

归算

- 归算公式（标高差）

$$\delta_2'' = \frac{e^2 H_2 \rho''}{2M_2} \cos^2 B_2 \sin 2A_1$$

由此可以看出需要纬度方位角等数据

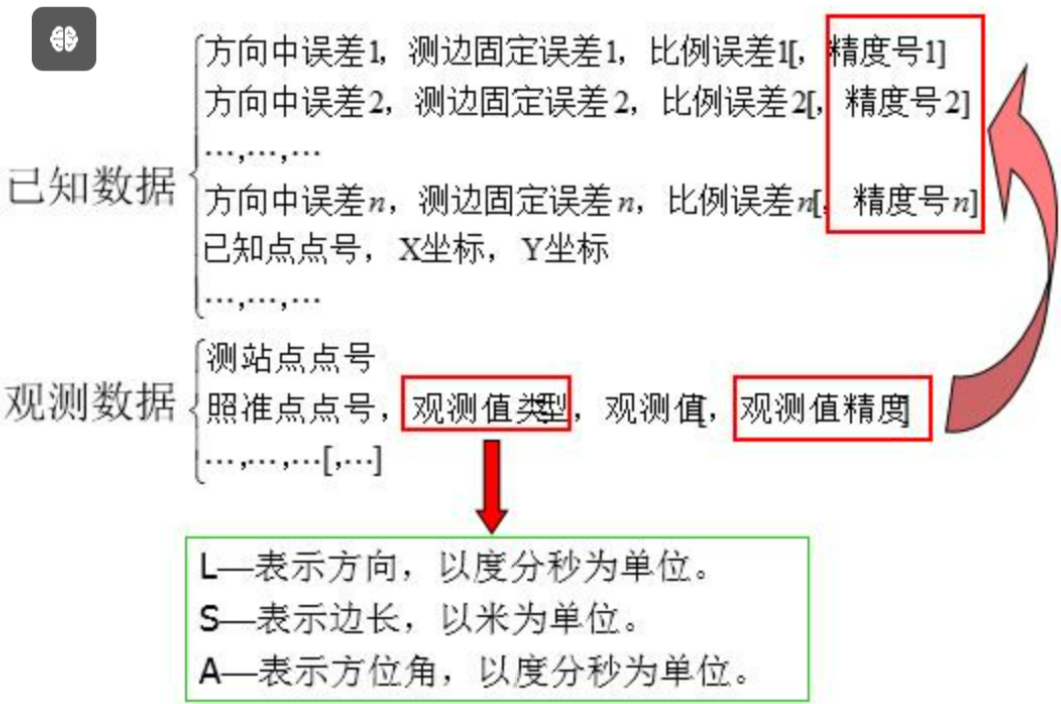
题中仅给出了部分已知平面坐标->坐标推算及反算

近似坐标推算

科傻系统实现

- 文件数据格式

平面观测文件：



平面网示例说明：

1.768,1,1 【1.768：方向中误差， 1：测边固定误差， 1：测边比例误差】

CPII-12,3377095.440,456568.755 【已知点号， 坐标X， 坐标Y】

CPII-11,3377481.525,456242.099 【已知点号， 坐标X， 坐标Y】

CPII-11 【测站点号】

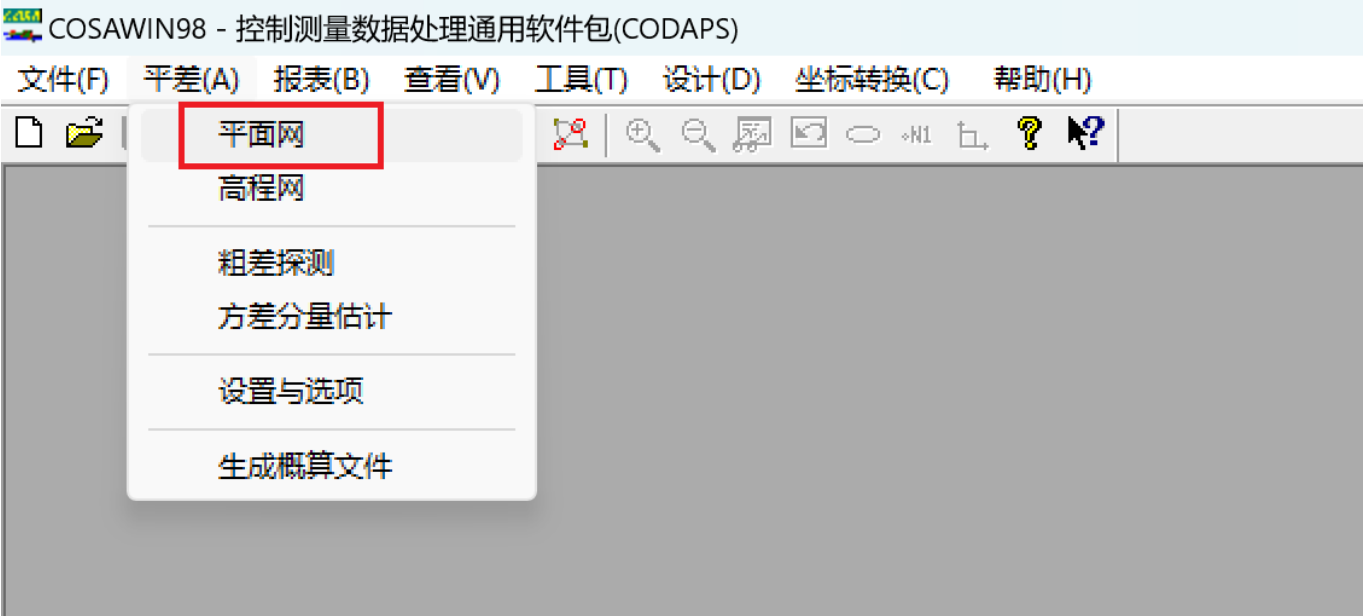
CPII-12,L,0 【后视点号， 角度归零】

JM-4,L,115.59005 【前视点号， 观测角度】

- 后面【】为格式说明，不需要输入

在新建文件中输入好后将文件命名及格式更改为“边角网.in2”

- 文件输入及近似坐标推算



近似坐标			
Name	X(m)	Y(m)	
G001	4590341.841	501783.982	
G009	4566778.255	509527.387	
G010	4564138.461	496046.167	
G005	4582057.551	513835.092	
G008	4571598.509	506343.007	
G006	4577753.054	507635.610	
G007	4577665.605	498252.594	
