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: | Кулагин Арсений Олегович |
| Группа: | РК6-73б |
| Тип задания: | Лабораторная работа |
| Название: | Метод конечных элементов |
| Вариант: | 54 |

Студент ____________ (подпись, дата) Кулагин А.О. (Фамилия, И.О.)

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

Оценка: ____________

Москва, 2022

# Содержание

# Метод конечных элементов

## 1 Цель выполнения лабораторной работы

**Цель выполнения лабораторной работы** – решение дифференциального уравнения методом конечных элементов (МКЭ), используя линейную и кубическую функции формы, и анализ точности относительной аналитического способа решения

## 2 Задание

Решить с помощью МКЭ уравнение 1

$$51\frac{d^2u}{dx^2} - 8u + 20 = 0, \tag{1}$$

при следующих граничных условиях (г. у.):

$$u(x = 5) = 0, \tag{2}$$

$$u'(x = 50) = 1. \tag{3}$$

Количество конечных элементов

- для первого расчета – 20,
- для второго – 40.

Также необходимо:

1. Сравнить результаты с аналитическим решением. Оценить максимальную погрешность.
2. Определить количество линейных КЭ, обеспечивающих такую же точность как и кубические.

# 3 Аналитическое решение

На рисунке 1 представлено аналитическое решение поставленной задачи.

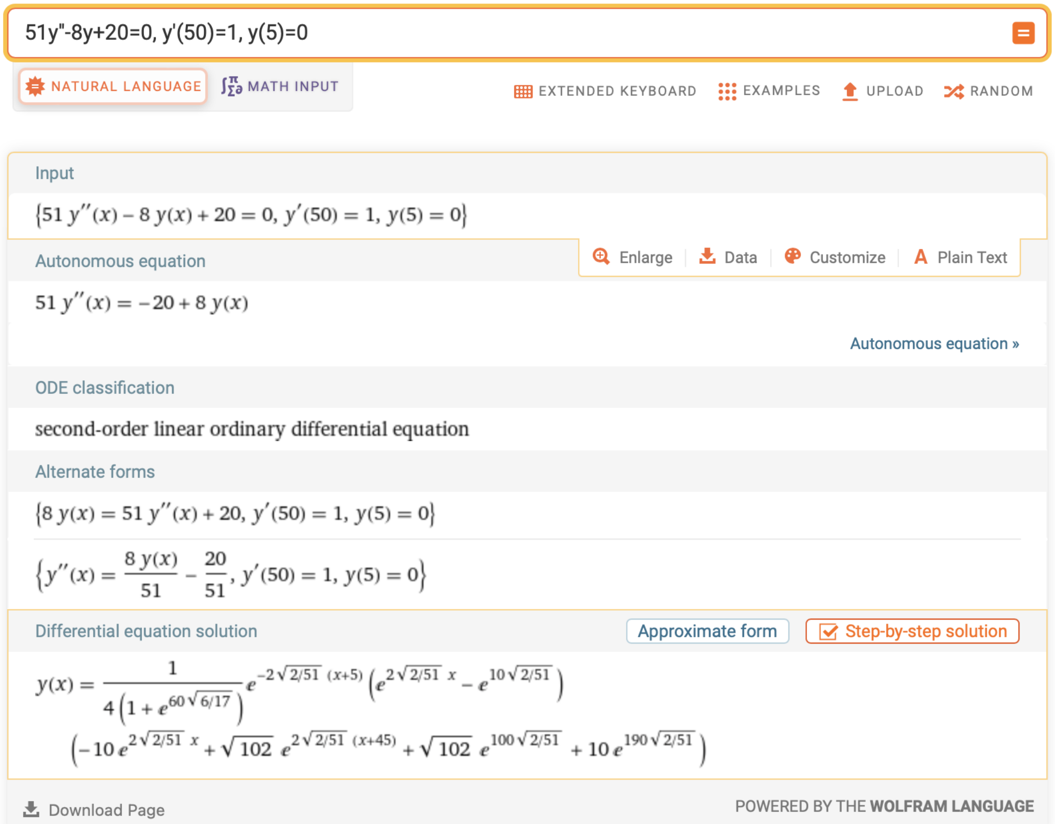


Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = \frac{e^{-2\sqrt{2/51}(x+5)}\left(e^{2\sqrt{2/51}x} - e^{10\sqrt{2/51}}\right)\left(-10e^{2\sqrt{2/51}x} + \sqrt{102}e^{2\sqrt{2/51}(x+45)} + \sqrt{102}e^{100\sqrt{2/51}} + 10e^{190\sqrt{2/51}}\right)}{4\left(1 + e^{60\sqrt{6/17}}\right)}.$$

# 4 Получение локальных матрицы жесткости и вектора нагрузок

Составим локальные матрицу жесткости и вектор нагрузок для уравнения 1.

