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

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

<u>ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ</u>

Студент	Никифорова Ирина Андреевна
Группа	РК6-71б
Тип задания	лабораторная работа
Тема работы	метод конечных элементов
Вариант	15

Студент _________________ **<u>Никифорова И. А.</u>**

подпись, дата *фамилия, и.о.*

Преподаватель _________________ **<u>Трудоношин В. А.</u>**

подпись, дата *фамилия, и.о.*

Оценка ___________________________________

Москва, 2019 г.

Оглавление

Задание на лабораторную работу

15.Методом конечных элементов решить уравнение:

$$2\frac{d^2u}{dx^2} - 5\frac{du}{dx} + u = 0 \qquad (1)$$

при следующих граничных условиях: u(1)=10, du/dx(x=0)=-6

Количество конечных элементов для

первого расчёта - 20

второго расчёта - 40

Сравнить результат с аналитическим решением, оценить максимальную погрешность.

Аналитическое решение ДУ

ДУ (1) является однородным ДУ второго порядка с постоянными коэффициентами. Для его решения было составлено характеристическое уравнение и найдено его решение:

$2\lambda^2 - 5\lambda + 1 = 0,\ \lambda_1 = \frac{5+\sqrt{17}}{4}\ ,\ \lambda_2 = \frac{5-\sqrt{17}}{4}$.

Характеристическое уравнение имеет 2 действительных решения, поэтому общим решением ДУ является следующее выражение:

$$u = C_1 e^{\lambda_1 x} + C_2 e^{\lambda_2 x}.$$

Первая производная решения будет иметь вид:

$$\frac{du}{dt} = e^{\lambda_1 x} + \lambda_1 C_1 e^{\lambda_1 x} + e^{\lambda_2 x} + \lambda_2 C_2 e^{\lambda_2 x}.$$

Из заданных граничных условий u(1)=10, du/dx(x=0)=-6 с помощью метода Гаусса можно найти константы C_1 и C_2 . Матричное уравнение будет иметь вид:

$$\begin{bmatrix} e^{\lambda_1} & e^{\lambda_2} \\ \lambda_1 & \lambda_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 10 \\ -8 \end{bmatrix}$$

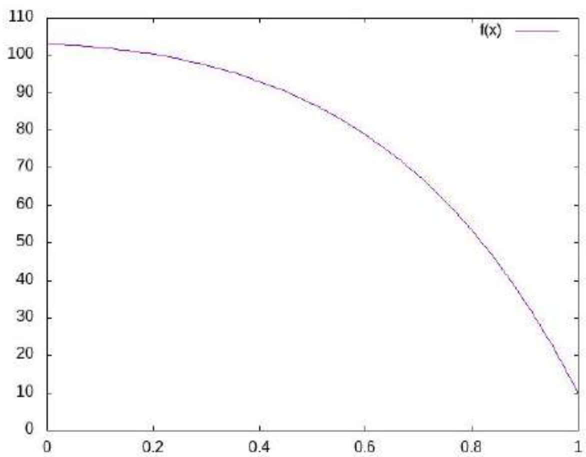

Рис. 1. График функции *u(x)*

Была сделана подстановка $\lambda_1 = \frac{5+\sqrt{17}}{4}$, $\lambda_2 = \frac{5-\sqrt{17}}{4}$, в результате чего, после использования метода Гаусса было получено решение: C_1 = -13.9, C_2 = 117.

Данная процедура была реализована с помощью кода в файле *analytics.cpp*. Далее был получен график аналитической функции (рис. 1) с помощью утилиты *gnuplot* и написанного для нее скрипта *analytics.gnu*.

Теоретическое решение

Исходный участок оси *Ox* [0;1] разбивается на *N* (= 20 для первого и = 40 для второго расчетов) конечных элементов. В каждом из узлов имеются значения функции, обозначаемые соответственно номеру узла: U_0 , U_1 … U_{N-1} .

Задается функция формы КЭ. Сначала была реализована линейная функция формы КЭ.

