using System.Data.OleDb;//添加类库

using System.IO;//添加类库 输入输出

using Excel=Microsoft.Office.Interop.Excel;

//封装角度转弧度

public double dmstorad(string s)

{ string[] ss = s.Split(new char[3] { '°', '′', '″' }, StringSplitOptions.RemoveEmptyEntries);

double[] d = new double[ss.Length];

for (int i = 0; i < d.Length; i++)

d[i] = Convert.ToDouble(ss[i]);

double sign = d[0] >= 0.0 ? 1.0 : -1.0;

double rad = 0;

if (d.Length == 1)

rad = (Math.Abs(d[0])) \* Math.PI / 180;

else if (d.Length == 2)

rad = ((Math.Abs(d[0])) + d[1] / 60) \* Math.PI / 180;

else

rad = ((Math.Abs(d[0])) + d[1] / 60 + d[2] / 60 / 60) \* Math.PI / 180;

rad = sign \* rad;

return rad;}

//弧度转角度

public string radtodms(double rad)

{double sign = rad >= 0.0 ? 1.0 : -1.0;

rad = Math.Abs(rad) \* 180 / Math.PI;

double[] d = new double[3];

d[0] = (int)rad;

d[1] = (int)((rad - d[0]) \* 60);

d[2] = (rad - d[0] - d[1] / 60) \* 60 \* 60;

d[2] = Math.Round(d[2], 2);

if (d[2] == 60)

{d[1] += 1;

d[2] -= 60;

if (d[1] == 60)

{ d[0] += 1;

d[1] -= 60;}}

d[0] = sign \* d[0];

string s = Convert.ToString(d[0]) + "°" + Convert.ToString(d[1]) + "′" + Convert.ToString(d[2]) + "″";

return s; }

//坐标方位角推算

public double fangweijiao(double[] sdr, double[] cr)

{ double sum = 0;

for (int i = 1; i < sdr.Length ; i++)//从第二行开始循环计算 坐标方位角、观测角度累加值

{ cr[i] = cr[i - 1] + sdr[i] - Math.PI;//计算坐标方位角/左角

if (cr[i] >= Math.PI \* 2)//判断坐标方位角是否在0到2PI之间

cr[i] -= Math.PI \* 2;

else if (cr[i] < 0.0) cr[i] += Math.PI \* 2;

sum += sdr[i];}

return sum;}

private void button1\_Click(object sender, EventArgs e)

{

string[] sd = new string[dataGridView1.RowCount - 5]; //新建一个数组存放观测角度的 原始值

double[] sdr = new double[sd.Length]; //新建一个数组存放观测角度的弧度值

double[] cr = new double[sd.Length ]; //新建一个数组存放计算的坐标方位角

double sum = 0;

cr[0] = dmstorad(Convert.ToString(dataGridView1.Rows[0].Cells[4].Value)); //获取第一个坐标方位角，并将其转换成弧度，放入cr[]数组第一个元素中

double acd = dmstorad(Convert.ToString (dataGridView1.Rows[dataGridView1.RowCount-6].Cells[4].Value)); //获取终边坐标方位角，并将其转换成弧度，放入放入acd中用于计算和检核

for (int i = 1; i < sd.Length; i++) //从第二行开始循环，将观测角度的原始值放入 sd[]数组中,并转换成弧度值存放在sdr数组中

{ sd[i] = Convert.ToString(dataGridView1.Rows[i].Cells[1].Value);

sdr[i] = dmstorad(sd[i]); }

sum = fangweijiao(sdr, cr); //计算改正前坐标方位角和观测角度总和，分别存储在 cr数组和sum中

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[1].Value = radtodms(sum); //将观测角度总和放入表格中

double fd, fdx;

fd = cr[cr.Length - 1] - acd;//计算角度闭合差，单位弧度

fdx = 60 \* Math.Sqrt(sd.Length - 1);//计算角度闭合差限差，单位秒

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[1].Value = Convert.ToString(Math.Round(fd \* 180 / Math.PI \* 3600, 2))+"″"; //将角度闭合差存入表格中

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[1].Value = Convert.ToString(Math.Round(fdx, 2))+"″";//将角度闭合差限差存入表格中

if (Math.Abs(fd \* 180 / Math.PI \* 3600) > fdx)//检查角度闭合差是否满足要求

MessageBox.Show("角度闭合差超限！"); else { double vd = -fd / (sd.Length - 1);//分配角度闭合差（观测左角）

double sumvd = 0; for (int i = 1; i < sdr.Length; i++) { sdr[i] += vd;//计算改正后的观测角度，并存入sdr数组中

sumvd += vd; dataGridView1.Rows[i].Cells[2].Value = Convert.ToString(Math.Round(vd \* 180 / Math.PI \* 3600, 2))+"″"; //将角度改正数存入表格中

