МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского» (ННГУ)

Институт информационных технологий, математики и механики

Кафедра дифференциальных уравнений, математического и численного анализа

Направление подготовки: «Фундаментальная информатика и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

на тему:

«Численное решение начально-краевой задачи для интегро-дифференциального уравнения в частных производных»

Выполнил: студент группы 381706-2 Паузин Леонид Павлович

____

Подпись

Проверил:
Ассистент кафедры ДУМЧА
Морозов Кирилл Евгеньевич

____

Подпись

Нижний Новгород
2020

Содержание

1. Введение

Краевая задача — это задача отыскания частного решения системы обыкновенных дифференциальных уравнений с дополнительными условиями, налагаемыми на значения функций не менее чем в двух точках отрезка. Свое название краевая задача получила по случаю, в котором дополнительные условия заданы на концах (краях) отрезка.

В качестве примера рассмотрим процесс нагревания тонкого однородного стрежня длиной l с теплоизолированными концами. На процесс изменения температуры стержня осуществляется некое воздействие для достижения определённых целей.

2. Постановка задачи

На множестве $Q = [0, l] \times [0, T],\ l > 0, T\ > 0$; нужно найти функцию $y(x, t)$ – температуру стержня – непрерывно дифференцируемую по t и дважды непрерывно дифференцируемую по x, которая является решением уравнения

$$y'_t(x, t) = a^2 y''_{xx}(x, t) + u(x, t) \tag{1}$$

удовлетворяющее однородным граничным условиям второго рода

$$y'_x(0, t) = y'_x(l, t) = 0 \tag{2}$$

и начальному условию

$$y(x, 0) = \varphi(x),\ \ \varphi(x) > 0 \tag{3}$$

Функция $\varphi(x)$ задает начальное распределение температуры, дважды непрерывно дифференцируема на отрезке $[0, \mathrm{l}]$, удовлетворяет условиям согласования (3) и условию

$$\int_0^l \varphi(x)\, dx = 1 \tag{4}$$

Непрерывная функция $u(x, t)$ – управление с обратной связью, которое представляется в одном из вариантов:

$$u(x, t) = b(x) y(x, t) \tag{5}$$

$$u(x, t) = b(x) y(x, t) - y(x, t) \int_0^l b(x) y(x, t)\, dx \tag{6}$$

где $b(x)$ – управляющая функция, непрерывная на отрезке $[0, l]$.

В качестве начальной функции $\varphi(x)$ и функции $b(x)$ в данной работе используются:

$$\varphi(x) = \frac{1}{l} + \varphi_1 \cos\left(\frac{\pi x}{l}\right) + \varphi_2 \cos\left(\frac{2\pi x}{l}\right) \tag{7}$$

$$b(x) = b_0 + b_1\, cos\left(\frac{\pi x}{l}\right) + b_2\, cos\left(\frac{2\pi x}{l}\right) \tag{8}$$

3.Описание метода

Мы должны составить разностную схему для решения задачи. После этого, найдем для этой схемы нулевой слой. Как уже писалось ранее, в качестве начальной функции берем (7).

Для вычисления последующих слоев нужно будет находить интеграл в точке. Перед вычислением каждого следующего слоя находим интеграл для значений последнего известного слоя по формуле Симпсона:

$$I_j = \frac{h}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_3 + \cdots + 2y_{K-2} + 4y_{K-1} + y_K),$$

где K = $\frac{l}{h}$ – количество шагов по x, предполагается четным.

После чего составим неявную разностную схему с погрешностью $O(\tau + h^2)$

$$\frac{y_k^{n+1} - y_k^n}{\tau} = \frac{y_{k+1}^{n+1} - 2y_k^{n+1} + y_{k-1}^{n+1}}{h^2} + u_k^n \tag{9}$$

Также мы должны проверить, что $\frac{\tau}{h^2} < ¼$, так как не доказано условие устойчивости.

