实验二 卫星位置的计算

一、实验目的

1、熟悉 GPS 星历

2、利用 GPS 星历计算卫星位置

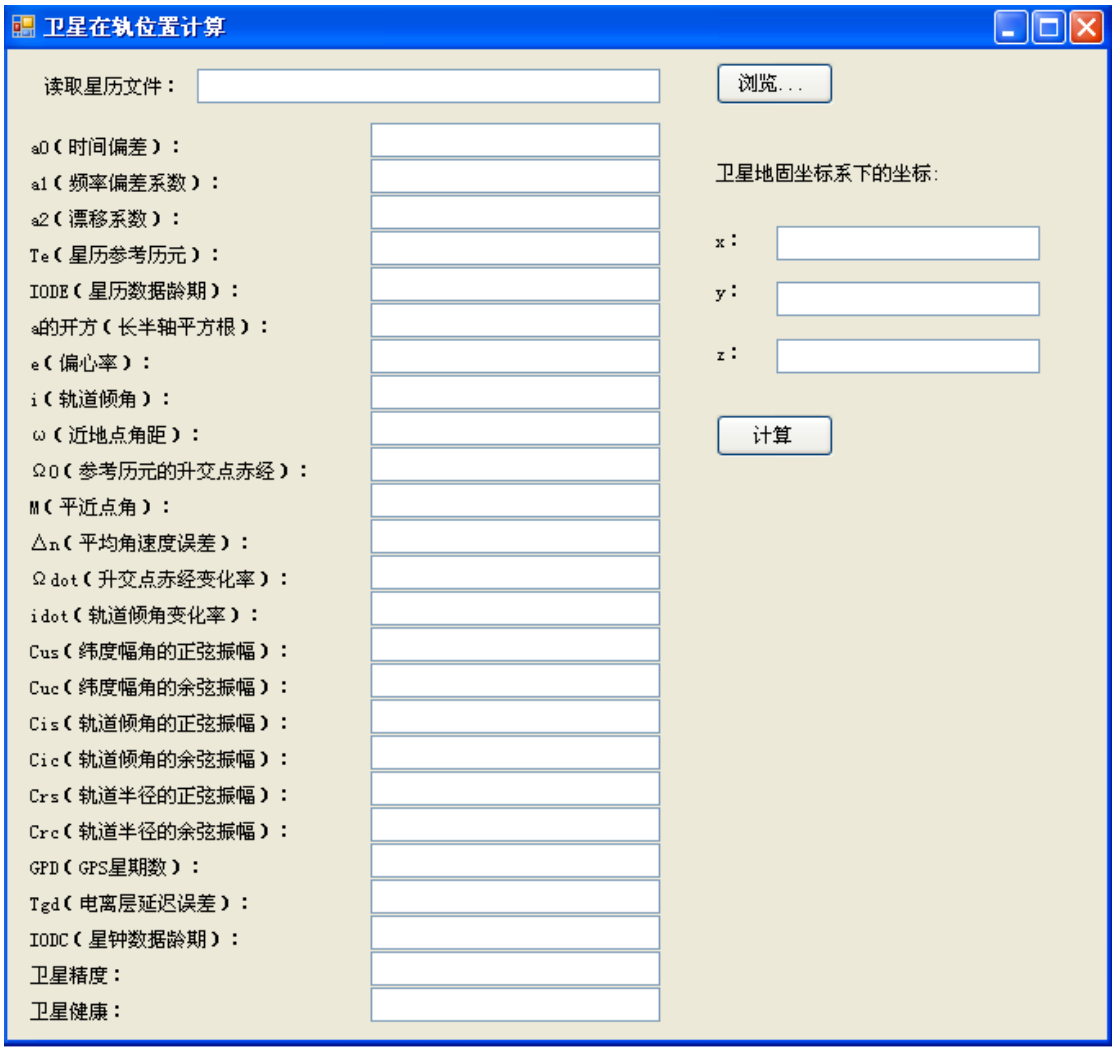
二、实验准备

1、台式电脑每人一台。

2、Visual studio2010 软件提前安装。

三、实验步骤

1、设置卫星位置计算的界面：



2、运用课本里面 48-50 页的公式，完成卫星位置的计算。

参考代码如下：

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

```
using System.IO;

namespace SatellitePositionComputation
{
    public partial class Form1 : Form
    {
        float u = float.Parse("0.3986005e+15");
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            openFileDialog1.DefaultExt = "txt";
            openFileDialog1.Filter = "文本文件  (*.txt)|*.txt";
            openFileDialog1.FileName = "";
            //openFileDialog1.InitialDirectory    = "c:\\";
            openFileDialog1.Title = "打开星历文件";

            if (openFileDialog1.ShowDialog()==System.Windows.Forms.DialogResult.OK    )
            {
                try
                {
                    textBox1.Text = openFileDialog1.FileName;
                    if (File.Exists(openFileDialog1.FileName))
                    {
                        string[] gData = File.ReadAllLines(openFileDialog1.FileName,
Encoding.Default);
                        a0.Text = gData[1];
                        a1.Text = gData[2];
                        a2.Text = gData[3];
                        te.Text = gData[4];
                        iode.Text = gData[5];
                        aRoot.Text = gData[6];
                        e0.Text = gData[7];
                        i0.Text = gData[8];
                        w0.Text = gData[9];
                        ascendingNode.Text = gData[10];
                        M.Text = gData[11];
                        meanMotionD.Text    = gData[12];
                        RateascendingN.Text = gData[13];
                        Ratei.Text = gData[14];
                        cus.Text = gData[15];
```

```
                    cuc.Text = gData[16];
                    cis.Text = gData[17];
                    cic.Text = gData[18];
                    crs.Text = gData[19];
                    crc.Text = gData[20];
                    gpd.Text = gData[21];
                    tgd.Text = gData[22];
                    iodc.Text = gData[23];
                    psatellite.Text = gData[24];
                    hsatellite .Text = gData[25];


                }
            }
            catch
            {
                MessageBox.Show("读取文本文件出现错误！");
                Console.WriteLine("The process failed: {0}", e.ToString());
            }
