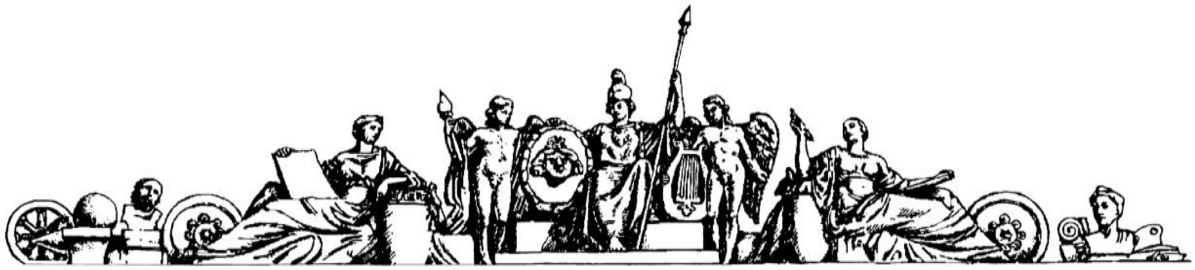


Министерство образования и науки Российской Федерации федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)»

Факультет «Робототехники и комплексной автоматизации»

Кафедра «Системы автоматизированного проектирования»

ОТЧЕТ

по лабораторной работе №2

по курсу «Модели и методы анализа проектных решений»

Вариант №5

Студент: Бурча Артём РК6-71

Преподаватель: к.т.н., доцент каф. РК6

Трудоношин В.А.

Москва, 2017 г.

Оглавление

Задание.

Методом конечных элементов решить уравнение:

$$3\frac{d^2u}{dx^2} - 5u + 10 = 0$$

при следующих граничных условиях:

u(2)=0 u(15)=10

количество конечных элементов для

первого расчёта-20

второго расчёта-40

Сравнить результат с аналитическим решением,

оценить максимальную погрешность.

Аналитическое решение.

$$3\frac{d^2u}{dx^2} - 5u + 10 = 0$$

при следующих граничных условиях:

u(2)=0 u(15)=10

Для получения аналитического решения воспользуемся программным комплексом Maple 2017.

restart;

$deq := 3\cdot diff(u(x), x\$2) - 5\cdot u(x) + 10 = 0;$

$$deq := 3\left(\frac{d^2}{dx^2}\,u(x)\right) - 5\,u(x) + 10 = 0$$

$cond := u(2) = 0, u(15) = 10;$

$$cond := u(2) = 0,\ u(15) = 10$$

$dsolve(deq, u(x));$

общее решение ДУ:

$$u(x) = e^{\frac{\sqrt{15}\,x}{3}}\,\_C2 + e^{-\frac{\sqrt{15}\,x}{3}}\,\_C1 + 2$$

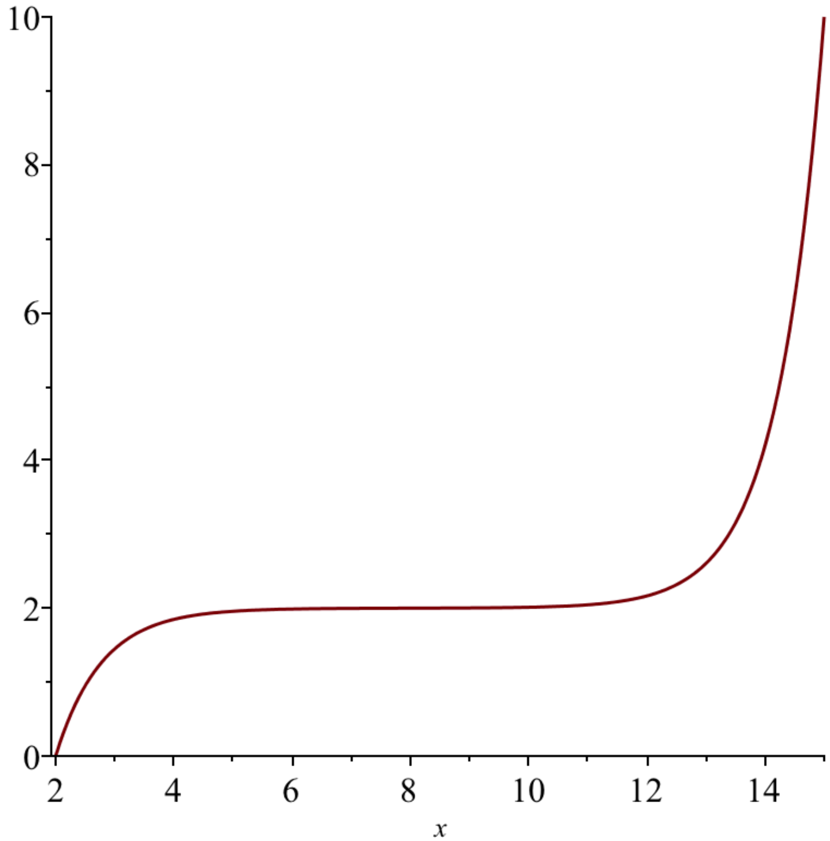
$dsolve(\{deq, cond\}, u(x));$

частное решение ДУ:

$$u(x) = -\frac{2\,e^{\frac{\sqrt{15}\,x}{3}}\left(e^{-5\sqrt{15}} + 4\,e^{-\frac{2\sqrt{15}}{3}}\right)}{e^{\frac{2\sqrt{15}}{3}}\,e^{-5\sqrt{15}} - e^{-\frac{2\sqrt{15}}{3}}\,e^{5\sqrt{15}}} + \frac{2\,e^{-\frac{\sqrt{15}\,x}{3}}\left(4\,e^{\frac{2\sqrt{15}}{3}} + e^{5\sqrt{15}}\right)}{e^{\frac{2\sqrt{15}}{3}}\,e^{-5\sqrt{15}} - e^{-\frac{2\sqrt{15}}{3}}\,e^{5\sqrt{15}}} + 2$$

построим график:

$y1 := rhs(\%) : plot(y1, x = 2..15, thickness = 1);$



упрощенная запись частного решения ДУ:

$$u(x)=\frac{2\,e^{-\frac{\sqrt{15}\,(x+2)}{3}}\left(e^{\frac{2\sqrt{15}\,x}{3}}-e^{\frac{\sqrt{15}\,(x+2)}{3}}+e^{\frac{\sqrt{15}\,(x+28)}{3}}+4\,e^{\frac{\sqrt{15}\,(2x+13)}{3}}-4\,e^{\frac{17\sqrt{15}}{3}}-e^{10\sqrt{15}}\right)}{e^{\frac{26\sqrt{15}}{3}}-1}$$

Именно эта запись будет использована в программе для сравнения аналитического решения с решением методом конечных элементов.

Решение методом конечных элементов.

Линейный конечный элемент.

В качестве функции формы возьмем функцию вида $u=a_0+a_1\cdot x$

Из граничных условий (x = 0, x = L) для данной функции получим следующую систему уравнений с двумя неизвестными a_0 и a_1:

$$\begin{cases}U_i=a_0\\ U_j=a_0+a_1\cdot L\end{cases}$$

Откуда $a_1=\dfrac{U_j-U_i}{L}$, а $u=U_i+\dfrac{U_j-U_i}{L}x=(1-\dfrac{x}{L})U_i+\dfrac{x}{L}U_j$

или же $u=N_l\cdot U$, где $N_l=\left[(1-\dfrac{x}{L}),\dfrac{x}{L}\right]$, $U=\begin{bmatrix}U_i\\ U_j\end{bmatrix}$.

Решим уравнение $\int_0^L N_l^T\,(3\,\frac{d^2u}{dx^2}-5u+10)=0$. Разобьем исходный интеграл на 3 части и решим по отдельности.

1 интеграл:

$$3\int_0^L N_l^T \frac{d^2u}{dx^2}dx = 3\int_0^L N_l^T d\left(\frac{du}{dx}\right) = 3N_l^T \frac{du}{dx}\Bigg|_0^L - 3\int_0^L \frac{du}{dx} dN_l^T =$$

$$3\begin{bmatrix} 1-\frac{x}{L} \\ \frac{x}{L} \end{bmatrix} \frac{du}{dx}\Bigg|_0^L - 3\int_o^L \frac{d}{dx}\begin{bmatrix} 1-\frac{x}{L} \\ \frac{x}{L} \end{bmatrix}\frac{du}{dx}dx = 3\begin{bmatrix} -\frac{du}{dx}\Big|0 \\ \frac{du}{dx}\Big|L \end{bmatrix} - 3\int_0^L \begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix}\begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}dx =$$

$$= 3\begin{bmatrix} -\frac{du}{dx}\Big|0 \\ \frac{du}{dx}\Big|L \end{bmatrix} - 3\int_0^L \begin{bmatrix} \frac{1}{L^2} & -\frac{1}{L^2} \\ -\frac{1}{L^2} & \frac{1}{L^2} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}dx = 3\begin{bmatrix} -\frac{du}{dx}\Big|0 \\ \frac{du}{dx}\Big|L \end{bmatrix} - 3L\begin{bmatrix} \frac{1}{L^2} & -\frac{1}{L^2} \\ -\frac{1}{L^2} & \frac{1}{L^2} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix} =$$

$$= 3\begin{bmatrix} -\frac{du}{dx}\Big|0 \\ \frac{du}{dx}\Big|L \end{bmatrix} + \begin{bmatrix} -\frac{3}{L} & \frac{3}{L} \\ \frac{3}{L} & -\frac{3}{L} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}$$