Составим трехточечные разностные производные первого порядка для краевых условий с погрешностью второго порядка.

$$\frac{y_1^{n+1} - y_0^{n+1}}{h} = \frac{y_K^{n+1} - y_{K-1}^{n+1}}{h} = 0$$

Уравнение (1) преобразуем к виду:

$$\frac{\partial^2 y}{\partial x^2} = \frac{\partial y}{\partial \tau} - u(x, \tau)$$

Подставив вторую производную в это выражение, получим два уравнения для правой и левой границы. Получим следующую систему:

$$\begin{cases} y^{j+1} = 0 \\ y_N^{j+1} = y_{N-1}^{j+1} + ht_{j+1} \\ \frac{\tau y_{i-1}^{j+1}}{h^2} - \left(1 + \frac{2\tau}{h^2}\right) y_i^{j+1} + \frac{\tau y_{i+1}^{j+1}}{h^2} = -(y_i^j + \tau x_i) \end{cases}$$

Матрицу коэффициентов мы преобразовываем к виду трехдиагональной матрицы. После чего можно решить систему методом прогонки

4.Выбор языка

В качестве языка программирования был выбран C#. По причине того, что в нем достаточно функций для выполнения данной задачи, а так же было интересно попрактиковаться в нем

5.Руководство пользователя

При запуске программа имеет следующий интерфейс:

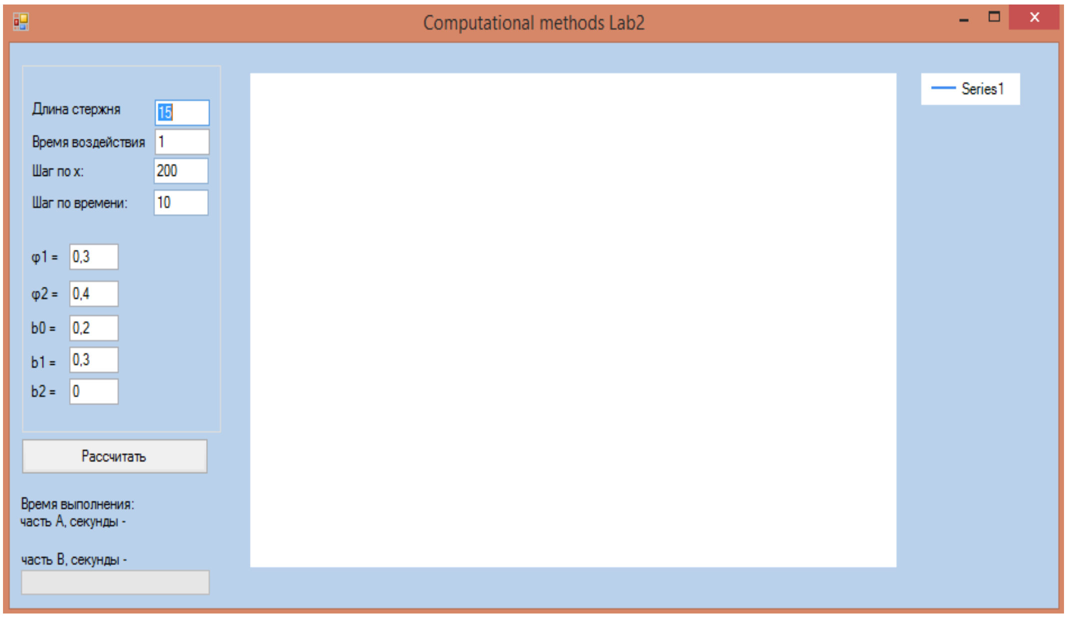

Рисунок 1

Пользователю предоставляется возможность указать длину стержня, время изменения температуры, количество точек, в которых производятся расчеты, а также параметры начального распределения температуры и управления обратной связью.

После заполнения всех нужных пользователю параметров можно приступать к решению задачи. Для этого нужно нажать кнопку «Рассчитать».

