



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:	Чернышева Елизавета Андреевна
Группа:	РК6-736
Тип задания:	Лабораторная работа
Название:	Метод конечных элементов
Вариант:	48

Студент

подпись, дата

Чернышева Е.А.

Фамилия, И.О.

Преподаватель

подпись, дата

Трудоношин В. А.

Фамилия, И.О.

Оценка:

Москва, 2022

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	7
Ансамблирование	7
Учет граничных условий	8
6 Анализ результатов	8
Линейная функция-формы	8
Кубическая функция-формы	11
Нахождение количества линейных КЭ, обеспечивающих ту же точность, что и 20 кубических	14
7 Код	14
8 Вывод	20

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**

$$12\frac{d^2u}{dx^2} - 45\frac{du}{dx} + 6 = 0, \quad (1)$$

при следующих граничных условиях (г. у.):

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

$$u'(x = 12) = 10. \quad (3)$$

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 $12y''-45y'+6=0, y(0)=10, y'(12)=10$. The interface includes tabs for "NATURAL LANGUAGE" and "MATH INPUT", and buttons for "EXTENDED KEYBOARD", "EXAMPLES", "UPLOAD", and "RANDOM". The solution is displayed in a structured format:

- Input:** $\{12 y''(x) - 45 y'(x) + 6 = 0, y(0) = 10, y'(12) = 10\}$
- Autonomous equation:** $12 y''(x) = -6 + 45 y'(x)$
- ODE classification:** second-order linear ordinary differential equation
- Alternate forms:**
 - $\{15 y'(x) = 4 y''(x) + 2, y(0) = 10, y'(12) = 10\}$
 - $\{y''(x) = \frac{15 y'(x)}{4} - \frac{1}{2}, y(0) = 10, y'(12) = 10\}$
 - $\{-3(-4 y''(x) + 15 y'(x) - 2) = 0, y(0) = 10, y'(12) = 10\}$
- Differential equation solution:** $y(x) = \frac{30 e^{45} (x + 75) + 592 e^{(15 x)/4} - 592}{225 e^{45}}$

Buttons for "Approximate form" and "Step-by-step solution" are visible. The bottom of the interface includes a "Download Page" button and the text "POWERED BY THE WOLFRAM LANGUAGE".

Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = y(x) = \frac{30e^{45}(x + 75) + 592e^{(15x)/4} - 592}{225e^{45}}.$$

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:

$$\int_0^L \mathbf{W}_e \left(12 \frac{d^2 \mathbf{u}}{dx^2} - 45 \frac{d\mathbf{u}}{dx} + 6 \right) dx = 0, \quad (4)$$

где $\mathbf{W}_e = \mathbf{N}_e^T$.

$$\int_0^L \mathbf{W}_e \left(12 \frac{d^2 \mathbf{u}}{dx^2} - 45 \frac{d\mathbf{u}}{dx} + 6 \right) dx = 12 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx - 45 \int_0^L \mathbf{W}_e \frac{d\mathbf{u}}{dx} dx + 6 \int_0^L \mathbf{W}_e dx = 0$$

Распишем каждое слагаемое отдельно:

$$\begin{aligned} 12 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx &= 12 \int_0^L \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d^2 \mathbf{u}}{dx^2} dx = 12 \left[\left(1 - \frac{x}{L} \right) \frac{d\mathbf{u}}{dx} \right]_0^L - \\ &- 12 \int_0^L \frac{d}{dx} \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d}{dx} \left[\left(1 - \frac{x}{L} \right); \frac{x}{L} \right] \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -12 \frac{d\mathbf{u}}{dx} |_i \\ 12 \frac{d\mathbf{u}}{dx} |_j \end{bmatrix} - 12 \begin{bmatrix} \frac{1}{L}, & -\frac{1}{L} \\ -\frac{1}{L}, & \frac{1}{L} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ -45 \int_0^L \mathbf{W}_e \frac{d\mathbf{u}}{dx} dx &= -45 \int_0^L \left[\left(1 - \frac{x}{L} \right) \frac{x}{L} \right] \frac{d\mathbf{u}}{dx} \begin{bmatrix} u_i \\ u_j \end{bmatrix} dx = \\ &= -\frac{45}{L} \int_0^L \begin{bmatrix} \left(1 - \frac{x}{L} \right) & \left(-1 + \frac{x}{L} \right) \\ -\frac{x}{L} & \frac{x}{L} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = -45 \begin{bmatrix} -\frac{1}{2}, & \frac{1}{2} \\ -\frac{1}{2}, & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ 6 \int_0^L \mathbf{W}_e dx &= 6 \left[\frac{L}{2} \right] \end{aligned}$$

