惯性导航解算

学院：测绘遥感信息工程国家重点实验室

专业：大地测量学与测量工程

班级：博士 2021 班

姓名：波波-zbzhao

学号：XXXXXXXX

2022 年 6 月

1. 基本原理

本次算法的全部内容均建立在北东地坐标系，注意和严老师的部分公式不同，严老师的代码坐标系是东北天坐标系。

1.1 姿态更新算法

以下算法中会对向量进行加粗，**加粗的量均为3×1的向量**

步骤 1：等效旋转矢量更新 b 系

$$\boldsymbol{\phi_k}=\boldsymbol{\Delta\theta_k}+\frac{1}{12}\boldsymbol{\Delta\theta_{k-1}}\times\boldsymbol{\Delta\theta_k}$$

其中，上述公式中等号右侧第二项为圆锥效应补偿项。将上式中的等效旋转矢量转换为四元数，在采用四元数进行计算时，$\|\boldsymbol{\phi_k}\|$表示的是对$\boldsymbol{\phi_k}$求模，将向量的每个元素平方再开方即可，转换关系如下：

$$q_{b(k)}^{b(k-1)}=\begin{bmatrix}\cos\|0.5\boldsymbol{\phi_k}\|\\ \dfrac{\sin\|0.5\boldsymbol{\phi_k}\|}{\|0.5\boldsymbol{\phi_k}\|}0.5\boldsymbol{\phi_k}\end{bmatrix}=\begin{bmatrix}\cos\|0.5\boldsymbol{\phi_k}\|\\ \dfrac{\sin\|0.5\boldsymbol{\phi_k}\|}{\|\boldsymbol{\phi_k}\|}\boldsymbol{\phi_k}\end{bmatrix}$$

步骤 2：等效旋转矢量更新 n 系，注意$q_{n(k-1)}^{n(k)}$和$q_{b(k)}^{b(k-1)}$上下标的区别。

$$\boldsymbol{\zeta_k}=[\boldsymbol{w}_{ie}^n(t_{k-1})+\boldsymbol{w}_{en}^n(t_{k-1})]\Delta t$$

$$q_{n(k-1)}^{n(k)}=\begin{bmatrix}\cos\|0.5\boldsymbol{\zeta_k}\|\\ -\dfrac{\sin\|0.5\boldsymbol{\zeta_k}\|}{\|0.5\boldsymbol{\zeta_k}\|}0.5\boldsymbol{\zeta_k}\end{bmatrix}=\begin{bmatrix}\cos\|0.5\boldsymbol{\zeta_k}\|\\ -\dfrac{\sin\|0.5\boldsymbol{\zeta_k}\|}{\|\boldsymbol{\zeta_k}\|}\boldsymbol{\zeta_k}\end{bmatrix}$$

步骤 3：计算当前姿态的四元数

$$q_{b(k)}^{n(k)}=q_{n(k-1)}^{n(k)}\circ q_{b(k-1)}^{n(k-1)}\circ q_{b(k)}^{b(k-1)}$$

步骤 4：对更新后的姿态四元数进行归一化处理

$$q_i=\frac{\hat{q}_i}{\sqrt{\hat{q}_0^2+\hat{q}_1^2+\hat{q}_2^2+\hat{q}_3^2}},i=0,1,2,3$$

其中

$$w_{ie}^n=[w_e\cos\varphi\quad 0\quad -w_e\sin\varphi]^T$$

$$w_{en}^n=\begin{bmatrix}v_E/(R_N+h)\\ -v_N/(R_M+h)\\ -v_E\tan\varphi/(R_N+h)\end{bmatrix}$$

$$\begin{cases}R_M=\dfrac{a(1-e^2)}{\sqrt{(1-e^2\sin^2\varphi)^3}}\\ R_N=\dfrac{a}{\sqrt{1-e^2\sin^2\varphi}}\end{cases}$$

式中，w_e为地球自转角速度，a为地球长半轴，e为地球第一偏心率。

1.2 速度更新算法

速度更新算法受到比力积分项和哥氏积分项两项影响，具体计算如下：

$$\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{n}} = \boldsymbol{v}_{k-1}^{n} + \Delta\boldsymbol{v}_{f,k}^{n} + \Delta\boldsymbol{v}_{g/cor,k}^{n}$$

1.2.1 重力/哥氏积分项对应的速度$\Delta\boldsymbol{v}_{\boldsymbol{g/cor,k}}^{\boldsymbol{n}}$