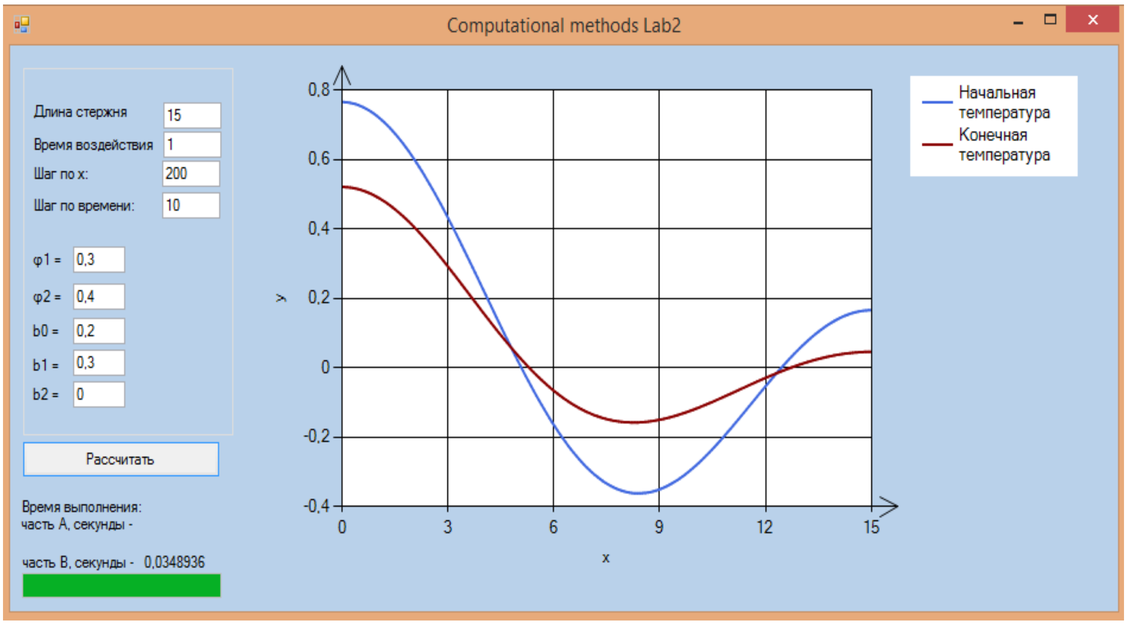

Рисунок 2

После нажаться кнопки «F» начинается решение с использованием части А

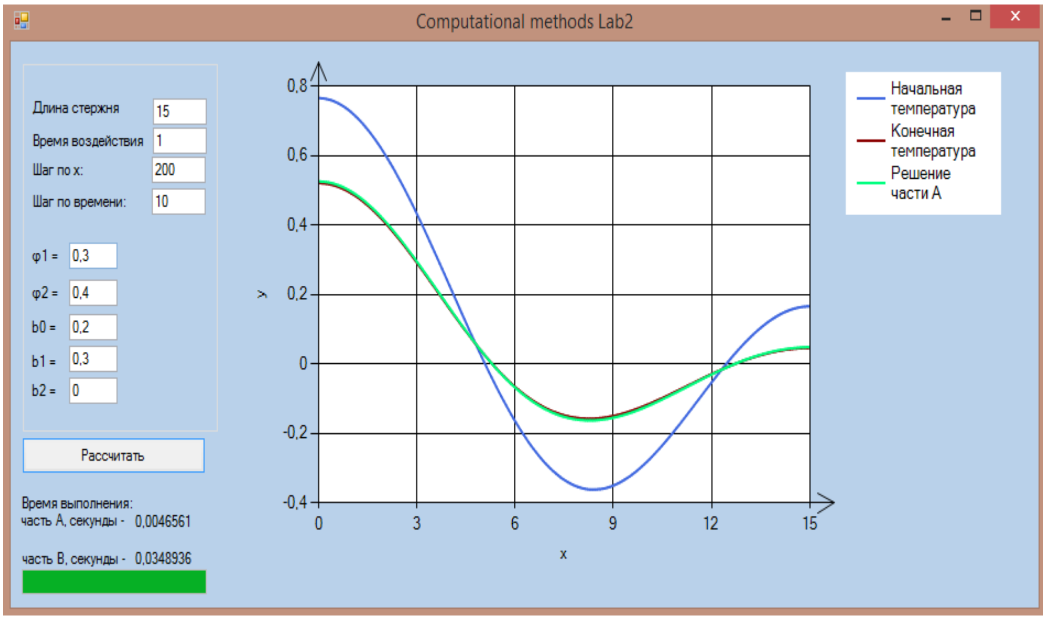

Рисунок 3

Зеленый график почти полностью совпадает с красным.

