中国科学院上海天文台
中国科学院大学
University of Chinese Academy of Sciences

空间飞行器精密定轨

宋叶志

2021秋季　　作业邮箱：song.yz@foxmail.com
课件地址：http://202.127.29.4/astrodynamics

第十讲 轨道摄动力（一）

- 地球非球形摄动加速度
- 固体潮
- 海潮
- 大气潮
- 自转引起的附加摄动
- 第三体引力摄动

卫星在地固坐标系下引力加速度

$$V(r,\varphi,\lambda)=\frac{GM_E}{r}\left[1+\sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{a_E}{r}\right)^{l}\bar{P}_{lm}(\sin\varphi)[\bar{C}_{lm}\cos m\lambda+\bar{S}_{lm}\sin m\lambda]\right]$$

$$\vec{f}_{CTS}=\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}=\begin{pmatrix} \frac{\partial V}{\partial x} \\ \frac{\partial V}{\partial y} \\ \frac{\partial V}{\partial z} \end{pmatrix}=\begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial \varphi}{\partial x} & \frac{\partial \lambda}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial \varphi}{\partial y} & \frac{\partial \lambda}{\partial y} \\ \frac{\partial r}{\partial z} & \frac{\partial \varphi}{\partial z} & \frac{\partial \lambda}{\partial z} \end{pmatrix}\begin{pmatrix} \frac{\partial V}{\partial r} \\ \frac{\partial V}{\partial \varphi} \\ \frac{\partial V}{\partial \lambda} \end{pmatrix}=\frac{\partial(r,\varphi,\lambda)}{\partial(x,y,z)}^{T}\cdot\begin{pmatrix} \frac{\partial V}{\partial r} \\ \frac{\partial V}{\partial \varphi} \\ \frac{\partial V}{\partial \lambda} \end{pmatrix}$$

引力位关于球坐标偏导数

$$\begin{cases} \dfrac{\partial V}{\partial r} = -\dfrac{GM_E}{r^2}\left[1+\sum_{l=2}^{\infty}\sum_{m=0}^{l}(l+1)\left(\dfrac{a_E}{r}\right)^l \bar{P}_{lm}(\sin\varphi)[\bar{C}_{lm}\cos m\lambda+\bar{S}_{lm}\sin m\lambda]\right] \\ \dfrac{\partial V}{\partial \varphi} = \dfrac{GM_E}{r}\left[\sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\dfrac{a_E}{r}\right)^l \quad \beta\cdot\bar{P}_{l(m+1)}(\sin\varphi)-m\cdot\tan\varphi\cdot\bar{P}_{lm}(\sin\varphi)\quad\cdot[\bar{C}_{lm}\cos m\lambda+\bar{S}_{lm}\sin m\lambda]\right] \\ \dfrac{\partial V}{\partial \lambda} = \dfrac{GM_E}{r}\left[\sum_{l=2}^{\infty}\sum_{m=0}^{l} m\cdot\left(\dfrac{a_E}{r}\right)^l \bar{P}_{lm}(\sin\varphi)[-\bar{C}_{lm}\sin m\lambda+\bar{S}_{lm}\cos m\lambda]\right] \end{cases}$$

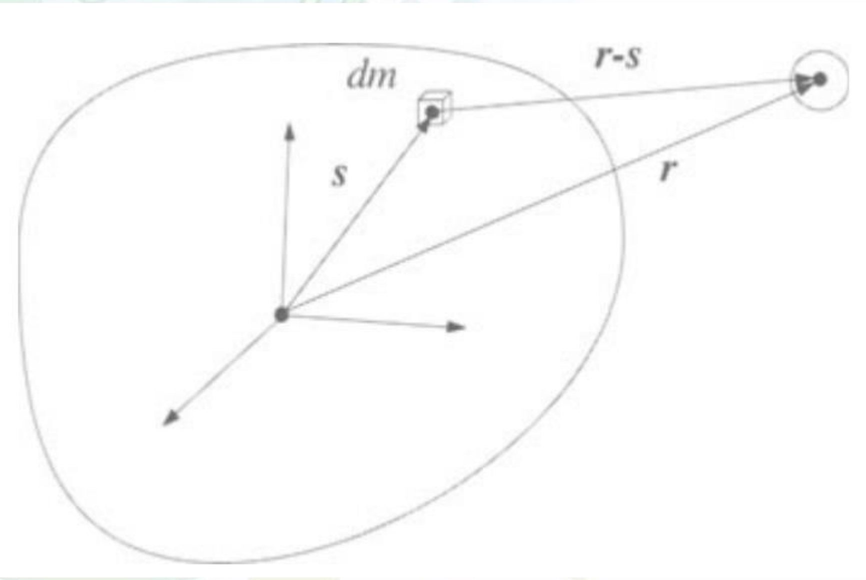
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos\varphi\cos\lambda \\ r\cos\varphi\sin\lambda \\ r\sin\varphi \end{pmatrix}, \quad \begin{pmatrix} r \\ \varphi \\ \lambda \end{pmatrix} = \begin{pmatrix} \sqrt{x^2+y^2+z^2} \\ \tan^{-1}(z/\sqrt{x^2+y^2}) \\ \tan^{-1}(y/x) \end{pmatrix}$$