G004	4573772.729	514952.911	

方向平差结果							
FROM	TO	TYPE	VALUE(dms)	M(sec)	V(sec)	RESULT(dms)	Ri
G001	G005	L	0.000000	1.30	-0.20	-0.000020	0.65
G001	G006	L	30.335200	1.30	-1.99	30.335001	0.66
G001	G007	L	71.033840	1.30	2.19	71.034059	0.65
G009	G005	L	0.000000	1.30	2.84	0.000284	0.78
G009	G004	L	22.032220	1.30	1.23	22.032343	0.76

根据推算坐标高斯反算大地坐标

- 详情见另一文件

角度观测值归算到椭球面

注意三差改正结果计算下来单位为秒，改正时需要注意角度转换

- 垂线偏差改正

$$\delta_1 = -(\xi \sin A - \eta \cos A) \cot z_1$$
$$= -(\xi \sin A - \eta \cos A) \tan \alpha_1$$

A为大地方位角，alpha为垂直角，ξ和η分别为垂线偏差分量

注：题中没有给垂线偏差和垂直角即为0，且没有给高程异常，则视为H为大地高

大地方位角（坐标方位角+子午线收敛角）

由平面坐标 x 、 y 计算子午线收敛角 γ 的公式【推导自学】

$$\gamma = \frac{1}{N_f} y t_f - \frac{y^3}{3N_f^3} t_f (1 + t_f^2 - \eta_f^2) + \frac{y^5}{15N_f^5} t_f (2 + 5t_f^2 + 3t_f^4)$$