Если пользователь попробует построить решение части А, без решения части В, то его ждет такое предупреждение

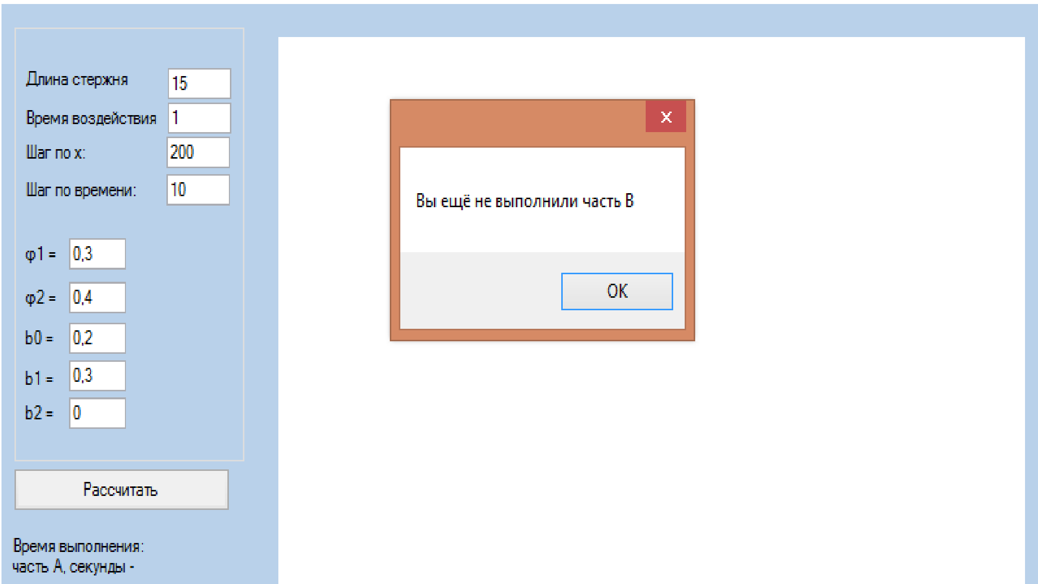

Рисунок 4

6.Вывод

В ходе выполнения лабораторной работы была решена начально-краевая задача для интегро-дифференциального уравнения нагревания стержня. Были получены практические навыки разработки пользовательского интерфейса в Windows Forms.

7.Литература

1. Самарский А.А., Гулин А.В. Введение в численные методы. — М.: Наука, 1989.
2. Учебно-методическое пособие - Лабораторная работа «Численное решение начально-краевой задачи для интегро-дифференциального уравнения в частных производных»

8.Приложение

```
//Вычисление интеграла по формуле Симпсона
private double Simp_integr (int t, int k, ref double[,] mas)    {
      double retValue, x = h;
      int iter = 1;
      if (k == 0)
      {
        int a;
        retValue = mas[0, t];
        while (iter < hnum)
        {
          a = 4;
          if (iter % 2 == 0)
            a = 2;
          retValue = retValue + a * mas[iter, t];
          x += h;
          iter++;
        }
        retValue = retValue + mas[hnum - 1, t];
      }
      else
      {
        int a;
        retValue = b(0) * mas[0, t];
        while (iter < hnum - 1)
        {
          a = 4;
          if (iter % 2 == 0)
            a = 2;
          retValue = retValue + b(x) * a * mas[iter, t];
          x += h;
          iter++;
        }
        retValue = retValue + b(x) * mas[hnum - 1, t];
      }
      retValue = retValue*(h / 3);
      return retValue;
}
```