### Линейная функция-формы КЭ

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

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:

$$\int_0^L \mathbf{W_e}\left(51\frac{d^2\mathbf{u}}{dx^2} - 8u + 20\right)dx = 0, \tag{4}$$

где $\mathbf{W_e} = \mathbf{N_e}^T$.

$$\int_0^L \mathbf{W_e}\left(51\frac{d^2\mathbf{u}}{dx^2} - 8u + 20\right)dx = 51\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx - 8\int_0^L \mathbf{W_e}\mathbf{u}dx + 20\int_0^L \mathbf{W_e}\mathbf{u} = 0$$

Распишем каждое слагаемое отдельно:

$$51\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 51\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx = 51\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d\mathbf{u}}{dx}|_0^L -$$

$$-51\int_0^L \frac{d}{dx}\begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\frac{d}{dx}\left[(1-\frac{x}{L});\quad \frac{x}{L}\right]\begin{bmatrix}u_i\\ u_j\end{bmatrix} = \begin{bmatrix}-51\frac{d\mathbf{u}}{dx}|_i\\ 51\frac{d\mathbf{u}}{dx}|_j\end{bmatrix} - 51\begin{bmatrix}\frac{1}{L}, & -\frac{1}{L}\\ -\frac{1}{L}, & \frac{1}{L}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix}$$

$$-8\int_0^L \mathbf{W_e}\mathbf{u}dx = -8\int_0^L \begin{bmatrix}(1-\frac{x}{L})\\ \frac{x}{L}\end{bmatrix}\mathbf{u}\begin{bmatrix}u_i\\ u_j\end{bmatrix}dx = -8\begin{bmatrix}\frac{L}{3}, & \frac{L}{6}\\ \frac{L}{6}, & \frac{L}{3}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix}$$

$$20\int_0^L \mathbf{W_e}dx = 20\begin{bmatrix}\frac{L}{2}\\ \frac{L}{2}\end{bmatrix}$$

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

$$\begin{bmatrix}51\frac{1}{L} - 8\frac{L}{3}, & -51\frac{1}{L} - 8\frac{L}{6}\\ -51\frac{1}{L} - 8\frac{L}{6}, & 51\frac{1}{L} - 8\frac{L}{3}\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} = \begin{bmatrix}-51\frac{du}{dx}|_i + 20\frac{L}{2}\\ 51\frac{du}{dx}|_j + 20\frac{L}{2}\end{bmatrix}$$

### 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.

$$51\int_0^L \mathbf{W_e}\frac{d^2\mathbf{u}}{dx^2}dx = 51\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d^2\mathbf{u}}{dx^2}dx =$$

$$= 51\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d\mathbf{u}}{dx}\Big|_0^L - 51\int_0^L \frac{d}{dx}\begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\frac{d}{dx}\mathbf{u} =$$

$$= \begin{bmatrix} -51\frac{d\mathbf{u}}{dx}\big|_i \\ 0 \\ 0 \\ 51\frac{d\mathbf{u}}{dx}\big|_l \end{bmatrix} - 51\begin{bmatrix} \frac{37}{10L} & -\frac{189}{40} & \frac{27}{20L} & -\frac{13}{40L} \\ -\frac{189}{40} & \frac{54}{5L} & -\frac{297}{40L} & \frac{27}{20L} \\ \frac{27}{20L} & -\frac{297}{40L} & \frac{54}{5L} & -\frac{189}{40} \\ -\frac{13}{40L} & \frac{27}{20L} & -\frac{189}{40} & \frac{37}{10L} \end{bmatrix}\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix}$$

$$-8\int_0^L \mathbf{W_e}\mathbf{u}dx = -8\int_0^L \begin{bmatrix} -\frac{9x^3}{2L^3} & + & \frac{18x^2}{2L^2} & - & \frac{11x}{2L} & + & 1 \\ \frac{27x^3}{2L^3} & - & \frac{45x^2}{2L^2} & + & \frac{9x}{L} & & \\ -\frac{27x^3}{2L^3} & + & \frac{36x^2}{2L^2} & - & \frac{9x}{2L} & & \\ \frac{9x^3}{2L^3} & - & \frac{9x^2}{2L^2} & - & \frac{x}{L} & & \end{bmatrix}\mathbf{u}dx =$$

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

$$20\int_0^L \mathbf{W_e}dx = 20\begin{bmatrix} \frac{L}{8} \\ \frac{3L}{8} \\ \frac{3L}{8} \\ \frac{L}{8} \end{bmatrix}$$

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

$$\begin{bmatrix} 51\frac{37}{10L} - 8\frac{8L}{105} & -51\frac{189}{40L} - 8\frac{33L}{560} & 51\frac{27}{20L} + 8\frac{3L}{140} & -51\frac{13}{40L} - 8\frac{119L}{1680} \\ -51\frac{189}{40L} - 8\frac{33L}{560} & 51\frac{54}{5L} + 0 - 8\frac{27L}{170} & -51\frac{297}{40L} + 8\frac{27L}{560} & 51\frac{27}{20L} + 8\frac{27L}{560} \\ 51\frac{27}{20L} + 8\frac{27L}{560} & -51\frac{297}{40L} + 8\frac{27L}{560} & 51\frac{54}{5L} + 0 - 8\frac{27L}{170} & -51\frac{189}{40L} - 8\frac{33L}{560} \\ -51\frac{13}{40L} - 8\frac{19L}{1680} & 51\frac{27}{20L}\frac{3}{10} - 8\frac{3L}{140} & -51\frac{189}{40L} - 8\frac{33L}{560} & 51\frac{37}{10L} - 8\frac{8L}{105} \end{bmatrix}.$$

$$\cdot \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} 20\frac{L}{8} - 51\frac{du}{dx}|_i \\ 20\frac{3L}{8} \\ 20\frac{3L}{8} \\ 20\frac{L}{8} + 51\frac{du}{dx}|_l \end{bmatrix} \tag{5}$$