带下标的 N 、 t 、 η 表示该值是用底点纬度 B_f 代替 B 计算得到的。上式计算子午线收敛角的精度可达 $0.001''$ 。

编程时采用 (4-419) 实现子午线收敛角计算。

- 坐标方位角

```
public static double Getfwangle(double x1, double x2, double y1, double y2)
{
    double x = x2 - x1;
    double y = y2 - y1;
    double fwangle = 0;
    if (x == 0 && y > 0)
    {
        fwangle = Math.PI * 0.5;
    }
    else if (x == 0 && y < 0)
    {
        fwangle = Math.PI * 1.5;
    }
    if (x > 0 && y > 0)
    {
        fwangle = Math.Atan(y / x);
    }
    else if (x < 0 && y > 0)
    {
        fwangle = Math.PI - Math.Abs(Math.Atan(y / x));
    }
    else if (x < 0 && y < 0)
    {
        fwangle = Math.PI + Math.Atan(y / x);
    }
    else if (x > 0 && y < 0)
    {
        fwangle = 2 * Math.PI - Math.Abs(Math.Atan(y / x));
    }
    return fwangle;
}
```

- 标高差改正

$$\delta_2'' = \frac{e^2 H_2 \rho''}{2M_2} \cos^2 B_2 \sin 2A_1$$

e2为**椭球参数**，H2、B2和M2分别为照准点的**大地高**、**大地纬度**和**子午圈曲率半径**；A1为测站点至照准点的**大地方位角**。ρ为角度与弧度转换量取**206265**

- 截面差改正

$$\delta_3'' = -\frac{e^2 S^2 \rho''}{12N_1^2} \cos^2 B_1 \sin 2A_1$$

观测改正

表 5-7 观测方向值化至椭球面

测站	方向	化至标石 中心的观 测方向值 /(° ' ")	三差改正			$\sum \delta$ /(")	归零 /(")	椭球面上 的方向值 /(° ' ")
			δ_1 /(")	δ_2 /(")	δ_3 /(")			
高山	龙山	0 00 00.00	+0.014	-0.266	+0.002	-0.250	0.000	0 00 00.00
	张庄	40 43 53.34	+0.009	-0.110	0.000	-0.100	+0.150	40 43 53.49
	王村	102 36 11.45	+0.039	+0.301	-0.003	+0.337	+0.587	102 36 12.04

- 代码实现（前提文件格式为仿照科傻输入）

```
string start = data.geoangle[0].StartP.name;
double a0 = 0;
bool if1 = true;
double a12 = 0;
for (int i = 0; i < data.geoangle.Count(); i++)
{
    data.geoangle[i].StartP.A = data.geoangle[i].A12;
    POINT p1 = new POINT(data.geoangle[i].StartP);
    POINT p2 = new POINT(data.geoangle[i].EndP);

    if (data.geoangle[i].StartP.name != start)
    {
        if1 = true;
        start = data.geoangle[i].StartP.name;
    }
}
```

```

    if (if1)
    {
        a0 = caculate.CalA1(p1, p2, data.geoangle[i].S);
        data.geoangle[i].A1 = 0;
        if1 = false;
    }
    else
    {
        data.geoangle[i].cangle = caculate.CalA1(p1, p2, data.geoangle[i].S);
        data.geoangle[i].A1 = (data.geoangle[i].cangle - a0) +
data.geoangle[i].A;
    }
}

```

距离观测值归算到椭球面

■ 短距离（小于30km） 斜距归算公式推导

$$S = d + \frac{d^3}{24R_A^2}, \quad d = D \sqrt{\frac{1 - \left(\frac{H_2 - H_1}{D}\right)^2}{\left(1 + \frac{H_1}{R_A}\right)\left(1 + \frac{H_2}{R_A}\right)}}$$

