МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева»

ИИТК/09.03.01/Информатика и вычислительная техника/Автоматизированные системы обработки информации и управления

институт/ факультет/ подразделение

Информатики и вычислительной техники

кафедра/ цикловая комиссия

ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ

Преподаватель ____________ Семенкина М. Е.

подпись, дата

Обучающиеся группы БИА21-01

Путинцев А.Ю.

Юрченко М.А.

Турсунов Д.М

Сухарева А.С.

Красноярск 2022

ВВЕДЕНИЕ

Цель работы: Изучение методов решения обыкновенных дифференциальных уравнений.

Задачи:

1. Решить обыкновенное дифференциальное уравнение аналитически. Построить задачу Коши (тестовую). Проверить выполняется ли условие существования и единственности решения задачи.

2. Решить задачу Коши численно (двумя методами).

3. Сравнить результаты.

ХОД РАБОТЫ

1. Сначала объявляем меню для выбора функций-формул.

исполняемый файл ode.cpp

Функция f(x). Код приведён ниже:

```
double function(double x, int s) { // rec
    switch (s)
    {
    case 1:
        return -x+(x/pow(M_E,x));
    case 2:
        return (x-pow(M_E,2))/ (x + pow(M_E, 2));
    case 3:
        return x / (pow(x, 2) + 1);
    case 5:
        return (x+1)/(1+x+1);
    default:
        return 0.0;
    }
}
```

// ОДУ

Функция y'. Код приведён ниже:

```
double ode(double x, double y, int s) { // rec
    switch (s)
    {
    case 1:
        return -(x+y-(y/x));
    case 2:
        return (1-pow(y,2))/(2*x);
    case 3:
        return -(2*x*pow(y,2)-y)/x;
    case 5:
        return (y/(1+x))-(pow(y,2)/(1+x));
    default:
        return 0.0;
    }
}
```

2. После этого мы пишем код для построения графика функции.

//Визуал графика функции

```
 dx = (double)(ubound - lbound) / n;
  double step = 0.01;
  a = -25;
  b = 25;
  x = a;

  int i, n;
  double x, y, h;
  double k1, k2, k3, k4;
  h = 0.5;
  n = 100;
```

// Визуал предиктор-корректор

```
x = 0.01;
y = function(x, s);
this->chart1->Series[4]->Points->AddXY(x, y);
for (i = 0; i < n; i++)
{
    k1 = h * ode(x, y, s);
    k2 = h * ode(x + (h / 2), y + (k1 / 2), s);
    y = y + k2;
    x += h;
    this->chart1->Series[4]->Points->AddXY(x, y);
}
```

//Визуал метода Рунге-Кутта 3 порядка

```
x = 0.01;
   y = function(x, s);
   this->chart1->Series[1]->Points->AddXY(x, y);
   for (i = 0; i < n; i++)
   {
       k1 = h * ode(x, y, s);
       k2 = h * ode(x + (h / 2), y + (k1 / 2), s);
       k3 = h * ode(x + h, y + (2 * k2) - k1, s);
       y += ((k1 + 4 * k2 + k3) / 6);
```

```
        x += h;
        this->chart1->Series[1]->Points->AddXY(x, y);
    }
```

// Визуал метода Рунге-Кутта 4 порядка

```
x = 0.01;
    y = function(x, s);
    this->chart1->Series[3]->Points->AddXY(x, y);
    for (i = 0; i < n; i++)
    {
        k1 = h * ode(x, y, s);
        k2 = h * ode(x + (h / 2), y + (k1 / 2), s);
        k3 = h * ode(x + (h/2), y + (k2/2), s);
        k4 = h * ode(x + h, y + k3, s);
        y = y + ((k1 + 2 * k2 + 2* k3 + k4) / 6);
        x = x + h;
        this->chart1->Series[3]->Points->AddXY(x, y);
    }
```

