



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:	Стройков Марк Николаевич
Группа:	РК6-726
Тип задания:	Лабораторная работа
Название:	Метод конечных элементов
Вариант:	38

Студент

подпись, дата

Стройков М.Н.

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	5
Линейная функция-формы КЭ	5
Кубическая функция-формы КЭ	6
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

$$4 \frac{d^2 u}{dx^2} - 9u + 10 = 0, \quad (1)$$

при следующих граничных условиях (г. у.):

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

$$u'(x = 14) = 1. \quad (3)$$

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.

$4y''-9y+10=0, y(2)=0, y'(14)=1$

NATURAL LANGUAGE

MATH INPUT

EXTENDED KEYBOARD

EXAMPLES

UPLOAD

RANDOM

Input

$\{4 y''(x) - 9 y(x) + 10 = 0, y(2) = 0, y'(14) = 1\}$

Autonomous equation

$4 y''(x) = -10 + 9 y(x)$

Autonomous equation »

ODE classification

second-order linear ordinary differential equation

Alternate forms

$\{9 y(x) = 4 y''(x) + 10, y(2) = 0, y'(14) = 1\}$

$\{y''(x) = \frac{9 y(x)}{4} - \frac{5}{2}, y(2) = 0, y'(14) = 1\}$

Differential equation solution

Approximate form

☒ Step-by-step solution

$$y(x) = \frac{2(-3e^{21-(3x)/2} - 5e^{39-(3x)/2} - 5e^{(3x)/2-3} + 3e^{(3x)/2+15} + 5 + 5e^{36})}{9(1+e^{36})}$$

Download Page

POWERED BY THE WOLFRAM LANGUAGE

Рис. 1. Аналитическое решение

Таким образом, получаем:

$$u(x) = \frac{2(-3e^{21-(3x)/2} - 5e^{39-(3x)/2} - 5e^{(3x)/2-3} + 3e^{(3x)/2+15} + 5 + 5e^{36})}{9(1+e^{36})}.$$

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:

$$\int_0^L \mathbf{W}_e \left(4 \frac{d^2 \mathbf{u}}{dx^2} - 9u + 10 \right) dx = 0, \quad (4)$$

где $\mathbf{W}_e = \mathbf{N}_e^T$.

$$\int_0^L \mathbf{W}_e \left(4 \frac{d^2 \mathbf{u}}{dx^2} - 9u + 10 \right) dx = 4 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx - 9 \int_0^L \mathbf{W}_e u dx + 10 \int_0^L \mathbf{W}_e dx = 0$$

Распишем каждое слагаемое отдельно:

$$\begin{aligned} 4 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx &= 4 \int_0^L \left[\left(1 - \frac{x}{L}\right) \right] \frac{d^2 \mathbf{u}}{dx^2} dx = 4 \left[\left(1 - \frac{x}{L}\right) \right] \frac{d\mathbf{u}}{dx} \Big|_0^L - \\ &- 4 \int_0^L \frac{d}{dx} \left[\left(1 - \frac{x}{L}\right) \right] \frac{d}{dx} \left[\left(1 - \frac{x}{L}\right); \frac{x}{L} \right] \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -4 \frac{d\mathbf{u}}{dx} \Big|_i \\ 4 \frac{d\mathbf{u}}{dx} \Big|_j \end{bmatrix} - 4 \begin{bmatrix} \frac{1}{L}, & -\frac{1}{L} \\ -\frac{1}{L}, & \frac{1}{L} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ -9 \int_0^L \mathbf{W}_e u dx &= -9 \int_0^L \left[\left(1 - \frac{x}{L}\right) \right] \mathbf{u} \begin{bmatrix} u_i \\ u_j \end{bmatrix} dx = -9 \begin{bmatrix} \frac{L}{3}, & \frac{L}{6} \\ \frac{L}{6}, & \frac{L}{3} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ 10 \int_0^L \mathbf{W}_e dx &= 10 \begin{bmatrix} \frac{L}{2} \\ \frac{L}{2} \end{bmatrix} \end{aligned}$$

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

$$\begin{bmatrix} 4\frac{1}{L} - 9\frac{L}{3}, & -4\frac{1}{L} - 9\frac{L}{6} \\ -4\frac{1}{L} - 9\frac{L}{6}, & 4\frac{1}{L} - 9\frac{L}{3} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -4 \frac{du}{dx} \Big|_i + 10 \frac{L}{2} \\ 4 \frac{du}{dx} \Big|_j + 10 \frac{L}{2} \end{bmatrix}$$

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}
4 \int_0^L \mathbf{W}_e \frac{d^2 \mathbf{u}}{dx^2} dx &= 4 \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 = \\
&= 4 \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 - 4 \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} -4 \frac{d\mathbf{u}}{dx} \Big|_i \\ 0 \\ 0 \\ 4 \frac{d\mathbf{u}}{dx} \Big|_l \end{bmatrix} - 4 \begin{bmatrix} \frac{37}{10L} & -\frac{189}{40} & \frac{27}{20L} & -\frac{13}{40L} \\ -\frac{189}{40} & \frac{5L}{27} & -\frac{40L}{54} & \frac{20L}{27} \\ \frac{20L}{27} & -\frac{40L}{54} & \frac{5L}{27} & -\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} \\
-9 \int_0^L \mathbf{W}_e \mathbf{u} dx &= -9 \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 = \\
&= -9 \begin{bmatrix} \frac{8L}{105} & \frac{33L}{560} & -\frac{3L}{140} & \frac{119L}{1680} \\ -\frac{33L}{560} & \frac{170}{27L} & -\frac{560}{27L} & -\frac{140}{33L} \\ -\frac{140}{33L} & -\frac{560}{27L} & \frac{170}{33L} & \frac{560}{8L} \\ \frac{119L}{1680} & -\frac{140}{33L} & \frac{560}{8L} & \frac{8L}{105} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} \\
10 \int_0^L \mathbf{W}_e dx &= 10 \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:

$$\begin{bmatrix} 4\frac{37}{10L} - 9\frac{8L}{105} & -4\frac{189}{40L} - 9\frac{33L}{560} & 4\frac{27}{20L} + 9\frac{3L}{140} & -4\frac{13}{40L} - 9\frac{119L}{1680} \\ -4\frac{189}{40L} - 9\frac{33L}{560} & 4\frac{54}{5L} + 0 - 9\frac{27L}{170} & -4\frac{297}{40L} + 9\frac{27L}{560} & 4\frac{27}{20L} + 9\frac{27L}{560} \\ 4\frac{27}{20L} + 9\frac{27L}{560} & -4\frac{297}{40L} + 9\frac{27L}{560} & 4\frac{54}{5L} + 0 - 9\frac{27L}{170} & -4\frac{189}{40L} - 9\frac{33L}{560} \\ -4\frac{13}{40L} - 9\frac{119L}{1680} & 4\frac{27}{20L} + 9\frac{27L}{560} & -4\frac{297}{40L} + 9\frac{27L}{560} & 4\frac{27}{20L} + 9\frac{27L}{560} \end{bmatrix}.$$

$$\begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} 10\frac{L}{8} - 4\frac{du}{dx}|_i \\ 10\frac{3L}{8} \\ 10\frac{3L}{8} \\ 10\frac{L}{8} + 4\frac{du}{dx}|_l \end{bmatrix} \quad (5)$$

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

$$\begin{bmatrix} a_{11} & 0 & 0 & a_{14} \\ a_{21} & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \\ u_k \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 4\frac{du}{dx}|_i \\ b_2 \\ b_3 \\ b_4 + 4\frac{du}{dx}|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix} \begin{bmatrix} u_i \\ u_l \end{bmatrix} = \begin{bmatrix} b_1 - 4\frac{du}{dx}|_i \\ b_4 + 4\frac{du}{dx}|_l \end{bmatrix}$$

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

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} b_1 - 4\frac{du}{dx}|_i \\ b_2 + 4\frac{du}{dx}|_j \end{bmatrix},$$

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

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

Учет граничных условий

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

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

6 Анализ результатов

Проведем сравнение результатов согласно заданию.

