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;

using System.IO;

using System.Data.OleDb;

namespace WindowsFormsApplication4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

string[] sd = new string[dataGridView1.RowCount - 5];

double[] sdr = new double[sd.Length];

double[] cr = new double[sd.Length];

double sum = 0;

cr[0] = dmstorad(Convert.ToString(dataGridView1.Rows[0].Cells[4].Value));

double acd = dmstorad(Convert.ToString(dataGridView1.Rows[dataGridView1.RowCount - 6].Cells[4].Value));

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

{

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

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

}

sum = fangweijiao(sdr, cr);

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[1].Value = radtodms(sum);

double fd, fdx;

fd = cr[cr.Length - 1] - acd;

fdx = 60 \* Math.Sqrt(sd.Length - 1);

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[1].Value = Convert.ToString(Math.Round(fd \* 180 / Math.PI \* 3600, 2)) + "″";

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[1].Value = Convert.ToString(Math.Round(fdx, 2)) + "″";

if (Math.Abs(fd \* 180 / Math.PI \* 3600) > fdx)

MessageBox.Show("角度闭合差超限！");

else

{

double vd = -fd / (sd.Length - 1);

double sumvd = 0;

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

{

sdr[i] += vd;

sumvd += vd;

dataGridView1.Rows[i].Cells[2].Value = Convert.ToString(Math.Round(vd \* 180 / Math.PI \* 3600, 2)) + "″";

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

}

if (Math.Round(sumvd, 8) != Math.Round(-fd, 8))

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)) + "″";

sum = fangweijiao(sdr, cr);

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);

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

dataGridView1.Rows[i].Cells[4].Value = radtodms(cr[i]);

}

}

double x2, y2, x3, y3; //存放已知两个点的x，y坐标

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

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

x3 = Convert.ToDouble(dataGridView1.Rows[sd.Length - 1].Cells[12].Value);

y3 = Convert.ToDouble(dataGridView1.Rows[sd.Length - 1].Cells[13].Value);

double[] sl = new double[sd.Length - 1]; //存放观测距离

double[] dx = new double[sl.Length]; //存放坐标增量

double[] dy = new double[sl.Length];

double suml = 0, sumdx = 0, 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

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

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

sumdx += dx[i]; //计算坐标增量总和#中函数

sumdy += dy[i];

}

double fx, fy, fxy, k1;

fx = sumdx - (x3 - x2); //计算坐标增量闭合差

fy = sumdy - (y3 - y2);

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导线全长相对闭合差：1/整数

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

double[] vy = new double[sl.Length];

double sumvx = 0, sumvy = 0;

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

double[] cy = new double[sl.Length];

double sumcx = 0, sumcy = 0;

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

double[] y = new double[sl.Length + 1];

x[1] = x2;

y[1] = y2;

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 i = 1; i < vx.Length; i++)

{

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

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

sumvx += vx[i]; //计算坐标增量改正数总和

sumvy += vy[i];

}

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！");

else

{

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

{

cx[i] = dx[i] + vx[i]; //计算改正后坐标增量

cy[i] = dy[i] + vy[i];

sumcx += cx[i]; //计算改正后坐标增量总和

sumcy += cy[i];

}

if (Math.Round(sumcx, 4) != Math.Round(x3 - x2, 4)

|| Math.Round(sumcy, 4) != Math.Round(y3 - y2, 4))

MessageBox.Show("改正后的坐标增量计算有误!");

else

{

for (int i = 2; i < x.Length; i++)

{

x[i] = x[i - 1] + cx[i - 1]; //计算x,y坐标

y[i] = y[i - 1] + cy[i - 1];

}

if (Math.Round(x[x.Length - 1], 4) != Math.Round(x3, 4)

|| Math.Round(y[y.Length - 1], 4) != Math.Round(y3, 4))

MessageBox.Show("坐标计算有误!");

else

{

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

{

dataGridView1.Rows[i].Cells[6].Value = Convert.ToString(Math.Round(dx[i], 4));

//将坐标增量放入表格

dataGridView1.Rows[i].Cells[7].Value = Convert.ToString(Math.Round(dy[i], 4));

dataGridView1.Rows[i].Cells[8].Value = Convert.ToString(Math.Round(vx[i], 4));

//将坐标增量改正数放入表格

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

dataGridView1.Rows[i].Cells[10].Value = Convert.ToString(Math.Round(cx[i], 4));

dataGridView1.Rows[i].Cells[11].Value = Convert.ToString(Math.Round(cy[i], 4));

//将改正后坐标增量放入表格

dataGridView1.Rows[i].Cells[12].Value = Convert.ToString(Math.Round(x[i], 3));

dataGridView1.Rows[i].Cells[13].Value = Convert.ToString(Math.Round(y[i], 3));

//将x,y坐标放入表格

}

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[5].Value =

Convert.ToString(Math.Round(suml, 4));

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[6].Value =

Convert.ToString(Math.Round(sumdx, 4));

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[7].Value =

Convert.ToString(Math.Round(sumdy, 4));

//将距离总和、坐标增量总和放入表格中

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

Convert.ToString(Math.Round(sumvx, 4));

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

Convert.ToString(Math.Round(sumvy, 4));

//将坐标增量改正数总和、改正后坐标增量总和放入表格中

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

Convert.ToString(Math.Round(sumcx, 4));

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

Convert.ToString(Math.Round(sumcy, 4));

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[7].Value =

Convert.ToString(Math.Round(fx, 4));

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[7].Value =

Convert.ToString(Math.Round(fy, 4));

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[10].Value =

Convert.ToString(Math.Round(fxy, 4));

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

}

}

}

}

}

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

{

Application.Exit();

}

private void excelToolStripMenuItem\_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 dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{ }

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;

}

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;

}

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

{

double sum = 0;

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

{

cr[i] = cr[i - 1] + sdr[i] - Math.PI;

if (cr[i] >= Math.PI \* 2)

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

else if (cr[i] < 0.0)

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

sum += sdr[i];

}

return sum;

}

private void excel文件ToolStripMenuItem\_Click\_1(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;

}

}

}