$$\frac{\partial(r,\varphi,\lambda)}{\partial(x,y,z)} = \begin{pmatrix} \dfrac{\partial r}{\partial x} & \dfrac{\partial r}{\partial y} & \dfrac{\partial r}{\partial z} \\ \dfrac{\partial \varphi}{\partial x} & \dfrac{\partial \varphi}{\partial y} & \dfrac{\partial \varphi}{\partial z} \\ \dfrac{\partial \lambda}{\partial x} & \dfrac{\partial \lambda}{\partial y} & \dfrac{\partial \lambda}{\partial z} \end{pmatrix} = \begin{pmatrix} \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \\ -\dfrac{1}{r}\sin\varphi\cos\lambda & -\dfrac{1}{r}\sin\varphi\sin\lambda & \dfrac{1}{r}\cos\varphi \\ -\dfrac{1}{r}\dfrac{\sin\lambda}{\cos\varphi} & \dfrac{1}{r}\dfrac{\cos\lambda}{\cos\varphi} & 0 \end{pmatrix}$$

等价计算方法

$$C_{nm} = \frac{2-\delta_{0m}}{M_\oplus}\frac{(n-m)!}{(n+m)!}\int \frac{s^n}{R_\oplus^n} P_{nm}(\sin\phi')\cos(m\lambda')\rho(\mathbf{s})\, d^3\mathbf{s}$$

$$S_{nm} = \frac{2-\delta_{0m}}{M_\oplus}\frac{(n-m)!}{(n+m)!}\int \frac{s^n}{R_\oplus^n} P_{nm}(\sin\phi')\sin(m\lambda')\rho(\mathbf{s})\, d^3\mathbf{s}$$



等价计算方法

$$U = \frac{GM_{\oplus}}{R_{\oplus}} \sum_{n=0}^{\infty} \sum_{m=0}^{n} (C_{nm} V_{nm} + S_{nm} W_{nm})$$

$$V_{nm} = \left(\frac{R_{\oplus}}{r}\right)^{n+1} \cdot P_{nm}(\sin\phi) \cdot \cos m\lambda$$

$$W_{nm} = \left(\frac{R_{\oplus}}{r}\right)^{n+1} \cdot P_{nm}(\sin\phi) \cdot \sin m\lambda$$

$$V_{mm} = (2m-1)\left\{\frac{xR_{\oplus}}{r^2} V_{m-1,m-1} - \frac{yR_{\oplus}}{r^2} W_{m-1,m-1}\right\}$$

$$W_{mm} = (2m-1)\left\{\frac{xR_{\oplus}}{r^2} W_{m-1,m-1} + \frac{yR_{\oplus}}{r^2} V_{m-1,m-1}\right\}$$

$$V_{nm} = \left(\frac{2n-1}{n-m}\right) \cdot \frac{zR_{\oplus}}{r^2} \cdot V_{n-1,m} - \left(\frac{n+m-1}{n-m}\right) \cdot \frac{R_{\oplus}^2}{r^2} \cdot V_{n-2,m}$$

$$W_{nm} = \left(\frac{2n-1}{n-m}\right) \cdot \frac{zR_{\oplus}}{r^2} \cdot W_{n-1,m} - \left(\frac{n+m-1}{n-m}\right) \cdot \frac{R_{\oplus}^2}{r^2} \cdot W_{n-2,m}$$

等价计算方法

$$\ddot{x} = \sum_{n,m} \ddot{x}_{nm} \quad , \quad \ddot{y} = \sum_{n,m} \ddot{y}_{nm} \quad , \quad \ddot{z} = \sum_{n,m} \ddot{z}_{nm}$$

$$\begin{aligned}
\ddot{x}_{nm} &\overset{(m=0)}{=} \frac{GM}{R_\oplus^2} \cdot \left\{ -C_{n0} V_{n+1,1} \right\} \\
&\overset{(m>0)}{=} \frac{GM}{R_\oplus^2} \cdot \frac{1}{2} \cdot \left\{ (-C_{nm} V_{n+1,m+1} - S_{nm} W_{n+1,m+1}) \right. \\
&\qquad \left. + \frac{(n-m+2)!}{(n-m)!} \cdot (+C_{nm} V_{n+1,m-1} + S_{nm} W_{n+1,m-1}) \right\}
\end{aligned}$$