2 интеграл:

$$-5\int_0^L N_l^T \cdot u\ dx = -5\int_0^L \begin{bmatrix} 1-\frac{x}{L} \\ \frac{x}{L} \end{bmatrix}\begin{bmatrix} 1-\frac{x}{L} & \frac{x}{L} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}dx = -5\int_0^L \begin{bmatrix} \left(1-\frac{x}{L}\right)^2 & \frac{x}{L}-\frac{x^2}{L^2} \\ \frac{x}{L}-\frac{x^2}{L^2} & \frac{x^2}{L^2} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}dx =$$

$$= -5\int_0^L \begin{bmatrix} 1-\frac{2x}{L}+\frac{x^2}{L^2} & \frac{x}{L}-\frac{x^2}{L^2} \\ \frac{x}{L}-\frac{x^2}{L^2} & \frac{x^2}{L^2} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}dx = -5\int_0^L \begin{bmatrix} x-\frac{x^2}{L}+\frac{x^3}{3L^2} & \frac{x^2}{2L}-\frac{x^3}{3L^2} \\ \frac{x^2}{2L}-\frac{x^3}{3L^2} & \frac{x^3}{3L^2} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}\Bigg|_0^L =$$

$$= -5\begin{bmatrix} \frac{L}{3} & \frac{L}{2}-\frac{L}{3} \\ \frac{L}{2}-\frac{L}{3} & \frac{L}{3} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix} = \begin{bmatrix} -\frac{5L}{3} & -\frac{5L}{6} \\ -\frac{5L}{6} & -\frac{5L}{3} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix}$$

3 интеграл:

$$10\int_0^L \mathrm{Nt}(x, L)\,dx \rightarrow \begin{pmatrix} 5\cdot L \\ 5\cdot L \end{pmatrix}$$

Просуммировав полученные результаты, получим следующий вид данного уравнения:

$$3\begin{bmatrix} -\dfrac{du}{dx}\Big|0 \\ \dfrac{du}{dx}\Big|L \end{bmatrix} + \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

или

$$3\begin{bmatrix} -\dfrac{du}{dx}\Big|0 \\ \dfrac{du}{dx}\Big|L \end{bmatrix} + \begin{bmatrix} -\dfrac{3}{L}-\dfrac{5L}{3} & \dfrac{3}{L}-\dfrac{5L}{6} \\ \dfrac{3}{L}-\dfrac{5L}{6} & -\dfrac{3}{L}-\dfrac{5L}{3} \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix} = \begin{bmatrix} -\dfrac{5}{L} \\ -\dfrac{5}{L} \end{bmatrix}$$

В результате, при разбиении объекта на n конечных элементов и при подстановке граничных условий мы будем иметь систему из n уравнений с n неизвестными:

$$3\begin{bmatrix} -\dfrac{du}{dx}\Big|_2 \\ 0 \\ \ldots \\ 0 \\ \dfrac{du}{dx}\Big|_{15} \end{bmatrix} + \begin{bmatrix} a & b & 0 & \ldots & \ldots & \ldots & 0 \\ c & d+a & b & 0 & \ldots & \ldots & 0 \\ 0 & c & d+a & b & 0 & \ldots & 0 \\ \ldots & \ldots & c & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & c & d+a & b & 0 \\ 0 & \ldots & \ldots & 0 & c & d+a & b \\ 0 & \ldots & \ldots & \ldots & 0 & c & d \end{bmatrix}\begin{bmatrix} U\big|_2 \\ U_3 \\ \\ \ldots \\ \\ U_{n-1} \\ U\big|_{15} \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta+\alpha \\ \ldots \\ \ldots \\ \ldots \\ \beta+\alpha \\ \beta \end{bmatrix}$$