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

$$\begin{bmatrix} 12 \frac{1}{L} - 45 \frac{1}{2}, & -12 \frac{1}{L} + 45 \frac{1}{2} \\ -12 \frac{1}{L} - 45 \frac{1}{2}, & 12 \frac{1}{L} + 45 \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -12 \frac{d\mathbf{u}}{dx} |_i + 6 \frac{L}{2} \\ 12 \frac{d\mathbf{u}}{dx} |_j + 6 \frac{L}{2} \end{bmatrix}$$

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}
12 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx &= 12 \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 = \\
&= 12 \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 - 12 \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} -12 \frac{d\mathbf{u}}{dx} \Big|_i \\ 0 \\ 0 \\ 12 \frac{d\mathbf{u}}{dx} \Big|_l \end{bmatrix} - 12 \begin{bmatrix} \frac{37}{10L} & -\frac{189}{40} & \frac{27}{20L} & -\frac{13}{40L} \\ -\frac{189}{40} & \frac{54}{297} & -\frac{40L}{54} & \frac{20L}{27} \\ \frac{40}{27} & -\frac{5L}{297} & \frac{54}{40L} & -\frac{189}{20L} \\ -\frac{20L}{13} & -\frac{40L}{27} & -\frac{5L}{189} & -\frac{40}{37} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} \\
-45 \int_0^L \mathbf{W}_e \frac{d\mathbf{u}}{dx} dx &= -45 \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 = \\
&= -45 \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} \\
6 \int_0^L \mathbf{W}_e dx &= 6 \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:

$$\begin{bmatrix} 12\frac{37}{10L} - 45\frac{1}{2} & -12\frac{189}{40L} + 45\frac{57}{80} & 12\frac{27}{20L} - 45\frac{3}{10} & -12\frac{13}{40L} + 45\frac{7}{80} \\ -12\frac{189}{40L} - 45\frac{57}{80} & 12\frac{54}{5L} + 0 & -12\frac{297}{40L} + 45\frac{81}{80} & 12\frac{27}{20L} - 45\frac{3}{10} \\ 12\frac{27}{20L} + 45\frac{3}{10} & -12\frac{297}{40L} - 45\frac{81}{80} & 12\frac{54}{5L} + 0 & -12\frac{189}{40L} + 45\frac{57}{80} \\ -12\frac{13}{40L} - 45\frac{7}{80} & 12\frac{27}{20L} + 45\frac{3}{10} & -12\frac{189}{40L} + 45\frac{57}{80} & 12\frac{37}{10L} + 45\frac{1}{2} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} 6\frac{L}{8} - 12\frac{du}{dx}|_i \\ 6\frac{3L}{8} \\ 6\frac{3L}{8} \\ 6\frac{L}{8} + 12\frac{du}{dx}|_l \end{bmatrix} \quad (5)$$

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

$$\begin{bmatrix} a_{11} & 0 & 0 & a_{14} \\ a_{21} & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 12\frac{du}{dx}|_i \\ b_2 \\ b_3 \\ b_4 + 12\frac{du}{dx}|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 12\frac{du}{dx}|_i \\ b_4 + 12\frac{du}{dx}|_l \end{bmatrix}$$