dataGridView1.Rows[i].Cells[3].Value = radtodms(sdr[i]); }

if (Math.Round(sumvd, 8) != Math.Round(-fd, 8)) //秒保留2位对应弧度是8位

MessageBox.Show("角度改正数分配有误！");

else dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[2].Value = Convert.ToString(Math.Round(sumvd \* 180 / Math.PI \* 3600, 2)) + "″"; //将角度改正数总和存入表格中

sum = fangweijiao(sdr, cr);//推算改正后的坐标方位角

if (Math.Round(cr[cr.Length - 1], 8) != Math.Round(acd, 8)) MessageBox.Show("坐标方位角推算有误！");

else

{

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[3].Value = radtodms(sum); //将改正后观测角度总和放入表格中

for (int i = 1; i < cr.Length-1; i++)//将改正后坐标方位角存入表格

dataGridView1.Rows[i].Cells[4].Value = radtodms(cr[i]); }//至此角度调整和计算完毕

double [] sl = new double[sd.Length - 1];

double suml = 0;

double [] Δx = new double[sd.Length - 1];

double [] Δy = new double[sd.Length - 1];

double sumdx = 0;

double sumdy = 0;

for (int i = 1;i< sl.Length;i++){

sl[i] = Convert .ToDouble (dataGridView1.Rows[i].Cells[5].Value);// 将观测距离放到 sl 数组中

suml += sl[ i ]; // 计算距离总和

Δx[ i ] = sl[ i ] \*Math.Cos ( cr[ i ]); // 利用距离和坐标方位角计算坐标增量

Δy[ i ] = sl[ i ] \* Math . Sin ( cr[ i ]);

sumdx += Δx[ i ]; // 计算坐标增量总和

sumdy += Δy[ i ];

dataGridView1.Rows[ i ].Cells[6].Value = Convert .ToString( Math .Round ( Δx[ i ],3 ));// 将坐标增量放入表格

dataGridView1.Rows[ i ].Cells[7].Value = Convert .ToString(Math .Round ( Δy[ i ],3));}

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[5].Value = Convert .ToString( suml );// 将距离总和放入表格中

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[6].Value = Convert .ToString( Math . Round (sumdx,3)); // 将坐标增量总和放入表格中

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[7].Value = Convert .ToString(Math .Round (sumdy,3));

double fx , fy , fxy , k1;

double x1, y1, x2, y2; // 存放已知两个点的x x ，y y 坐标

x1 = Convert.ToDouble(dataGridView1.Rows[1].Cells[12].Value);

y1 = Convert.ToDouble(dataGridView1.Rows[1].Cells[13].Value);

x2 = Convert.ToDouble(dataGridView1.Rows[sd.Length - 1].Cells[12].Value);

y2 = Convert.ToDouble(dataGridView1.Rows[sd.Length - 1].Cells[13].Value);

fx = sumdx - (x2 - x1); // 计算坐标增量闭合差

fy = sumdy - (y2 - y1);

fxy = Math .Sqrt(fx \* fx + fy \* fy ); // 计算导线全长闭合差

k1 = suml / fxy ; // 计算导线全长相对闭合差分母

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[7].Value =Convert .ToString( Math .Round (fx,3));

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[7].Value =Convert .ToString( Math .Round (fy,3));

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[10].Value =Convert .ToString( Math .Round( fxy , 3));

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[11].Value =Convert .ToString (( int )k1);//导线全长相对闭合差分母取整

double [] vx = new double[sd.Length - 1]; // 定义数组用于存放坐标增量的改正数及总和

double [] vy = new double[sd.Length - 1];

double sumvx = 0;

double sumvy = 0;

double [] cx = new double[sd.Length - 1]; // 定义数组用于存放改正后的坐标增量及总和

double [] cy = new double[sd.Length - 1];

double sumcx = 0;

double sumcy = 0;

double [] x = new double[sd.Length - 1]; // 定义数组用于存放x , y 坐标

double [] y = new double[sd.Length - 1];

x[1] = x1;

y[1] = y1;

if (k1 < 2000 ) // 判断导线全长相对闭合差是否超限

MessageBox .Show( " 导线全长相对闭合差超限！" );

else{

for(int j=1;j< vx.Length;j++){

vx [j] = -fx \* sl [j] / suml ; // 计算坐标增量改正数

vy [j] = -fy \* sl [j] / suml; ;

sumvx += vx [j]; // 计算坐标增量改正数总和

sumvy += vy [j];

dataGridView1.Rows[j].Cells[8].Value = Convert .ToString( Math .Round( vx [j], 4 ));// 将坐标增量改正数放入表格

dataGridView1.Rows[j].Cells[9].Value = Convert .ToString( Math .Round( vy [j], 4));

cx[j] = Δx[j] + vx [j]; // 计算改正后坐标增量

cy[j] = Δy[j] + vy[j];

sumcx += cx[j]; // 计算改正后坐标增量总和

sumcy += cy[j];

dataGridView1.Rows[j].Cells[10].Value = Convert .ToString(Math .Round (cx[j], 3));// 将改正后坐标增量放入表格

dataGridView1.Rows[j].Cells[11].Value = Convert .ToString(Math .Round (cy[j], 3));}

if ( Math .Round(sumvx , 4) != Math .Round(- fx , 4) || Math .Round(sumvy , 4) !=Math .Round(- fy , 4))

