7 \left[\begin{pmatrix} 1 - \frac{x}{L} \end{pmatrix} \right] \frac{d\mathbf{u}}{dx} \Big|_{0}^{L} - \frac{1}{L} \int_{0}^{L} \frac{d}{dx} \left[\begin{pmatrix} 1 - \frac{x}{L} \end{pmatrix} \right] \frac{d}{dx} \left[\begin{pmatrix} 1 - \frac{x}{L} \end{pmatrix}; \quad \frac{x}{L} \right] \left[\begin{pmatrix} u_{i} \\ u_{j} \end{pmatrix} \right] = \left[\begin{pmatrix} -7\frac{d\mathbf{u}}{dx} \\ 7\frac{d\mathbf{u}}{dx} \\ 7\frac{d\mathbf{u}}{dx} \\ 3 \end{pmatrix} \right] - 7 \left[\begin{pmatrix} \frac{1}{L}, & -\frac{1}{L} \\ -\frac{1}{L}, & \frac{1}{L} \end{pmatrix} \right] \left[u_{i} \\ u_{j} \right] - 1 \int_{0}^{L} \mathbf{W_{e}} \mathbf{u} dx = -1 \int_{0}^{L} \left[\begin{pmatrix} 1 - \frac{x}{L} \\ \frac{x}{L} \end{pmatrix} \right] \left[\begin{pmatrix} 1 - \frac{x}{L} \end{pmatrix}; \quad \frac{x}{L} \right] \left[\begin{pmatrix} u_{i} \\ u_{j} \end{pmatrix} dx = -1 \left[\frac{L}{\frac{3}{2}}, & \frac{L}{\frac{6}{2}} \right] \left[u_{i} \\ u_{j} \right] \right] - 1 \int_{0}^{L} \mathbf{W_{e}} \mathbf{u} dx = 12 \left[\frac{L}{\frac{2}{2}} \right]$$

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

$$\begin{bmatrix} 7\frac{1}{L} + 1\frac{L}{3}, & -7\frac{1}{L} + 1\frac{L}{3} \\ -7\frac{1}{L} + 1\frac{L}{3}, & 7\frac{1}{L} + 1\frac{L}{3} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -7\frac{du}{dx}|_i + 12\frac{L}{2} \\ 7\frac{du}{dx}|_j + 12\frac{L}{2} \end{bmatrix}$$

Кубическая функция-формы КЭ

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

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

$$7 \int_{0}^{L} \mathbf{W_{e}} \frac{d^{2}\mathbf{u}}{dx^{2}} dx = 7 \int_{0}^{L} \begin{bmatrix} -\frac{9x^{3}}{2L^{3}} + \frac{18x^{2}}{2L^{2}} - \frac{11x}{2L} + 1 \\ \frac{27x^{3}}{2L^{3}} - \frac{45x^{2}}{2L^{2}} + \frac{9x}{2L} \\ \frac{27x^{3}}{2L^{3}} + \frac{21x^{2}}{2L} - \frac{11x}{2L} \end{bmatrix} \frac{d^{2}\mathbf{u}}{dx^{2}} dx = \begin{bmatrix} -\frac{9x^{3}}{2L^{3}} + \frac{18x^{2}}{2L} - \frac{11x}{2L} + 1 \\ \frac{27x^{3}}{2L^{3}} - \frac{45x^{2}}{2L^{2}} - \frac{11x}{2L} + 1 \\ \frac{27x^{3}}{2L^{3}} - \frac{45x^{2}}{2L^{2}} - \frac{9x}{L} \end{bmatrix} \frac{d\mathbf{u}}{dx} \begin{bmatrix} -7 \int_{0}^{L} \frac{d}{dx} \begin{bmatrix} -\frac{9x^{3}}{2L^{3}} + \frac{18x^{2}}{2L^{2}} - \frac{11x}{2L} + 1 \\ \frac{27x^{3}}{2L^{3}} - \frac{45x^{2}}{2L^{2}} + \frac{9x}{L} \\ \frac{27x^{3}}{2L^{3}} - \frac{45x^{2}}{2L^{2}} + \frac{9x}{L} \end{bmatrix} \frac{d\mathbf{u}}{dx} \mathbf{u} = \begin{bmatrix} -7\frac{d\mathbf{u}}{dx}|_{l} \\ 0 \\ 7\frac{d\mathbf{u}}{dx}|_{l} \end{bmatrix} - 7\begin{bmatrix} \frac{3\mathbf{u}}{dx} - \frac{189}{40} & \frac{27}{20L} - \frac{13}{40L} \\ \frac{27}{20L} - \frac{297}{40L} & \frac{5L}{40} & \frac{189}{40} \\ \frac{27}{30L} - \frac{189}{40} & \frac{27}{30L} \end{bmatrix} \begin{bmatrix} u_{i} \\ u_{k} \\ u_{l} \end{bmatrix}$$