Линейная функция-формы

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

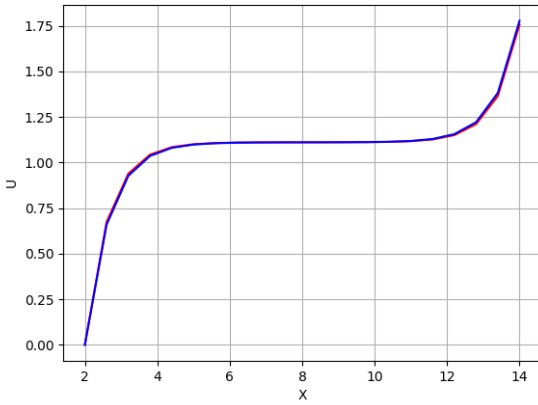


Рис. 2. Результат работы программы для 20
КЭ

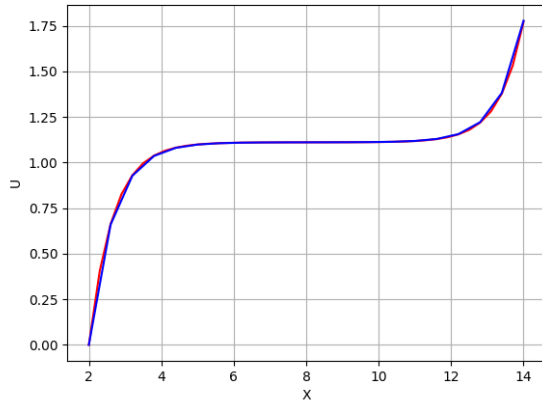


Рис. 3. Результат работы программы для 40
КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
2.000000	0.000000e+00	0.000000e+00	0.000000e+00
2.600000	6.593671e-01	6.742184e-01	1.485135e-02
3.200000	9.274457e-01	9.393234e-01	1.187769e-02
3.800000	1.036438e+00	1.043564e+00	7.125203e-03
4.400000	1.080752e+00	1.084551e+00	3.799613e-03
5.000000	1.098769e+00	1.100668e+00	1.899500e-03
5.600000	1.106095e+00	1.107006e+00	9.111370e-04
6.200000	1.109076e+00	1.109500e+00	4.236239e-04
6.800000	1.110295e+00	1.110485e+00	1.898670e-04
7.400000	1.110807e+00	1.110884e+00	7.659023e-05
8.000000	1.111056e+00	1.111070e+00	1.369272e-05
8.600000	1.111258e+00	1.111217e+00	4.023546e-05
9.200000	1.111586e+00	1.111465e+00	1.215424e-04
9.800000	1.112326e+00	1.112043e+00	2.832939e-04
10.400000	1.114118e+00	1.113493e+00	6.249816e-04
11.000000	1.118516e+00	1.117175e+00	1.340669e-03
11.600000	1.129326e+00	1.126535e+00	2.791664e-03
12.200000	1.155915e+00	1.150337e+00	5.577321e-03
12.800000	1.221310e+00	1.210872e+00	1.043853e-02
13.400000	1.382158e+00	1.364824e+00	1.733377e-02
14.000000	1.777778e+00	1.756356e+00	2.142143e-02

