Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Отчет по лабораторной работе №7
по курсу «Численные методы»

Работу выполнил:
М8О-409Б-19 Худяков О. С. __________ 2 вариант
Руководитель: **________/** Пивоваров Д.Е.

Оценка: ________

Дата: 25.11.2022

Задание: Решить краевую задачу для дифференциального уравнения эллиптического типа. Аппроксимацию уравнения произвести с использованием центрально-разностной схемы. Для решения дискретного аналога применить следующие методы: метод простых итераций (метод Либмана), метод Зейделя, метод простых итераций с верхней релаксацией. Вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением U(x, y) . Исследовать зависимость погрешности от сеточных параметров hx, hy.

Вариант:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

$$u_x(0, y) = 0;$$

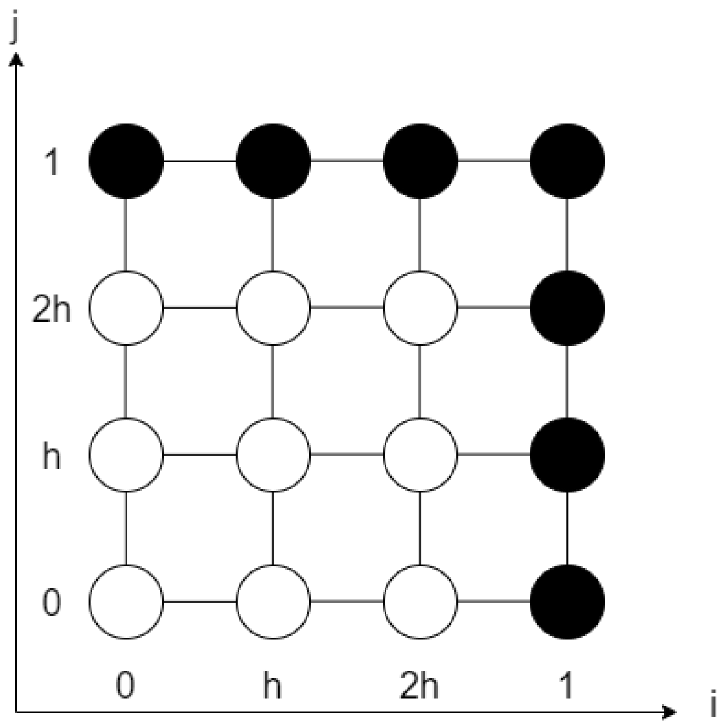
$$u(1, y) = 1 - y^2,$$

$$u_y(x, 0) = 0;$$

$$u(x, 1) = x^2 - 1$$

$$U = \mathrm{x}^2 + y^2$$

Решение: Рассмотрим решение данной задачи на примере, для случая $N_x = N_y = 3$.



На графе черным отмечены точки, значение которых мы знаем точно из граничных условий. Из граничных условий: $u_{1j} - u_{0j} = 0; u_{Nj} = 1 - (jh)^2; u_{i1} - u_{i0} = 0; u_{iN} = (ih)^2 - 1$. Остальные u_{ij} находим следующим образом: аппроксимируем $\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} + O(h_x^2)$, $\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2} + O(h_y^2)$. Тогда $\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h_x^2} + \frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h_y^2} + O\left(h_x^2 + h_y^2\right) = 0$, где $h_x = h_y = h = 1/N$. Получаем СЛАУ которая имеет пяти-диагональный вид: $u_{i-1,j} + u_{i+1,j} - 4u_{ij} + u_{i,j+1} + u_{i,j-1}$. В результате получена СЛАУ, содержащая $(N_x + 1)\left(N_y + 1\right)$ уравнений относительно неизвестных u_{ij}:

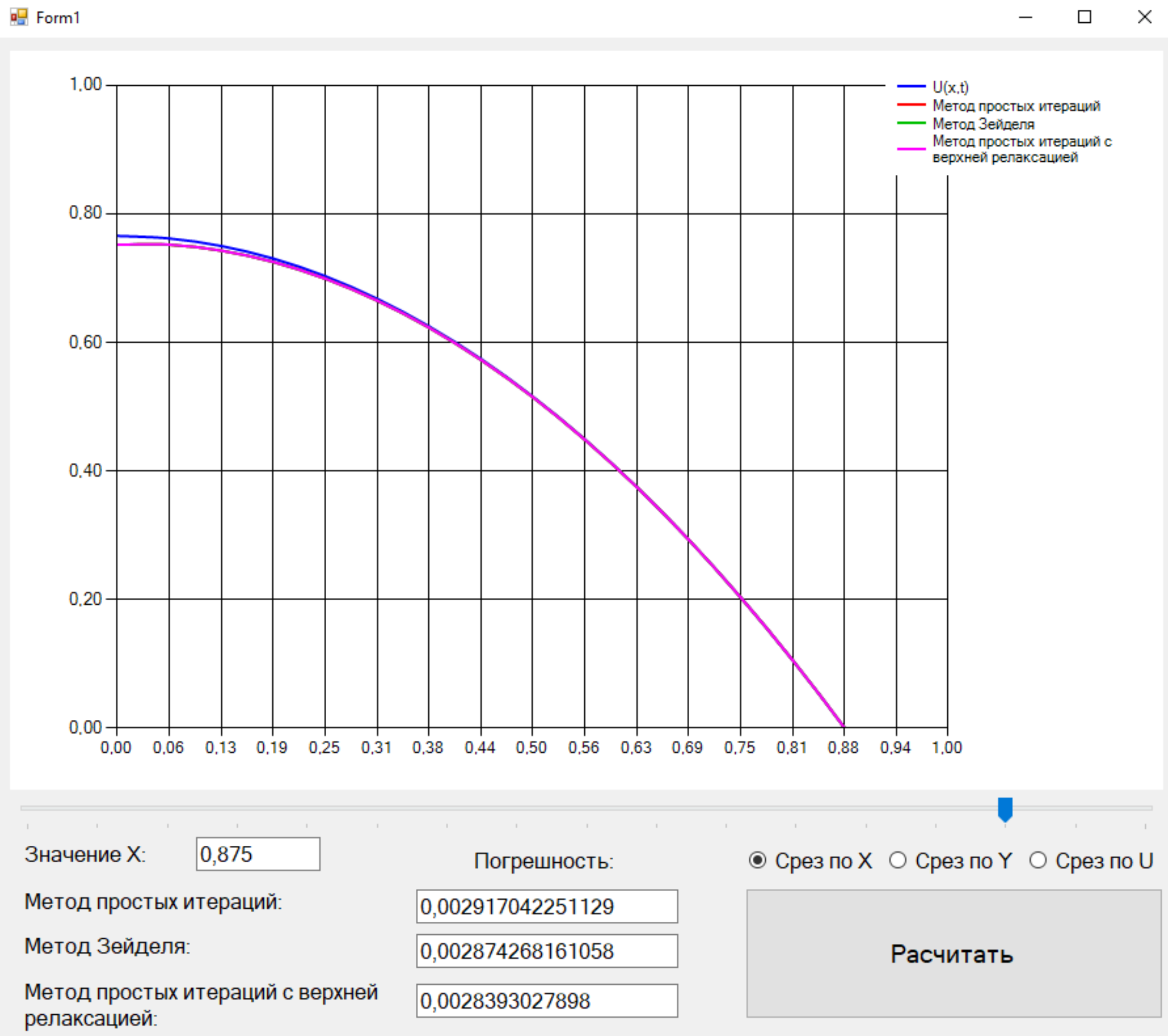
$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} = \begin{pmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ h^2 - 1 \\ 0 \\ 0 \\ 0 \\ 4h^2 - 1 \\ 1 \\ 1 - h^2 \\ 1 - 4h^2 \\ 0
\end{pmatrix}$$