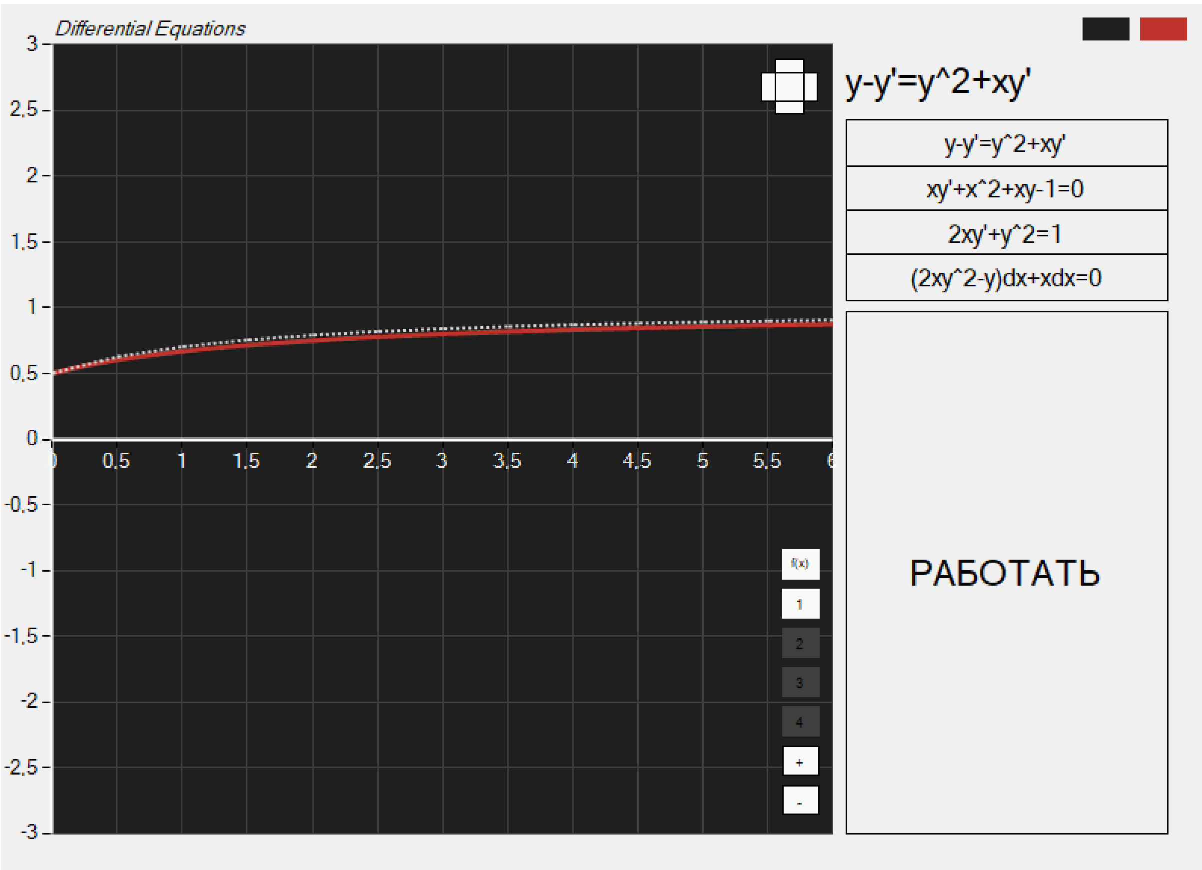
// Визуал метода Эйлера

```
x = 0.01;
    y = function(x,s);
    this->chart1->Series[2]->Points->AddXY(x, y); // eiler
    for (i = 0; i < n; i++)
    {
        if (x == 0) {
            y += h * ode(x + 0.01, y, s); // вычисление yi
            x += h;
            this->chart1->Series[2]->Points->AddXY(x, y);
            y += h * ode(x - 0.01, y, s); // вычисление yi
            x += h;
            this->chart1->Series[2]->Points->AddXY(x, y);
            i++;
            i++;
        }
        y += h * ode(x, y, s); // вычисление yi
        x += h;
        this->chart1->Series[2]->Points->AddXY(x, y);
    }
    h = 0.1;
```

```
n = 100*5;
x = 0.01;
y = function(x, s);
this->chart1->Series[0]->Points->AddXY(x, y);
for (i = 0; i < n; i++) // function
{
    x += h;
    y = function(x, s);
    this->chart1->Series[0]->Points->AddXY(x, y);
}
```