Таблица 1. 20 линейных КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
2.000000	0.000000e+00	0.000000e+00	0.000000e+00
2.300000	4.026354e-01	4.053828e-01	2.747381e-03
2.600000	6.593671e-01	6.628639e-01	3.496820e-03
2.900000	8.230664e-01	8.264044e-01	3.338024e-03
3.200000	9.274457e-01	9.302781e-01	2.832391e-03
3.500000	9.940010e-01	9.962541e-01	2.253140e-03
3.800000	1.036438e+00	1.038159e+00	1.720655e-03
4.100000	1.063498e+00	1.064775e+00	1.277505e-03
4.400000	1.080752e+00	1.081681e+00	9.291186e-04
4.700000	1.091753e+00	1.092419e+00	6.651624e-04
5.000000	1.098769e+00	1.099239e+00	4.702849e-04
5.300000	1.103242e+00	1.103571e+00	3.291294e-04
5.600000	1.106095e+00	1.106323e+00	2.283645e-04
5.900000	1.107915e+00	1.108072e+00	1.572360e-04
6.200000	1.109076e+00	1.109184e+00	1.074484e-04
6.500000	1.109819e+00	1.109892e+00	7.278711e-05
6.800000	1.110295e+00	1.110344e+00	4.868540e-05
7.100000	1.110604e+00	1.110635e+00	3.182274e-05
7.400000	1.110807e+00	1.110827e+00	1.978525e-05
7.700000	1.110949e+00	1.110959e+00	1.079052e-05
8.000000	1.111056e+00	1.111060e+00	3.466549e-06
8.300000	1.111153e+00	1.111149e+00	3.330181e-06
8.600000	1.111258e+00	1.111247e+00	1.067372e-05
8.900000	1.111393e+00	1.111373e+00	1.972158e-05
9.200000	1.111586e+00	1.111554e+00	3.187878e-05
9.500000	1.111877e+00	1.111828e+00	4.899116e-05
9.800000	1.112326e+00	1.112253e+00	7.358758e-05
10.100000	1.113025e+00	1.112916e+00	1.091922e-04
10.400000	1.114118e+00	1.113958e+00	1.607273e-04
10.700000	1.115831e+00	1.115596e+00	2.350218e-04
11.000000	1.118516e+00	1.118174e+00	3.414242e-04
11.300000	1.122725e+00	1.122233e+00	4.924814e-04
11.600000	1.129326e+00	1.128622e+00	7.045588e-04
11.900000	1.139679e+00	1.138681e+00	9.981215e-04
12.200000	1.155915e+00	1.154517e+00	1.397084e-03
12.500000	1.181377e+00	1.179451e+00	1.926078e-03
12.800000	1.221310e+00	1.218707e+00	2.603467e-03
13.100000	1.283938e+00	1.280512e+00	3.426146e-03
13.400000	1.382158e+00	1.377818e+00	4.339017e-03
13.700000	1.536197e+00	1.531020e+00	5.176583e-03
14.000000	1.777778e+00	1.772223e+00	5.554790e-03

Таблица 2. 40 линейных КЭ

Максимальная абсолютная погрешность $2.142143\text{e-}02$ и $5.554790\text{e-}03$ соответственно.