$$-1 \int_{0}^{L} \mathbf{W_{e}} \mathbf{u} dx = -1 \int_{0}^{L} \begin{bmatrix} -\frac{9x^{3}}{2L^{3}} + \frac{18x^{2}}{2L^{2}} - \frac{11x}{2L} + 1 \\ \frac{27x^{3}}{40L} - \frac{297}{20L} & \frac{5L}{40} & \frac{189}{40} \\ \frac{5L}{20L} - \frac{189}{40} & \frac{27}{30L} \end{bmatrix} \begin{bmatrix} u_{i} \\ u_{k} \\ u_{l} \end{bmatrix}$$

$$u_{k} \\ u_{i} \end{bmatrix}$$

$$u_{k} \\ u_{i} \end{bmatrix}$$

$$u_{k} \\ u_{i} \end{bmatrix}$$

$$12 \int_{0}^{L} \mathbf{W_{e}} dx = 12 \begin{bmatrix} \frac{L}{3} \\ \frac{3L}{3} \\ \frac{3L}{8} \\ \frac{3L}{8} \end{bmatrix}$$

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

$$\begin{bmatrix} 7\frac{37}{10L} + 1\frac{8L}{105} & -7\frac{189}{40L} + 1\frac{33L}{560} & 7\frac{27}{20L} - 1\frac{3L}{140} & -7\frac{13}{40L} + 1\frac{19L}{1680} \\ -7\frac{189}{40L} + 1\frac{33L}{560} & 7\frac{54}{5L} + 1\frac{27L}{70} & -7\frac{297}{29L} - 1\frac{27L}{560} & 7\frac{27}{20L} - 1\frac{3L}{340} \\ 7\frac{27}{20L} - 1\frac{3L}{140} & -7\frac{297}{40L} - 1\frac{27L}{560} & 7\frac{5L}{5L} + 1\frac{27L}{70} & -7\frac{189}{40L} + 1\frac{33L}{560} \\ -7\frac{13}{40L} + 1\frac{19L}{1680} & 7\frac{27}{20L} - 1\frac{3L}{140} & -7\frac{189}{40L} + 1\frac{33L}{560} & 7\frac{31}{10L} + 1\frac{8L}{105} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} 12\frac{L}{8} - 7\frac{du}{dx}|_i \\ 12\frac{3L}{8} \\ 12\frac{3L}{8} \\ 12\frac{L}{8} + 7\frac{du}{dx}|_l \end{bmatrix}$$
 (5)

Локальные матрица жесткости и вектор нагрузок могут быть представлены в виде:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 2\frac{du}{dx}|_i \\ b_2 \\ b_3 \\ b_4 + 2\frac{du}{dx}|_l \end{bmatrix}$$

Выполним матричные преобразования.