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

$$\begin{bmatrix} a_{11} & 0 & 0 & a_{14} \\ a_{21} & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 51\frac{du}{dx}|_i \\ b_2 \\ b_3 \\ b_4 + 51\frac{du}{dx}|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 51\frac{du}{dx}|_i \\ b_4 + 51\frac{du}{dx}|_l \end{bmatrix}$$

# 5 Получение глобальных матрицы жесткости и вектора нагрузок

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

### Ансамблирование

Пусть локальные матрица жесткости и вектор неизвестных заданы следующим образом

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} b_1 - 51\frac{du}{dx}|_i \\ b_2 + 51\frac{du}{dx}|_l \end{bmatrix},$$

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

$$
\begin{bmatrix}
a_{11}^1 & a_{12}^1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\
0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\
0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\
\vdots & \vdots & \vdots & \vdots & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\
0 & 0 & 0 & 0 & 0 & 0 & a_{21}^n & a_{22}^n
\end{bmatrix}
\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}
=
\begin{bmatrix}
b_1^1 - 51\frac{du}{dx}|_0 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ b_2^n + 51\frac{du}{dx}|_L
\end{bmatrix}
$$

### Учет граничных условий

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

$$
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_{21}^1 & a_{22}^1 + a_{11}^2 & a_{12}^2 & 0 & \cdots & 0 & 0 & 0 \\
0 & a_{21}^2 & a_{22}^2 + a_{11}^3 & a_{12}^3 & 0 & \cdots & 0 & 0 \\
0 & 0 & a_{21}^3 & a_{22}^3 + \cdots & & & & 0 \\
\vdots & \vdots & \vdots & \vdots & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots + a_{11}^n & a_{12}^n \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}
=
\begin{bmatrix}
0 \\ b_2^1 + b_1^2 \\ b_2^2 + b_1^3 \\ b_2^3 + b_1^4 \\ \vdots \\ b_2^{n-1} + b_1^n \\ b_2^n + 51 \cdot 1
\end{bmatrix}
$$

## 6 Анализ результатов

Проведем сравнение результатов согласно заданию.

### Линейная функция-формы

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

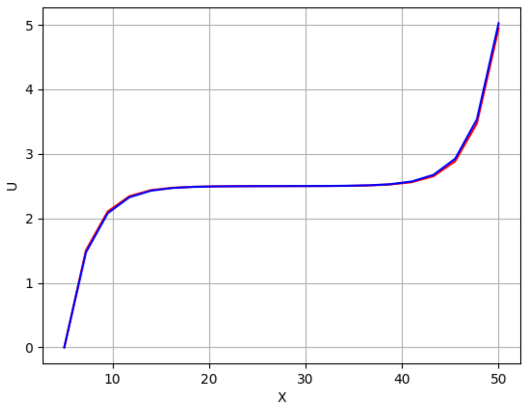


Рис. 2. Результат работы программы для 20 КЭ

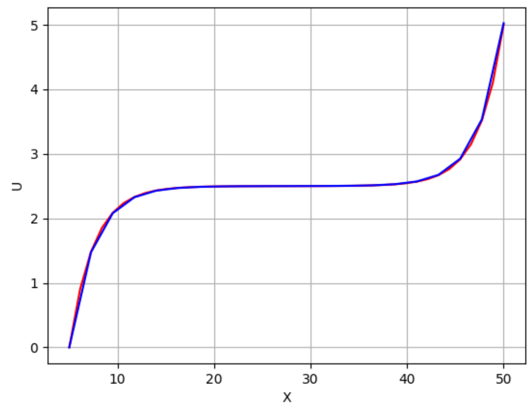


Рис. 3. Результат работы программы для 40 КЭ

| X | Аналитическое решение | МКЭ-решение | Абсолютная погрешность |
|---|---|---|---|
| 5.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 7.250000e+00 | 1.474523e+00 | 1.507202e+00 | 3.267876e-02 |
| 9.500000e+00 | 2.079359e+00 | 2.105741e+00 | 2.638181e-02 |
| 1.175000e+01 | 2.327457e+00 | 2.343432e+00 | 1.597490e-02 |
| 1.400000e+01 | 2.429226e+00 | 2.437825e+00 | 8.598776e-03 |
| 1.625000e+01 | 2.470972e+00 | 2.475311e+00 | 4.338560e-03 |
| 1.850000e+01 | 2.488101e+00 | 2.490201e+00 | 2.099340e-03 |
| 2.075000e+01 | 2.495139e+00 | 2.496121e+00 | 9.822609e-04 |
| 2.300000e+01 | 2.498054e+00 | 2.498491e+00 | 4.376530e-04 |
| 2.525000e+01 | 2.499318e+00 | 2.499481e+00 | 1.628628e-04 |
| 2.750000e+01 | 2.500003e+00 | 2.499995e+00 | 8.694016e-06 |
| 2.975000e+01 | 2.500692e+00 | 2.500504e+00 | 1.879828e-04 |
| 3.200000e+01 | 2.501967e+00 | 2.501474e+00 | 4.928542e-04 |
| 3.425000e+01 | 2.504910e+00 | 2.503793e+00 | 1.116689e-03 |
| 3.650000e+01 | 2.512017e+00 | 2.509585e+00 | 2.432750e-03 |
| 3.875000e+01 | 2.529316e+00 | 2.524148e+00 | 5.168143e-03 |
| 4.100000e+01 | 2.571478e+00 | 2.560814e+00 | 1.066440e-02 |
| 4.325000e+01 | 2.674259e+00 | 2.653140e+00 | 2.111984e-02 |
| 4.550000e+01 | 2.924827e+00 | 2.885627e+00 | 3.919965e-02 |
| 4.775000e+01 | 3.535681e+00 | 3.471061e+00 | 6.461993e-02 |
| 5.000000e+01 | 5.024876e+00 | 4.945263e+00 | 7.961335e-02 |