Нам известны граничные условия для U(2) и U(15), поэтому мы можем перенести их из матрицы жесткости в вектор нагрузок (удаляем соответствующие столбцы из матрицы жесткости и переносим их в вектор нагрузок). Значения производных $-\frac{du}{dx}\Big|_2$ и $\frac{du}{dx}\Big|_{15}$ нам не известны, поэтому перенесем их в вектор неизвестных, дополнив матрицу жесткости соответствующими столбцами. Модифицированная СЛАУ примет вид:

$$
\begin{bmatrix}
-3 & b & 0 & \dots & \dots & \dots & 0 \\
0 & d+a & b & 0 & \dots & \dots & 0 \\
0 & c & d+a & b & 0 & \dots & 0 \\
\dots & \dots & c & \dots & \dots & \dots & \dots \\
0 & \dots & 0 & c & d+a & b & 0 \\
0 & \dots & \dots & 0 & c & d+a & 0 \\
0 & \dots & \dots & \dots & 0 & c & 3
\end{bmatrix}
\begin{bmatrix}
\left.\frac{du}{dx}\right|_{2} \\
U_3 \\
\dots \\
U_{n-1} \\
\left.\frac{du}{dx}\right|_{15}
\end{bmatrix}
=
\begin{bmatrix}
\alpha - a\cdot U|_2 \\
\beta + \alpha - c\cdot U|_2 \\
\dots \\
\dots \\
\dots \\
\beta + \alpha - b\cdot U|_2 \\
\beta - d\cdot U|_2
\end{bmatrix}
$$

Кубический конечный элемент.

В качестве функции формы возьмем функцию вида:

$$u = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$$

Из граничных условий (x = 0, x = $\frac{L}{3}$, x = $\frac{2L}{3}$, x = L) для данной функции

получим следующую систему уравнений с четырьмя неизвестными a_0, a_1, a_2, a_3.

Далее большинство расчетов будет проводиться в MathCAD.

$$
\begin{cases}
U_i = a_0 \\
U_k = a_0 + a_1 \cdot \frac{L}{3} + a_2 \cdot \frac{L^2}{9} + a_3 \cdot \frac{L^3}{27} \\
U_l = a_0 + a_1 \cdot \frac{2L}{3} + a_2 \cdot \frac{4L^2}{9} + a_3 \cdot \frac{8L^3}{27} \\
U_j = a_0 + a_1 \cdot L + a_2 \cdot L^2 + a_3 \cdot L^3
\end{cases}
$$

Решаем систему:

$a_0 = U_i$,

$$a_1 = \frac{2U_j - 11U_i + 18U_k - 9U_l}{2L},$$

$$a_2 = -\frac{9U_j - 18U_i + 45U_k - 36U_l}{2L^2},$$

$$a_3 = -\frac{9U_i - 9U_j - 27U_k + 27U_l}{2L^3}.$$

Приводим подобные слагаемые:

$u = b_0 \cdot U_k + b_1 \cdot U_l + b_2 \cdot U_i + b_3 \cdot U_j$, где

$$b_0 = \frac{18L^2 \cdot x - 45L \cdot x^2 + 27 \cdot x^3}{2L^3},$$

$$b_1 = -\frac{9L^2 \cdot x - 36L \cdot x^2 + 27 \cdot x^3}{2L^3},$$

$$b_2 = \frac{2L^3 - 11L^2 \cdot x + 18L \cdot x^2 - 9 \cdot x^3}{2L^3},$$

$$b_3 = \frac{2L^2 \cdot x - 9L \cdot x^2 + 9 \cdot x^3}{2L^3}.$$

Иными словами $u = N_l \cdot U$, где $N_l = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix}$, $U = \begin{bmatrix} U_k \\ U_l \\ U_i \\ U_j \end{bmatrix}$.

Решим уравнение $\int_0^L N_l^T \left(3\frac{d^2u}{dx^2} - 5u + 10 \right) = 0$. Разобьем исходный интеграл на 3 части и решим по отдельности.

1 интеграл:

$$3\int_0^L N_l^T \frac{d^2u}{dx^2}dx = 3\int_0^L N_l^T d\left(\frac{du}{dx}\right) = 3N_l^T \frac{du}{dx}\Bigg|_0^L - 3\int_0^L \frac{du}{dx} dN_l^T =$$

$$= 3\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}\frac{du}{dx}\Bigg|_0^L - 3\int_0^L \frac{dN_l^T}{dx}\cdot\frac{d}{dx}\left(N_l U\right)\, dx =$$

$$= 3\begin{bmatrix} 0 \\ 0 \\ -\left.\dfrac{du}{dx}\right|_0 \\ \left.\dfrac{du}{dx}\right|_L \end{bmatrix} - 3\int_0^L \frac{d}{dx}\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}\frac{d}{dx}\begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix}\begin{bmatrix} U_k \\ U_l \\ U_i \\ U_j \end{bmatrix}dx =$$

$$= 3\begin{bmatrix} 0 \\ 0 \\ -\left.\dfrac{du}{dx}\right|_0 \\ \left.\dfrac{du}{dx}\right|_L \end{bmatrix} + \begin{array}{|c|c|c|c|} \hline -\dfrac{162}{5L} & \dfrac{891}{40L} & \dfrac{567}{40L} & -\dfrac{81}{20L} \\ \hline \dfrac{891}{40L} & -\dfrac{162}{5L} & -\dfrac{81}{20L} & \dfrac{567}{40l} \\ \hline \dfrac{567}{40L} & -\dfrac{81}{20L} & -\dfrac{111}{10L} & \dfrac{39}{40L} \\ \hline -\dfrac{81}{20L} & \dfrac{567}{40l} & \dfrac{39}{40L} & -\dfrac{111}{10L} \\ \hline \end{array}\begin{bmatrix} U_k \\ U_l \\ U_i \\ U_j \end{bmatrix}$$

2 интеграл:

$$-5\int_0^L N_l^T\cdot u\ dx = -5\int_0^L \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}\begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix}\begin{bmatrix} U_k \\ U_l \\ U_i \\ U_j \end{bmatrix}dx =$$

$$= \begin{array}{|c|c|c|c|} \hline -\dfrac{27L}{14} & \dfrac{27L}{112} & -\dfrac{33L}{112} & \dfrac{3L}{28} \\ \hline \dfrac{27L}{112} & -\dfrac{27L}{14} & \dfrac{3L}{28} & -\dfrac{33L}{112} \\ \hline -\dfrac{33L}{112} & \dfrac{3L}{28} & -\dfrac{8L}{21} & -\dfrac{19L}{336} \\ \hline \dfrac{3L}{28} & -\dfrac{33L}{112} & -\dfrac{19L}{336} & -\dfrac{8L}{21} \\ \hline \end{array}\begin{bmatrix} U_k \\ U_l \\ U_i \\ U_j \end{bmatrix}$$

3 интеграл:

$$10\int_0^L Nt(x,L)\,dx \rightarrow \begin{pmatrix} \frac{15\cdot L}{4} \\ \frac{15\cdot L}{4} \\ \frac{5\cdot L}{4} \\ \frac{5\cdot L}{4} \end{pmatrix}$$

Просуммируем полученные результаты. В ансамблировании участвуют только крайние узлы, поэтому с помощью прямого хода метода Гаусса обнуляем в уравнениях этих узлов коэффициенты перед U_l и U_k.

$$3\begin{bmatrix} 0 \\ 0 \\ -\left.\frac{du}{dx}\right|_0 \\ \left.\frac{du}{dx}\right|_L \end{bmatrix} + \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \\ 0 & l_1 & l_2 & l_3 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix}\begin{bmatrix} U_k \\ U_l \\ U_i \\ U_j \end{bmatrix} = \begin{bmatrix} k_5 \\ l_4 \\ \alpha \\ \beta \end{bmatrix}$$

Матрица, полученная из матрицы жесткости и вектора нагрузок, после прямого хода Гаусса с обнуленными коэффициентами перед U_l и U_k:

$$\begin{bmatrix} -\frac{27\cdot L}{14}-\frac{162}{5\cdot L} & \frac{27\cdot(5\cdot L^2+462)}{560\cdot L} & \frac{567}{40\cdot L}-\frac{33\cdot L}{112} & \frac{3\cdot(5\cdot L^2-189)}{140\cdot L} & \frac{15\cdot L}{4} \\ 0 & -\frac{243\cdot(5\cdot L^2+126)\cdot(L^2+6)}{128\cdot L\cdot(5\cdot L^2+84)} & \frac{9\cdot(5\cdot L^4-276\cdot L^2+6804)}{128\cdot L\cdot(5\cdot L^2+84)} & \frac{9\cdot(177\cdot L^2-5\cdot L^4+3402)}{32\cdot L\cdot(5\cdot L^2+84)} & \frac{135\cdot L\cdot(5\cdot L^2+126)}{32\cdot(5\cdot L^2+84)} \\ 0 & 0 & -\frac{\frac{5\cdot L^6}{3}+135\cdot L^4+1728\cdot L^2+2268}{L\cdot(5\cdot L^2+126)\cdot(L^2+6)} & -\frac{5\cdot L^6-90\cdot L^4+1944\cdot L^2-27216}{12\cdot L\cdot(5\cdot L^2+126)\cdot(L^2+6)} & \frac{5\cdot L\cdot(L^2+36)}{6\cdot(L^2+6)} \\ 0 & 0 & -\frac{5\cdot L^6-90\cdot L^4+1944\cdot L^2-27216}{12\cdot L\cdot(5\cdot L^2+126)\cdot(L^2+6)} & -\frac{\frac{5\cdot L^6}{3}+135\cdot L^4+1728\cdot L^2+2268}{L\cdot(5\cdot L^2+126)\cdot(L^2+6)} & \frac{5\cdot L\cdot(L^2+36)}{6\cdot(L^2+6)} \end{bmatrix}$$