Кубическая функция-формы

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

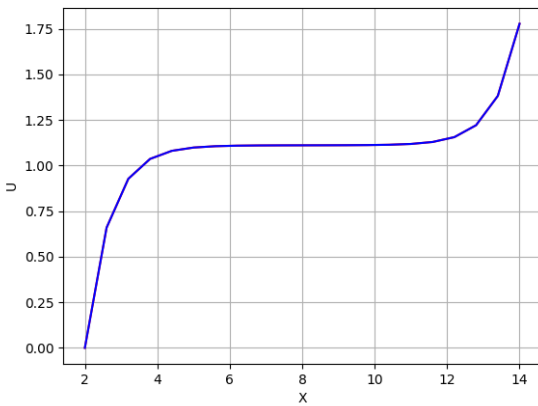


Рис. 4. Результат работы программы для 20
КЭ

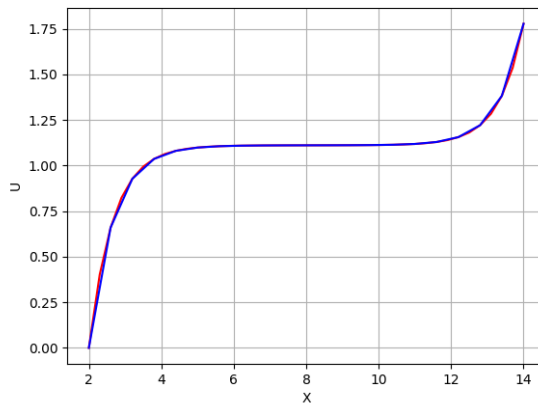


Рис. 5. Результат работы программы для 40
КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
2.000000	0.000000e+00	0.000000e+00	0.000000e+00
2.600000	6.593671e-01	6.593682e-01	1.102376e-06
3.200000	9.274457e-01	9.274466e-01	8.963823e-07
3.800000	1.036438e+00	1.036439e+00	5.466572e-07
4.400000	1.080752e+00	1.080752e+00	2.963268e-07
5.000000	1.098769e+00	1.098769e+00	1.505689e-07
5.600000	1.106095e+00	1.106095e+00	7.339519e-08
6.200000	1.109076e+00	1.109076e+00	3.466394e-08
6.800000	1.110295e+00	1.110295e+00	1.576349e-08
7.400000	1.110807e+00	1.110807e+00	6.429195e-09
8.000000	1.111056e+00	1.111056e+00	1.141615e-09
8.600000	1.111258e+00	1.111258e+00	3.431914e-09
9.200000	1.111586e+00	1.111586e+00	1.024358e-08
9.800000	1.112326e+00	1.112326e+00	2.354799e-08
10.400000	1.114118e+00	1.114118e+00	5.116208e-08
11.000000	1.118516e+00	1.118515e+00	1.080251e-07
11.600000	1.129326e+00	1.129326e+00	2.213609e-07
12.200000	1.155915e+00	1.155914e+00	4.351771e-07
12.800000	1.221310e+00	1.221309e+00	8.014704e-07
13.400000	1.382158e+00	1.382156e+00	1.309885e-06
14.000000	1.777778e+00	1.777776e+00	1.594959e-06

Таблица 3. 20 кубических КЭ

Х	Аналитическое решение	МКЭ- решение	Абсолютная погрешность
2.000000	0.000000e+00	0.000000e+00	0.000000e+00
2.300000	4.026354e-01	4.026354e-01	1.322480e-08
2.600000	6.593671e-01	6.593671e-01	1.686501e-08
2.900000	8.230664e-01	8.230664e-01	1.613039e-08
3.200000	9.274457e-01	9.274458e-01	1.371356e-08
3.500000	9.940010e-01	9.940010e-01	1.093015e-08
3.800000	1.036438e+00	1.036438e+00	8.363189e-09
4.100000	1.063498e+00	1.063498e+00	6.221286e-09
4.400000	1.080752e+00	1.080752e+00	4.533431e-09
4.700000	1.091753e+00	1.091754e+00	3.251776e-09
5.000000	1.098769e+00	1.098769e+00	2.303503e-09
5.300000	1.103242e+00	1.103242e+00	1.615200e-09
5.600000	1.106095e+00	1.106095e+00	1.122830e-09
5.900000	1.107915e+00	1.107915e+00	7.745582e-10
6.200000	1.109076e+00	1.109076e+00	5.302745e-10
6.500000	1.109819e+00	1.109819e+00	3.598510e-10
6.800000	1.110295e+00	1.110295e+00	2.410896e-10
7.100000	1.110604e+00	1.110604e+00	1.578089e-10
7.400000	1.110807e+00	1.110807e+00	9.821410e-11
7.700000	1.110949e+00	1.110949e+00	5.357270e-11
8.000000	1.111056e+00	1.111056e+00	1.714517e-11
8.300000	1.111153e+00	1.111153e+00	1.669931e-11
8.600000	1.111258e+00	1.111258e+00	5.325451e-11
8.900000	1.111393e+00	1.111393e+00	9.822476e-11
9.200000	1.111586e+00	1.111586e+00	1.585221e-10
9.500000	1.111877e+00	1.111877e+00	2.432075e-10
9.800000	1.112326e+00	1.112326e+00	3.646636e-10
10.100000	1.113025e+00	1.113025e+00	5.401055e-10
10.400000	1.114118e+00	1.114118e+00	7.935190e-10
10.700000	1.115831e+00	1.115831e+00	1.158100e-09
11.000000	1.118516e+00	1.118516e+00	1.679178e-09
11.300000	1.122725e+00	1.122725e+00	2.417426e-09
11.600000	1.129326e+00	1.129326e+00	3.451748e-09
11.900000	1.139679e+00	1.139679e+00	4.880472e-09
12.200000	1.155915e+00	1.155915e+00	6.817980e-09
12.500000	1.181377e+00	1.181377e+00	9.381222e-09
12.800000	1.221310e+00	1.221310e+00	1.265572e-08
13.100000	1.283938e+00	1.283938e+00	1.662209e-08
13.400000	1.382158e+00	1.382157e+00	2.100914e-08
13.700000	1.536197e+00	1.536196e+00	2.501401e-08
14.000000	1.777778e+00	1.777778e+00	2.678539e-08

Таблица 4. 40 кубических КЭ

Максимальная абсолютная погрешность $1.594959\text{e-}06$ и $2.678539\text{e-}08$ 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.