        }

}

private void comp_Click(object sender, EventArgs e)
{
    try
    {
        //mean angular speed n
        double    n = Math.Pow(u,0.5)/Math.Pow(float.Parse(aRoot.Text ),3);
        n = n + double.Parse(meanMotionD.Text);
        //time tk
        double tk = 0;
        //mean anomaly Mk
        double Mk = double.Parse(M.Text )+n*tk;
        //Eccentric anomaly Ek
        double Ek=Mk;
        double Ekt =0;
        while(Math.Abs( Ek -Ekt) >0.000000000001)
        {
            Ekt = Ek;
            Ek = Mk + double.Parse(e0.Text )*Math.Sin(Ekt );
        }
        //true anomaly Vk
        double e00 = double.Parse(e0.Text );
```

```
                double Vk =
Math.Atan(Math.Pow(1-Math.Pow(e00,2 ),0.5)*Math.Sin(Ek)/(Math .Cos(Ek )-e00));
                //argument of satellite and ascension node
                double Qk = Vk + double.Parse(w0.Text );
                //diturbed satellite motion correction term
                double Cu = double.Parse(cuc.Text) *
Math.Cos(2*Qk)+double.Parse(cus.Text )*Math.Sin(2*Qk );
                double Cr = double.Parse(crc.Text) *
Math.Cos(2*Qk)+double.Parse(crs.Text )*Math.Sin(2*Qk);
                double Ci = double.Parse(cic.Text) *
Math.Cos(2*Qk)+double.Parse(cis.Text )*Math.Sin(2*Qk);

                //argument of latitude
                double Uk = Qk + Cu;
                double Rk = Math.Pow(double.Parse(aRoot.Text),2 )*(1-e00*Math.Cos(Ek ))+Cr;
                double Ik = double.Parse(i0.Text )+double.Parse (Ratei.Text )*tk +Ci ;

                //orbit plane coordinate
                double OrbitX = Rk * Math.Cos(Uk );
                double OrbitY = Rk * Math.Sin(Uk );

                //longitude of ascension node
                double we = double.Parse("7.29211567E-5");
                double W = double.Parse(ascendingNode.Text) + (double.Parse(RateascendingN.Text) -
we) * tk - we * double.Parse(te.Text);

                //Earth coordinate
                double EarthX= OrbitX*Math.Cos(W)-OrbitY*Math.Cos(Ik)*Math.Sin(W);
                double EarthY= OrbitX*Math.Sin(W)+OrbitY*Math.Cos(Ik)*Math.Cos(W);
                double EarthZ = OrbitY * Math.Sin(Ik );

                xcoord.Text = EarthX.ToString();
                ycoord.Text = EarthY.ToString();
                zcoord.Text = EarthZ.ToString();

                //

            }
            catch
            {
                Console.WriteLine("此操作失败:{0}",e.ToString());
            }
        }
    }
```

}