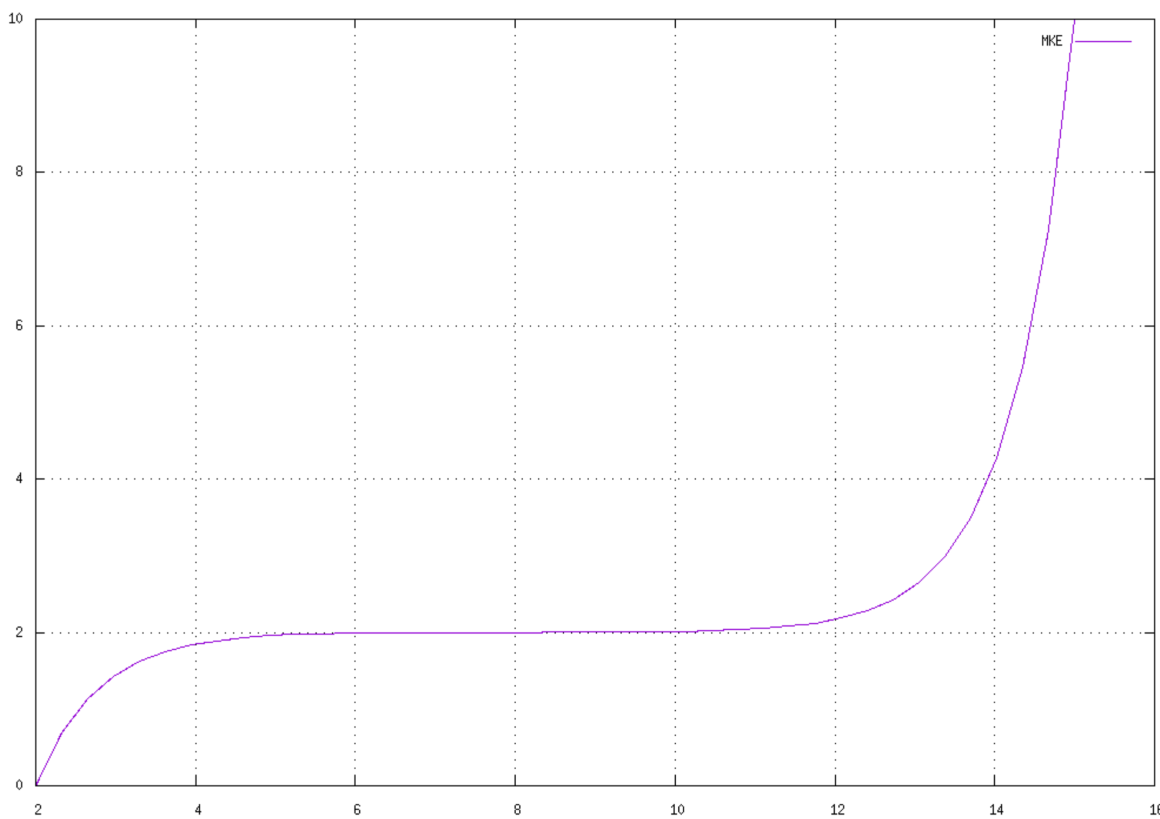
Таким образом, переменные U_k и U_l можно исключить. Получим:

$$3\begin{bmatrix} -\left.\dfrac{du}{dx}\right|_0 \\ \left.\dfrac{du}{dx}\right|_L \end{bmatrix} + \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} U_i \\ U_j \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Осталось провести ансамблирование аналогично случаю с линейным конечным элементом и найти значения неизвестных фазовых переменных методом Гаусса.

Результат работы программы.

Программа реализует метод конечных элементов для решения данного уравнения с заданными граничными условиями, видом конечного элемента и количеством конечных элементов. Результатом работы программы является следующий график, построенный в gnuplot 5.2 patchlevel 2:



Ниже приведена максимальная погрешность для 20 и 40 конечных элементов с использованием линейной и кубической функции формы.

	Линейная	Кубическая
20	0.0912163	5.14831e-06
40	0.0216382	7.89793e-08

Листинг программы.