1. Получение линейной функции формы

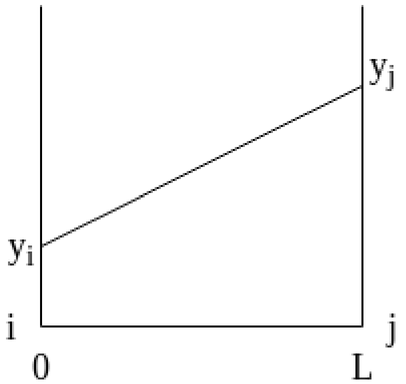

Рис. 2. Конечный элемент с линейной функцией формы.

$$u = a_0 + a_1 x$$

Необходимо найти коэффициенты a_0 и a_1:

$$a_0 = u_i ,$$
$$u_j = a_0 + a_1 L , \ a_1 = \frac{u_j - u_i}{L}$$

$$u = u_i + \frac{u_j - u_i}{L} x = [1 - \tfrac{x}{L}, \ \tfrac{x}{L}] \begin{bmatrix} u_i \\ u_j \end{bmatrix} \quad (2)$$

Отсюда линейная функция формы получается равна:

$$N_l = [1 - \tfrac{x}{L}, \ \tfrac{x}{L}] \quad (3)$$

2. Нахождение уравнения для линейной функции формы

Чтобы получить решение необходимо рассмотреть следующее уравнение:

$$\int_0^L N_l^T \left(2\frac{d^2u}{dx^2} - 5\frac{du}{dx} + u\right) dx = 0 \quad (4)$$

Разделим интеграл из уравнения (4) на слагаемые и найдем каждое из них отдельно:

$$\int_0^L N_l^T (2\frac{d^2u}{dx^2} - 5\frac{du}{dx} + u)\, dx \ =$$

$$= \ 2\int_0^L N_l^T \frac{d^2u}{dx^2}\, dx \ - \tag{5}$$

$$- 5\int_0^L N_l^T \frac{du}{dx}\, dx \ + \tag{6}$$

$$+ \int_0^L N_l^T u\, dx \tag{7}$$

Рассмотрим слагаемое (5). Для его решения воспользуемся формулой интегрирования по частям для матриц в определенном интеграле:

$$\int_{x_0}^{x_1} F(x)G`(x)dx = F(x_1)G(x_1) - F(x_0)G(x_0) - \int_{x_0}^{x_1} F`(x)G(x)dx \tag{8}$$

Здесь $x_0 = 0, x_1 = L,\ F(x) = N_l^T\ ,\ G`(x) = \frac{d^2u}{dx^2}$ => $G(x) = \frac{du}{dx}$

из формулы (2):

$$u \ = \ [1 - \tfrac{x}{L},\ \tfrac{x}{L}] \begin{bmatrix} u_i \\ u_j \end{bmatrix}$$, поэтому $$\frac{du}{dx} \ = \ [-\tfrac{1}{L},\ \tfrac{1}{L}] \begin{bmatrix} u_i \\ u_j \end{bmatrix} .$$

Найдем также *F(x)G(x)*:

$$F(x)G(x) \ = \begin{bmatrix} 1 - \frac{x}{L} \\ \frac{x}{L} \end{bmatrix} [-\tfrac{1}{L},\ \tfrac{1}{L}] \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} -\frac{1}{L} + \frac{x}{L^2} & \frac{1}{L} - \frac{x}{L^2} \\ -\frac{x}{L^2} & \frac{x}{L^2} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix}$$

Тогда

$$F(x_1)G(x_1) \ = \ = \begin{bmatrix} 0 & 0 \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{du}{dx}\big|_L \end{bmatrix}$$

Аналогично:

$$F(x_0)G(x_0) \ = \ = \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = \begin{bmatrix} \frac{du}{dx}\big|_0 \\ 0 \end{bmatrix}$$

$$F(x) = N_l^T = \begin{bmatrix} 1-\frac{x}{L} \\ \frac{x}{L} \end{bmatrix}, \text{ значит } F'(x) = \begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix}$$

Отсюда

$$\int_{x_0}^{x_1} F`(x)G(x)dx = \int_0^L \begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix} [-\tfrac{1}{L},\ \tfrac{1}{L}] \begin{bmatrix} u_i \\ u_j \end{bmatrix} dx$$