其公式为：

$$\Delta\boldsymbol{v}_{g/cor,k}^{n} = \left[\boldsymbol{g}_{\boldsymbol{p}}^{\boldsymbol{n}} - (\boldsymbol{2}\boldsymbol{w}_{\boldsymbol{ie}}^{\boldsymbol{n}} + \boldsymbol{w}_{\boldsymbol{en}}^{\boldsymbol{n}}) \times \boldsymbol{v}^{\boldsymbol{n}}\right]_{t_{k-\frac{1}{2}}}(t_k - t_{k-1})$$

注意该式子中的$\boldsymbol{g}_{\boldsymbol{p}}^{\boldsymbol{n}}$、$\boldsymbol{2}\boldsymbol{w}_{\boldsymbol{ie}}^{\boldsymbol{n}} + \boldsymbol{w}_{\boldsymbol{en}}^{\boldsymbol{n}}$和$v^n$均为中间时刻的变量，$\boldsymbol{2}\boldsymbol{w}_{\boldsymbol{ie}}^{\boldsymbol{n}} + \boldsymbol{w}_{\boldsymbol{en}}^{\boldsymbol{n}}$这两个变量均与中间时刻的位置和速度有关。因此在计算时，可以采用上一时刻的位置和速度和加速度进行外推，具体外推的速度和位置公式如下：

上一时刻的加速度公式如下,其中$T = (t_k - t_{k-1})$，为采样间隔：

$$\boldsymbol{a}_{k-1}^{n} = \frac{\Delta\boldsymbol{v}_{f,k}^{n} + \Delta\boldsymbol{v}_{g/cor,k}^{n}}{T}$$

那么**中间时刻的速度**为：

$$\boldsymbol{v}_{k-1/2}^{n} = \boldsymbol{v}_{k-1}^{n} + \boldsymbol{a}_{k-1}^{n}\frac{T}{2}$$

接下来推导中间时刻的位置。

纬度，经度和高程的微分方程如下：

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \dfrac{V_N}{R_M + h} \\ \dfrac{V_E}{(R_N + h)\cos\varphi} \\ -V_D \end{bmatrix}$$

那么纬度经度和高程的变化率乘以时间即为位置的变化量：

$$\Delta\boldsymbol{P} = \begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix}\Delta t = \begin{bmatrix} \dfrac{V_N}{R_M + h} \\ \dfrac{V_E}{(R_N + h)\cos\varphi} \\ -V_D \end{bmatrix}\Delta t$$

则**中间时刻的位置**的公式如下：

$$\boldsymbol{P}_{k-1/2}^{n} = \boldsymbol{P}_{k-1}^{n} + \begin{bmatrix} \dfrac{V_{N_{k-1}}}{R_{M_{k-1}} + h_{k-1}} \\ \dfrac{V_{E_{k-1}}}{(R_M + h)_{k-1}\cos\varphi_{k-1}} \\ -V_{D_{k-1}} \end{bmatrix}\frac{T}{2}$$

由上述$\boldsymbol{v}_{\boldsymbol{k-1/2}}^{\boldsymbol{n}}$和$\boldsymbol{P}_{\boldsymbol{k-1/2}}^{\boldsymbol{n}}$即可求得与中间位置和速度相关的$\boldsymbol{g}_{\boldsymbol{p}}^{\boldsymbol{n}}$、$\boldsymbol{2}\boldsymbol{w}_{\boldsymbol{ie}}^{\boldsymbol{n}} + \boldsymbol{w}_{\boldsymbol{en}}^{\boldsymbol{n}}$和$\boldsymbol{v}^{\boldsymbol{n}}$，$g_p^n$为正常重力，正常重力的求法可以参考牛小骥老师 PPT 第二部分的附录

最后一页，需要注意的是，严恭敏老师代码中的正常重力的求法和牛小骥老师的不同，求出来会有细微的差别。同时正常重力是有符号的，如果取北东地坐标系，则正常重力的符号为正，若取东北天坐标系，正常重力的符号为负。

1.2.2 比力积分项对应的速度$\Delta\boldsymbol{v}_{f,k}^{n}$

其公式如下：

$$\Delta\boldsymbol{v}_{f,k}^{n}=\left[I-\frac{1}{2}(\boldsymbol{\zeta}_{n(k-1),n(k)}\times)\right]C_{b(k-1)}^{n(k-1)}\Delta\boldsymbol{v}_{f,k}^{b(k-1)}$$

在上式中，

$$\boldsymbol{\zeta}_{n(k-1),n(k)}=(\boldsymbol{w}_{ie}^{n}+\boldsymbol{w}_{en}^{n})_{t_{k-\frac{1}{2}}}(t_k-t_{k-1})$$

$$\Delta\boldsymbol{v}_{f,k}^{b(k-1)}=\Delta\boldsymbol{v}_k+\frac{1}{2}\Delta\boldsymbol{\theta}_k\times\Delta\boldsymbol{v}_k+\frac{1}{12}(\Delta\boldsymbol{\theta}_{k-1}\times\Delta\boldsymbol{v}_k+\Delta\boldsymbol{v}_{k-1}\times\Delta\boldsymbol{\theta}_k)$$

需要注意的是上式中$(\boldsymbol{w}_{ie}^{n}+\boldsymbol{w}_{en}^{n})_{t_{k-\frac{1}{2}}}$为中间时刻位置对应的角速率。$\Delta\boldsymbol{v}_{f,k}^{b(k-1)}$式子中后面两项分别对应的是旋转效应补偿项和划桨效应补偿项。$\Delta\boldsymbol{\theta}_{k-1}$、$\Delta\boldsymbol{\theta}_k$是经过圆锥补偿之后的计算结果。

1.3 位置更新算法

步骤 1：高程更新

$$h_k=h_{k-1}-\frac{1}{2}(v_{D,k-1}+v_{D,k})(t_k-t_{k-1})$$

步骤 2：纬度更新

$$\varphi_k=\varphi_{k-1}+\frac{v_{N,k-1}+v_{N,k}}{2(R_{M,k-1/2}+\bar{h})}(t_k-t_{k-1})$$

步骤 3：经度更新

$$\lambda_k=\lambda_{k-1}+\frac{v_{E,k-1}+v_{E,k}}{2(R_{N,k-1/2}+\bar{h})\cos\bar{\varphi}}(t_k-t_{k-1})$$

上式中：

$$\bar{h}=\frac{1}{2}(h_k+h_{k-1}),\bar{\varphi}=\frac{1}{2}(\varphi_k+\varphi_{k-1})$$

位置更新也可以写成另一种矩阵形式，矩阵中涉及到的中间时刻的$R_{M,k-1/2}$，$h_{k-1/2}$，$R_{N,k-1/2}$均可以用速度更新算法中的中间时刻位置求得：

$$\begin{bmatrix}\varphi_k\\ \lambda_k\\ h_k\end{bmatrix}=\begin{bmatrix}\varphi_{k-1}\\ \lambda_{k-1}\\ h_{k-1}\end{bmatrix}+\begin{bmatrix}\frac{1}{R_{M,k-1/2}+h_{k-1/2}} & 0 & 0\\ 0 & \frac{1}{(R_{N,k-1/2}+h_{k-1/2})\cos\varphi_{k-1/2}} & 0\\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}\frac{v_{N,k-1}+v_{N,k}}{2}\\ \frac{v_{E,k-1}+v_{E,k}}{2}\\ \frac{v_{D,k-1}+v_{D,k}}{2}\end{bmatrix}(t_k-t_{k-1})$$