$$\begin{bmatrix} a_{11} - \frac{a_{12}}{a_{22}} a_{21} & a_{12} - \frac{a_{12}}{a_{22}} a_{22} & a_{13} - \frac{a_{12}}{a_{22}} a_{23} & a_{14} - \frac{a_{12}}{a_{22}} a_{24} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} - \frac{a_{32}}{a_{22}} a_{21} & a_{32} - \frac{a_{32}}{a_{22}} a_{22} & a_{33} - \frac{a_{32}}{a_{22}} a_{23} & a_{34} - \frac{a_{32}}{a_{22}} a_{24} \\ a_{41} - \frac{a_{42}}{a_{22}} a_{21} & a_{42} - \frac{a_{42}}{a_{22}} a_{22} & a_{43} - \frac{a_{42}}{a_{22}} a_{23} & a_{44} - \frac{a_{42}}{a_{22}} a_{24} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - \frac{a_{12}}{a_{22}} b_2 - 7\frac{du}{dx}|_i \\ b_2 \\ b_3 - \frac{a_{32}}{a_{22}} b_2 \\ b_4 - \frac{a_{42}}{a_{22}} b_2 + 7\frac{du}{dx}|_l \end{bmatrix}$$

$$\begin{bmatrix} a'_{11} - \frac{a'_{13}}{a'_{33}} a'_{31} & 0 - \frac{a'_{13}}{a'_{33}} 0 & a'_{13} - -\frac{a'_{13}}{a'_{33}} a'_{33} & a'_{14} - \frac{a'_{13}}{a'_{33}} a'_{34} \\ a'_{21} - \frac{a'_{23}}{a'_{33}} a'_{31} & a'_{22} - \frac{a'_{23}}{a'_{33}} 0 & a'_{23} - \frac{a'_{23}}{a'_{33}} a'_{33} & a'_{24} - \frac{a'_{23}}{a'_{33}} a'_{34} \\ a'_{31} & 0 & a'_{33} & a'_{34} \\ a'_{41} - \frac{a'_{43}}{a'_{23}} a'_{31} & 0 - \frac{a'_{43}}{a'_{23}} 0 & a'_{43} - \frac{a'_{43}}{a'_{23}} a'_{33} & a'_{44} - \frac{a'_{43}}{a'_{33}} a'_{34} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b'_1 - \frac{a'_{13}}{a'_{33}} b'_3 - 7\frac{du}{dx} |_i \\ b'_2 - \frac{a'_{23}}{a'_{33}} b'_3 \\ b'_3 \\ b'_4 - \frac{a'_{43}}{a'_{23}} b'_3 + 7\frac{du}{dx} |_i \end{bmatrix}$$

Итого получаем:

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

Для упрощения расчетов преобразуем систему выше, исключив внутренние узлы. Таким образом, при использовании кубической функции-формы, получаем:

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

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

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

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

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

Учет граничных условий

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

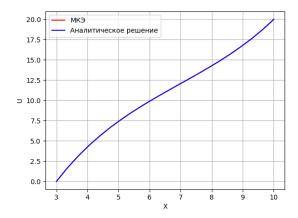
$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} 0 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_{n-1}^{n-1} + b_n^n \\ 20 \end{bmatrix}$$

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

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



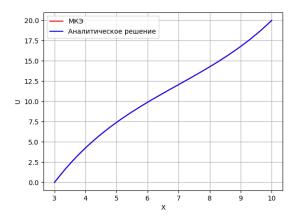


Рис. 2. Результат работы программы для 20 $\,$ Рис. 3. Результат работы программы для 40 $\,$ КЭ $\,$ КЭ

X	Аналитическое	МКЭ-	Абсолютная
	решение	решение	погрешность
3.000000	0.000000	0.000000	0.000000e+00
3.350000	1.654420	1.655201	7.811001e-04
3.700000	3.127528	3.128839	1.310491e-03
4.050000	4.445142	4.446777	1.635257e-03
4.400000	5.630352	5.632147	1.794902e-03
4.750000	6.703931	6.705754	1.822635e-03
5.100000	7.684694	7.686440	1.746486e-03
5.450000	8.589829	8.591419	1.590279e-03
5.800000	9.435198	9.436573	1.374494e-03
6.150000	10.235618	10.236735	1.117033e-03
6.500000	11.005116	11.005950	8.339115e-04
6.850000	11.757178	11.757718	5.399021e-04
7.200000	12.504985	12.505234	2.491390e-04
7.550000	13.261642	13.261618	2.429586e-05
7.900000	14.040410	14.040144	2.657894e-04
8.250000	14.854937	14.854477	4.596007e-04
8.600000	15.719498	15.718910	5.882344e-04
8.950000	16.649246	16.648614	6.317676e-04
9.300000	17.660474	17.659907	5.671116e-04
9.650000	18.770905	18.770538	3.671881e-04
10.000000	20.000000	20.000000	2.131628e-14