Аналогично были получены выражения для других слагаемых (6) и (7), в результате итоговое выражение для уравнения (4) приняло вид:

$$2 \begin{bmatrix} \frac{du}{dx}\big|_i \\ \frac{du}{dx}\big|_j \end{bmatrix} + \begin{bmatrix} -\frac{2}{L}+\frac{5}{2}+\frac{L}{3} & \frac{2}{L}-\frac{5}{2}+\frac{L}{6} \\ \frac{2}{L}+\frac{5}{2}+\frac{L}{6} & -\frac{2}{L}-\frac{5}{2}+\frac{L}{3} \end{bmatrix} \begin{bmatrix} u_i \\ u_j \end{bmatrix} = 0 \qquad (9)$$

3. Анасамблирование и решение СЛАУ

Далее необходимо составить СЛАУ, матрица коэффициентов в которой анасамблируется из выражения (9).

В вектор неизвестных входят значения функции в каждом из узлов, а также ее производные на концах отрезка. Размерность вектора неизвестных, как и вектора правых частей, будет равна (N+3). Анасамблируемая матрица коэффициентов будет иметь размерность (N+3) * (N+3).

Например, для 4х конечных элементов СЛАУ будет иметь вид:

$$\begin{bmatrix} -3.4 & 3.6 & & & -2 & \\ 8.6 & -11.8 & 3.6 & & & \\ & 8.6 & -11.8 & 3.6 & & \\ & & 8.6 & -8.4 & & 2 \\ & & & & 1 & \\ & & & 1 & & \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \frac{du}{dx}\Big|_{x=0} \\ \frac{du}{dx}\Big|_{x=1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -6 \\ 10 \end{bmatrix}$$

4. Получение квадратичной функции формы

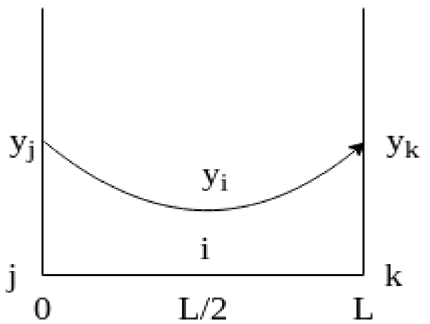

Рис. 3. Конечный элемент с квадратичной функцией формы.

$$u = a_0 + a_1 x + a_2 x^2$$

Найдем коэффициенты из следующих уравнений:

$$u_j = a_0 ,$$
$$u_i = a_0 + a_1 \tfrac{L}{2} + a_2 \tfrac{L^2}{4} ,$$
$$u_k = a_0 + a_1 L + a_2 L^2 ,$$

Получим:

$$u = \left[1 - \frac{3x}{L} + \frac{2x^2}{L^2}, - \frac{x}{L} + \frac{2x^2}{L^2},\ \frac{4x}{L} - \frac{4x^2}{L^2}\right] \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix} \quad (10)$$

Отсюда квадратичная функция формы выражается как:

$$N_l = \left[1 - \frac{3x}{L} + \frac{2x^2}{L^2}, - \frac{x}{L} + \frac{2x^2}{L^2},\ \frac{4x}{L} - \frac{4x^2}{L^2}\right] \quad (11)$$

5. Нахождение уравнения для квадратичной функции формы

Разделим интеграл из уравнения (4) на слагаемые (5), (6) и (7) и найдем каждое из них отдельно.

Используя (8), (11) и аналогично линейной функции формы, получим для (5):

$$2\int_0^L N_l^T \frac{d^2u}{dx^2}\, dx = 2N_l^T \frac{du}{dx}\Big|_0^L - 2\int_0^L N_l^{T\,\backprime} \frac{du}{dx} dx =$$

$$= 2 \begin{bmatrix} -\frac{du}{dx}\big|_j \\ \frac{du}{dx}\big|_k \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{14}{3L} & -\frac{2}{3L} & \frac{16}{3L} \\ -\frac{2}{3L} & -\frac{14}{3L} & \frac{16}{3L} \\ \frac{16}{3L} & \frac{16}{3L} & -\frac{32}{3L} \end{bmatrix} \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix} \quad (12)$$