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

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

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

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

Учет граничных условий

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

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

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

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

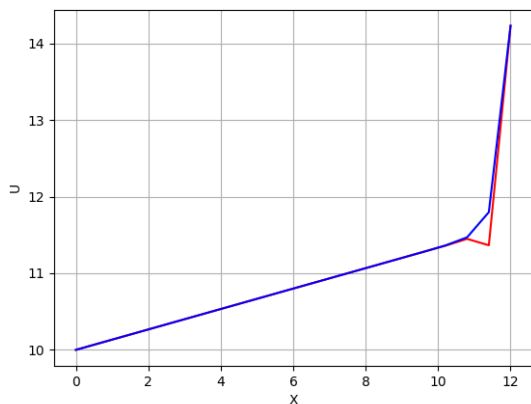


Рис. 2. Результат работы программы для 20 КЭ

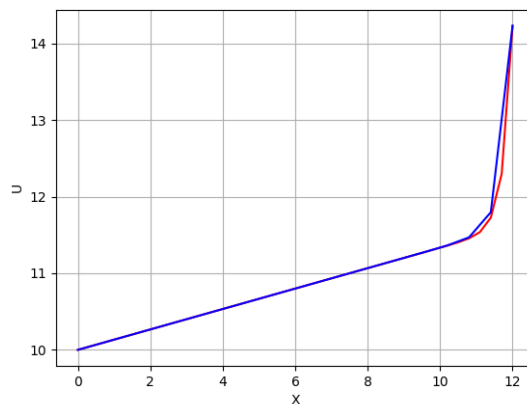


Рис. 3. Результат работы программы для 40 КЭ

X	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.600000	10.080000	10.080000	0.000000
1.200000	10.160000	10.160000	0.000000
1.800000	10.240000	10.240000	0.000000
2.400000	10.320000	10.320000	0.000000
3.000000	10.400000	10.400000	0.000000
3.600000	10.480000	10.480000	0.000000
4.200000	10.560000	10.560000	0.000000
4.800000	10.640000	10.640000	0.000000
5.400000	10.720000	10.720000	0.000000
6.000000	10.800000	10.800000	0.000000
6.600000	10.880000	10.880000	0.000000
7.200000	10.960000	10.960000	0.000000
7.800000	11.040000	11.040000	0.000000
8.400000	11.120004	11.120000	0.000003
9.000000	11.200034	11.199998	0.000036
9.600000	11.280325	11.280032	0.000293
10.200000	11.363081	11.359464	0.003616
10.800000	11.469229	11.449104	0.020125
11.400000	11.797317	11.365229	0.432088
12.000000	14.231111	14.231111	0.000000

Таблица 1. 20 линейных КЭ

X	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.300000	10.040000	10.040000	0.000000
0.600000	10.080000	10.080000	0.000000
0.900000	10.120000	10.120000	0.000000
1.200000	10.160000	10.160000	0.000000
1.500000	10.200000	10.200000	0.000000
1.800000	10.240000	10.240000	0.000000
2.100000	10.280000	10.280000	0.000000
2.400000	10.320000	10.320000	0.000000
2.700000	10.360000	10.360000	0.000000
3.000000	10.400000	10.400000	0.000000
3.300000	10.440000	10.440000	0.000000
3.600000	10.480000	10.480000	0.000000
3.900000	10.520000	10.520000	0.000000
4.200000	10.560000	10.560000	0.000000
4.500000	10.600000	10.600000	0.000000
4.800000	10.640000	10.640000	0.000000
5.100000	10.680000	10.680000	0.000000
5.400000	10.720000	10.720000	0.000000
5.700000	10.760000	10.760000	0.000000
6.000000	10.800000	10.800000	0.000000
6.300000	10.840000	10.840000	0.000000
6.600000	10.880000	10.880000	0.000000
6.900000	10.920000	10.920000	0.000000
7.200000	10.960000	10.960000	0.000000
7.500000	11.000000	11.000000	0.000000
7.800000	11.040000	11.040000	0.000000
8.100000	11.080001	11.080000	0.000001
8.400000	11.120004	11.120001	0.000003
8.700000	11.160011	11.160002	0.000009
9.000000	11.200034	11.200008	0.000026
9.300000	11.240105	11.240028	0.000078
9.600000	11.280325	11.280099	0.000225
9.900000	11.321000	11.320355	0.000645
10.200000	11.363081	11.361268	0.001813
10.500000	11.409489	11.404528	0.004961
10.800000	11.469229	11.456172	0.013057
11.100000	11.570032	11.537758	0.032274
11.400000	11.797317	11.726279	0.071038
11.700000	12.414197	12.296711	0.117486
12.000000	14.231111	14.231111	0.000000

Таблица 2. 40 линейных КЭ

Максимальная абсолютная погрешность $4.320883\text{e-}01$ и $1.174856\text{e-}01$ соответственно.