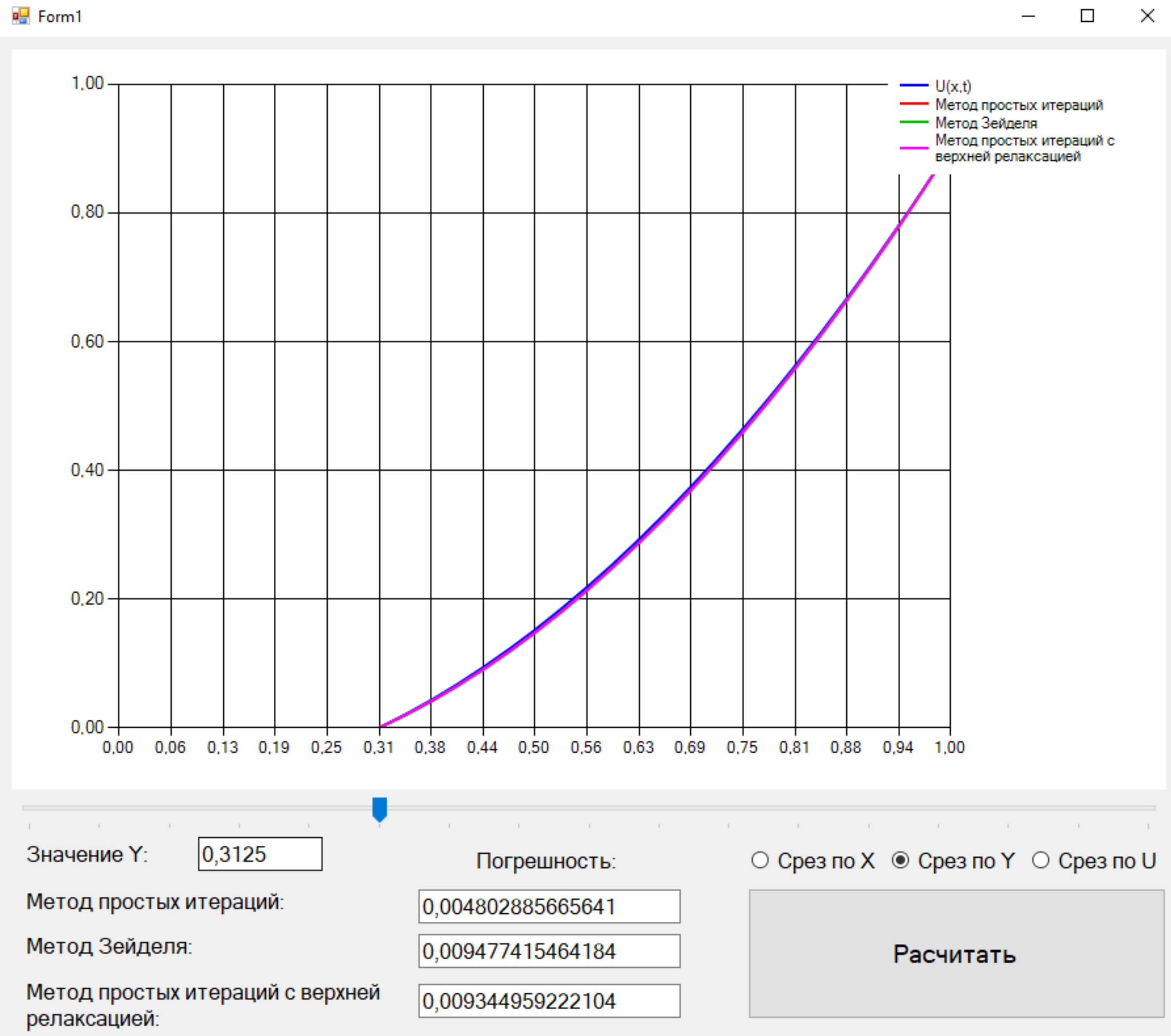
Код :

```
private void CRS()
{
    double h = l / N;
    double[,] u = new double[K + 1, N + 1];
    double[,] matr = new double[(N + 1) * (N + 1), (N + 1) * (N + 1)];
    double[] d = new double[(N + 1) * (N + 1)];
    for (int j = 0; j <= N - 1; j++)
    {
        matr[j, j] = -1; matr[j, N + 1 + j] = 1;
    }
    matr[N, N] = 1; d[N] = - 1;
    matr[(N + 1) * (N + 1) - 1, (N + 1) * (N + 1) - 1] = 1; d[(N + 1) * (N + 1) - 1] = 0;
    for (int j = (N+1)*(N+1)-2; j > (N + 1) * (N + 1) - N - 2; j--)
    {
        matr[j, j] = 1;
        d[j] = 1 - Math.Pow(h * (j - (N + 1) * (N + 1) + N + 1), 2);
    }
    for (int i = 1; i <= N - 1; i++)
    {
```

```
            matr[(N + 1) * i, (N + 1) * i] = -1;
            matr[(N + 1) * i, (N + 1) * i + 1] = 1;
            matr[(N + 1) * i + N, (N + 1) * i + N] = 1;
            d[(N + 1) * i + N] = Math.Pow(h * i, 2) - 1;
            for (int j = 1; j <= N - 1; j++)
            {
                matr[(N + 1) * i + j, (N + 1) * i - N - 1 + j] = 1;
                matr[(N + 1) * i + j, (N + 1) * i + N + 1 + j] = 1;
                matr[(N + 1) * i + j, (N + 1) * i + j] = -4;
                matr[(N + 1) * i + j, (N + 1) * i - 1 + j] = 1;
                matr[(N + 1) * i + j, (N + 1) * i + 1 + j] = 1;
            }
        }
        double[] xP = Prost(matr, d);
        double[] xZ = Zeydel(matr, d);
        double[] xPU = ProstUpgrade(matr, d);
        for (int i = 0; i < N + 1; i++)
        {
            for (int j = 0; j < N + 1; j++)
            {
                U[i, j] = xP[(N + 1) * i + j];
                dt[i] += Math.Abs(U[i, j] - f(h * i, h * j));
                U2[i, j] = xZ[(N + 1) * i + j];
                dt2[i] += Math.Abs(U2[i, j] - f(h * i, h * j));
                U3[i, j] = xPU[(N + 1) * i + j];
                dt3[i] += Math.Abs(U3[i, j] - f(h * i, h * j));
            }
        }
        Console.Write(Pcount + "\t" + Zcount + "\t" + PUcount);
        Console.WriteLine();
    }
```

Результаты:





Вывод: Мной было реализовано решение краевой задачи для дифференциального уравнения эллиптического типа, с использованием центрально-разностной схемы и применяя следующие методы: метод простых итераций (метод Либмана), метод Зейделя, метод простых итераций с верхней релаксацией, а также вычислена погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением.