C#程序课程内容 （边角网平差程序节选）

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[] dis = new double[dataGridView1.RowCount - 6];

double[] dx = new double[dis.Length];

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

double[] x = new double[dis.Length + 1];

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

double sumdis = 0;

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

{

dis[j] = Convert.ToDouble (dataGridView1.Rows[j].Cells[5].Value);//距离赋值

sumdis = sumdis+dis[j];//求距离和

}

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[5].Value = Math.Round(sumdis, 4);//和

sum = xzuobiao(cr, dis, dx);//求△x

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[6].Value = Math.Round(sum, 4);//和

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

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

sum = yzuobiao(cr, dis, dy);//求△y

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[7].Value = Math.Round(sum,4);//和

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

dataGridView1.Rows[i].Cells[7].Value = Math.Round(dy[i], 4); /\*△x,y求完毕，开始求改正\*/

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

double mm = Convert.ToDouble(dataGridView1.Rows[dataGridView1.RowCount - 6].Cells[12].Value);

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

{

x[i] = x[i - 1] + dx[i - 1];

}

double fd1 = x[x.Length - 1] - mm;

double sumv = 0;

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

{

double vdis = dis[i] / sumdis \* (-fd1);

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

dx[i] += vdis;//

dataGridView1.Rows[i].Cells[8].Value = Math.Round(vdis, 4);//x改正数

sumv += vdis;

}

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[8].Value = Math.Round ( sumv,4);//x改正数和

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[7].Value = Math.Round(fd1, 4);//x坐标增量闭合差

sum = 0;

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

{

sum = sum + dx[i];

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

}//求dx改正后结果

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[10].Value = Math.Round(sum, 4);////求dx改正后结果和

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

{

x[i] = x[i - 1] + dx[i - 1];

dataGridView1.Rows[i].Cells[12].Value = Math.Round(x[i], 4);//x最终值

}

//x结束

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

double nn = Convert.ToDouble(dataGridView1.Rows[dataGridView1.RowCount - 6].Cells[13].Value);

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

{

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

}

double fd2 = y[y.Length - 1] - nn;

sumv = 0;

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

{

double vdis = dis[i] / sumdis \* (-fd2);

dy[i] += vdis;//

dataGridView1.Rows[i].Cells[9].Value = Math.Round(vdis, 4);//y改正数

sumv += vdis;

}

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[9].Value = Math.Round(sumv, 4);//y改正数和

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[7].Value = Math.Round(fd2, 4);//y坐标增量闭合差

sum = 0;

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

{

sum = sum + dy[i];

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

}//求dy改正后结果

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[11].Value = Math.Round(sum, 4);////求dy改正后结果和

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

{

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

dataGridView1.Rows[i].Cells[13].Value = Math.Round(y[i], 4);//y最终值

}

//y结束

double f = Math.Sqrt(Math.Abs(fd1) \* Math.Abs(fd1) + Math.Abs(fd2) \* Math.Abs(fd2));//求导线全长闭合差

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[10].Value = Math.Round(f, 4);

int k;

k = Convert.ToInt32(sumdis / f/10)\*10;

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[11].Value = k;//求导线全长相对闭合差

//if (k < 2000) MessageBox.Show("导线相对闭合差超限");

}