2. 实现步骤

用 Matlab 将数据文件进行读取，由于初始状态时间为 91620.0s，因此将拿

到的惯导数据从 91620.005s 开始积分。

解题思路：

（1）对进入循环中的角度增量先做圆锥补偿，对速度增量做旋转补偿和划桨补偿；

（2）采用 1.2 中的速度更新算法进行速度更新；

（3）采用 1.3 中的位置更新算法进行位置更新；

（4）采用 1.1 中的姿态更新算法进行位置更新。

代码中主函数是 SINS.m 文件，运行时将数据文件和该文件放在一个文件夹下面即可运行，具体详细代码见压缩包。接下来将核心算法部分贴上来，

```
function ins = insUpdateAlog(ins, imu)
% SINS Updating Alogrithm including attitude, velocity and position
% updating.
%
% Prototype: ins = insupdate(ins, imu)
% Inputs: ins - SINS structure array created by function 'insinit'
%         imu - gyro & acc incremental sample(s)
% Output: ins - SINS structure array after updating
    nn = size(imu,1);
    nts = nn*ins.ts;  nts2 = nts/2;  ins.nts = nts;
    [phim, dvbm] = cnscl(imu,2);    % coning & sculling compensation  圆锥效应补偿后的角度增量 和 划桨效应补偿后的速度增量
    %% earth & angular rate updating
    %根据上一时刻的速度和上一个时刻的加速度外推中间时刻的速度
    vn01 = ins.vn+ins.an*nts2;
    %根据上一时刻的位置和纬度、经度、高度微分方程外推  ins.Mpv 为纬度 经度 高程微分方程的系数阵
    pos01 = ins.pos+ins.Mpv*vn01*nts2;
    ins.eth = ethupdate(ins.eth, pos01, vn01);
    ins.wib = phim/nts; ins.fb = dvbm/nts;  % 计算陀螺的三轴角速度 Wib 和计算 Fb 比力
    %% (1)velocity updating
    % 严老师的计算公式
    ins.fn = qmulv(ins.qnb, ins.fb); %将 Fb 转换成 Fn
    ins.an = qmulv(rv2q(-ins.eth.wnin*nts2),ins.fn) + ins.eth.gcc;%先将等效旋转矢量转换成四元数，然后将 fn 通过四元数进行旋转
    vn1 = ins.vn + ins.an*nts; %当前时刻的速度

    % 牛老师的计算公式
    vectorWnin = ins.eth.wnin * ins.ts;
    ins.vnfk = (eye(3) - 1.0/2 * setMat(vectorWnin)) * ins.Cnb * dvbm;
```

```
    ins.ngcork = ins.eth.gcc * ins.ts;
    vn1 = ins.vn + ins.vnfk + ins.ngcork;
    ins.an = (ins.vnfk + ins.ngcork) / ins.ts;
    % 对比结果发现二者相同

    %% (2)position updating
    %用处理好的系数阵和速度相乘
    ins.Mpv = [ 1/ins.eth.RMh, 0,0; 0, 1/ins.eth.clRNh, 0; 0, 0, -1];
    ins.Mpvvn = ins.Mpv*(ins.vn+vn1)/2; %ins.vn 为上一时刻的速度 vn1 为当前时刻的速度
    ins.pos1 = ins.pos + ins.Mpvvn*nts;

    % 展开计算位置
    ins.lastPos = ins.pos;
    ins.pos(3) = ins.pos(3) - 0.5 * (ins.vn(3) + vn1(3)) *nts;
    ins.pos(1) = ins.pos(1) + 0.5 * (ins.vn(1) + vn1(1)) *nts / ...
        (ins.eth.RM + 0.5 *(ins.pos(3) + ins.lastPos(3)));
    ins.pos(2) = ins.pos(2) + 0.5 * (ins.vn(2) + vn1(2)) *nts / ...
        ((ins.eth.RN + 0.5 *(ins.pos(3) + ins.lastPos(3)))*cos(0.5*(ins.pos(1) + ins.lastPos(1))));
    %对比完之后，二者结果一致
    ins.vn = vn1;
    ins.an0 = ins.an;
    %% (3)attitude updating
    ins.Cnb0 = ins.Cnb;
    % 传进去的参数分别为上一个时刻的 qnb，惯导器件经过圆锥补偿的三个姿态角，n 系相对 i 系在 n 系投影的三个角度
    ins.qnb = qupdt2(ins.qnb, phim, ins.eth.wnin*nts);
    [ins.qnb, ins.att, ins.Cnb] = attsyn(ins.qnb);
    %% 整理最终结果
    ins.avp = [ins.att; ins.vn; ins.pos];
end
```