Таблица 1. 20 линейных КЭ

| X | Аналитическое решение | МКЭ-решение | Абсолютная погрешность |
|---|---|---|---|
| 5.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 6.125000e+00 | 8.988465e-01 | 9.048714e-01 | 6.024942e-03 |
| 7.250000e+00 | 1.474523e+00 | 1.482226e+00 | 7.702960e-03 |
| 8.375000e+00 | 1.843222e+00 | 1.850608e+00 | 7.386249e-03 |
| 9.500000e+00 | 2.079359e+00 | 2.085655e+00 | 6.295608e-03 |
| 1.062500e+01 | 2.230596e+00 | 2.235627e+00 | 5.030626e-03 |
| 1.175000e+01 | 2.327457e+00 | 2.331316e+00 | 3.859008e-03 |
| 1.287500e+01 | 2.389494e+00 | 2.392372e+00 | 2.877996e-03 |
| 1.400000e+01 | 2.429226e+00 | 2.431328e+00 | 2.102516e-03 |
| 1.512500e+01 | 2.454673e+00 | 2.456185e+00 | 1.511909e-03 |
| 1.625000e+01 | 2.470972e+00 | 2.472046e+00 | 1.073656e-03 |
| 1.737500e+01 | 2.481413e+00 | 2.482167e+00 | 7.546166e-04 |
| 1.850000e+01 | 2.488101e+00 | 2.488627e+00 | 5.256946e-04 |
| 1.962500e+01 | 2.492388e+00 | 2.492751e+00 | 3.632130e-04 |
| 2.075000e+01 | 2.495139e+00 | 2.495388e+00 | 2.487614e-04 |
| 2.187500e+01 | 2.496908e+00 | 2.497077e+00 | 1.684327e-04 |
| 2.300000e+01 | 2.498054e+00 | 2.498166e+00 | 1.119046e-04 |
| 2.412500e+01 | 2.498806e+00 | 2.498878e+00 | 7.156510e-05 |
| 2.525000e+01 | 2.499318e+00 | 2.499360e+00 | 4.175615e-05 |
| 2.637500e+01 | 2.499692e+00 | 2.499710e+00 | 1.814146e-05 |
| 2.750000e+01 | 2.500003e+00 | 2.500001e+00 | 2.823632e-06 |
| 2.862500e+01 | 2.500316e+00 | 2.500291e+00 | 2.435678e-05 |
| 2.975000e+01 | 2.500692e+00 | 2.500642e+00 | 4.978991e-05 |
| 3.087500e+01 | 2.501207e+00 | 2.501124e+00 | 8.303351e-05 |
| 3.200000e+01 | 2.501967e+00 | 2.501838e+00 | 1.291148e-04 |
| 3.312500e+01 | 2.503123e+00 | 2.502928e+00 | 1.948468e-04 |
| 3.425000e+01 | 2.504910e+00 | 2.504620e+00 | 2.896930e-04 |
| 3.537500e+01 | 2.507688e+00 | 2.507261e+00 | 4.268961e-04 |
| 3.650000e+01 | 2.512017e+00 | 2.511392e+00 | 6.249395e-04 |
| 3.762500e+01 | 2.518772e+00 | 2.517863e+00 | 9.093871e-04 |
| 3.875000e+01 | 2.529316e+00 | 2.528001e+00 | 1.315084e-03 |
| 3.987500e+01 | 2.545778e+00 | 2.543889e+00 | 1.888559e-03 |
| 4.100000e+01 | 2.571478e+00 | 2.568788e+00 | 2.690149e-03 |
| 4.212500e+01 | 2.611606e+00 | 2.607811e+00 | 3.794811e-03 |
| 4.325000e+01 | 2.674259e+00 | 2.668970e+00 | 5.289414e-03 |
| 4.437500e+01 | 2.772085e+00 | 2.764822e+00 | 7.262330e-03 |
| 4.550000e+01 | 2.924827e+00 | 2.915049e+00 | 9.777481e-03 |
| 4.662500e+01 | 3.163314e+00 | 3.150495e+00 | 1.281860e-02 |
| 4.775000e+01 | 3.535681e+00 | 3.519503e+00 | 1.617839e-02 |
| 4.887500e+01 | 4.117086e+00 | 4.097838e+00 | 1.924807e-02 |
| 5.000000e+01 | 5.024876e+00 | 5.004246e+00 | 2.063028e-02 |

Таблица 2. 40 линейных КЭ

Максимальная абсолютная погрешность 7.961335e-02 и 2.063028e-02 соответственно.