Для (6) получим:

$$-5\int_0^L N_l^T \frac{du}{dx}\,dx =$$

$$=-5\int_0^L \begin{bmatrix} 1-\frac{3x}{L}+\frac{2x^2}{L^2} \\ -\frac{x}{L}+\frac{2x^2}{L^2} \\ \frac{4x}{L}-\frac{4x^2}{L^2} \end{bmatrix} \left[-\frac{3}{L}+\frac{4x}{L^2}, -\frac{1}{L}+\frac{4x}{L^2},\ \frac{4}{L}-\frac{8x}{L^2}\right] \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix} dx =$$

$$= \begin{bmatrix} \frac{5}{2} & \frac{5}{6} & -\frac{10}{3} \\ -\frac{5}{6} & -\frac{5}{2} & \frac{10}{3} \\ \frac{10}{3} & -\frac{10}{3} & 0 \end{bmatrix} \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix} \qquad (13)$$

Для (7) получим:

$$\int_0^L N_l^T u\,dx =$$

$$=\int_0^L \begin{bmatrix} 1-\frac{3x}{L}+\frac{2x^2}{L^2} \\ -\frac{x}{L}+\frac{2x^2}{L^2} \\ \frac{4x}{L}-\frac{4x^2}{L^2} \end{bmatrix} \left[1-\frac{3x}{L}+\frac{2x^2}{L^2}, -\frac{x}{L}+\frac{2x^2}{L^2},\ \frac{4x}{L}-\frac{4x^2}{L^2}\right] \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix} dx =$$

$$= \qquad (14)$$

$$\begin{bmatrix} \frac{2L}{15} & -\frac{L}{30} & \frac{L}{15} \\ -\frac{L}{30} & \frac{2L}{15} & \frac{L}{15} \\ \frac{L}{15} & \frac{L}{15} & \frac{8L}{15} \end{bmatrix} \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix}$$

Объединив (12), (13) и (14), получим:

$$\int_0^L N_l^T \left(2\frac{d^2u}{dx^2} - 5\frac{du}{dx} + u\right) dx =$$

$$= 2\begin{bmatrix} -\frac{du}{dx}\Big|_j \\ \frac{du}{dx}\Big|_k \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{14}{3L} + \frac{5}{2} + \frac{2L}{15} & -\frac{2}{3L} + \frac{5}{6} - \frac{L}{30} & \frac{16}{3L} - \frac{10}{3} + \frac{L}{15} \\ -\frac{2}{3L} - \frac{5}{6} - \frac{L}{30} & -\frac{14}{3L} - \frac{5}{2} + \frac{2L}{15} & \frac{16}{3L} + \frac{10}{3} + \frac{L}{15} \\ \frac{16}{3L} + \frac{10}{3} + \frac{L}{15} & \frac{16}{3L} - \frac{10}{3} + \frac{L}{15} & -\frac{32}{3L} + \frac{8L}{15} \end{bmatrix} \begin{bmatrix} u_j \\ u_k \\ u_i \end{bmatrix} = 0$$

Внутренний узел необходимо исключить из уравнения. Для этого, с помощью прямого хода метода Гаусса нужно обнулить значения в третьем столбце у первых двух строк, а последнюю строку - убрать совсем.

Добавим к первой строке третью, умноженную на такой коэффициент K_1, что будет верно выражение:

$$\frac{16}{3L} - \frac{10}{3} + \frac{L}{15} + K_1\left(-\frac{32}{3L} + \frac{8L}{15}\right) = 0,\ K_1 = \frac{-80+50L+L^2}{-160+8L^2}$$

Аналогично, ко второй строке добавим третью, умноженную на коэффициент K_2:

$$\frac{16}{3L} + \frac{10}{3} + \frac{L}{15} + K_2\left(-\frac{32}{3L} + \frac{8L}{15}\right) = 0,\ K_2 = \frac{-80-50L+L^2}{-160+8L^2}$$

