



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:	РК6-746
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	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	4
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	21

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 относительно аналитического способа решения

2 Задание

Решить с помощью МКЭ уравнение 1

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

при следующих граничных условиях (г. у.):

$$u(x = 0) = 10, \quad (2)$$

$$u'(x = 7) = -5. \quad (3)$$

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.

The screenshot shows the Wolfram Language interface with the input $7y''+6y'-5=0, y(0)=10, y'(7)=-5$. The interface includes buttons for "NATURAL LANGUAGE", "MATH INPUT", "EXTENDED KEYBOARD", "EXAMPLES", "UPLOAD", and "RANDOM". The input is processed into the following sections:

- Input:** $\{7 y''(x) + 6 y'(x) - 5 = 0, y(0) = 10, y'(7) = -5\}$
- Autonomous equation:** $7 y''(x) = 5 - 6 y'(x)$
- ODE classification:** second-order linear ordinary differential equation
- Alternate form:** $\{y''(x) = \frac{5}{7} - \frac{6 y'(x)}{7}, y(0) = 10, y'(7) = -5\}$
- Differential equation solution:** $y(x) = \frac{5}{36} (6 x + 49 e^{6-(6 x)/7} - 49 e^6 + 72)$

Buttons for "Approximate form" and "Step-by-step solution" are visible. At the bottom, there is a "Download Page" button and the text "POWERED BY THE WOLFRAM LANGUAGE".

Рис. 1. Аналитическое решение

Таким образом, получаем:

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

4 Получение локальных матрицы жесткости и вектора нагрузок

Составим локальные матрицу жесткости и вектор нагрузок для уравнения 1.

Линейная функция-формы КЭ

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

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:

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

где $\mathbf{W}_e = \mathbf{N}_e^T$.

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

Распишем каждое слагаемое отдельно:

$$\begin{aligned} 7 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx &= 7 \int_0^L \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d^2 \mathbf{u}}{dx^2} dx = 7 \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d\mathbf{u}}{dx} \Big|_0^L - \\ &- 7 \int_0^L \frac{d}{dx} \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d}{dx} \left[\left(1 - \frac{x}{L} \right); \frac{x}{L} \right] \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -7 \frac{d\mathbf{u}}{dx} \Big|_i \\ 7 \frac{d\mathbf{u}}{dx} \Big|_j \end{bmatrix} - 7 \begin{bmatrix} \frac{1}{L}, & -\frac{1}{L} \\ -\frac{1}{L}, & \frac{1}{L} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ &+ 6 \int_0^L \mathbf{W}_e \frac{d\mathbf{u}}{dx} dx = +6 \int_0^L \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d\mathbf{u}}{dx} \begin{bmatrix} u_i \\ u_j \end{bmatrix} dx = \\ &= \frac{6}{L} \int_0^L \begin{bmatrix} \left(1 - \frac{x}{L} \right) & \left(-1 + \frac{x}{L} \right) \\ -\frac{x}{L} & \frac{x}{L} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} dx = 6 \begin{bmatrix} -\frac{1}{2}, & \frac{1}{2} \\ -\frac{1}{2}, & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ &- 5 \int_0^L \mathbf{W}_e dx = -5 \left[\frac{L}{2} \right] \end{aligned}$$

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.

$$\begin{aligned}
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{40L}{297} & \frac{20L}{27} \\ \frac{40}{27} & -\frac{5L}{297} & \frac{54}{40L} & -\frac{189}{20L} \\ -\frac{20L}{13} & \frac{40L}{27} & -\frac{5L}{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}
\end{aligned}$$

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

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

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.

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

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

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:

$$\begin{bmatrix} a''_{11} & a''_{14} \\ a''_{41} & a''_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b''_1 - 7\frac{du}{dx}|_i \\ b''_4 + 7\frac{du}{dx}|_l \end{bmatrix}$$

5 Получение глобальной матрицы жесткости и вектора нагрузок

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

Ансамблирование

Пусть локальная 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 разбиении области на n КЭ, глобальная матрица жесткости будет иметь размерность $(n+1) \cdot (n+1)$:

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

Учет граничных условий

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

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_{21}^3 & a_{22}^3 + \dots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots + a_{11}^n & a_{12}^n \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{21}^n & a_{22}^n \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} 10 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ b_2^n + 7 \cdot -5 \end{bmatrix}$$

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

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

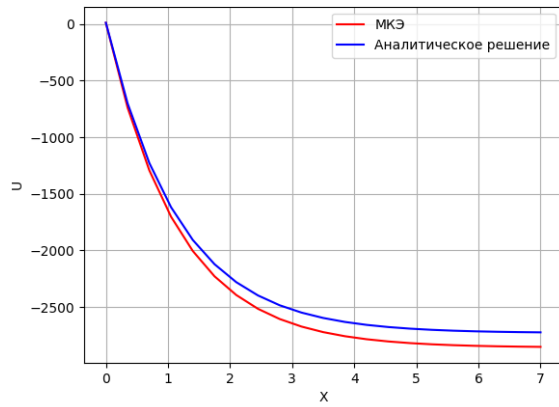


Рис. 2. Результат работы программы для 20 КЭ

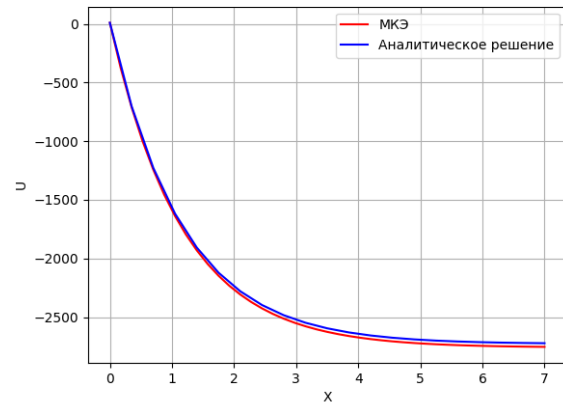


Рис. 3. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000e+00
0.350000	-701.306699	-739.369977	3.806328e+01
0.700000	-1228.180068	-1293.176047	6.499598e+01
1.050000	-1618.421864	-1702.434881	8.401302e+01
1.400000	-1907.444503	-2004.854454	9.740995e+01
1.750000	-2121.482145	-2228.305878	1.068237e+02
2.100000	-2279.969536	-2393.389539	1.134200e+02
2.450000	-2397.304288	-2515.331810	1.180275e+02
2.800000	-2484.152416	-2605.386967	1.212346e+02
3.150000	-2548.415496	-2671.873387	1.234579e+02
3.500000	-2595.947163	-2720.939437	1.249923e+02
3.850000	-2631.083893	-2757.129561	1.260457e+02
4.200000	-2657.038228	-2783.802696	1.267645e+02
4.550000	-2676.190077	-2803.441535	1.272515e+02
4.900000	-2690.302522	-2817.881111	1.275786e+02
5.250000	-2700.681683	-2828.477755	1.277961e+02
5.600000	-2708.295160	-2836.233970	1.279388e+02
5.950000	-2713.859768	-2841.890737	1.280310e+02
6.300000	-2717.906536	-2845.995739	1.280892e+02
6.650000	-2720.828861	-2848.953784	1.281249e+02
7.000000	-2722.918178	-2851.064078	1.281459e+02

Таблица 1. 20 линейных КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000e+00
0.175000	-372.288366	-377.304143	5.015777e+00
0.350000	-701.306699	-710.545499	9.238800e+00
0.525000	-984.475089	-997.267712	1.279262e+01
0.700000	-1228.180068	-1243.961826	1.578176e+01
0.875000	-1437.918573	-1456.213156	1.829458e+01
1.050000	-1618.421864	-1638.827673	2.040581e+01
1.225000	-1773.762173	-1795.940745	2.217857e+01
1.400000	-1907.444503	-1931.110715	2.366621e+01
1.575000	-2022.485637	-2047.399409	2.491377e+01
1.750000	-2121.482145	-2147.441426	2.595928e+01
1.925000	-2206.668916	-2233.503742	2.683483e+01
2.100000	-2279.969536	-2307.537014	2.756748e+01
2.275000	-2343.039651	-2371.219714	2.818006e+01
2.450000	-2397.304288	-2425.996106	2.869182e+01
2.625000	-2443.989981	-2473.108932	2.911895e+01
2.800000	-2484.152416	-2513.627527	2.947511e+01
2.975000	-2518.700230	-2548.472015	2.977178e+01
3.150000	-2548.415496	-2578.434133	3.001864e+01
3.325000	-2573.971350	-2604.195141	3.022379e+01
3.500000	-2595.947163	-2626.341241	3.039408e+01
3.675000	-2614.841607	-2645.376839	3.053523e+01
3.850000	-2631.083893	-2661.735958	3.065207e+01
4.025000	-2645.043444	-2675.792061	3.074862e+01
4.200000	-2657.038228	-2687.866498	3.082827e+01
4.375000	-2667.341920	-2698.235781	3.089386e+01
4.550000	-2676.190077	-2707.137838	3.094776e+01
4.725000	-2683.785443	-2714.777399	3.099196e+01
4.900000	-2690.302522	-2721.330626	3.102810e+01
5.075000	-2695.891510	-2726.949101	3.105759e+01
5.250000	-2700.681683	-2731.763253	3.108157e+01
5.425000	-2704.784310	-2735.885315	3.110100e+01
5.600000	-2708.295160	-2739.411856	3.111670e+01
5.775000	-2711.296663	-2742.425972	3.112931e+01
5.950000	-2713.859768	-2744.999166	3.113940e+01
6.125000	-2716.045539	-2747.192960	3.114742e+01
6.300000	-2717.906536	-2749.060295	3.115376e+01
6.475000	-2719.487998	-2750.646723	3.115872e+01
6.650000	-2720.828861	-2751.991440	3.116258e+01
6.825000	-2721.962639	-2753.128173	3.116553e+01
7.000000	-2722.918178	-2754.085943	3.116776e+01