3. 计算结果

将计算出来的姿态、速度和纬度经度和参考的计算结果做差，其差值图如下所示：

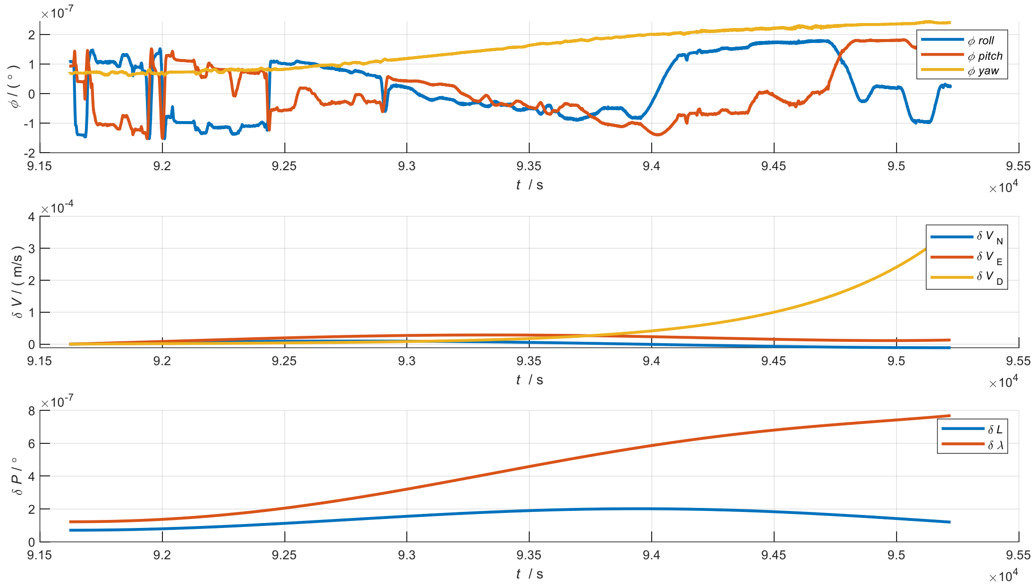

图 1 和参考计算值的差值

上图中的姿态角和参考计算值的差值级别在10^{-7}°，速度的差值级别在10^{-4}m/s，纬度和经度差值级别在10^{-7}°。纬度、经度和高程无法在一个图中展示，因此将纬度和经度的差值转换为北方向和东方向的差值。纬度和经度转换为北方向和东方向的差值的计算原理如下：

纬度，经度和高程的微分方程为：

$$\begin{bmatrix}\dot{\varphi}\\ \dot{\lambda}\\ \dot{h}\end{bmatrix}=\begin{bmatrix}\dfrac{V_N}{R_M+h}\\ \dfrac{V_E}{(R_N+h)\cos\varphi}\\ -V_D\end{bmatrix}$$

那么对上式左右两侧乘以时间,目前我们求出来的差值就是下式中的最左侧一项：

$$\begin{bmatrix}\varphi_{dif}\\ \lambda_{dif}\\ h_{dif}\end{bmatrix}=\begin{bmatrix}\dot{\varphi}t\\ \dot{\lambda}t\\ \dot{h}t\end{bmatrix}=\begin{bmatrix}\dfrac{V_N t}{R_M+h}\\ \dfrac{V_E t}{(R_N+h)\cos\varphi}\\ -V_D t\end{bmatrix}$$

$V_N t$、$V_E t$和$V_D t$即为我们要的在 NED 方向的当地差值，即为：

$$\begin{bmatrix}P_N\\ P_E\\ P_D\end{bmatrix}=\begin{bmatrix}V_N t\\ V_E t\\ V_D t\end{bmatrix}=\begin{bmatrix}(R_M+h)\varphi_{dif}\\ \lambda_{dif}(R_N+h)\cos\varphi\\ -h_{dif}\end{bmatrix}$$