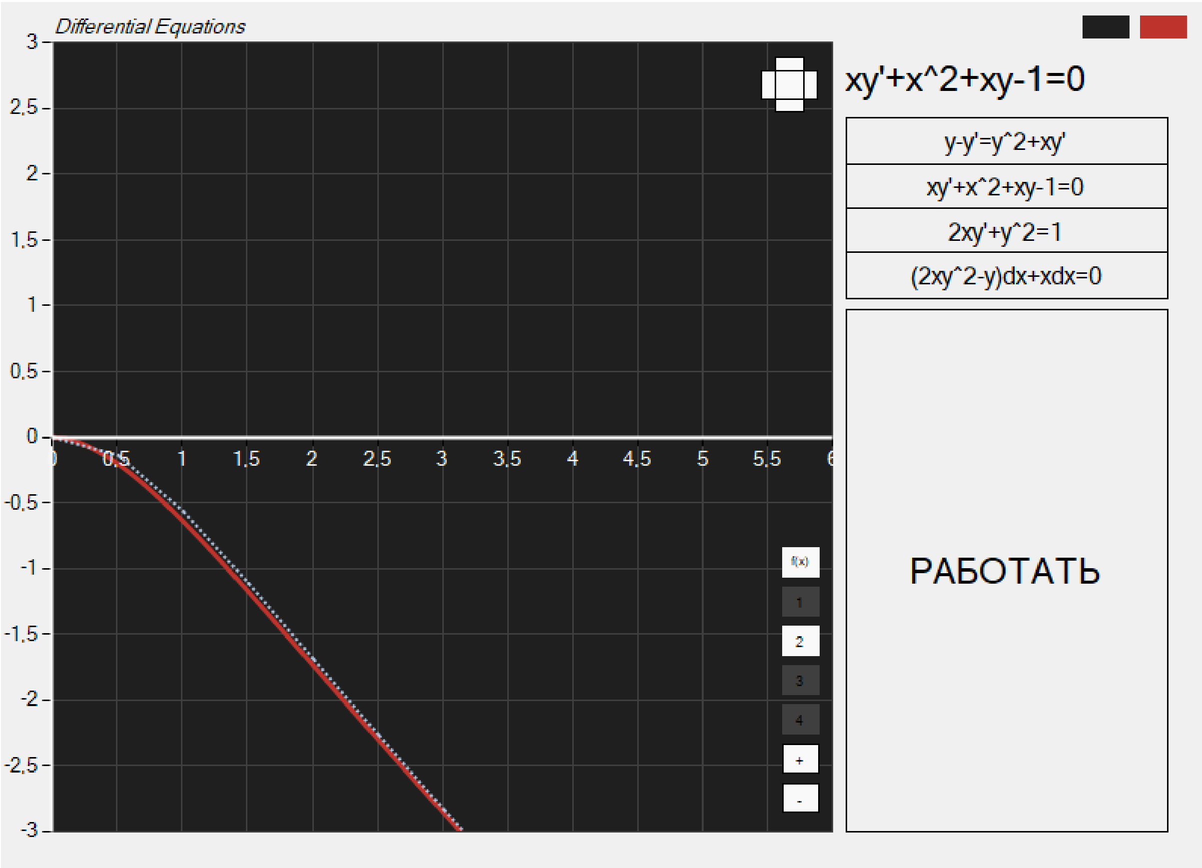
Графики:

1. Пример работы программы функции $y - y' = y^2 + xy'$



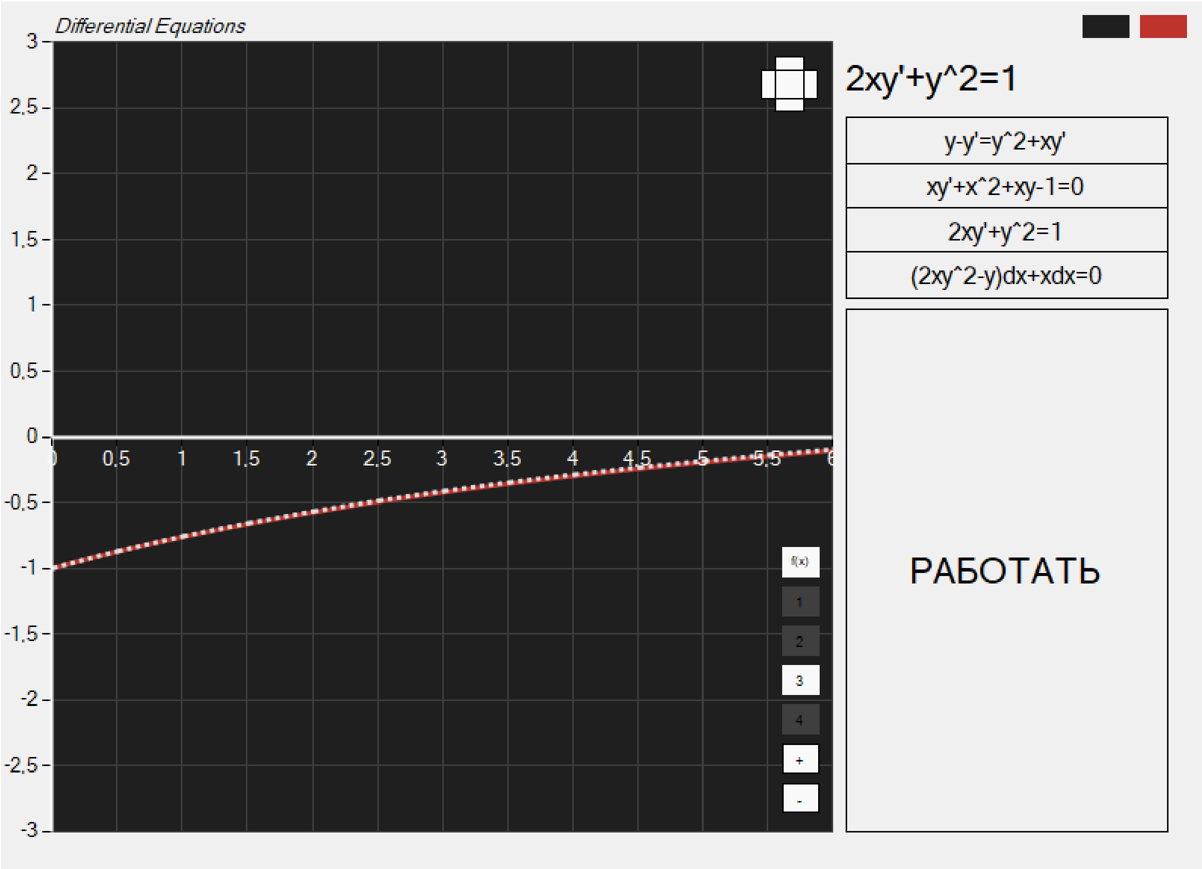
2.Пример работы программы функции

$$xy' - x^2 + xy - 1 = 0$$



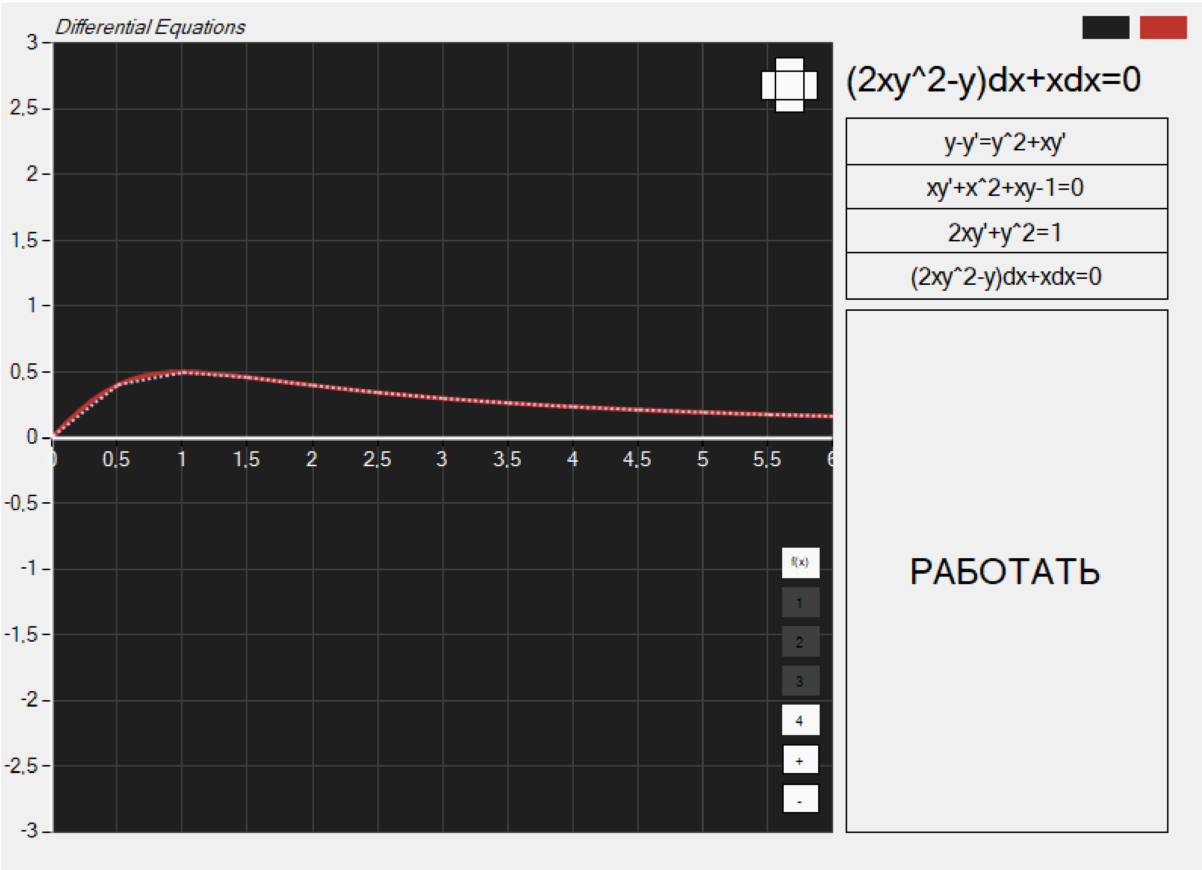
3. Пример работы программы функции

$$2xy' + y^2 = 1$$



4. Пример работы программы функции

$$(2x^2 - y)dx + xdx = 0$$



Выводы

Заключением нашей работы стало создание программы, которая позволяет пользователю:

1) аналитически решить обыкновенное дифференциальное уравнение на 4-ёх примерах (задача Коши);