```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace LaplaceSolution
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            GraphicalSetup gs = new GraphicalSetup(this);
            FreeSpace region = new FreeSpace(gs);
            do
            {
                region.Relaxation(gs);
                textBox1.Text = region.sigmav.ToString();
                textBox1.Refresh();
            } while (region.sigmav > region.deltav);
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace LaplaceSolution
{
    class FreeSpace
    {
        public float lbv, rbv, ubv, bbv, t, delt, sigmav,deltav,ave;
        int size;
        float[,] region; float[,] oldv;
        public FreeSpace(GraphicalSetup gs)
        {
            size = 10; deltav = 0.0005f; ave = 0;
            lbv = -1; rbv = 1; ubv = 1; bbv = -1;
            region = new float[size, size];
            oldv = new float[size, size];

            for (int i = 0; i < size; i++)
            {
                for (int j = 0; j < size; j++)
                {


                    region[0, j] = ubv;
                    region[size - 1, j] = bbv;
```

```
            if (i != 0 && j != 0 && i != size - 1 && j != size - 1)
                region[i, j] = 0;

        }
        region[i, 0] = lbv;
        region[i, size - 1] = rbv;
    }
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            //gs.gg.FillEllipse(gs.bred, 300 + j* 20, 300 + i * 20, 6, 6);
            gs.gg.DrawString(region[i, j].ToString(), gs.f, gs.bblue, 160 + j * 70,
                40 + i * 70);
            System.Threading.Thread.Sleep(1);
        }
    }
    t = 0; delt = 0.001f; sigmav = 0.001f;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            oldv[i, j] = region[i, j];
        }
    }
}
public void Relaxation(GraphicalSetup gs)
{
    ave = 0;
    for (int i = 1; i < size - 1; i++)
    {
        for (int j = 1; j < size - 1; j++)
        {
            gs.gg.DrawString(region[i, j].ToString(), gs.f, gs.bwhite, 160 + j * 70,
              40 + i * 70);
            region[i, j] = (region[i - 1, j] + region[i + 1, j]
                + region[i, j - 1] + region[i, j + 1]) / 4;
            System.Threading.Thread.Sleep(1);
            gs.gg.DrawString(Math.Round(region[i, j],2).ToString(), gs.f, gs.bblue, 160 + j * 70,
               40 + i * 70);
            ave = ave + region[i, j];
             }
    }//loop ends
    ave=ave/(size*size-4*size+4);


}//relaxation ends
public void sigma()
{
    sigmav=0;
    for (int i = 1; i < size - 1; i++)
    {
        for (int j = 1; j < size - 1; j++)
        {
            sigmav = sigmav + (region[i, j] - ave) * (region[i, j] - ave);
        }
    }//loop ends
    sigmav=(float)(Math.Sqrt(sigmav/(size*size-4*size+4)));
```

```
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;

namespace LaplaceSolution
{
    class GraphicalSetup
    {
        //Data
        public Graphics gg;
        public SolidBrush bred, bblue, bwhite;
      public  Pen pblue, pred, pwhite;
       public Font f;
        public float x0, y0;
        //constructor
        public GraphicalSetup(Form1 frm)
        {
            gg = frm.CreateGraphics();
            bred = new SolidBrush(Color.Red);
            bblue = new SolidBrush(Color.Blue);
            bwhite = new SolidBrush(Color.White);
            pred = new Pen(Color.Red);
            pblue = new Pen(Color.Blue);
            pwhite = new Pen(Color.White);
            f = new Font("Arial",16);
            x0 = frm.ClientSize.Width / 2;
            y0 = frm.ClientSize.Height / 2;
        }
        //Other Functions
        public void DrawAxes(float xlen,float ylen,
            string xlabel,string ylabel,float deltax,
            float deltay)
        {
            gg.DrawLine(pred, x0, y0, x0 + xlen, y0);
            gg.DrawLine(pred, x0, y0, x0, y0 - ylen);
            gg.DrawString(xlabel,f,bblue,
                x0 + xlen / 2, y0 + 10);
            gg.DrawString(ylabel, f, bblue,
               x0 -50, y0-ylen/2);

        }
        public void plotter(float hv,float vv,
            float hscale,float vscale)
        {
            gg.FillEllipse(bred, x0 + hv * hscale,
                y0 - vv * vscale-5, 5, 5);
        }
        public void eraser(float hv, float vv,
            float hscale, float vscale)
        {
            gg.FillEllipse(bwhite, x0 + hv * hscale,
```

```
                y0 - vv * vscale-5, 5, 5);
        }
    }
}
```