### Кубическая функция-формы

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

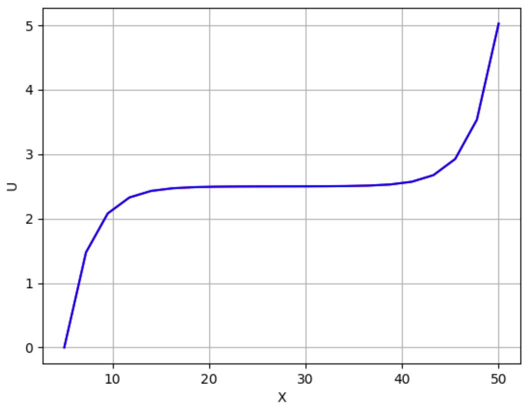


Рис. 4. Результат работы программы для 20 КЭ

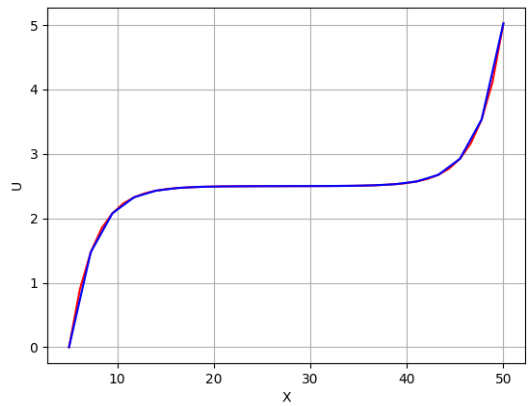


Рис. 5. Результат работы программы для 40 КЭ

| X | Аналитическое решение | МКЭ-решение | Абсолютная погрешность |
|---|---|---|---|
| 5.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 7.250000e+00 | 1.474523e+00 | 1.474525e+00 | 2.333584e-06 |
| 9.500000e+00 | 2.079359e+00 | 2.079361e+00 | 1.914420e-06 |
| 1.175000e+01 | 2.327457e+00 | 2.327459e+00 | 1.177895e-06 |
| 1.400000e+01 | 2.429226e+00 | 2.429227e+00 | 6.441669e-07 |
| 1.625000e+01 | 2.470972e+00 | 2.470973e+00 | 3.301771e-07 |
| 1.850000e+01 | 2.488101e+00 | 2.488101e+00 | 1.622645e-07 |
| 2.075000e+01 | 2.495139e+00 | 2.495139e+00 | 7.706170e-08 |
| 2.300000e+01 | 2.498054e+00 | 2.498054e+00 | 3.478267e-08 |
| 2.525000e+01 | 2.499318e+00 | 2.499318e+00 | 1.302114e-08 |
| 2.750000e+01 | 2.500003e+00 | 2.500003e+00 | 8.453287e-10 |
| 2.975000e+01 | 2.500692e+00 | 2.500692e+00 | 1.541325e-08 |
| 3.200000e+01 | 2.501967e+00 | 2.501967e+00 | 3.986086e-08 |
| 3.425000e+01 | 2.504910e+00 | 2.504910e+00 | 8.903815e-08 |
| 3.650000e+01 | 2.512017e+00 | 2.512017e+00 | 1.910709e-07 |
| 3.875000e+01 | 2.529316e+00 | 2.529316e+00 | 3.996966e-07 |
| 4.100000e+01 | 2.571478e+00 | 2.571477e+00 | 8.120248e-07 |
| 4.325000e+01 | 2.674259e+00 | 2.674258e+00 | 1.583199e-06 |
| 4.550000e+01 | 2.924827e+00 | 2.924824e+00 | 2.892981e-06 |
| 4.775000e+01 | 3.535681e+00 | 3.535676e+00 | 4.695994e-06 |
| 5.000000e+01 | 5.024876e+00 | 5.024870e+00 | 5.702698e-06 |