Текст программы

```cpp
#include <iostream>
#include <cmath>
#include "matrix.hpp"
#include "csv_parser.hpp"

#define ABS(a) (((a) < 0) ? -(a) : (a))

double analytics(double x) {
   double l_1 = (5 + sqrt(17)) / 4;
   double l_2 = (5 - sqrt(17)) / 4;
   double c_1 = -13.9;
   double c_2 = 117;
   return c_1 * exp(l_1 * x) + c_2 * exp(l_2 * x);
}

Matrix<double> lin_experiment(map<string, double>& C, double N) {
   // получение матрицы и коэффициента
   // для анасамблирования
   double l = 1.0 / N;

   double a[2][2];
   a[0][0] = -2/l + 2.5 + l/3;
   a[0][1] = 2/l - 2.5 + l/6;
   a[1][0] = 2/l + 2.5 + l/6;
   a[1][1] = -2/l - 2.5 + l/3;

   double coef = 2;


   // анасамблирование всех матриц
   // создание
   Matrix<double> A(N + 3, N + 3);
   Matrix<double> R(N + 3, 1);

   // наложение граничных условий
   R.set(N + 1, 0, C["dU_0"]);
   R.set(N + 2, 0, C["U_1"]);

   A.set(N + 1, N + 1, 1);
   A.set(N + 2, N, 1);

   // учет границ
   A.set(0, N + 1, -coef);
   A.set(N, N + 2, coef);

   // добавление КЭ-матрицы по диагонали
   for (int i = 0; i <= N - 1; i++) {
       A.set(i, i, A.get(i, i) + a[0][0]);
       A.set(i, i + 1, A.get(i, i + 1) + a[0][1]);
       A.set(i + 1, i, A.get(i + 1, i) + a[1][0]);
       A.set(i + 1, i + 1, A.get(i + 1, i + 1) + a[1][1]);
   }

   // решение методом Гаусса
```

```
    A.Gauss(&R);

    return R;
}

Matrix<double> quadr_experiment(map<string, double>& C, double N) {
    // получение матрицы и коэффициента
    // для анасамблирования
    double l = 1.0 / N;

    double a[3][3];

    // заполняем третью строку сначала, чтобы ее можно было вычитать
    a[2][0] = 16.0/(3.0 * l) + 10.0/3.0 + l/15.0;
    a[2][1] = 16.0/(3.0 * l) - 10.0/3.0 + l/15.0;
    a[2][2] = -32.0/(3.0 * l) + (8.0 * l)/15.0;


    // заполнение первой и второй строки начальными значениями
    a[0][0] = -14.0/(3.0 * l) + 2.5 + (2.0 * l)/15.0;
    a[0][1] = -2.0/(3.0 * l) + 5.0/6.0 - l/30.0;
    a[0][2] = 16.0/(3.0 * l) - 10.0/3.0 + l/15.0;

    a[1][0] = -2.0/(3.0 * l) - 5.0/6.0 - l/30.0;
    a[1][1] = -14.0/(3.0 * l) - 2.5 + (2.0 * l)/15.0;
    a[1][2] = 16.0/(3.0 * l) + 10.0/3.0 + l/15.0;

    // расчет коэффициентов и прямой ход метода Гаусса
    //double k_1 = (-80 + 50 * l + l * l)/(-160 + 8 * l * l);

    double k1 = -a[0][2]/a[2][2];

    for (int i = 0; i < 3; i++) {
        a[0][i] += a[2][i] * k1;
    }

    //double k_2 = (-80 - 50 * l + l * l)/(-160 + 8 * l * l);

    double k2 = -a[1][2]/a[2][2];

    for (int i = 0; i < 3; i++) {
        a[1][i] += a[2][i] * k2;
    }


    double coef = 2;


    // анасамблирование всех матриц
    // создание
    Matrix<double> A(N + 3, N + 3);
    Matrix<double> R(N + 3, 1);

    // наложение граничных условий
    R.set(N + 1, 0, C["dU_0"]);
    R.set(N + 2, 0, C["U_1"]);
```

```
    A.set(N + 1, N + 1, 1);
    A.set(N + 2, N, 1);

    // учет границ
    A.set(0, N + 1, -coef);
    A.set(N, N + 2, coef);

    // добавление КЭ-матрицы по диагонали
    for (int i = 0; i <= N - 1; i++) {
        A.set(i, i, A.get(i, i) + a[0][0]);
        A.set(i, i + 1, A.get(i, i + 1) + a[0][1]);
        A.set(i + 1, i, A.get(i + 1, i) + a[1][0]);
        A.set(i + 1, i + 1, A.get(i + 1, i + 1) + a[1][1]);
    }

