Power GUI 2019

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


namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            comboBox1.Text = "Active";
            comboBox2.Text = "Lagging";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            progressBar1.Value = 0;
            //checking valid input
            string Text1 = Not_Empty(textBox1.Text);
            string Text2 = Not_Empty(textBox2.Text);
            string Text3 = Not_Empty(textBox3.Text);
            string Text4 = Not_Empty(textBox4.Text);
            string Text5 = Not_Empty(textBox5.Text);
            string Text6 = Not_Empty(textBox6.Text);
            string Text7 = Not_Empty(textBox7.Text);

            if (progressBar1.Value == 70)
            {
            //saving inputs from user
            double R = Convert.ToDouble(Text1);
            double L = Convert.ToDouble(Text2);
            double Cc = Convert.ToDouble(Text3);
            double lenght = Convert.ToDouble(Text4);

            double Vrr   = Convert.ToDouble(Text7) * Math.Pow(10, 3) / Math.Sqrt(3);
            double power = Convert.ToDouble(Text6) * Math.Pow(10, 3);
            double pf    = Convert.ToDouble(Text5);

            //Complex parameters variables
            double w = 2 * 3.14 * 60;
            double LL = w * L * lenght * Math.Pow(10, -3);
            Complex Z = new Complex(R * lenght , LL);
            double YY = w * Cc * lenght * Math.Pow(10, -6);
            Complex Y = new Complex(0, YY);
```

```
            //MessageBox.Show(" Z= " + Convert.ToString(Z) + ", Y= " +
Convert.ToString(Y));//impedances test

            //Sending voltage & current variables
            double Vss;
            Complex Vs;

            double Iss;
            Complex Is;

            //power variables
            double dir;
            double Pr;
            double Ps;

            double VR;
            double efficiency;

            //pf type
            if (comboBox2.Text == "Lagging") dir = -1;
            else if (comboBox2.Text == "Unity") { dir = 1; textBox5.Text = "1"; }
            else dir = 1;

            //power type
            switch (comboBox1.Text)
                {
                case "Active":
                    Pr = power;
                    break;
                case "Reactive":
                    Pr = power /  Math.Sin(Math.Acos(pf)) * pf ;
                    break;
                case "Apparent":
                    Pr = power * pf;
                    break;
                default:
                    Pr = 1;
                    MessageBox.Show("Something is Wrong","Warning");
                    break;
                }

            Complex Vr = new Complex(Vrr, 0);
            double Irr = Pr / (3 * Vrr * pf);
            Complex Ir = new Complex(Irr * pf, Irr * dir * Math.Sin(Math.Acos(pf)) );
            //MessageBox.Show(" Vr= " + Convert.ToString(Vr) + ", Ir= " +
Convert.ToString(Ir));//Vr & Ir test

            //ABCD parameters calculation
            Complex A,B,C,D;
            if (lenght < 80)        //short model
            {
                A = D = 1; B = Z; C = 0;
            }
            else if(lenght < 250)   //medium model
            {
                A = D = (1 + Z * Y / 2); B = Z; C = Y * (1 + Z * Y / 4);
            }
            else                    //long model
```

```
            {
                //new parameters
                int n = 2; double new_lenght=lenght;
                while (new_lenght > 250)
                    {
                        new_lenght = lenght / n;
                        n++;
                    } n--;

                LL = w * L * new_lenght * Math.Pow(10, -3);
                Z = new Complex(R * new_lenght, LL);
                YY = w * Cc * new_lenght * Math.Pow(10, -6);
                Y = new Complex(0, YY);
                A = (1 + Z * Y / 2); C = Y * (1 + Z * Y / 4);

                Complex[,] Matrix = { { A,Z }, { C,A } }; //initial matrix
                //MessageBox.Show("A= " + Convert.ToString(A) + ", C= " +
Convert.ToString(C) + ", B= " + Convert.ToString(Z));

