



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

<b>Метод конечных элементов</b>	<b>3</b>
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 . . . . .	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 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 results are displayed in a structured format:

- 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:

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

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).

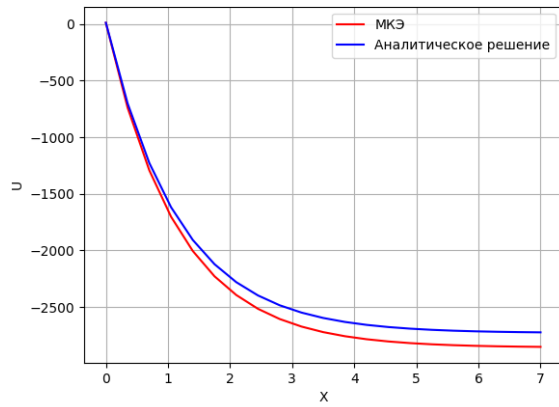


Рис. 2. Результат работы программы для 20 КЭ

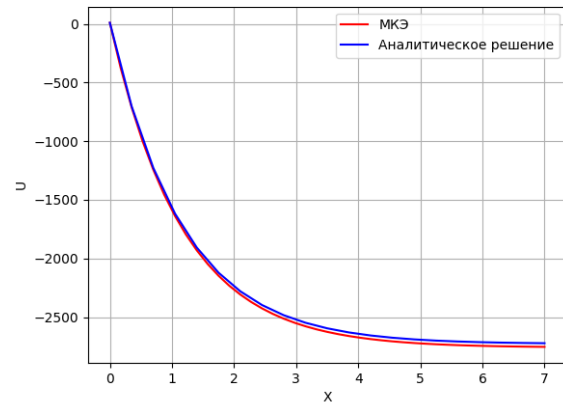


Рис. 3. Результат работы программы для 40 КЭ

Х	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).

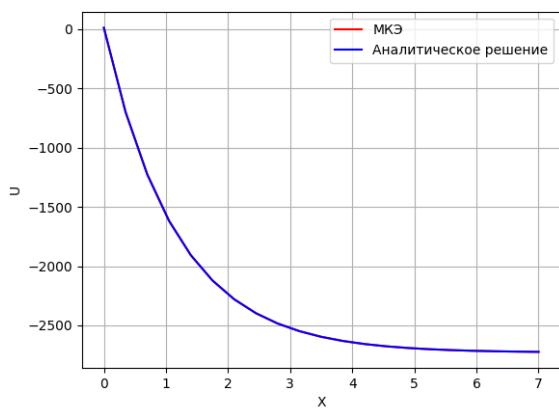


Рис. 4. Результат работы программы для 20  
КЭ

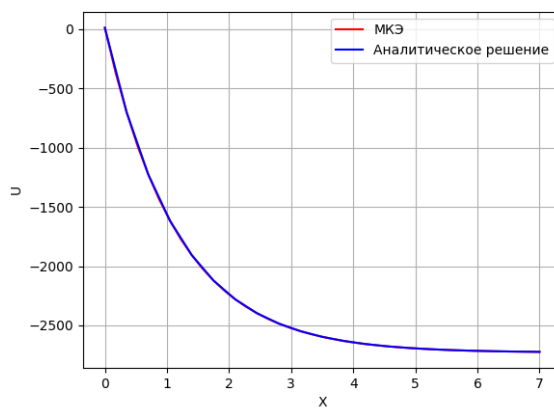


Рис. 5. Результат работы программы для 40  
КЭ

Х	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.

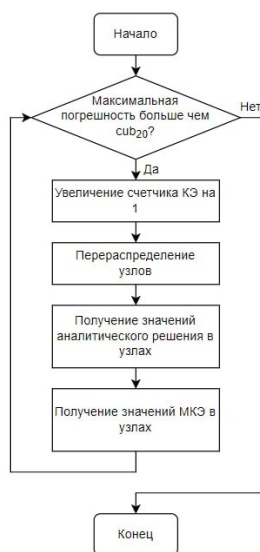


Рис. 6. Алгоритм нахождения количества КЭ, заданную точность ( $cub_{20} = 1.195516e - 04$ )

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 double EPS = 1e-16;
7 double X_BEGIN = 0.0;
8 double X_END = 7.0;
9 size_t ELEMS_NUM = 20;
10 double L = (X_END - X_BEGIN) / ELEMS_NUM;
11

```