Таблица 1. 20 линейных КЭ

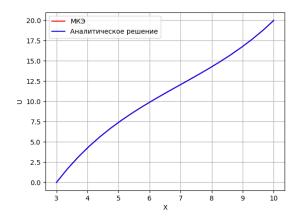
X	Аналитическое	МКЭ-	Абсолютная
	решение	решение	погрешность
3.000000	0.000000	0.000000	0.0000000e+00
3.175000	0.851606	0.851712	1.061539e-04
3.350000	1.654420	1.654615	1.949669e-04
3.525000	2.411956	2.412224	2.681101e-04
3.700000	3.127528	3.127856	3.271136e-04
3.875000	3.804270	3.804643	3.733787e-04
4.050000	4.445142	4.445550	4.081886e-04
4.225000	5.052949	5.053382	4.327195e-04
4.400000	5.630352	5.630800	4.480495e-04
4.575000	6.179878	6.180333	4.551678e-04
4.750000	6.703931	6.704386	4.549834e-04
4.925000	7.204806	7.205254	4.483323e-04
5.100000	7.684694	7.685130	4.359855e-04
5.275000	8.145696	8.146114	4.186554e-04
5.450000	8.589829	8.590226	3.970023e-04
5.625000	9.019037	9.019408	3.716408e-04
5.800000	9.435198	9.435541	3.431455e-04
5.975000	9.840134	9.840446	3.120561e-04
6.150000	10.235618	10.235897	2.788833e-04
6.325000	10.623380	10.623624	2.441135e-04
6.500000	11.005116	11.005324	2.082139e-04
6.675000	11.382499	11.382670	1.716368e-04
6.850000	11.757178	11.757313	1.348250e-04
7.025000	12.130796	12.130894	9.821570e-05
7.200000	12.504985	12.505047	6.224527e-05
7.375000	12.881385	12.881412	2.735390e-05
7.550000	13.261642	13.261636	6.010012e-06
7.725000	13.647421	13.647384	3.738521e-05
7.900000	14.040410	14.040344	6.629291e-05
8.075000	14.442329	14.442237	9.223201e-05
8.250000	14.854937	14.854822	1.146742e-04
8.425000	15.280040	15.279907	1.330586e-04
8.600000	15.719498	15.719352	1.467870e-04
8.775000	16.175236	16.175080	1.552174e-04
8.950000	16.649246	16.649088	1.576589e-04
9.125000	17.143604	17.143451	1.533651e-04
9.300000	17.660474	17.660333	1.415277e-04
9.475000	18.202118	18.201996	1.212693e-04
9.650000	18.770905	18.770814	9.163616e-05
9.825000	19.369326	19.369275	5.159009e-05
10.000000	20.000000	20.000000	2.131628e-14

Таблица 2. 40 линейных КЭ 11

Максимальная абсолютная погрешность 1.822635e-03 и 4.551678e-04 соответственно.

Кубическая функция-формы

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



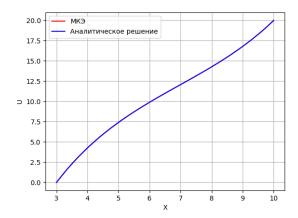


Рис. 4. Результат работы программы для 20 $\,$ Рис. 5. Результат работы программы для 40 $\,$ КЭ $\,$ КЭ