    // решение методом Гаусса
    A.Gauss(&R);

    return R;
}

Matrix<double> additional_experiment(map<string, double>& C, double N) {
    // получение матрицы и коэффициента
    // для анасамблирования
    double l = 1 / N;

    double a[2][2];
    a[0][0] = -2/l + 2.5 + l/3;
    a[0][1] = 2/l - 2.5 + l/6;
    a[1][0] = 2/l + 2.5 + l/6;
    a[1][1] = -2/l - 2.5 + l/3;

    double coef = 2;


    // анасамблирование всех матриц
    // создание
    Matrix<double> A(N + 3, N + 3);
    Matrix<double> R(N + 3, 1);

    // наложение граничных условий
    R.set(N + 1, 0, C["dU_0"]);
    R.set(N + 2, 0, 0);

    A.set(N + 1, N + 1, 1);
    A.set(N + 2, N, -1);
    A.set(N + 2, N + 2, 1);

    // учет границ
    A.set(0, N + 1, -coef);
    A.set(N, N + 2, coef);

    // добавление КЭ-матрицы по диагонали
    for (int i = 0; i <= N - 1; i++) {
        A.set(i, i, A.get(i, i) + a[0][0]);
        A.set(i, i + 1, A.get(i, i + 1) + a[0][1]);
        A.set(i + 1, i, A.get(i + 1, i) + a[1][0]);
```

```
            A.set(i + 1, i + 1, A.get(i + 1, i + 1) + a[1][1]);
        }

        // решение методом Гаусса
        A.Gauss(&R);

        return R;
}

void accuracy(Matrix<double>& R, string filename) {
        std::ofstream result_file;
        result_file.open(filename);
        if (!result_file) {
            cout << "ОШИБКА: Невозможно открыть файл для записи погрешностей: "
<< filename << endl;
            return;
        }

        double N = R.get_rows() - 2;
        double l = 1 / N;
        double max_acc = 0;
        for (int i = 0; i < N; i++) {
            double time = i * l;
            result_file << time << "\t" << ABS(R.get(i, 0) - analytics(time)) <<
endl;
            if (ABS(R.get(i, 0) - analytics(time)) > max_acc) {
                max_acc = ABS(R.get(i, 0) - analytics(time));
            }
        }

        result_file.close();
        cout << filename << ":" << endl;
        cout << max_acc << endl << endl;
}

void graph(Matrix<double>& R, string filename) {
        std::ofstream result_file;
        result_file.open(filename);
        if (!result_file) {
            cout << "ОШИБКА: Невозможно открыть файл для записи результатов: " <<
filename << endl;
            return;
        }

        double N = R.get_rows() - 2;
        double l = 1 / N;
        for (int i = 0; i < N; i++) {
            result_file << i * l << "\t" << R.get(i, 0) << endl;
        }

        result_file.close();
}

int main() {
        map<string, double> C = create_map_from_file("constants.dat");

        Matrix<double> R1 = lin_experiment(C, C["N_1"]);
```

```
    graph(R1, "res/lin_result_N1.dat");
    accuracy(R1, "res/lin_accuracy_N1.dat");

    Matrix<double> R2 = lin_experiment(C, C["N_2"]);
    graph(R2, "res/lin_result_N2.dat");
    accuracy(R2, "res/lin_accuracy_N2.dat");

    /*Matrix<double> R3 = additional_experiment(C, C["N_2"]);
    graph(R3, "res/additional_N2.dat");*/

    Matrix<double> R4 = quadr_experiment(C, C["N_1"]);
    graph(R4, "res/quadr_result_N1.dat");
    accuracy(R4, "res/quadr_accuracy_N1.dat");

    Matrix<double> R5 = quadr_experiment(C, C["N_2"]);
    graph(R5, "res/quadr_result_N2.dat");
    accuracy(R5, "res/quadr_accuracy_N2.dat");

    return 0;
}
```