Кубическая функция-формы

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

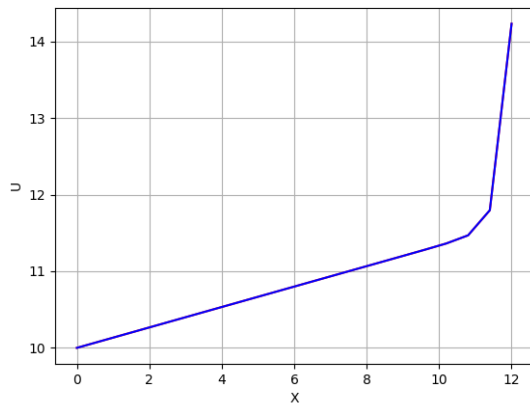


Рис. 4. Результат работы программы для 20
КЭ

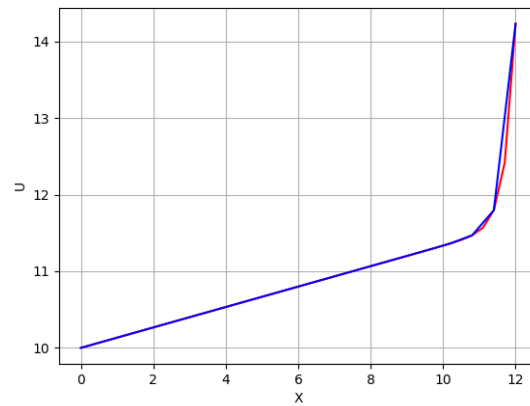


Рис. 5. Результат работы программы для 40
КЭ

X	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.600000	10.080000	10.080000	0.000000
1.200000	10.160000	10.160000	0.000000
1.800000	10.240000	10.240000	0.000000
2.400000	10.320000	10.320000	0.000000
3.000000	10.400000	10.400000	0.000000
3.600000	10.480000	10.480000	0.000000
4.200000	10.560000	10.560000	0.000000
4.800000	10.640000	10.640000	0.000000
5.400000	10.720000	10.720000	0.000000
6.000000	10.800000	10.800000	0.000000
6.600000	10.880000	10.880000	0.000000
7.200000	10.960000	10.960000	0.000000
7.800000	11.040000	11.040000	0.000000
8.400000	11.120004	11.120004	0.000000
9.000000	11.200034	11.200034	0.000001
9.600000	11.280325	11.280320	0.000005
10.200000	11.363081	11.363048	0.000032
10.800000	11.469229	11.469024	0.000205
11.400000	11.797317	11.796344	0.000973
12.000000	14.231111	14.231111	0.000000

Таблица 3. 20 кубических КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
0.000000	10.000000	10.000000	0.000000
0.300000	10.040000	10.040000	0.000000
0.600000	10.080000	10.080000	0.000000
0.900000	10.120000	10.120000	0.000000
1.200000	10.160000	10.160000	0.000000
1.500000	10.200000	10.200000	0.000000
1.800000	10.240000	10.240000	0.000000
2.100000	10.280000	10.280000	0.000000
2.400000	10.320000	10.320000	0.000000
2.700000	10.360000	10.360000	0.000000
3.000000	10.400000	10.400000	0.000000
3.300000	10.440000	10.440000	0.000000
3.600000	10.480000	10.480000	0.000000
3.900000	10.520000	10.520000	0.000000
4.200000	10.560000	10.560000	0.000000
4.500000	10.600000	10.600000	0.000000
4.800000	10.640000	10.640000	0.000000
5.100000	10.680000	10.680000	0.000000
5.400000	10.720000	10.720000	0.000000
5.700000	10.760000	10.760000	0.000000
6.000000	10.800000	10.800000	0.000000
6.300000	10.840000	10.840000	0.000000
6.600000	10.880000	10.880000	0.000000
6.900000	10.920000	10.920000	0.000000
7.200000	10.960000	10.960000	0.000000
7.500000	11.000000	11.000000	0.000000
7.800000	11.040000	11.040000	0.000000
8.100000	11.080001	11.080001	0.000000
8.400000	11.120004	11.120004	0.000000
8.700000	11.160011	11.160011	0.000000
9.000000	11.200034	11.200034	0.000000
9.300000	11.240105	11.240105	0.000000
9.600000	11.280325	11.280325	0.000000
9.900000	11.321000	11.321000	0.000000
10.200000	11.363081	11.363080	0.000000
10.500000	11.409489	11.409488	0.000001
10.800000	11.469229	11.469226	0.000003
11.100000	11.570032	11.570025	0.000006
11.400000	11.797317	11.797304	0.000013
11.700000	12.414197	12.414176	0.000020
12.000000	14.231111	14.231111	0.000000

Таблица 4. 40 кубических КЭ

Максимальная абсолютная погрешность $9.731184e-04$ и $2.029513e-05$ 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.