X	Аналитическое	МКЭ-	Абсолютная
	решение	решение	погрешность
3.000000	0.000000	0.000000	0.000000e+00
3.350000	1.654420	1.654420	2.850054e-11
3.700000	3.127528	3.127528	4.784440e-11
4.050000	4.445142	4.445142	5.972911e-11
4.400000	5.630352	5.630352	6.561240e-11
4.750000	6.703931	6.703931	6.668266e-11
5.100000	7.684694	7.684694	6.394441e-11
5.450000	8.589829	8.589829	5.831069e-11
5.800000	9.435198	9.435198	5.047696e-11
6.150000	10.235618	10.235618	4.115996e-11
6.500000	11.005116	11.005116	3.086242e-11
6.850000	11.757178	11.757178	2.015277e-11
7.200000	12.504985	12.504985	9.579892e-12
7.550000	13.261642	13.261642	3.677059e-13
7.900000	14.040410	14.040410	9.173107e-12
8.250000	14.854937	14.854937	1.627676e-11
8.600000	15.719498	15.719498	2.101253e-11
8.950000	16.649246	16.649246	2.264855e-11
9.300000	17.660474	17.660474	2.039258e-11
9.650000	18.770905	18.770905	1.321609e-11
10.000000	20.000000	20.000000	2.131628e-14

Таблица 3. 20 кубических КЭ

X	Аналитическое	МКЭ-	Абсолютная
	решение	решение	погрешность
3.000000	0.000000	0.000000	0.000000e+00
3.175000	0.851606	0.851606	3.621548e-13
3.350000	1.654420	1.654420	6.739054e-13
3.525000	2.411956	2.411956	9.698908e-13
3.700000	3.127528	3.127528	1.206590e-12
3.875000	3.804270	3.804270	1.426859e-12
4.050000	4.445142	4.445142	1.604938e-12
4.225000	5.052949	5.052949	1.758593e-12
4.400000	5.630352	5.630352	1.910472e-12
4.575000	6.179878	6.179878	2.017053e-12
4.750000	6.703931	6.703931	2.121858e-12
4.925000	7.204806	7.204806	2.185807e-12
5.100000	7.684694	7.684694	2.235545e-12
5.275000	8.145696	8.145696	2.287948e-12
5.450000	8.589829	8.589829	2.305711e-12
5.625000	9.019037	9.019037	2.302158e-12
5.800000	9.435198	9.435198	2.295053e-12
5.975000	9.840134	9.840134	2.287948e-12
6.150000	10.235618	10.235618	2.277289e-12
6.325000	10.623380	10.623380	2.231104e-12
6.500000	11.005116	11.005116	2.181366e-12
6.675000	11.382499	11.382499	2.126299e-12
6.850000	11.757178	11.757178	2.044587e-12
7.025000	12.130796	12.130796	1.982414e-12
7.200000	12.504985	12.504985	1.897149e-12
7.375000	12.881385	12.881385	1.834977e-12
7.550000	13.261642	13.261642	1.731948e-12
7.725000	13.647421	13.647421	1.621814e-12
7.900000	14.040410	14.040410	1.531220e-12
8.075000	14.442329	14.442329	1.408651e-12
8.250000	14.854937	14.854937	1.289635e-12
8.425000	15.280040	15.280040	1.174172e-12
8.600000	15.719498	15.719498	1.046274e-12
8.775000	16.175236	16.175236	9.450218e-13
8.950000	16.649246	16.649246	8.064660e-13
9.125000	17.143604	17.143604	6.608047e-13
9.300000	17.660474	17.660474	5.329071e-13
9.475000	18.202118	18.202118	3.943512e-13
9.650000	18.770905	18.770905	2.771117e-13
9.825000	19.369326	19.369326	1.421085e-13
10.000000	20.000000	20.000000	2.131628e-14

Таблица 4. 40 кубических К
Э14

Максимальная абсолютная погрешность 6.668266e-11 и 2.305711e-12 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 ($cub_{20} = 6.668266e - 11$)

Реализовав данный алгоритм с начальным количеством K9=20 и увеличивая счетчик всегда на 1 получаем необходимое количество K9, равное 2.