Результаты работы программы

В результате работы программы для линейной функции формы были получены следующие графики:

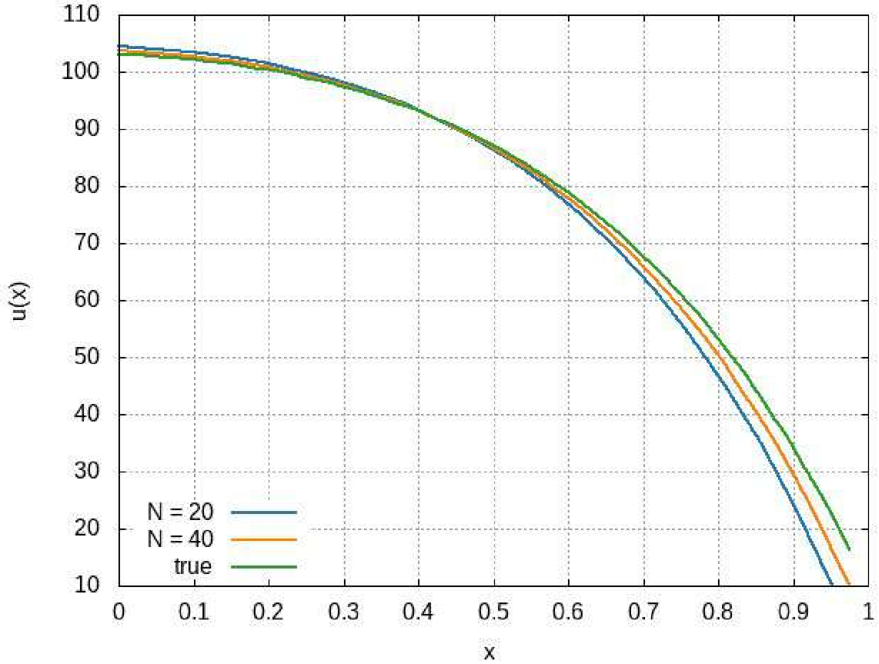

Рис. 4. Результаты работы программы для линейной функции формы. Здесь: синий - при 20 КЭ, оранжевый - при 40 КЭ, зеленый - аналитическое решение.

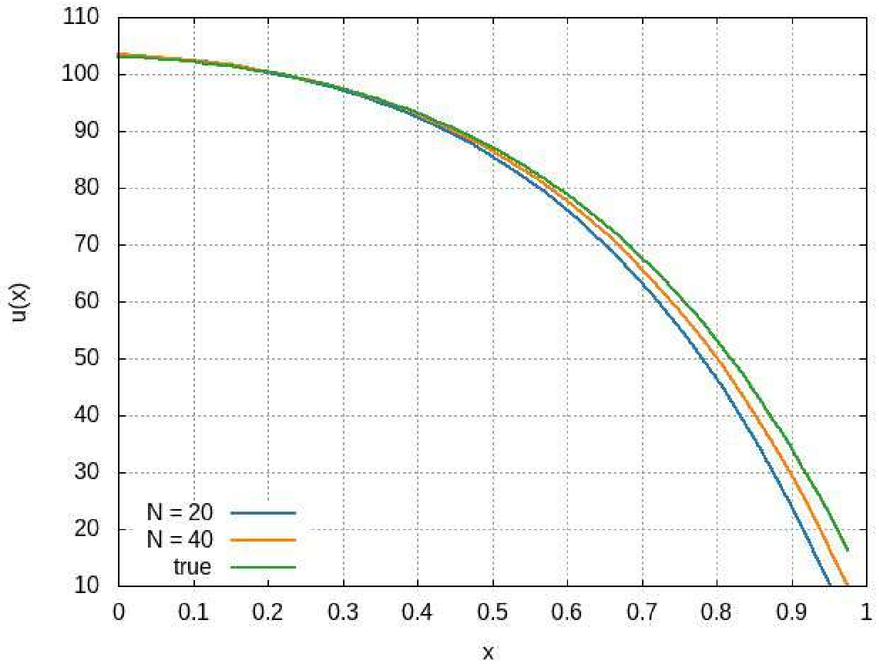

Рис. 5. Результаты работы программы для квадратичной функции формы. Здесь: синий - при 20 КЭ, оранжевый - при 40 КЭ, зеленый - аналитическое решение.