Таблица 3. 20 кубических КЭ

| X | Аналитическое решение | МКЭ-решение | Абсолютная погрешность |
|---|---|---|---|
| 5.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 6.125000e+00 | 8.988465e-01 | 8.988465e-01 | 2.788281e-08 |
| 7.250000e+00 | 1.474523e+00 | 1.474523e+00 | 3.571569e-08 |
| 8.375000e+00 | 1.843222e+00 | 1.843222e+00 | 3.431173e-08 |
| 9.500000e+00 | 2.079359e+00 | 2.079359e+00 | 2.930037e-08 |
| 1.062500e+01 | 2.230596e+00 | 2.230596e+00 | 2.345706e-08 |
| 1.175000e+01 | 2.327457e+00 | 2.327457e+00 | 1.802779e-08 |
| 1.287500e+01 | 2.389494e+00 | 2.389494e+00 | 1.347013e-08 |
| 1.400000e+01 | 2.429226e+00 | 2.429226e+00 | 9.859032e-09 |
| 1.512500e+01 | 2.454673e+00 | 2.454673e+00 | 7.102849e-09 |
| 1.625000e+01 | 2.470972e+00 | 2.470972e+00 | 5.053373e-09 |
| 1.737500e+01 | 2.481413e+00 | 2.481413e+00 | 3.558344e-09 |
| 1.850000e+01 | 2.488101e+00 | 2.488101e+00 | 2.483428e-09 |
| 1.962500e+01 | 2.492388e+00 | 2.492388e+00 | 1.718945e-09 |
| 2.075000e+01 | 2.495139e+00 | 2.495139e+00 | 1.179342e-09 |
| 2.187500e+01 | 2.496908e+00 | 2.496908e+00 | 7.998153e-10 |
| 2.300000e+01 | 2.498054e+00 | 2.498054e+00 | 5.321419e-10 |
| 2.412500e+01 | 2.498806e+00 | 2.498806e+00 | 3.406648e-10 |
| 2.525000e+01 | 2.499318e+00 | 2.499318e+00 | 1.988010e-10 |
| 2.637500e+01 | 2.499692e+00 | 2.499692e+00 | 8.613465e-11 |
| 2.750000e+01 | 2.500003e+00 | 2.500003e+00 | 1.408917e-11 |
| 2.862500e+01 | 2.500316e+00 | 2.500316e+00 | 1.171028e-10 |
| 2.975000e+01 | 2.500692e+00 | 2.500692e+00 | 2.386757e-10 |
| 3.087500e+01 | 2.501207e+00 | 2.501207e+00 | 3.973311e-10 |
| 3.200000e+01 | 2.501967e+00 | 2.501967e+00 | 6.168026e-10 |
| 3.312500e+01 | 2.503123e+00 | 2.503123e+00 | 9.292247e-10 |
| 3.425000e+01 | 2.504910e+00 | 2.504910e+00 | 1.379108e-09 |
| 3.537500e+01 | 2.507688e+00 | 2.507688e+00 | 2.028588e-09 |
| 3.650000e+01 | 2.512017e+00 | 2.512017e+00 | 2.964224e-09 |
| 3.762500e+01 | 2.518772e+00 | 2.518772e+00 | 4.305408e-09 |
| 3.875000e+01 | 2.529316e+00 | 2.529316e+00 | 6.214528e-09 |
| 3.987500e+01 | 2.545778e+00 | 2.545778e+00 | 8.907803e-09 |
| 4.100000e+01 | 2.571478e+00 | 2.571478e+00 | 1.266485e-08 |
| 4.212500e+01 | 2.611606e+00 | 2.611606e+00 | 1.783181e-08 |
| 4.325000e+01 | 2.674259e+00 | 2.674259e+00 | 2.480806e-08 |
| 4.437500e+01 | 2.772085e+00 | 2.772085e+00 | 3.399685e-08 |
| 4.550000e+01 | 2.924827e+00 | 2.924827e+00 | 4.568398e-08 |
| 4.662500e+01 | 3.163314e+00 | 3.163314e+00 | 5.977885e-08 |
| 4.775000e+01 | 3.535681e+00 | 3.535681e+00 | 7.530167e-08 |
| 4.887500e+01 | 4.117086e+00 | 4.117086e+00 | 8.941383e-08 |
| 5.000000e+01 | 5.024876e+00 | 5.024876e+00 | 9.563975e-08 |

Таблица 4. 40 кубических КЭ

Максимальная абсолютная погрешность 5.702698e-06 и 9.563975e-08 соответственно.

### Нахождение количества линейных КЭ, обеспечивающих ту же точность, что и 20 кубических

Так как очевидно, что при увлечении числа КЭ точность растет, найдем искомое следуя алгоритму, представленному на рисунке 6.

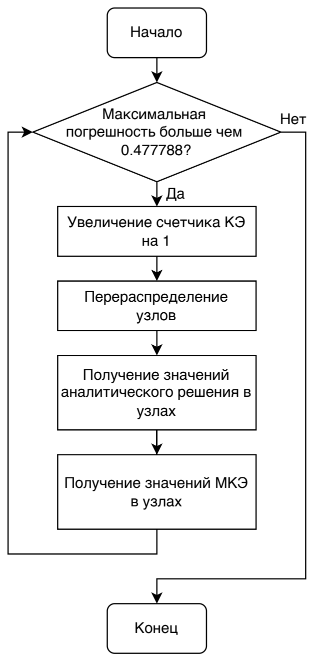


Рис. 6. Алгоритм нахождения количества КЭ, заданную точность

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 2422 КЭ.

# 7 Код

Листинг 1. *Реализация МКЭ*

```

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

double EPS = 1e-16;
double X_BEGIN = 5.0;
double X_END = 50.0;
size_t ELEMS_NUM = 20;
```

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

double a = 51.0, B = 0.0, C = -8.0, D = 20.0, usl_left = 0.0, usl_right = 1.0; // au''+Bu'+Cu+D=0

std::vector<double> solve_with_gauss(std::vector<std::vector<double>>& A, std::vector<double>& b){
    size_t row_size = A.size();
    size_t col_size = A.back().size();

    // Прямой ход Гаусса
    double pivot = 0.;
    for (size_t i = 0; i < row_size; i++) {
        for (size_t j = i + 1; j < col_size; j++) {
            if (std::abs(A.at(j).at(i)) < EPS) {
                continue;
            }
            pivot = A.at(j).at(i) / A.at(i).at(i);
            b.at(j) -= pivot * b.at(i);
            for (size_t k = 0; k < row_size; k++) {
                A.at(j).at(k) -= pivot * A.at(i).at(k) ;
            }
        }
    }