```

12 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
13
14 std::vector<double> solve_with_gauss(std::vector<std::vector<double>>& A,
    std::vector<double>& b){
15     size_t row_size = A.size();
16     size_t col_size = A.back().size();
17
18     // Прямой ход Гаусса
19     double pivot = 0.;
20     for (size_t i = 0; i < row_size; i++) {
21         for (size_t j = i + 1; j < col_size; j++) {
22             if (std::abs(A.at(j).at(i)) < EPS) {
23                 continue;
24             }
25             pivot = A.at(j).at(i) / A.at(i).at(i);
26             b.at(j) -= pivot * b.at(i);
27             for (size_t k = 0; k < row_size; k++) {
28                 A.at(j).at(k) -= pivot * A.at(i).at(k) ;
29             }
30         }
31     }
32
33     // Обратный ход Гаусса
34     std::vector<double> x(row_size);
35     for (int i = row_size - 1; i >= 0; i--) {
36         x.at(i) = b.at(i);
37         for (size_t j = i + 1; j < row_size; j++) {
38             x.at(i) -= x.at(j) * A.at(i).at(j);
39         }
40         x.at(i) /= A.at(i).at(i);
41     }
42
43     return x;
44 }
45
46 double analytical_solution(double x) {
47     return 5. / 36. * (6. * x + 49. * exp(6. - (6. * x) / 7.) - 49. * exp(6.) + 72.);
48 }
49
50 std::vector<double> build_analytical_solution(std::vector<double>& x_vec) {
51     size_t x_vec_size = x_vec.size();
52     std::vector<double> y_vec = std::vector<double>(x_vec_size);
53     for (size_t i = 0; i < x_vec_size; i++) {
54         y_vec.at(i) = analytical_solution(x_vec.at(i));
55     }

```



```

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

```

```

102     std::vector<double> res = solve_with_gauss(A, b);
103     return res;
104 }
105
106 std::vector<double> build_cube_solution(size_t elems_num) {
107     double L = (X_END - X_BEGIN) / elems_num;
108     size_t size = elems_num + 1;
109     std::vector< std::vector<double> > A(size, std::vector<double>(size));
110     std::vector<double> b(size);
111
112     // Локальная матрица жесткости для кубического КЭ
113     std::vector< std::vector<double> > local_matrix = {
114         { 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},
115         { -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},
116         { 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},
117         { -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}
118     };
119
120
121
122     // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего
       элементов учитываются далее)
123     std::vector<double> local_b = { D * L / 8.0,
124                                     D * 3.0 * L / 8.0,
125                                     D * 3.0 * L / 8.0,
126                                     D * L / 8.0 };
127
128
129     // Производим матричные преобразования для обнуления элементов локальной
       матрицы жесткости, относящихся к внутренним узлам
130     for (size_t i = 1; i < 3; i++) {
131         for (size_t j = 0; j < 4; j++) {
132             if (std::fabs(local_matrix.at(j).at(i)) > EPS && i != j) {
133                 double val = local_matrix.at(j).at(i) / local_matrix.at(i).at(i);
134                 local_b.at(j) -= val * local_b.at(i);
135                 for (size_t k = 0; k < 4; k++) {
136                     local_matrix.at(j).at(k) -= val * local_matrix.at(i).at(k);
137                 }

```

```

138         }
139         continue;
140     }
141 }
142
143
144 // Исключаем внутренние узлы из рассмотрения
145 std::vector< std::vector<double> > local_matrix_mod = { { local_matrix.at(0).at(0),
146                                                         local_matrix.at(0).at(3) },
                                                         { local_matrix.at(3).at(0),
                                                         local_matrix.at(3).at(3)
                                                         } };
147
148 std::vector<double> local_b_mod = { local_b.at(0),
149                                   local_b.at(3)
150                                   };
151
152 // Ансамблирование и получение глобальной матрицы жесткости для кубического КЭ
153 for (size_t i = 0; i < elems_num; i++) {
154     for (size_t j = 0; j < 2; j++) {
155         for (size_t k = 0; k < 2; k++) {
156             A.at(i + j).at(i + k) += local_matrix_mod.at(j).at(k);
157         }
158     }
159 }
160
161 for (size_t i = 0; i < elems_num; i++) {
162     b.at(i) += local_b_mod.at(0);
163     b.at(i+1) += local_b_mod.at(1);
164 }
165
166 // Учет ГУ
167 if (0 == 1 ) {
168     b.at(0) = local_b_mod.at(0) - a * usl_left;
169 } else {
170     b.at(0) = usl_left;
171     A.at(0).at(0) = 1.;
172     A.at(0).at(1) = 0.;
173 }
174
175 if (1 == 1 ) {
176     b.at(size - 1) = local_b_mod.at(1) + a * usl_right;
177 } else {
178     b.at(size - 1) = usl_right;
179     A.at(size - 1).at(size - 1) = 1.;
180     A.at(size - 1).at(size - 2) = 0.;
181 }

```