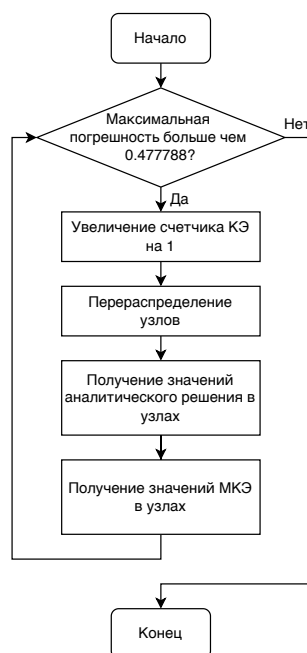


Рис. 6. Алгоритм нахождения количества КЭ, заданную точность

Реализовав данный алгоритм с начальным количеством КЭ=20 и увеличивая счетчик всегда на 1 получаем необходимое количество КЭ, равное 2421 КЭ.

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 = 2.0;
8 double X_END = 14.0;
9 size_t ELEMS_NUM = 20;
  
```

```

10 double L = (X_END - X_BEGIN) / ELEMS_NUM;
11
12 double a = 4.0, B = 0.0, C = -9.0, D = 10.0, usl_left = 0.0, usl_right = 1.0; //
    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 (2. * (-3. * exp(21. - (3. * x)/2.) - 5. * exp(39. - (3. * x)/2.) - 5. * exp((3.
        * x)/2. - 3.) + 3. * exp((3. * x)/2. + 15.) + 5. + 5. * exp(36)))/(9. * (1. +
        exp(36)));
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.) - C * L / 3., -(a / L) - (B / 2.) - C * L / 6.},
68         { -(a / L) + (B / 2.) - C * L / 6., (a / L) - (B / 2.) - C * L / 3.},
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
86     b.at(0) = usl_left;
87     A.at(0).at(0) = 1;
88     A.at(0).at(1) = 0;
89
90
91
92     b.at(size - 1) = D * L / 2. + a*usl_right;
93
94
95     // Решение полученной СЛАУ методом Гаусса
96     std::vector<double> res = solve_with_gauss(A, b);
97     return res;

```



```

98 }
99
100 std::vector<double> build_cube_solution(size_t elems_num) {
101     double L = (X_END - X_BEGIN) / elems_num;
102     size_t size = elems_num + 1;
103     std::vector< std::vector<double> > A(size, std::vector<double>(size));
104     std::vector<double> b(size);
105
106     // Локальная матрица жесткости для кубического КЭ
107     std::vector< std::vector<double> > local_matrix = {
108         { 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},
109         { -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},
110         { 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},
111         { -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}
112     };
113
114     // Локальный вектор нагрузок (дополнительные слагаемые для первого и последнего
        элементов учитываются далее)
115     std::vector<double> local_b = { D * L / 8.0,
116                                     D * 3.0 * L / 8.0,
117                                     D * 3.0 * L / 8.0,
118                                     D * L / 8.0 };
119
120
121     // Производим матричные преобразования для обнуления элементов локальной
        матрицы жесткости, относящихся к внутренним узлам
122     for (size_t i = 1; i < 3; i++) {
123         for (size_t j = 0; j < 4; j++) {
124             if (std::fabs(local_matrix.at(j).at(i)) > EPS && i != j) {
125                 double val = local_matrix.at(j).at(i) / local_matrix.at(i).at(i);
126                 local_b.at(j) -= val * local_b.at(i);
127                 for (size_t k = 0; k < 4; k++) {
128                     local_matrix.at(j).at(k) -= val * local_matrix.at(i).at(k);
129                 }
130             }
131             continue;
132         }
133     }

```

