

1 ddres双差计算及观测矩阵的构建

流程：ddres→baseline→ecef2pos→ionmapf→prectrop→validobs→varerr→ddcov

$$v = (y_i^r - y_i^b) - (y_j^r - y_j^b) - N$$

- ① 调用baseline计算基站和流动站间的基线长度，并将二者坐标从ECEF转化为WGS-84
- ② 进行中间变量的初始化，并将双差伪距/载波相位残差置0
- ③ opt中包含电离层状态量则调用ionmapf计算基站和流动站出的投影系数，并将其平均值作为“单差垂直电离层延迟状态量”的投影系数
- ⑤ 对各个系统循环处理：
 - a. 首先找到高度角最大的卫星作为参考星；
 - b. 利用zdres函数中计算的载波相位/伪距非差残差，计算每颗卫星的载波相位/伪距双差残差，并对量测矩阵H中和位置状态量相关的部分进行赋值；
 - c. 以GPS L1为基准，计算当前频率的电离层单差延迟量，更新上一步中计算的双差，并对H中和电离层相关的部分进行赋值；
 - d. 从双差残差中去除对流层双差湿分量，并对H中和对流层相关的部分进行赋值；
 - e. 如果不是无电离层，则在双差残差中去除双差模糊度，并对H中和模糊度相关的部分进行赋值；
 - f. 保存伪距/载波相位残差。
- ⑥ 对上述计算的双差残差进行阈值检测，超过阈值则无效
- ⑦ 计算伪距/载波相位单差的量测噪声方差
- ⑧ 设置卫星有效标志
- ⑨ PMODE_MOVEB模式下增加基线长度约束，调用ddcov计算双差量测噪声

1.1 ecef2pos

PI: π RE_WGS84: 地球半长轴 FE_WGS84: 地球扁平率 e: 地球偏心率

$$r = \sqrt{a^2 + b^2}$$

$$\sin p = \frac{z}{\sqrt{r^2 + z^2}} \quad p \text{ 表示计算得到的大地纬度}$$

$$v = \frac{R_E}{\sqrt{1 - e^2 \sin^2 p}} \quad \text{地球表面上的纬度对应的曲率半径}$$

$$z = r \sin p + e^2 v \sin p$$

1.1 电离层投影系数

$$z' = \arcsin\left(\frac{R_E}{R_E + H}\right) \sin z \quad H \text{ 为电离层高度}$$

$$z = \frac{\pi}{2} - El_r^s \quad \text{天顶角}$$

$$I_{r,i}^s = \frac{1}{\cos z'} \frac{40.3 \times 10^{16}}{f_i} TEC(t, \phi_{IPP}, \lambda_{IPP})$$

$$TEC(t, \phi_{IPP}, \lambda_{IPP}) = \frac{(t - t_i)TEC(t_i, \phi_{IPP}, \lambda_{IPP} + \omega(t - t_i)) + (t_{i+1} - t)TEC(t_{i+1}, \phi_{IPP}, \lambda_{IPP} + \omega(t - t_{i+1}))}{t_{i+1} - t_i}$$

1.2 对流层延迟湿分量

首先找到对流层天顶方向延迟状态量的索引，然后使用NMF计算湿延迟投影函数

$$m_w = m_0 + m_0 * \cot(El_r^s) * \cos Az + m_0 * \cot(El_r^s) * G_E * \sin Az$$

NMF求得的对流层湿延迟投影函数+北向电离层梯度系数+东向电离层梯度系数

后续会添加到量测矩阵

1.3 单差量测噪声方差

$$2.0 * (\text{opt} \rightarrow \text{iono} \text{opt} == \text{IONOOPT_IFLC?} 3.0 : 1.0) * (a * a + b * b / \sin^2 \text{el} + c * c) + d * d;$$

$$R_i = 2.0 * (a^2 + \frac{b^2}{\sin^2 \text{el}^i} + c^2) + d^2$$

$$R_j = 2.0 * (a^2 + \frac{b^2}{\sin^2 \text{el}^j} + c^2) + d^2$$

1.4 双差量测噪声方差

$$R[k+i+(k+j)*nv]=R_i[k+i]+(i==j?R_j[k+i]:0.0);$$

$$R = \begin{bmatrix} R_i[0] + R_j[0] & R_i[0] & \cdots & R_i[0] \\ R_i[1] & R_i[1] + R_j[1] & \cdots & R_i[1] \\ \vdots & \vdots & \ddots & \vdots \\ R_i[n] & R_i[n] & \cdots & R_i[n] + R_j[n] \end{bmatrix}$$

2 filter卡尔曼测量更新

$$Q = H^T * P * H + R$$

$$K = P * H Q^{-1}$$

$$x_p = x + K * v$$

$$P_p = (I - K * H) * P$$