那么在 NED 方向的差值如下：

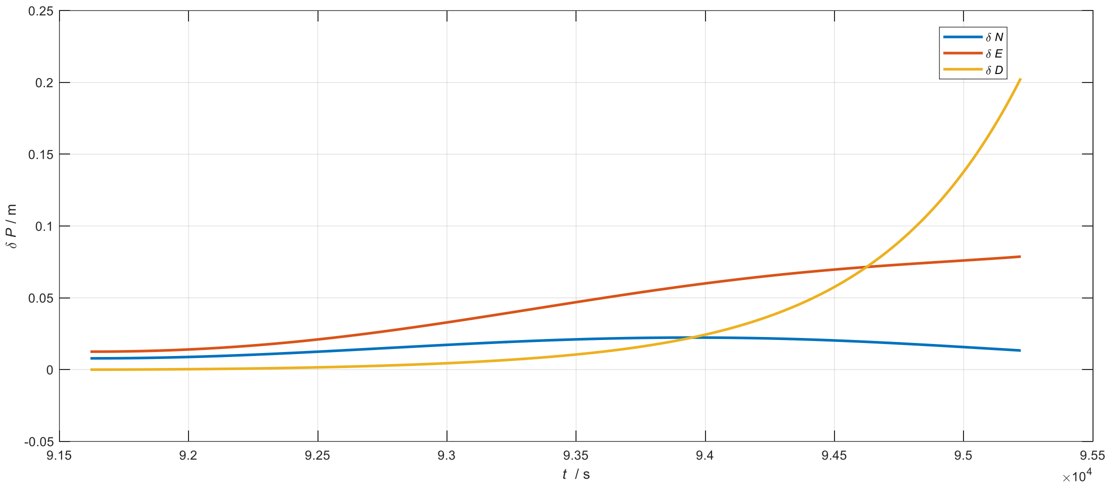

图 2 和参考计算值的在 NED 方向的差值

可以看到在 NED 方向与参考计算值的差值在 0.25m 以下,计算结果较好。

4. 心得体会

1. 对于速度改正项$\Delta \boldsymbol{v}_{f,k}^{n}$的理解

$$\Delta \boldsymbol{v}_{f,k}^{n}=\left[I-\frac{1}{2}(\boldsymbol{\zeta}_{n(k-1),n(k)}\times)\right]C_{b(k-1)}^{n(k-1)}\Delta \boldsymbol{v}_{f,k}^{b(k-1)}$$

上式中这一项，严格按照牛小骥老师的计算公式进行计算，是采用 DCM 的计算方式，也就是先左乘$C_{b(k-1)}^{n(k-1)}$，转换成 n 系下的$\Delta \boldsymbol{v}_{f,k}^{n(k-1)}$，之后再左乘矩阵。我也研究了下严老师对这一项的计算，严老师的计算是将上式转换成了加速度，如下：

$$\Delta \boldsymbol{f}_{f,k}^{n}=\left[I-\frac{1}{2}(\boldsymbol{\zeta}_{n(k-1),n(k)}\times)\right]C_{b(k-1)}^{n(k-1)}\Delta \boldsymbol{f}_{f,k}^{b(k-1)}$$

并且没有用矩阵的算法，是先将$\Delta \boldsymbol{f}_{f,k}^{b(k-1)}$通过$q_{b(k-1)}^{n(k-1)}$四元数进行转换，转换为$\Delta \boldsymbol{f}_{f,k}^{n(k-1)}$,之后将$\left[I-\frac{1}{2}(\boldsymbol{\zeta}_{n(k-1),n(k)}\times)\right]$视为 DCM，$-\frac{1}{2}\boldsymbol{\zeta}_{n(k-1),n(k)}$这一项视为等效旋转矢量，之后再转换为四元数。注意：$-\frac{1}{2}\boldsymbol{\zeta}_{n(k-1),n(k)}$这一项整体是等效旋转矢量。因此相当于将$\Delta \boldsymbol{f}_{f,k}^{b(k-1)}$通过两个四元数，旋转为了一个$3\times 1$的向量，感觉也是蛮巧妙的。

这里简单做一个类比：q_b^{n*}为q_b^n的共轭，后三个量取反即为q_b^n的共轭。

（1）DCM 和等效旋转矢量的下标表示如下：

$$C_b^n\approx I+(\phi_{nb}\times)$$

等效旋转矢量和四元数对应关系为：

$$\phi_{nb}\rightarrow q_b^n$$

那么v^n通过四元数和 DCM 的求法如下：

$$v^n = q_b^n \circ v^b \circ q_b^{n*}$$
$$v^n = C_b^n v^b$$

同理另一种形式:

（2）DCM 和等效旋转矢量的下标表示如下：

$$C_n^b = {C_b^n}^T = (I + (\phi_{nb} \times))^T = I^T + (\phi_{nb} \times)^T = I - (\phi_{nb} \times)$$

这里$(\phi_{nb} \times)^T = -(\phi_{nb} \times)$。

等效旋转矢量和四元数对应关系为：

$$-\phi_{nb} \to q_n^b$$

那么v^b通过四元数和 DCM 的求法如下：

$$v^b = q_n^b \circ v^n \circ q_n^{b*}$$

$$v^b = C_n^b v^n$$

由上面（1）和（2）两种对应关系可得，

$$\Delta \boldsymbol{f}_{f,k}^n = \left[I - \frac{1}{2}(\boldsymbol{\zeta}_{n(k-1),n(k)} \times)\right] C_{b(k-1)}^{n(k-1)} \Delta \boldsymbol{f}_{f,k}^{b(k-1)}$$

该式子中$\left[I - \frac{1}{2}(\boldsymbol{\zeta}_{n(k-1),n(k)} \times)\right]$对应的等效旋转矢量为$-\frac{1}{2}\boldsymbol{\zeta}_{n(k-1),n(k)}$，也就是说写成$I \pm (\phi_{nb} \times)$的形式的 DCM，其对应的等效旋转矢量为$\pm\phi_{nb}$，必须包含前面的正负号。

2. 推导了严老师和牛老师的双子样算法

之前研究过严老师的代码，严老师的代码在做圆锥补偿和划桨效应补偿的时候，前面的系数是$\frac{2}{3}$，而牛老师前面的系数是$\frac{1}{12}$，最后请教了陈启金老师，理解了二者，二者主要区别是采用的双子样数据不用。前者是将两个历元视为一个历元进行解算，结果是可能浪费了一个历元的采样，后者是对一个历元进行解算。之后对二者的算法进行了一一推导。

3. 纬度和经度差转换为北方向和东方向的误差的关系

纬度和经度差转换为北东地方向的误差可以见 3 计算结果中，推导完之后才理解了二者之间可以通过下面的式子来进行联系。

纬度，经度和高程的微分方程如下：

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \dfrac{V_N}{R_M + h} \\ \dfrac{V_E}{(R_N + h)\cos\varphi} \\ -V_D \end{bmatrix}$$

4. 正常重力模型对结果的影响较大

起初我用的重力模型是严老师代码中的正常重力模型，发现计算出来的结果对结果影响较大，在高程方向的结果和参考的结果在高程方向的差值达到了 1km。后来改用了牛老师的重力模型，发现结果和参考的结果接近了。

对于代码的一点心得：（1）起初在计算正常重力模型的函数中传入的参数应该是纬度和高程，但是我传入的是高程和纬度，导致计算错误，一行代码就导致了很大的问题。（2）最开始自己根据严老师开源 PSINS 的代码，后来发现结果跟参考的计算结果差别较大。随后自己在严老师的基础上重新写代码，利用了 PSINS 工具箱中的四元数等一些计算函数。当然，DCM 和欧拉角之间的转换这些与坐标系相关的函数，我已经对它重写了，最终形成了一套北东地的坐标系的纯惯导的算法。期间还得到了李涛的帮助，利用它的代码和我的代码一步步对照计算结果，找到了我的计算 bug，是其中的一个计算正常重力模型的函数传参错误。

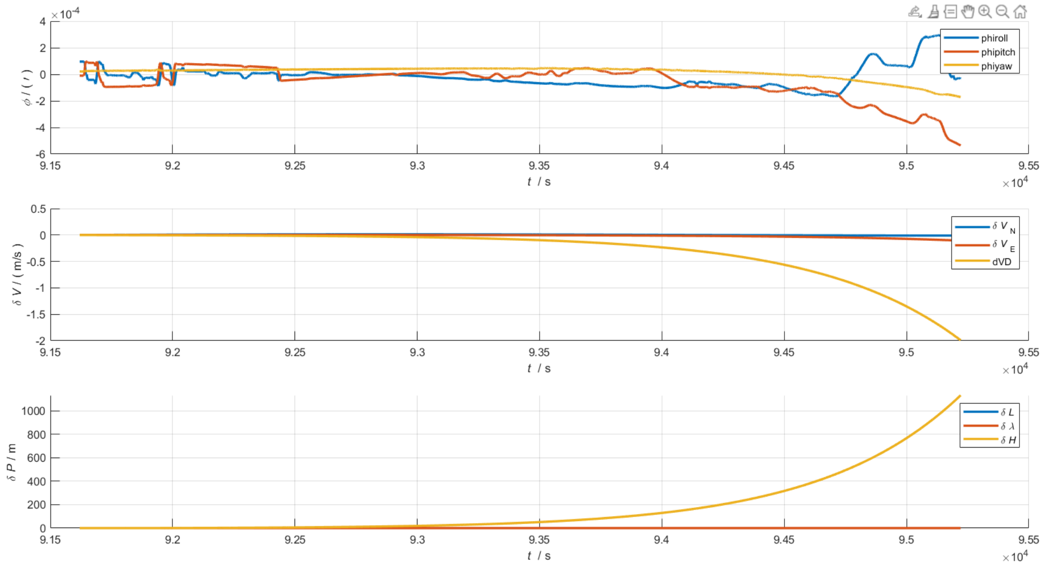

图 3 grs80 正常重力模型计算的结果和参考计算值的差值