Заголовочный файл «classdefinition.h»

```
#include <vector>

using namespace std;

class GlobalMatrix{
private:
    double a;
    double b;
    double c;
    double d;
    double alpha;
    double beta;
    int nodeAmount;
    double elementLength;
    double **GM;
    vector<double> solution;
    double special;

public:
    GlobalMatrix();
    GlobalMatrix(double aa, double bb, double cc, double dd, double inalpha,
double inbeta,int elementAmount,double elementLength,double a);
    ~GlobalMatrix();
    void InitMatrix();
    void Ensembling();
    void Gauss();
    double analyticSolution(double x);
    void Compare();
    void PrintMatrix();
    void toFile();
};
```

Файл с определениями методов класса «methods.cpp»

```
#include "classdefinition.h"
#include <iostream>
#include <cmath>
#include <fstream>
```

```
using namespace std;

GlobalMatrix::GlobalMatrix() {cout<<"NO PARAMETERS INPUT IN CONSTRUCTOR!!!";
exit(0);}

GlobalMatrix::GlobalMatrix(double aa, double bb, double cc, double dd,
                           double inalpha, double inbeta, int elem, double L,
double a):
a(aa),b(bb),c(cc),d(dd),alpha(inalpha),beta(inbeta),nodeAmount(elem+1),elementLe
ngth(L),special(a)
{
    GM = new double* [nodeAmount];
    for (int i = 0; i < nodeAmount; i++) {
        GM[i] = new double[nodeAmount + 1];
    }
    InitMatrix();
}

void GlobalMatrix::InitMatrix() {
        for (int i = 0; i < nodeAmount; i++) {
            for (int j = 0; j < nodeAmount +1 ; j++)
                GM[i][j] = 0;
        }
}

void GlobalMatrix::PrintMatrix() {
    cout.precision(3);
    cout.setf(ios::left);
    cout.width(7);
    for (int i = 0; i < nodeAmount; i++) {
        for (int j = 0; j < nodeAmount +1 ; j++)
            cout<<GM[i][j]<<"\t";
        cout<<endl;
    }
}

void GlobalMatrix::Ensembling() {
   // double x=2.0;
    for (int i=0; i<nodeAmount-1; i++){

        GM[i][i]     += a;
        GM[i][i+1]   += b;
        GM[i+1][i]   += c;
        GM[i+1][i+1] += d;
        GM[i][nodeAmount]   += alpha;
        GM[i+1][nodeAmount] += beta;
    }

    GM[0][0] = -3.0;
    GM[1][0] = 0.0;
    GM[nodeAmount-2][nodeAmount-1]=0.0;
    GM[nodeAmount-1][nodeAmount-1]=3.0;

    //GM[0][nodeAmount]-=a*0; //u(2)=0; GM[1][nodeAmount] doesn't change too
    GM[nodeAmount-2][nodeAmount] -=b*10.0; //u(15) = 10;
    GM[nodeAmount-1][nodeAmount] -=d*10.0;
}
```

```
void GlobalMatrix::Gauss() {
        int      i, j, k, maxInd;
        double    max;
        double    tmp;
        double    eps = 0.00001;
        /*Direct*/
        for(i = 0; i < nodeAmount; i++) {
            max = fabs(GM[i][i]);
            maxInd = i;
            // Search max diag element
            for(k = i + 1; k < nodeAmount; k++) {
                if(fabs(GM[k][i]) > max) {
                    maxInd = k;
                    max = fabs(GM[k][i]);
                }
            }
            // if max is zero
            if(max < eps) {
                puts("Can'GM solve system of equations: zero column");
                exit(5);
            }
            // Swap current and max lines
            for(j = 0; j < nodeAmount + 1; j++) {
                tmp = GM[maxInd][j];
                GM[maxInd][j] = GM[i][j];
                GM[i][j] = tmp;
            }
            // Divide each element in diag line
            tmp = GM[i][i];
            for(j = i; j < nodeAmount + 1; j++) {
                GM[i][j] /= tmp;
            }
            // Subtraction multiple first line from each line
            for(k = i + 1; k < nodeAmount; k++) {
                tmp = GM[k][i];
                for(j = i; j < nodeAmount + 1; j++) {
                    GM[k][j] -= GM[i][j] * tmp;
                }
            }
        }
        // Reverse
        for(i = nodeAmount-1; i >=0; i--) {
            double tmp = GM[i][nodeAmount];
            solution.push_back(tmp);
            for(j = 0; j < i; j++) {
                GM[j][nodeAmount] -= GM[j][i] * tmp;
            }
        }
}

double GlobalMatrix::analyticSolution(double x) {
    double u;
    u = (-1.0/(exp(26.0*sqrt(15.0)/3.0) - 1.0)) *
            (2.0*(-4.0*exp((x-2.0)*sqrt(15.0))
                - exp(-19.0*sqrt(15.0)/3.0 + sqrt(15.0)*x)
                - exp(3.0*sqrt(15.0)+2.0/3.0*sqrt(15.0)*x)
                + exp((2.0*x - 17.0)*sqrt(15.0)/3.0)
```

```
                + exp((11.0+x)*sqrt(15.0)/3.0)
                + 4.0*exp((x-2.0)*sqrt(15.0)/3.0))
             *exp((-2.0*x+17.0)*sqrt(15.0)/3.0));
    return u;
}

void GlobalMatrix::Compare() {
    cout.precision(6);
    ofstream graph("../data.txt");
    graph<<2.0<<"\t"<<0.0<<endl;
    double currentX = 2.0;
    double maxDiff = 0.0;
    double difference;
    double currentAnalytic;
    cout<<"result"<<"\t\t"<<"analytic"<<"\t\t"<<"difference"<<endl;
    for (int i=2; i<nodeAmount; i++)
    {
        currentX += elementLength;
        currentAnalytic = analyticSolution(currentX);
        difference = fabs(currentAnalytic - solution[nodeAmount-i]);
        cout<<solution[nodeAmount-
i]<<"\t\t"<<currentAnalytic<<"\t\t"<<difference<<endl;
        maxDiff = (maxDiff < difference) ? difference : maxDiff;
        graph<<currentX<<"\t"<<solution[nodeAmount-i]<<endl;
    }
    graph<<15.0<<"\t"<<10.0<<endl;
    cout<<endl<<"Max difference = "<<maxDiff<<endl;
}
```