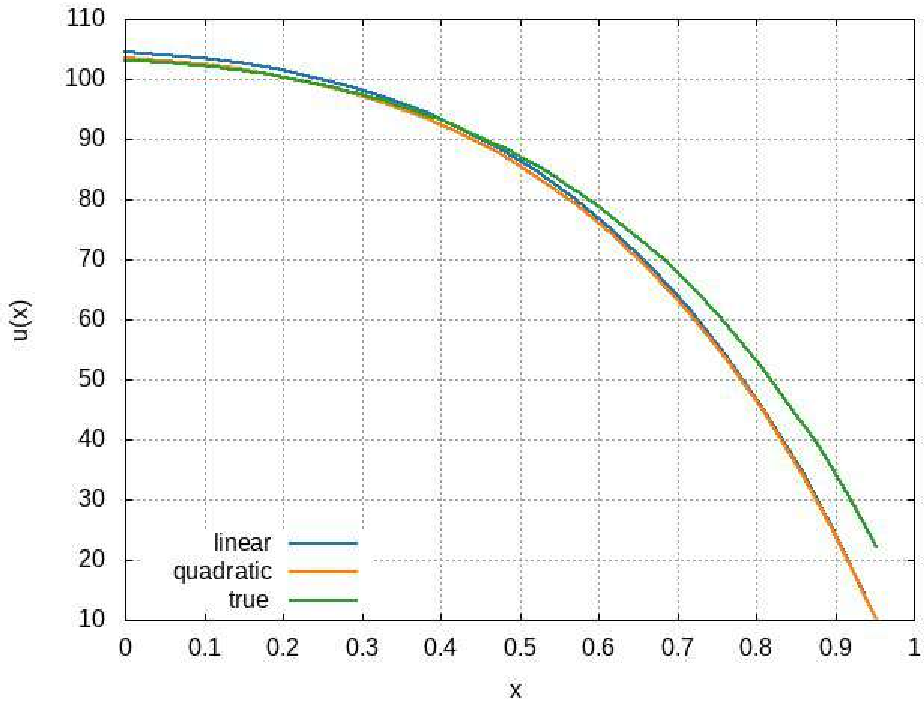

Рис. 6. Сравнение аналитического решения(зеленый), решения МКЭ с линейной функцией формы(синий) и с квадратичной функцией формы(оранжевый) при N=20.

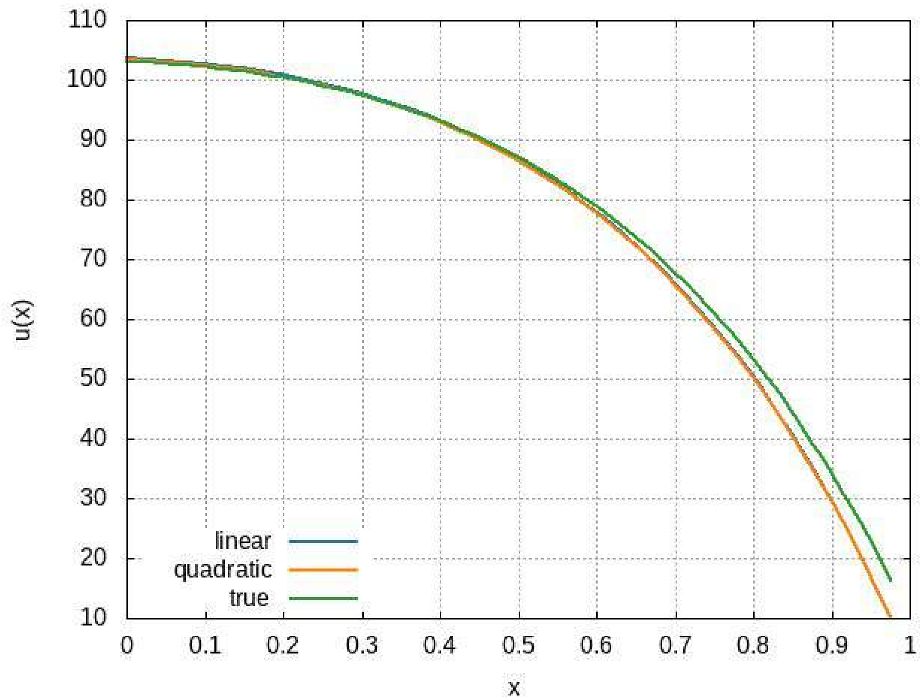

Рис. 7. Сравнение аналитического решения(зеленый), решения МКЭ с линейной функцией формы(синий) и с квадратичной функцией формы(оранжевый) при N=40.