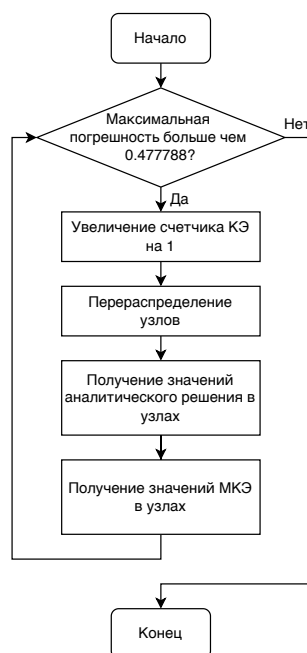


Рис. 6. Алгоритм нахождения количества КЭ, заданную точность

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 411 .

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 = 12.0;
9  size_t ELEMS_NUM = 20;
  
```

```

10 double L = (X_END - X_BEGIN) / ELEMS_NUM;
11
12 double a = 12.0, B = -45.0, C = 0.0, D = 6.0, usl_left = 10.0, usl_right = 10.0; //
    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 (30. * exp(45.) * (x + 75.) + 592. * exp(15. * x / 4.) - 592.) / 225. / exp(45.);
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) + (B / 2.), -(a / L) - (B / 2.) },
68         { -(a / L) + (B / 2.), (a / L) - (B / 2.) },
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./(10.*L) + B / 2., -a * 189./(40.*L) - B * 57./80., a * 27./(20.*L) + B
          * 3./10., -a * 13./(40.*L) - B * 7./80. },
115         { -a * 189./(40.*L) + B * 57./80., a * 54./(5.*L) + 0., -a * 297./(40.*L) - B *
          81./80., a * 27./(20.*L) + B * 3./10. },
116         { a * 27./(20.*L) - B * 3./10., -a * 297./(40.*L) + B * 81./80., a * 54./(5.*L) -
          0., -a * 189./(40.*L) - B * 57./80. },
117         { -a * 13./(40.*L) + B * 7./80., a * 27./(20.*L) - B * 3./10., -a * 189./(40.*L)
          + B * 57./80., a * 37./(10.*L) - B * 1./2. }
118     };
119
120     // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего
        элементов учитываются далее)
121     std::vector<double> local_b = { D * L / 8.0,
122                                     D * 3.0 * L / 8.0,
123                                     D * 3.0 * L / 8.0,
124                                     D * L / 8.0 };
125
126
127     // Производим матричные преобразования для обнуления элементов локальной
        матрицы жесткости, относящихся к внутренним узлам
128     for (size_t i = 1; i < 3; i++) {
129         for (size_t j = 0; j < 4; j++) {
130             if (std::fabs(local_matrix.at(j).at(i)) > EPS && i != j) {
131                 double val = local_matrix.at(j).at(i) / local_matrix.at(i).at(i);
132                 local_b.at(j) -= val * local_b.at(i);
133                 for (size_t k = 0; k < 4; k++) {
134                     local_matrix.at(j).at(k) -= val * local_matrix.at(i).at(k);
135                 }
136             }
137             continue;
138         }
139     }

```

```

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

```

```

183 }
184
185 double calc_abs_error(const std::vector<double>& y_real, const std::vector<double>&
    y) {
186     double max_err = 0.0;
187     for (size_t i = 0; i < y_real.size(); i++) {
188         double err = std::fabs(y_real.at(i) - y.at(i));
189         if (err > max_err) {
190             max_err = err;
191         }
192     }
193     return max_err;
194 }
195
196 int main() {
197
198     std::vector<double> x(ELEMS_NUM + 1);
199     for (size_t i = 0; i < x.size(); i++) {
200         x.at(i) = X_BEGIN + i * L;
201     }
202     size_t x_size = x.size();
203
204     std::vector<double> y;
205     if (true) {
206         y = build_linear_solution(ELEMS_NUM);
207     } else {
208         y = build_cube_solution(ELEMS_NUM);
209     }
210     std::vector<double> y_real = build_analytical_solution(x);
211
212
213     FILE* gp;
214     FILE* ab;
215     FILE* pgr;
216     FILE* tab;
217     if (true) {
218         if(ELEMS_NUM == 20) {
219             gp = fopen("res/labs/text/graph/lin_20.txt", "w");
220             ab = fopen("res/labs/text/graph/abs.txt", "w");
221             for (size_t i = 0; i < x_size; i++) {
222                 fprintf(ab, "%lf %lf\n", x.at(i), y_real.at(i));
223             }
224             pgr = fopen("res/labs/text/pgr/lin_20.txt", "w");
225             tab = fopen("res/labs/text/tab/lin_20.txt", "w");
226         }
227         if(ELEMS_NUM == 40) {

```

```

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

```

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, Чернышева Е.А.

2022, осенний семестр