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).

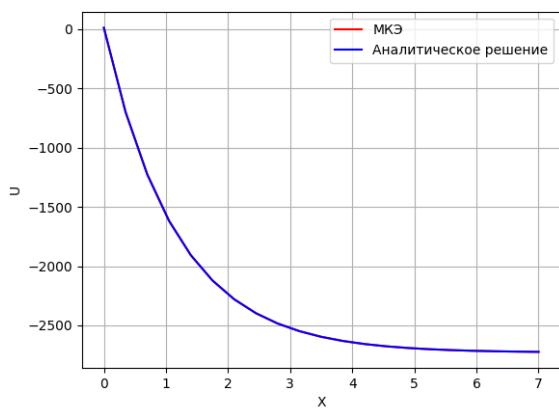


Рис. 4. Результат работы программы для 20
КЭ

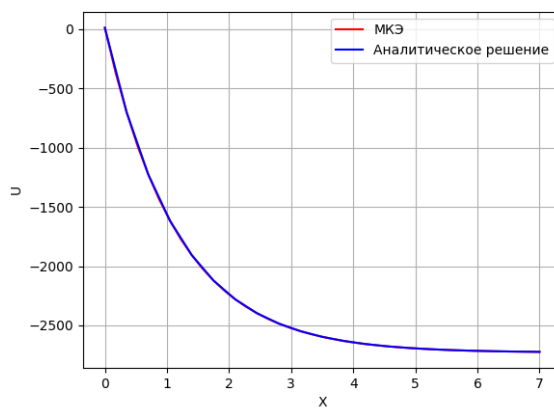


Рис. 5. Результат работы программы для 40
КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000e+00
0.350000	-701.306699	-701.306735	3.541402e-05
0.700000	-1228.180068	-1228.180128	6.050159e-05
1.050000	-1618.421864	-1618.421943	7.823664e-05
1.400000	-1907.444503	-1907.444594	9.074518e-05
1.750000	-2121.482145	-2121.482245	9.954509e-05
2.100000	-2279.969536	-2279.969642	1.057185e-04
2.450000	-2397.304288	-2397.304398	1.100358e-04
2.800000	-2484.152416	-2484.152529	1.130444e-04
3.150000	-2548.415496	-2548.415612	1.151327e-04
3.500000	-2595.947163	-2595.947279	1.165756e-04
3.850000	-2631.083893	-2631.084010	1.175674e-04
4.200000	-2657.038228	-2657.038346	1.182450e-04
4.550000	-2676.190077	-2676.190196	1.187046e-04
4.900000	-2690.302522	-2690.302641	1.190138e-04
5.250000	-2700.681683	-2700.681802	1.192196e-04
5.600000	-2708.295160	-2708.295279	1.193548e-04
5.950000	-2713.859768	-2713.859887	1.194423e-04
6.300000	-2717.906536	-2717.906656	1.194976e-04
6.650000	-2720.828861	-2720.828981	1.195316e-04
7.000000	-2722.918178	-2722.918297	1.195516e-04