                //Multiple Matrix Multiplication
                Complex[,] Matrix2 = new Complex[2, 2];  //temp matrix
                Complex[,] Matrix3 = new Complex[2, 2];  //final result matrix
                Matrix2 = Matrix;
                for (int h = 1; h < n; h++)
                {
                    for (int i = 0; i < 2; i++)
                    {
                        for (int j = 0; j < 2; j++)
                        {
                            Matrix3[i, j] = 0;
                            for (int k = 0; k < 2; k++)
                            {
                                Matrix3[i, j] += Matrix2[i, k] * Matrix[k, j];
                            }
                        }
                    }
                    Matrix2 = Matrix3;
                }

                //final matrix values
                A = Matrix3[0,0]; B = Matrix3[0,1];
                C = Matrix3[1,0]; D = Matrix3[1,1];
            }
            //MessageBox.Show("A= " + Convert.ToString(A) + ", C= " + Convert.ToString(C)
+ ", B= " + Convert.ToString(B) + ", D= " + Convert.ToString(D));

            //Vs & Is Calculation
            Vs = A * Vr + B * Ir;
            Is = C * Vr + D * Ir;

            Vss = Complex.Abs(Vs);
            Iss = Complex.Abs(Is);

            Ps = 3 * Vss * Iss * Math.Cos(Vs.Phase - Is.Phase) ;
            //VR & Efficiency Calculation
            VR = (Vss/Complex.Abs(A) - Vrr) / Vrr * 100;
            VR = Math.Round(VR, 2);
            efficiency = Pr / Ps * 100 ;
```

```
        efficiency = Math.Round(efficiency, 2);

        //Rounding to 2 decimal places
        Vss = Math.Round(Vss / 1000, 2);
        double Vphase = Math.Round(Vs.Phase * 180.00 / 3.14, 2);
        Iss = Math.Round(Iss, 2);
        double Iphase = Math.Round(Is.Phase * 180.00 / 3.14, 2);


        //display
        ListViewItem lv1 = new ListViewItem((Vss).ToString() + " Kv");
        lv1.SubItems.Add((Vphase).ToString());
        lv1.SubItems.Add(Iss.ToString() + " A");
        lv1.SubItems.Add((Iphase).ToString());
        lv1.SubItems.Add(efficiency.ToString() + "%");
        lv1.SubItems.Add(VR.ToString() + "%");
        listView1.Items.Add(lv1);
        }
        button2.Enabled = true;

    }

    string Not_Empty(string s)
    {
        if (s == "")
            MessageBox.Show("Enter missing value", "Alert");
        else
        {
            try
            {
                Convert.ToDouble(s);
                progressBar1.PerformStep();
            }
            catch
            {
                MessageBox.Show("Enter numbers only", "Alert");
            }
        }
        return s;
    }


    private void button2_Click(object sender, EventArgs e)
    {
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
        textBox5.Text = "";
        textBox6.Text = "";
        textBox7.Text = "";
        progressBar1.Value = 0;
        button2.Enabled = false;
    }

    private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (comboBox2.Text == "Unity") { textBox5.Text = "1"; }
```

```
        }

        private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
        {
            switch (comboBox3.Text)
            {
                case "Ex1: Small":
                    //small test
                    textBox1.Text = "0.11";
                    textBox2.Text = "1.11";
                    textBox3.Text = "0";
                    textBox4.Text = "50";
                    textBox5.Text = "0.8";
                    textBox6.Text = "20000";
                    textBox7.Text = "69";
                    break;

                case "Ex2: Medium":
                    //medium test
                    textBox1.Text = "0.035";
                    textBox2.Text = "0.9";
                    textBox3.Text = "0.015";
                    textBox4.Text = "200";
                    textBox5.Text = "0.8";
                    textBox6.Text = "400000";
                    textBox7.Text = "380";
                    break;

                case "Ex3: Long":
                    //Long test
                    textBox1.Text = "0.0125";
                    textBox2.Text = "0.35";
                    textBox3.Text = "0.0050";
                    textBox4.Text = "400";
                    textBox5.Text = "0.8";
                    textBox6.Text = "900000";
                    textBox7.Text = "500";
                    break;

                default:
                    //MessageBox.Show("No preset", "Warning");
                    break;
            }
            progressBar1.Value = 0;
        }
    }
}
```