Файл с функцией main «main.cpp»

```
#include <iostream>
#include <cmath>
#include "classdefinition.h"
#define LEFTBORDER 2.0
#define RIGHTBORDER 15.0
#define N 40.0
#define QUBIC  //LINEAR, QUBIC
//#define LINEAR

using namespace std;

int main() {

    double A;
    double B;
    double C;
    double D;
    double ALPH;
    double BET;

#ifdef LINEAR
    //LL = Linear length of element
    double start = 6.0;
    double end = 10.0;
    double special = end - start;
    double LL = (RIGHTBORDER - LEFTBORDER)/(N);
```

```
    A = -5.0*LL/3.0 - 3.0/LL;
    B = 3.0/LL - 5.0*LL/6.0;
    C = 3.0/LL - 5.0*LL/6.0;
    D = -5.0*LL/3.0 - 3.0/LL;
    ALPH = -5.0*LL;
    BET = -5.0*LL;
    GlobalMatrix *linear = new GlobalMatrix(A,B,C,D,ALPH,BET,N,LL,special);
    linear->Ensembling();
    linear->PrintMatrix();
    linear->Gauss();
    cout<<endl;
    linear->PrintMatrix();
    cout<<LL<<endl;
    cout<<endl;
    linear->Compare();
#endif

#ifdef QUBIC
    double LQ = (RIGHTBORDER - LEFTBORDER)/N;
     A = - (5.0/3.0 * pow(LQ,6.0) + 135.0*pow(LQ,4.0) + 1728.0*pow(LQ,2.0) +
2268.0) /
             (LQ*(5.0*pow(LQ,2.0)+126.0)*(pow(LQ,2.0)+6.0));
     B = - (5.0*pow(LQ,6.0) - 90.0*pow(LQ,4.0) + 1944.0*pow(LQ,2.0) - 27216.0) /
             (12.0*LQ*(5.0*pow(LQ,2.0)+126.0)*(pow(LQ,2.0)+6.0));
     C =  - (5.0*pow(LQ,6.0) - 90.0*pow(LQ,4.0) + 1944.0*pow(LQ,2.0) - 27216.0)
/
          (12.0*LQ*(5.0*pow(LQ,2.0)+126.0)*(pow(LQ,2.0)+6.0));
     D = - (5.0/3.0 * pow(LQ,6.0) + 135.0*pow(LQ,4.0) + 1728.0*pow(LQ,2.0) +
2268.0) /
         (LQ*(5.0*pow(LQ,2.0)+126.0)*(pow(LQ,2.0)+6.0));
     ALPH = -5.0*LQ*(pow(LQ,2.0) + 36.0)/(6.0*(pow(LQ,2.0)+6.0));
     BET = -5.0*LQ*(pow(LQ,2.0) + 36.0)/(6.0*(pow(LQ,2.0)+6.0));

    GlobalMatrix *quadratic = new GlobalMatrix(A,B,C,D,ALPH,BET,N,LQ,1);
    quadratic->Ensembling();
    cout<<LQ<<endl;
    quadratic->Gauss();
    cout<<endl;
    quadratic->Compare();
#endif
    return 0;
}
```