7 Код

Листинг 1. Реализация МКЭ

```
1
2 #include <iostream>
3 #include <vector>
4 #include <cmath>
5
6 //define OSN
7 //define TABLE
8 //define CUBE
9
10 constexpr double EPS = 1e-16,CUB= 6.668266e-11;
11 double EPS = 1e-16;
```

```
12 double X BEGIN = 3.0;
13 double X END = 10.0;
14 \text{ size } t \text{ ELEMS } NUM = ;
15 double L = (X END - X_BEGIN) / ELEMS_NUM;
17 constexpr double a = 7.0, B = 0.0, C = -1.0, D = 12.0, usl left = 0.0, usl right = 20.0;
         au''+Bu'+Cu+D=0
18
19 std::vector<double> solve with gauss(std::vector<std::vector<double>>& A,
         std::vector<double>& b){
         size t row size = A.size();
20
21
         size t col size = A.back().size();
22
         // Прямой ход Гаусса
23
         double pivot = 0.0;
         for (size t i = 0; i < row size; i++) {
24
              for (size t j = i + 1; j < col_size; j++) {
25
26
                    if (std::abs(A.at(j).at(i)) < EPS)  {
                         continue;
27
28
29
                    pivot = A.at(j).at(i) / A.at(i).at(i);
30
                    b.at(j) = pivot * b.at(i);
                   for (size t k = 0; k < row size; k++) {
31
                         A.at(j).at(k) = pivot * A.at(i).at(k);
32
                    }
33
              }
34
         }
35
         // Обратный ход Гаусса
36
         std::vector<double> x(row size);
37
         for (int i = row size -1.0; i >= 0; i—) {
38
              x.at(i) = b.at(i);
39
              for (size t j = i + 1; j < row_size; j++) {
40
                    x.at(i) = x.at(j) * A.at(i).at(j);
41
42
              x.at(i) /= A.at(i).at(i);
43
44
         }
45
         return x;
46 }
47
48 double analytical solution(double x) {
         double rez = (4. * \exp(-(x + 3.) / \operatorname{sqrt}(7.)) * (3. * \exp((2. * x) / \operatorname{sqrt}(7.)) - 3. * \exp((x + 3.) / \operatorname{sqrt}(7.)) + (3. * \exp((2. * x) / \operatorname{sqrt}(7.))) + (3. * \exp((2. * x) / \operatorname{sqrt}(7.))) + (3. * \exp((2. * x) / \operatorname{sqrt}(7.))) + (3. * \exp((2. * x) / \operatorname{sqrt}(7.)))) + (3. * \exp((2. * x) / \operatorname{sqrt}(7.))))
49
              +3.)/sqrt(7.)) + 3. * exp((x + 17.)/sqrt(7.)) + 2. * exp((2. * x + 7.)/sqrt(7.)) -
              2. * \exp(13. / \operatorname{sqrt}(7.)) - 3. * \exp(20. / \operatorname{sqrt}(7.))) / (\exp(2. * \operatorname{sqrt}(7.)) - 1.);
50
         return rez;
51 }
52
```

```
53 std::vector<double> build analytical solution(std::vector<double>& x vec) {
      size t \times vec size = x vec.size();
54
55
      std::vector<double> y vec = std::vector<double>(x vec size);
      for (size t i = 0; i < x vec size; i++) {
56
           y vec.at(i) = analytical solution(x vec.at(i));
57
58
      }
59
      return y vec;
60 }
61
62 std::vector<double> build linear solution(size t elems num) {
      double L = (X END - X BEGIN) / elems num;
63
      size t \text{ size} = elems \text{ num} + 1;
64
      std::vector< std::vector<double> > A(size, std::vector<double>(size));
65
      std::vector<double> b(size);
66
67
68
      // Локальная матрица жесткости для линейного КЭ
      std::vector< std::vector<double> > local matrix = {
69
           \{a/L - C * L/3.0 + B*1.0/2.0, -a/L - C * L/6.0 - B*1.0/2.0\},\
70
           \{-a/L - C * L/6.0 + B*1.0/2.0, a/L - C*L/3.0 - B*1.0/2.0\},\
71
72
      };
73
      // Ансамблирование и получение глобальной матрицы жесткости для линейного КЭ
74
      for (size t i = 0; i < elems num; i++) {
75
76
           for (size t j = 0; j < 2; j++) {
               for (size t k = 0; k < 2; k++) {
77
                   A.at(i + j).at(i + k) += local\_matrix.at(j).at(k);
78
79
           }
80
81
      }
82
83
           for (size t i = 0; i < size; i++) {
           b.at(i) = D * L;
84
      }
85
86
      // Учет ГУ
87
      b.at(0) = usl left;
88
      A.at(0).at(0) = 1;
89
      A.at(0).at(1) = 0;
90
91
      b.at(size - 1) = usl right;
92
      A.at(size -1).at(size -1) = 1;
93
      A.at(size -1).at(size -2) = 0;
94
95
96
      // Решение полученной СЛАУ методом Гаусса
      std::vector<double> res = solve_with_gauss(A, b);
97
      return res;
98
```

```
99 }
100
101 std::vector<double> build cube solution(size t elems num) {
                        double L = (X END - X BEGIN) / elems num;
102
                        size t \text{ size} = elems \text{ num} + 1;
103
                        std::vector< std::vector<double> > A(size,std::vector<double>(size));
104
                        std::vector<double> b(size);
105
106
                        // Локальная матрица жесткости для кубического КЭ
107
                                  std::vector< std::vector< double> > local matrix = {
108
109
                                     \{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\}
110
                                                  -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,
111
                                                  -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*3*L/140.0 - B*3.0/(40.0*L) + C*3*L/140.0 - B*3*L/140.0 - B*3.0/(40.0*L) + C*3*L/140.0 - B*3.0/
112
                                                  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*19.0*L/1680.0 + C*19.0*L
113
                                                  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
114
                     };
115
116
                         // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего
117
                                      элементов учитываются далее)
                        std::vector<double> local b = { D * L / 8.0,
118
                                                                                                                            D*3.0 * L / 8.0
119
                                                                                                                             D*3.0 * L / 8.0,
120
121
                                                                                                                             D * L / 8.0 };
122
123
124
                         // Производим матричные преобразования для обнуления элементов локальной
                                      матрицы жесткости, относящихся к внутренним узлам
                        for (size t i = 1; i < 3; i++) {
125
                                     for (size t = 0; j < 4; j++) {
126
                                                 if (std::fabs(local matrix.at(j).at(i)) > EPS \&\& i!= j) {
127
                                                              double val = local matrix.at(j).at(i) /local_matrix.at(i).at(i);
128
                                                              local b.at(i) = val * local b.at(i);
129
                                                              for (size t k = 0; k < 4; k++) {
130
                                                                          local matrix.at(j).at(k) -= val *local matrix.at(i).at(k);
131
                                                              }
132
133
134
                                                 continue;
```

```
}
135
136
137
138
        // Исключаем внутренние узлы из рассмотрения
139
        std::vector < std::vector < double > > local matrix mod = { { local matrix.at(0).at(0),
140
            local matrix.at(0).at(3) },
141
                                                                   { local matrix.at(3).at(0),
                                                                       local matrix.at(3).at(3)
                                                                       } };
        std::vector<double> local_b_mod = { local_b.at(0),
142
                                             local_b.at(3)};
143
144
        // Ансамблирование и получение глобальной матрицы жесткости для кубического КЭ
145
        for (size t i = 0; i < elems num; i++) {
146
147
            for (size t j = 0; j < 2; j++) {
                for (size t k = 0; k < 2; k++) {
148
                    A.at(i + j).at(i + k) += local matrix mod.at(j).at(k);
149
150
            }
151
152
        }
            for (size t i = 0; i < elems num; i++) {
153
            b.at(i) += local b mod.at(0);