Таблица 3. 20 кубических КЭ

X	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000e+00
0.175000	-372.288366	-372.288367	2.949657e-07
0.350000	-701.306699	-701.306700	5.432628e-07
0.525000	-984.475089	-984.475090	7.521662e-07
0.700000	-1228.180068	-1228.180069	9.278363e-07
0.875000	-1437.918573	-1437.918574	1.075475e-06
1.050000	-1618.421864	-1618.421866	1.199487e-06
1.225000	-1773.762173	-1773.762175	1.303588e-06
1.400000	-1907.444503	-1907.444505	1.390919e-06
1.575000	-2022.485637	-2022.485639	1.464132e-06
1.750000	-2121.482145	-2121.482147	1.525467e-06
1.925000	-2206.668916	-2206.668918	1.576811e-06
2.100000	-2279.969536	-2279.969538	1.619760e-06
2.275000	-2343.039651	-2343.039652	1.655655e-06
2.450000	-2397.304288	-2397.304290	1.685628e-06
2.625000	-2443.989981	-2443.989983	1.710634e-06
2.800000	-2484.152416	-2484.152417	1.731475e-06
2.975000	-2518.700230	-2518.700232	1.748825e-06
3.150000	-2548.415496	-2548.415498	1.763254e-06
3.325000	-2573.971350	-2573.971351	1.775239e-06
3.500000	-2595.947163	-2595.947165	1.785180e-06
3.675000	-2614.841607	-2614.841609	1.793415e-06
3.850000	-2631.083893	-2631.083894	1.800227e-06
4.025000	-2645.043444	-2645.043446	1.805853e-06
4.200000	-2657.038228	-2657.038229	1.810492e-06
4.375000	-2667.341920	-2667.341922	1.814307e-06
4.550000	-2676.190077	-2676.190079	1.817441e-06
4.725000	-2683.785443	-2683.785445	1.820008e-06
4.900000	-2690.302522	-2690.302523	1.822107e-06
5.075000	-2695.891510	-2695.891511	1.823818e-06
5.250000	-2700.681683	-2700.681685	1.825209e-06
5.425000	-2704.784310	-2704.784312	1.826336e-06
5.600000	-2708.295160	-2708.295162	1.827245e-06
5.775000	-2711.296663	-2711.296665	1.827976e-06
5.950000	-2713.859768	-2713.859770	1.828561e-06
6.125000	-2716.045539	-2716.045541	1.829027e-06
6.300000	-2717.906536	-2717.906538	1.829396e-06
6.475000	-2719.487998	-2719.488000	1.829687e-06
6.650000	-2720.828861	-2720.828863	1.829913e-06
6.825000	-2721.962639	-2721.962641	1.830088e-06
7.000000	-2722.918178	-2722.918180	1.830221e-06

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.

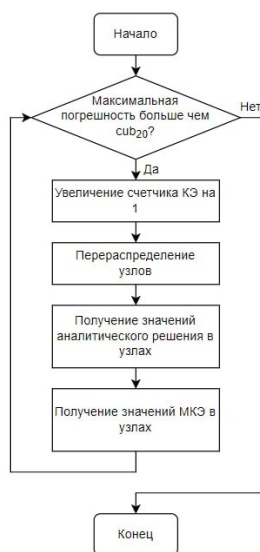


Рис. 6. Алгоритм нахождения количества КЭ, заданную точность ($cub_{20} = 1.195516e - 04$)

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 555 .

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 = 0.0001195516;
11 double EPS = 1e-16;
  
```

```

12 double X_BEGIN = 0.0;
13 double X_END = 7.0;
14 size_t ELEMS_NUM = ;
15 double L = (X_END - X_BEGIN) / ELEMS_NUM;
16
17 constexpr 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
18
19 std::vector<double> solve_with_gauss(std::vector<std::vector<double>>& A,
    std::vector<double>& b){
20     size_t row_size = A.size();
21     size_t col_size = A.back().size();
22     // Прямой ход Гаусса
23     double pivot = 0.0;
24     for (size_t i = 0; i < row_size; i++) {
25         for (size_t j = i + 1; j < col_size; j++) {
26             if (std::abs(A.at(j).at(i)) < EPS) {
27                 continue;
28             }
29             pivot = A.at(j).at(i) / A.at(i).at(i);
30             b.at(j) -= pivot * b.at(i);
31             for (size_t k = 0; k < row_size; k++) {
32                 A.at(j).at(k) -= pivot * A.at(i).at(k) ;
33             }
34         }
35     }
36     // Обратный ход Гаусса
37     std::vector<double> x(row_size);
38     for (int i = row_size - 1.0; i >= 0; i--) {
39         x.at(i) = b.at(i);
40         for (size_t j = i + 1; j < row_size; j++) {
41             x.at(i) -= x.at(j) * A.at(i).at(j);
42         }
43         x.at(i) /= A.at(i).at(i);
44     }
45     return x;
46 }
47
48 double analytical_solution(double x) {
49     double rez = 5. / 36. * (6. * x + 49. * exp(6. - (6. * x) / 7.) - 49. * exp(6.) + 72.);
50     return rez;
51 }
52
53 std::vector<double> build_analytical_solution(std::vector<double>& x_vec) {
54     size_t x_vec_size = x_vec.size();

```