$$R_A = \frac{N}{1 + e'^2 \cos^2 B_1 \cos^2 A_{12}}$$

注:和方向改化不一样的是, 这里的**S就是最终结果**而不是改正值

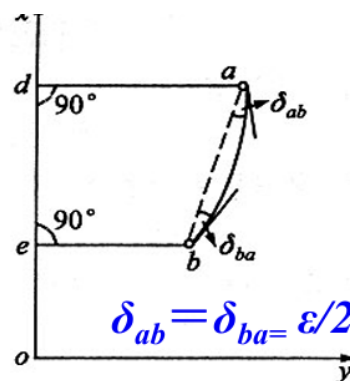
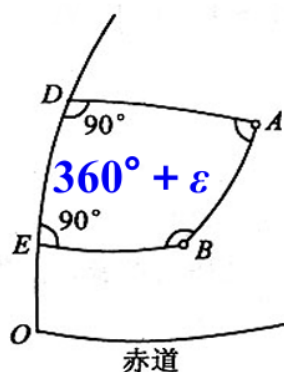
椭球面角度观测值归算到高斯平面

注意: 凡是涉及y坐标都要记得减去500000才是它的真正平面坐标

■ 等角投影

大地线在高斯平面上的投影是一条**凹向中央子午线**的曲线。

球面角超的计算公式：



$$\varepsilon'' = \frac{F}{R^2} \rho'' \quad F = \left| \frac{y_a + y_b}{2} (x_a - x_b) \right| = |y_m (x_a - x_b)|$$

$$\delta_{ab} = \delta_{ba} = \frac{1}{2} \varepsilon'' = \frac{|y_m (x_a - x_b)|}{2R_m^2} \quad (4-426)$$

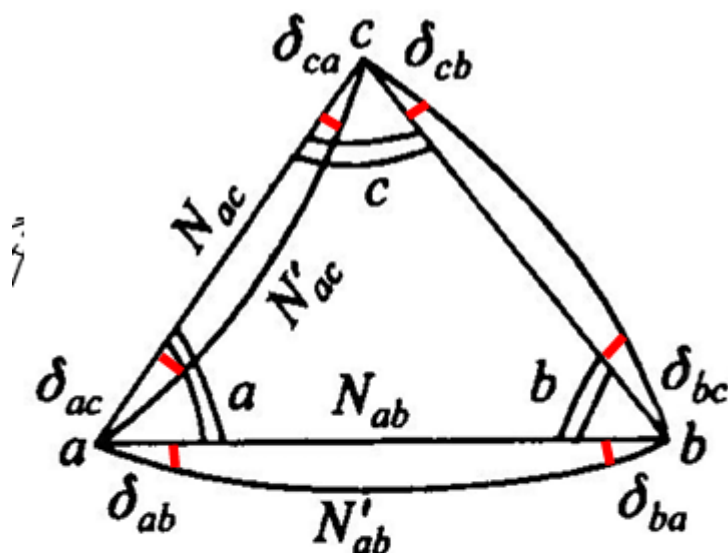
顾及符号后（度盘为顺时针）： δ_{ab} 应取正号， δ_{ba} 应取负号

$$\delta_{ab} = \frac{\rho''}{2R_m^2} y_m (x_a - x_b), \quad \delta_{ba} = -\frac{\rho''}{2R_m^2} y_m (x_a - x_b) = \frac{1}{2R_m^2} y_m (x_b - x_a)$$

通用形式为

$$\delta_{i,j}'' = -\frac{\rho''}{2R_m^2} y_m (x_j - x_i)$$

该式的误差小于0.1", 适用于三、四等三角测量



$$\begin{aligned} a &= (N_{ab} - N_{ac}) \\ &= (N'_{ab} + \delta_{ab}) - (N'_{ac} + \delta_{ac}) \\ &= (N'_{ab} - N'_{ac}) + \delta_{ab} - \delta_{ac} \end{aligned}$$

椭球面距离观测值归算到高斯平面

$$D = S(1 + \frac{y_m^2}{2R_m^2} + \frac{\Delta y^2}{24R_m^2} + \frac{y_m^4}{24R_m^4}) \tag{4-459}$$

S为椭球面距离，ym为两点y坐标平均值，Rm为起点平均曲率半径

输出结果并再次带入到科傻进行计算并平差

编程实现

完整代码：<https://github.com/rockli666/GuiSuan>

测绘地信遥感导航AI交流qq群:592418364

LSH from SWPU