154
            b.at(i+1) += local b mod.at(1);
155
156
        // Учет ГУ
157
        b.at(0) = usl left;
158
        A.at(0).at(0) = 1;
159
160
        A.at(0).at(1) = 0;
161
162
        b.at(size - 1) = usl right;
        A.at(size -1).at(size -1) = 1;
163
        A.at(size - 1).at(size - 2) = 0;
164
165
        // Решение полученной СЛАУ методом Гаусса
166
        std::vector<double> res = solve with gauss(A, b);
167
168
        return res;
169 }
170
171 double calc abs error(const std::vector<double>& y real, const std::vector<double>&
        double max_err = 0.0;
172
        for (size t i = 0; i < y real.size(); i++) {
173
174
            double err = std::fabs(y real.at(i) - y.at(i));
            if (err > max err) {
175
176
                \max \ err = err;
```

```
}
177
178
179
        return max err;
180 }
181
182 int main() {
183 #ifdef OSN
184
        std::vector < double > x(ELEMS NUM + 1);
185
        for (size t i = 0; i < x.size(); i++) {
186
187
            x.at(i) = X BEGIN + i * L;
188
189
        size t \times size = x.size();
190
191
192 #ifdef CUBE
        std::vector<double> y = build cube solution(ELEMS NUM);
193
194 #else
        std::vector<double> y = build linear solution(ELEMS NUM);
195
196 #endif
197
        std::vector<double> y real = build analytical solution(x);
198
199 #ifdef TABLE
200
        for (size t i = 0; i < x.size(); i++) {
            std::cout<<x.at(i)<<" & "<<y real.at(i)<<" & "<<y.at(i)<<"
201
                \&"<<std::fabs(y real.at(i) - y.at(i))<<"\\\\"<<std::endl;
        }
202
203 #endif
204
        FILE* gp = popen("gnuplot -persist", "w");
205
206
        fprintf(gp, "$predict << EOD\n");</pre>
        for (size t i = 0; i < x size; i++) {
207
            fprintf(gp, "%lf %lf\n", x.at(i), y.at(i));
208
209
        fprintf(gp, "EOD\n");
210
        fprintf(gp, "$real << EOD\n");</pre>
211
        for (size t i = 0; i < x size; i++) {
212
            fprintf(gp, "%lf %lf\n", x.at(i), y real.at(i));
213
214
        fprintf(gp, "EOD\n");
215
        fprintf(gp, "set grid\n");
216
         fprintf(gp, "plot '$predict' using 1:2 with lp lc '#ba55d3' lw 1.5 pt 7 ps 0.5 title
217
             МКЭрешение'— (%zu KЭ)'," "'$real' using 1:2 with lines lc rgb '#afdafc' lt 1 lw 2
             title аналитическое' решение(%zu KЭ)',\n", ELEMS_NUM, ELEMS_NUM);
         printf("Абсолютная погрешность: %e\n", calc_abs_error(y_real, y));
218
219
```

```
220
        //нахождение количества линейных КЭ
221
222 #else
        int N=20,n=10000:
223
224
        double err=10;
        FILE* gp = popen("gnuplot -persist", "w");
225
       fprintf(gp, "$predict << EOD\n");</pre>
226
227
        while (err>CUB && n<=19000){
228
229
            double L = (X END - X BEGIN) / N;
            std::vector < double > x(N + 1);
230
            for (size t i = 0; i < x.size(); i++) {
231
                x.at(i) = X BEGIN + i * L;
232
            }
233
234
235
            std::vector<double> y r(N + 1);
            std::vector < double > y s(N + 1);
236
237
           y = build linear solution(N);
238
           y r = build analytical solution(x);
239
240
            err=calc abs error(y r, y s);
241
242
            fprintf(gp, "%d %e\n", N, err);
243
            printf("Абсолютная погрешность: %e количествоКЭ: %d\n", calc abs error(y r,
244
                y_s), N);
            N+=1;
245
            n+=1;
246
247
        }
248
249
        fprintf(gp, "EOD\n");
        fprintf(gp, "set grid\n");
250
       fprintf(gp, "set logscale y 2\n");
251
        fprintf(gp, "plot '$predict' using 1:2 with lp lc '#cd853f' lw 1.5 pt 7 ps 0.5 title
252
            Абсолютная' погрешность', \n");
        std::cout<<"Количество линейныхКЭ"<<N-1<<std::endl;
253
        printf("Абсолютная погрешность: %e\n", err);
254
255
256 #endif
257
        return 0;
258 }
```

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: \bigcirc доцент кафедры PK-6, кандидат технических наук, до-

цент, Трудоношин В.А.

Решение и вёрстка: С студент группы РК6-716, Коротина К. В.

2023, осенний семестр