```

55     std::vector<double> y_vec = std::vector<double>(x_vec_size);
56     for (size_t i = 0; i < x_vec_size; i++) {
57         y_vec.at(i) = analytical_solution(x_vec.at(i));
58     }
59     return y_vec;
60 }
61
62 std::vector<double> build_linear_solution(size_t elems_num) {
63     double L = (X_END - X_BEGIN) / elems_num;
64     size_t size = elems_num + 1;
65     std::vector< std::vector<double> > A(size, std::vector<double>(size));
66     std::vector<double> b(size);
67
68     // Локальная матрица жесткости для линейного КЭ
69     std::vector< std::vector<double> > local_matrix = {
70         { a/L - C * L/3.0 + B*1.0/2.0, -a/L - C * L/6.0 - B*1.0/2.0},
71         { -a/L - C * L/6.0 + B*1.0/2.0, a/L - C*L/3.0 - B*1.0/2.0},
72     };
73
74     // Ансамблирование и получение глобальной матрицы жесткости для линейного КЭ
75     for (size_t i = 0; i < elems_num; i++) {
76         for (size_t j = 0; j < 2; j++) {
77             for (size_t k = 0; k < 2; k++) {
78                 A.at(i + j).at(i + k) += local_matrix.at(j).at(k);
79             }
80         }
81     }
82
83     for (size_t i = 0; i < size ; i++) {
84         b.at(i) = D * L;
85     }
86
87     // Учет ГУ
88     b.at(0) = usl_left;
89     A.at(0).at(0) = 1;
90     A.at(0).at(1) = 0;
91
92     b.at(size - 1) = D * L / 2. + a*usl_right;
93
94     // Решение полученной СЛАУ методом Гаусса
95     std::vector<double> res = solve_with_gauss(A, b);
96     return res;
97 }
98
99 std::vector<double> build_cube_solution(size_t elems_num) {
100     double L = (X_END - X_BEGIN) / elems_num;

```

```

101 size_t size = elems_num + 1;
102 std::vector< std::vector<double> > A(size,std::vector<double>(size));
103 std::vector<double> b(size);
104
105 // Локальная матрица жесткости для кубического КЭ
106 std::vector< std::vector<double> > local_matrix = {
107
108     { 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},
109     { -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},
110     { 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},
111     { -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}
112 };
113
114
115 // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего
    элементов учитываются далее)
116 std::vector<double> local_b = { D * L / 8.0,
117                                D*3.0 * L / 8.0,
118                                D*3.0 * L / 8.0,
119                                D * L / 8.0 };
120
121
122 // Производим матричные преобразования для обнуления элементов локальной
    матрицы жесткости, относящихся к внутренним узлам
123 for (size_t i = 1; i < 3; i++) {
124     for (size_t j = 0; j < 4; j++) {
125         if (std::fabs(local_matrix.at(j).at(i)) > EPS && i != j) {
126             double val = local_matrix.at(j).at(i) / local_matrix.at(i).at(i);
127             local_b.at(j) -= val * local_b.at(i);
128             for (size_t k = 0; k < 4; k++) {
129                 local_matrix.at(j).at(k) -= val * local_matrix.at(i).at(k);
130             }
131         }
132         continue;
133     }
134 }
135
136

```

```

137 // Исключаем внутренние узлы из рассмотрения
138 std::vector< std::vector<double> > local_matrix_mod = { { local_matrix.at(0).at(0),
139                                                         local_matrix.at(0).at(3) },
                                                         { local_matrix.at(3).at(0),
                                                         local_matrix.at(3).at(3)
                                                         } };
140 std::vector<double> local_b_mod = { local_b.at(0),
141                                   local_b.at(3)};
142
143 // Ансамблирование и получение глобальной матрицы жесткости для кубического КЭ
144 for (size_t i = 0; i < elems_num; i++) {
145     for (size_t j = 0; j < 2; j++) {
146         for (size_t k = 0; k < 2; k++) {
147             A.at(i + j).at(i + k) += local_matrix_mod.at(j).at(k);
148         }
149     }
150 }
151 for (size_t i = 0; i < elems_num; i++) {
152     b.at(i) += local_b_mod.at(0);
153     b.at(i+1) += local_b_mod.at(1);
154 }
155 // Учет ГУ
156 b.at(0) = usl_left;
157 A.at(0).at(0) = 1;
158 A.at(0).at(1) = 0;
159
160 b.at(size - 1) = local_b_mod.at(1) + a * usl_right;
161
162 // Решение полученной СЛАУ методом Гаусса
163 std::vector<double> res = solve_with_gauss(A, b);
164 return res;
165 }
166
167 double calc_abs_error(const std::vector<double>& y_real, const std::vector<double>&
168 y) {
169     double max_err = 0.0;
170     for (size_t i = 0; i < y_real.size(); i++) {
171         double err = std::fabs(y_real.at(i) - y.at(i));
172         if (err > max_err) {
173             max_err = err;
174         }
175     }
176     return max_err;
177 }
178 int main() {

```

```

179 #ifdef OSN
180
181     std::vector<double> x(ELEMS_NUM + 1);
182     for (size_t i = 0; i < x.size(); i++) {
183         x.at(i) = X_BEGIN + i * L;
184     }
185     size_t x_size = x.size();
186
187
188 #ifdef CUBE
189     std::vector<double> y = build_cube_solution(ELEMS_NUM);
190 #else
191     std::vector<double> y = build_linear_solution(ELEMS_NUM);
192 #endif
193     std::vector<double> y_real = build_analytical_solution(x);
194
195 #ifdef TABLE
196     for (size_t i = 0; i < x.size(); i++) {
197         std::cout<<x.at(i)<<" &"<<y_real.at(i)<<" &"<<y.at(i)<<"
198             &"<<std::fabs(y_real.at(i) - y.at(i))<<"\\\\"<<std::endl;
199     }
200 #endif
201
202     FILE* gp = popen("gnuplot --persist", "w");
203     fprintf(gp, "$predict << EOD\n");
204     for (size_t i = 0; i < x_size; i++) {
205         fprintf(gp, "%lf %lf\n", x.at(i), y.at(i));
206     }
207     fprintf(gp, "EOD\n");
208     fprintf(gp, "$real << EOD\n");
209     for (size_t i = 0; i < x_size; i++) {
210         fprintf(gp, "%lf %lf\n", x.at(i), y_real.at(i));
211     }
212     fprintf(gp, "EOD\n");
213     fprintf(gp, "set grid\n");
214     fprintf(gp, "plot '$predict' using 1:2 with lp lc '#ba55d3' lw 1.5 pt 7 ps 0.5 title
215         МКЭрешение'— (%zu КЭ)', '$real' using 1:2 with lines lc rgb '#afdafc' lt 1 lw 2
216         title аналитическое' решение(%zu КЭ)',\n", ELEMS_NUM, ELEMS_NUM);
217     printf("Абсолютная погрешность: %e\n", calc_abs_error(y_real, y));
218
219 //нахождение количества линейных КЭ
220 #else
221     int N=20,n=10000;
222     double err=10;
223     FILE* gp = popen("gnuplot --persist", "w");

```

```

222 fprintf(gp, "$predict << EOD\n");
223 while (err>CUB && n<=19000){
224
225     double L = (X_END - X_BEGIN) / N;
226     std::vector<double> x(N + 1);
227     for (size_t i = 0; i < x.size(); i++) {
228         x.at(i) = X_BEGIN + i * L;
229     }
230
231     std::vector<double> y_r(N + 1);
232     std::vector<double> y_s(N + 1);
233
234     y_s = build_linear_solution(N);
235     y_r = build_analytical_solution(x);
236
237     err=calc_abs_error(y_r, y_s);
238
239     fprintf(gp, "%d %e\n", N, err);
240     printf("Абсолютная погрешность: %e количествоКЭ: %d\n", calc_abs_error(y_r,
241         y_s), N);
242     N+=1;
243     n+=1;
244 }
245 fprintf(gp, "EOD\n");
246 fprintf(gp, "set grid\n");
247 fprintf(gp, "set logscale y 2\n");
248 fprintf(gp, "plot '$predict' using 1:2 with lp lc '#cd853f' lw 1.5 pt 7 ps 0.5 title
249     Абсолютная' погрешность',\n" );
250 std::cout<<"Количество линейныхКЭ"<<N-1<<std::endl;
251 printf("Абсолютная погрешность: %e\n", err);
252 #endif
253 return 0;
254 }

```

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