    // Обратный ход Гаусса
    std::vector<double> x(row_size);
    for (int i = row_size - 1.; i >= 0; i--) {
        x.at(i) = b.at(i);
        for (size_t j = i + 1; j < row_size; j++) {
            x.at(i) -= x.at(j) * A.at(i).at(j);
        }
        x.at(i) /= A.at(i).at(i);
    }

    return x;
}

double analytical_solution(double x) {
    return (exp(-2. * sqrt(2./51.) * (x + 5.)) * (exp(2. * sqrt(2./51.) * x) - exp(10. * sqrt(2./51.))) * (-10. * exp(2 * sqrt(2./51.) * x) + sqrt(102) * exp(2. * sqrt(2./51.) * (x + 45.)) + sqrt(102) * exp(100. * sqrt(2./51.)) + 10. * exp(190. * sqrt(2./51.))))/(4. * (1. + exp(60. * sqrt(6./17.))));
}

std::vector<double> build_analytical_solution(std::vector<double>& x_vec) {
```

```
    size_t x_vec_size = x_vec.size();
    std::vector<double> y_vec = std::vector<double>(x_vec_size);
    for (size_t i = 0; i < x_vec_size; i++) {
        y_vec.at(i) = analytical_solution(x_vec.at(i));
    }
    return y_vec;
}

std::vector<double> build_linear_solution(size_t elems_num) {
    double L = (X_END - X_BEGIN) / elems_num;
    size_t size = elems_num + 1;
    std::vector< std::vector<double> > A(size, std::vector<double>(size));
    std::vector<double> b(size);

    // Локальная матрица жесткости для линейного КЭ
    std::vector< std::vector<double> > local_matrix = {
        { (a / L) + (B / 2.) - C * L / 3., -(a / L) - (B / 2.) - C * L / 6.},
        { -(a / L) + (B / 2.) - C * L / 6., (a / L) - (B / 2.) - C * L / 3.},
    };

    // Ансамблирование и получение глобальной матрицы жесткости для линейного КЭ
    for (size_t i = 0; i < elems_num; i++) {
        for (size_t j = 0; j < 2; j++) {
            for (size_t k = 0; k < 2; k++) {
                A.at(i + j).at(i + k) += local_matrix.at(j).at(k);
            }
        }
    }

    for (size_t i = 0; i < size ; i++) {
        b.at(i) = D * L;
    }

    // Учет ГУ
    if ( 0 == 1 ) {
        b.at(0) = D * L /2. - a*usl_left;
    } else {
        b.at(0) = usl_left;
        A.at(0).at(0) = 1;
        A.at(0).at(1) = 0;
    }

    if ( 1 == 1 ) {
        b.at(size - 1) = D * L /2. + a*usl_right;
    } else {
        b.at(size - 1) = usl_right;
```

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

    // Решение полученной СЛАУ методом Гаусса
    std::vector<double> res = solve_with_gauss(A, b);
    return res;
}

std::vector<double> build_cube_solution(size_t elems_num) {
    double L = (X_END - X_BEGIN) / elems_num;
    size_t size = elems_num + 1;
    std::vector< std::vector<double> > A(size,std::vector<double>(size));
    std::vector<double> b(size);

    // Локальная матрица жесткости для кубического КЭ
    std::vector< std::vector<double> > local_matrix = {
        { a * 37./(10.*L) + B / 2. - C * 8. / 105. * L, -a * 189./(40.*L) - B * 57./80. - C * 33. / 560. * L, a * 27./(20.*L) + B * 3./10. + C * 3. / 140. * L, -a * 13./(40.*L) - B * 7./80. - C * 19. / 1680. * L},
        { -a * 189./(40.*L) + B * 57./80. - C * 33./ 560. * L, a * 54./(5.*L) + 0. - C * 27. / 70. * L, -a * 297./(40.*L) - B * 81./80. + C * 27. / 560. * L, a * 27./(20.*L) + B * 3./10. + C * 3. / 140. * L},
        { a * 27./(20.*L) - B * 3./10. + C * 3. / 140. * L, -a * 297./(40.*L) + B * 81./80. + C * 27. / 560. * L, a * 54./(5.*L) - 0. - C * 27. / 70. * L, -a * 189./(40.*L) - B * 57./80. - C * 33. / 560. * L},
        { -a * 13./(40.*L) + B * 7./80. - C * 19./ 1680.* L, a * 27./(20.*L) - B * 3./10. + C * 3. / 140. * L, -a * 189./(40.*L) + B * 57./80. - C * 33. / 560. * L, a * 37./(10.*L) - B * 1./2. - C * 8. / 105. * L}
    };

    // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего элементов учитываются далее)
    std::vector<double> local_b = { D * L / 8.0,
                                    D * 3.0 * L / 8.0,
                                    D * 3.0 * L / 8.0,
                                    D * L / 8.0 };


    // Производим матричные преобразования для обнуления элементов локальной матрицы жесткости, относящихся к внутренним узлам
    for (size_t i = 1; i < 3; i++) {
        for (size_t j = 0; j < 4; j++) {
            if (std::fabs(local_matrix.at(j).at(i)) > EPS && i != j) {
                double val = local_matrix.at(j).at(i) /local_matrix.at(i).at(i);
                local_b.at(j) -= val * local_b.at(i);
```

```
                for (size_t k = 0; k < 4; k++) {
                    local_matrix.at(j).at(k) -= val *local_matrix.at(i).at(k);
                }
            }
            continue;
        }
    }


    // Исключаем внутренние узлы из рассмотрения
    std::vector< std::vector<double> > local_matrix_mod = { { local_matrix.at(0).at(0),
        local_matrix.at(0).at(3) },
                                                            { local_matrix.at(3).at(0),
                                                                local_matrix.at(3).at(3)
                                                                } };
    std::vector<double> local_b_mod = { local_b.at(0),
                                        local_b.at(3)
                                        };

    // Ансамблирование и получение глобальной матрицы жесткости для кубического КЭ
    for (size_t i = 0; i < elems_num; i++) {
        for (size_t j = 0; j < 2; j++) {
            for (size_t k = 0; k < 2; k++) {
                A.at(i + j).at(i + k) += local_matrix_mod.at(j).at(k);
            }
        }
    }

    for (size_t i = 0; i < elems_num; i++) {
        b.at(i) += local_b_mod.at(0);
        b.at(i+1) += local_b_mod.at(1);
    }

    // Учет ГУ
    if (0 == 1 ) {
        b.at(0) = local_b_mod.at(0) - a * usl_left;
    } else {
        b.at(0) = usl_left;
        A.at(0).at(0) = 1.;
        A.at(0).at(1) = 0.;
    }

    if (1 == 1 ) {
        b.at(size - 1) = local_b_mod.at(1) + a * usl_right;
    } else {
        b.at(size - 1) = usl_right;
```