MessageBox .Show(" 坐标增量分配有误！ " );

if ( Math .Round(sumcx , 4) != Math .Round (x2 - x1, 4) || Math .Round(sumcy , 4) !=Math .Round (y2 - y1, 4))

MessageBox .Show( " 改正后坐标增量计算有误！" );

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[8].Value =

Convert .ToString( Math .Round( sumvx , 3)); // 将坐标增量改正数总和放入表格中

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[9].Value =

Convert .ToString( Math .Round( sumvy , 3));

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[10].Value =

Convert .ToString( Math .Round( sumcx , 3)); // 将改正后坐标增量总和放入表格中

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[11].Value =

Convert .ToString( Math .Round( sumcy , 3));

for(int j=1;j< x.Length- 1;j++ ){

x[j + 1] = x[j] + cx[j]; // 计算 x,y 坐标

y[j + 1] = y[j] + cy[j];

dataGridView1.Rows[j+1].Cells[12].Value= Convert .ToString( Math .Round (x[j+1], 3));// 将 x,y 坐标放入表格

dataGridView1.Rows[j+1].Cells[13].Value= Convert .ToString( Math .Round (y[j+1], 3));}

if(Math .Round (x[x.Length - 1]+cx[cx.Length - 1],3)!= Math .Round (x2,3) || Math .Round

(y[y.Length - 1]+cy[cy.Length - 1],3)!=Math .Round ( y2,3))

MessageBox .Show(" 坐标计算有误！");

} }}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();//关闭

}

private void excel导入ToolStripMenuItem\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = null;

dataGridView1.Rows.Clear();

dataGridView1.Columns.Clear();

OpenFileDialog file = new OpenFileDialog();

file.Filter = "Excel文件|\*.xls|Excel文件|\*.xlsx";

if (file.ShowDialog() == DialogResult.OK)

{

string fname = file.FileName;

string strSource = "provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=" + fname + ";Extended Properties='Excel 8.0;HDR=Yes;IMEX=1'";

OleDbConnection conn = new OleDbConnection(strSource);

string sqlstring = "SELECT \* FROM [Sheet1$]";

OleDbDataAdapter adapter = new OleDbDataAdapter(sqlstring, conn);

DataSet da = new DataSet();

adapter.Fill(da);

dataGridView1.DataSource = da.Tables[0];

}

else return;

}

private void txt导入ToolStripMenuItem\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = null;

dataGridView1.Rows.Clear();

dataGridView1.Columns.Clear();

OpenFileDialog file = new OpenFileDialog();

file.Filter = "文本文件|\*.txt";

if (file.ShowDialog() == DialogResult.OK)

{ StreamReader sr = new StreamReader(file.FileName, System.Text.Encoding.Default);

textBox1.Text = sr.ReadToEnd();

sr.Close(); } else return;

string[] str = textBox1.Text.Split(new string[] { "\r\n" }, StringSplitOptions.RemoveEmptyEntries);

string[][] k = new string[str.Length][];

for (int i = 0; i < str.Length; i++) k[i] = str[i].Split(',');

dataGridView1.RowCount = k.Length;

dataGridView1.ColumnCount = k[0].Length;

for (int i = 0; i < k[0].Length; i++)

dataGridView1.Columns[i].HeaderText = k[0][i];

for (int i = 1; i < k.Length; i++)

{ for (int j = 0; j < k[i].Length; j++) dataGridView1.Rows[i-1].Cells[j].Value = k[i][j];

}

}

private void excel文件导出ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Excel.Application ex = new Excel.Application(); //声明一个Excel.Application对象

ex.Visible = true; //使ex可见

ex.Application.Workbooks.Add(true); //在ex中增加一个工作簿

for (int i = 0; i < dataGridView1.ColumnCount; i++) //把dataGridView1中的列名存入ex中

{ ex.Cells[1, i + 1] = dataGridView1.Columns[i].HeaderText; }

for (int i = 0; i < dataGridView1.RowCount; i++) //把dataGridView1中的数据存入ex中

{ for (int j = 0; j < dataGridView1.ColumnCount; j++)

ex.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value; }

MessageBox.Show(“数据输出已完成!");

}

}

}