```

181
182 // Решение полученной СЛАУ методом Гаусса
183 std::vector<double> res = solve_with_gauss(A, b);
184 return res;
185 }
186
187 double calc_abs_error(const std::vector<double>& y_real, const std::vector<double>&
    y) {
188     double max_err = 0.0;
189     for (size_t i = 0; i < y_real.size(); i++) {
190         double err = std::fabs(y_real.at(i) - y.at(i));
191         if (err > max_err) {
192             max_err = err;
193         }
194     }
195     return max_err;
196 }
197
198 int main() {
199
200     std::vector<double> x(ELEMS_NUM + 1);
201     for (size_t i = 0; i < x.size(); i++) {
202         x.at(i) = X_BEGIN + i * L;
203     }
204     size_t x_size = x.size();
205
206     std::vector<double> y;
207     if (true) {
208         y = build_linear_solution(ELEMS_NUM);
209     } else {
210         y = build_cube_solution(ELEMS_NUM);
211     }
212     std::vector<double> y_real = build_analytical_solution(x);
213
214
215     FILE* gp;
216     FILE* ab;
217     FILE* pgr;
218     FILE* tab;
219     if (true) {
220         if(ELEMS_NUM == 20) {
221             gp = fopen("res/labs/text/graph/lin_20.txt", "w");
222             ab = fopen("res/labs/text/graph/abs.txt", "w");
223             for (size_t i = 0; i < x_size; i++) {
224                 fprintf(ab, "%lf %lf\n", x.at(i), y_real.at(i));
225             }

```

```

226         pgr = fopen("res/labs/text/pgr/lin_20.txt", "w");
227         tab = fopen("res/labs/text/tab/lin_20.txt", "w");
228     }
229     if(ELEMS_NUM == 40) {
230         gp = fopen("res/labs/text/graph/lin_40.txt", "w");
231         pgr = fopen("res/labs/text/pgr/lin_40.txt", "w");
232         tab = fopen("res/labs/text/tab/lin_40.txt", "w");
233     }
234 } else {
235     if(ELEMS_NUM == 20) {
236         gp = fopen("res/labs/text/graph/cub_20.txt", "w");
237         pgr = fopen("res/labs/text/pgr/cub_20.txt", "w");
238         tab = fopen("res/labs/text/tab/cub_20.txt", "w");
239     }
240     if(ELEMS_NUM == 40) {
241         gp = fopen("res/labs/text/graph/cub_40.txt", "w");
242         pgr = fopen("res/labs/text/pgr/cub_40.txt", "w");
243         tab = fopen("res/labs/text/tab/cub_40.txt", "w");
244     }
245 }
246
247 for (size_t i = 0; i < x.size()-1; i++) {
248     fprintf(tab, "%lf & %lf & %lf & %lf \\\n", x.at(i), y_real.at(i), y.at(i),
249         std::fabs(y_real.at(i) - y.at(i)));
250 }
251 fprintf(tab, "%lf & %lf & %lf & %lf", x.at(x.size()-1), y_real.at(x.size()-1),
252     y.at(x.size()-1), std::fabs(y_real.at(x.size()-1) - y.at(x.size()-1)));
253
254 for (size_t i = 0; i < x_size; i++) {
255     fprintf(gp, "%lf %lf\n", x.at(i), y.at(i));
256 }
257
258 fprintf(pgr, "%e", calc_abs_error(y_real, y));
259 fclose(gp);
260 fclose(ab);
261 fclose(pgr);
262 fclose(tab);
263
264 return 0;
265 }

```

---

## 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*