```

134
135
136 // Исключаем внутренние узлы из рассмотрения
137 std::vector< std::vector<double> > local_matrix_mod = { { local_matrix.at(0).at(0),
138 local_matrix.at(0).at(3) },
{ local_matrix.at(3).at(0),
local_matrix.at(3).at(3)
} };
139
140 std::vector<double> local_b_mod = { local_b.at(0),
141 local_b.at(3)
142 };
143
144 // Ансамблирование и получение глобальной матрицы жесткости для кубического КЭ
145 for (size_t i = 0; i < elems_num; i++) {
146     for (size_t j = 0; j < 2; j++) {
147         for (size_t k = 0; k < 2; k++) {
148             A.at(i + j).at(i + k) += local_matrix_mod.at(j).at(k);
149         }
150     }
151 }
152
153 for (size_t i = 0; i < elems_num; i++) {
154     b.at(i) += local_b_mod.at(0);
155     b.at(i+1) += local_b_mod.at(1);
156 }
157
158 // Учет ГУ
159 if (0 == 1) {
160     b.at(0) = local_b_mod.at(0) - a * usl_left;
161 } else {
162     b.at(0) = usl_left;
163     A.at(0).at(0) = 1.;
164     A.at(0).at(1) = 0.;
165 }
166
167 if (1 == 1) {
168     b.at(size - 1) = local_b_mod.at(1) + a * usl_right;
169 } else {
170     b.at(size - 1) = usl_right;
171     A.at(size - 1).at(size - 1) = 1.;
172     A.at(size - 1).at(size - 2) = 0.;
173 }
174
175 // Решение полученной СЛАУ методом Гаусса
176 std::vector<double> res = solve_with_gauss(A, b);
177 return res;

```

```

177 }
178
179 double calc_abs_error(const std::vector<double>& y_real, const std::vector<double>&
    y) {
180     double max_err = 0.0;
181     for (size_t i = 0; i < y_real.size(); i++) {
182         double err = std::fabs(y_real.at(i) - y.at(i));
183         if (err > max_err) {
184             max_err = err;
185         }
186     }
187     return max_err;
188 }
189
190 int main() {
191
192     std::vector<double> x(ELEMS_NUM + 1);
193     for (size_t i = 0; i < x.size(); i++) {
194         x.at(i) = X_BEGIN + i * L;
195     }
196     size_t x_size = x.size();
197
198     std::vector<double> y;
199     if (true) {
200         y = build_linear_solution(ELEMS_NUM);
201     } else {
202         y = build_cube_solution(ELEMS_NUM);
203     }
204     std::vector<double> y_real = build_analytical_solution(x);
205
206     FILE* gp;
207     FILE* ab;
208     FILE* pgr;
209     FILE* tab;
210
211
212     gp = fopen("res/labs/text/graph/lin_20.txt", "w");
213     ab = fopen("res/labs/text/graph/abs.txt", "w");
214     for (size_t i = 0; i < x_size; i++) {
215         fprintf(ab, "%lf %lf\n", x.at(i), y_real.at(i));
216         printf("%lf %lf\n", x.at(i), y_real.at(i));
217     }
218     pgr = fopen("res/labs/text/pgr/lin_20.txt", "w");
219     tab = fopen("res/labs/text/tab/lin_20.txt", "w");
220
221

```

```

222
223     for (size_t i = 0; i < x.size()-1; i++) {
224         fprintf(tab, "%f & %le & %le & %le \\\n", x.at(i), y_real.at(i), y.at(i),
                std::fabs(y_real.at(i) - y.at(i)));
225     }
226     fprintf(tab, "%f & %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)));
227
228     for (size_t i = 0; i < x_size; i++) {
229         fprintf(gp, "%lf %lf\n", x.at(i), y.at(i));
230     }
231
232     fprintf(pgr, "%e", calc_abs_error(y_real, y));
233
234     fclose(gp);
235     fclose(ab);
236     fclose(pgr);
237     fclose(tab);
238
239     return 0;
240 }

```

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 РК6-726, Стройков М.Н.

2022, осенний семестр