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

    // Решение полученной СЛАУ методом Гаусса
    std::vector<double> res = solve_with_gauss(A, b);
    return res;
}

double calc_abs_error(const std::vector<double>& y_real, const std::vector<double>& y) {
    double max_err = 0.0;
    for (size_t i = 0; i < y_real.size(); i++) {
        double err = std::fabs(y_real.at(i) - y.at(i));
        if (err > max_err) {
            max_err = err;
        }
    }
    return max_err;
}

int main() {

     std::vector<double> x(ELEMS_NUM + 1);
     for (size_t i = 0; i < x.size(); i++) {
         x.at(i) = X_BEGIN + i * L;
     }
     size_t x_size = x.size();

    std::vector<double> y;
    if (true) {
        y = build_linear_solution(ELEMS_NUM);
    } else {
        y = build_cube_solution(ELEMS_NUM);
    }
     std::vector<double> y_real = build_analytical_solution(x);


     FILE* gp;
     FILE* ab;
     FILE* pgr;
     FILE* tab;
     if (true) {
        if(ELEMS_NUM == 20) {
            gp = fopen("res/labs/text/graph/lin_20.txt", "w");
            ab = fopen("res/labs/text/graph/abs.txt", "w");
```

```
            for (size_t i = 0; i < x_size; i++) {
                fprintf(ab, "%lf %lf\n", x.at(i), y_real.at(i));
            }
            pgr = fopen("res/labs/text/pgr/lin_20.txt", "w");
            tab = fopen("res/labs/text/tab/lin_20.txt", "w");
        }
        if(ELEMS_NUM == 40) {
            gp = fopen("res/labs/text/graph/lin_40.txt", "w");
            pgr = fopen("res/labs/text/pgr/lin_40.txt", "w");
            tab = fopen("res/labs/text/tab/lin_40.txt", "w");
        }
    } else {
        if(ELEMS_NUM == 20) {
            gp = fopen("res/labs/text/graph/cub_20.txt", "w");
            pgr = fopen("res/labs/text/pgr/cub_20.txt", "w");
            tab = fopen("res/labs/text/tab/cub_20.txt", "w");
        }
        if(ELEMS_NUM == 40) {
            gp = fopen("res/labs/text/graph/cub_40.txt", "w");
            pgr = fopen("res/labs/text/pgr/cub_40.txt", "w");
            tab = fopen("res/labs/text/tab/cub_40.txt", "w");
        }
    }

    for (size_t i = 0; i < x.size()-1; i++) {
        fprintf(tab, "%le & %le & %le & %le \\\\\n", x.at(i), y_real.at(i), y.at(i),
            std::fabs(y_real.at(i) - y.at(i)));
    }
    fprintf(tab, "%le & %le & %le & %le", x.at(x.size()-1), y_real.at(x.size()-1),
        y.at(x.size()-1), std::fabs(y_real.at(x.size()-1) - y.at(x.size()-1)));

    for (size_t i = 0; i < x_size; i++) {
        fprintf(gp, "%lf %lf\n", x.at(i), y.at(i));
    }

    fprintf(pgr, "%e", calc_abs_error(y_real, y));
    fclose(gp);
    fclose(ab);
    fclose(pgr);
    fclose(tab);

    return 0;
}
```

## 8 Вывод

В ходе выполнения лабораторной работы был реализован МКЭ для различных функций форм, а также найдено количество линейных КЭ обеспечивающих точность 20ти кубических КЭ.


Постановка: © *доцент кафедры РК-6, кандидат технических наук, доцент, Трудоношин В.А.*

Решение и вёрстка: © *студент группы РК6-73б, Кулагин А.О.*

*2022, осенний семестр*