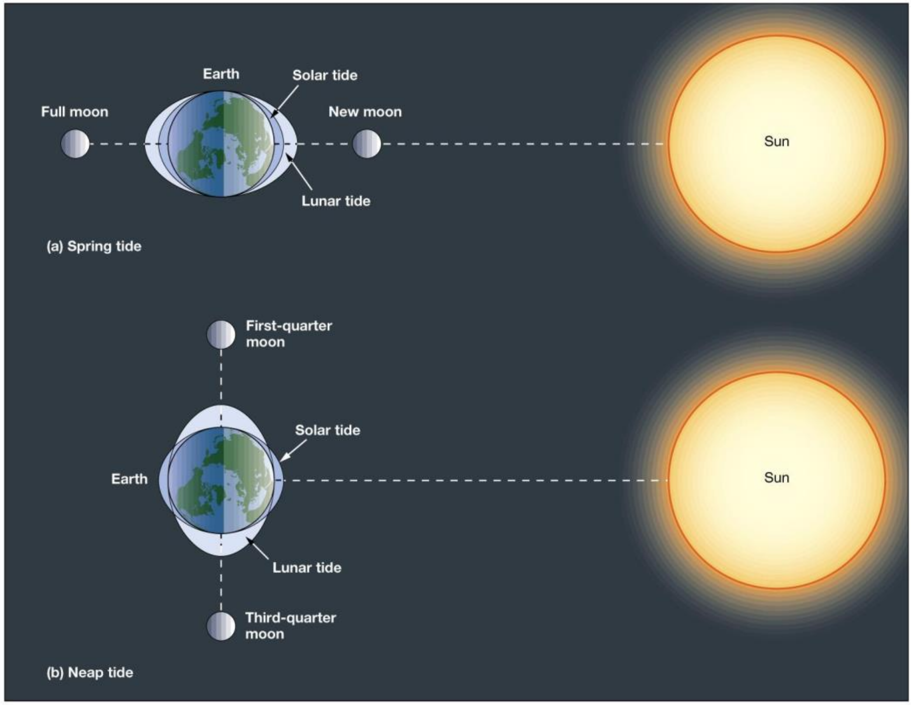
$$\begin{aligned}
\ddot{y}_{nm} &\overset{(m=0)}{=} \frac{GM}{R_\oplus^2} \cdot \left\{ -C_{n0} W_{n+1,1} \right\} \\
&\overset{(m>0)}{=} \frac{GM}{R_\oplus^2} \cdot \frac{1}{2} \cdot \left\{ (-C_{nm} \cdot W_{n+1,m+1} + S_{nm} \cdot V_{n+1,m+1}) \right. \\
&\qquad \left. + \frac{(n-m+2)!}{(n-m)!} \cdot (-C_{nm} W_{n+1,m-1} + S_{nm} V_{n+1,m-1}) \right\}
\end{aligned}$$

$$\ddot{z}_{nm} = \frac{GM}{R_\oplus^2} \cdot \left\{ (n-m+1) \cdot (-C_{nm} V_{n+1,m} - S_{nm} W_{n+1,m}) \right\} \quad .$$

$$\begin{array}{ccccccccc}
V_{00}, W_{00} & & & & & & & & \\
\downarrow & \searrow & & & & & & & \\
V_{10}, W_{10} & & V_{11}, W_{11} & & & & & & \\
\downarrow & & \downarrow & \searrow & & & & & \\
V_{20}, W_{20} & & V_{21}, W_{21} & & V_{22}, W_{22} & & & & \\
\downarrow & & \downarrow & & \downarrow & \searrow & & & \\
\vdots & & \vdots & & \vdots & & \ddots & & \\
\downarrow & & \downarrow & & \downarrow & & & \searrow & \\
V_{n0}, W_{n0} & & V_{n1}, W_{n1} & & V_{n2}, W_{n2} & & \cdots & & V_{nn}, W_{nn}
\end{array}$$

固体潮汐

地球并非刚体，由于日月引力的影响
$\Rightarrow$ 地球弹性形变 $\Rightarrow$ 固体潮
$\Rightarrow$ 地球体积和密度分布之改变
$\Rightarrow$ 引力位的变化 $\Rightarrow$ 固体潮摄动。



$$\Delta\bar{C}_{lm} - i\Delta\bar{S}_{lm} = \frac{k_{lm}}{2l+1}\sum_{j=2}^{3}\frac{GM_j}{GM_E}\left(\frac{a_E}{r_j}\right)^{l+1}\bar{P}_{lm}(\sin\phi_j)e^{-im\lambda_j}$$

$$\begin{Bmatrix}\Delta\bar{C}_{lm}\\ \Delta\bar{S}_{lm}\end{Bmatrix}_{st} = \frac{k_{lm}}{2l+1}\sum_{j=2}^{3}\left(\frac{GM_j}{GM_E}\right)\left(\frac{a_E}{r_j}\right)^{l+1}\bar{P}_{lm}\sin(\varphi_j)\begin{Bmatrix}\cos m\lambda_j\\ \sin m\lambda_j\end{Bmatrix}$$

$$\Delta\bar{S}_{l0} = 0$$

固体潮汐改正步骤

固体潮模型以Wahr模型为基础，固体潮引起 发生变化。Wahr固体潮模型是弹性、椭球成层，旋转及自引力地球模型，且考虑到液核的动力学影响，因而不能用一组简单的Love数来表达出地面点的运动，Wahr模型对不同的分潮波（主要是全日波）将有不同的勒夫数。Wahr模型中Love数是随分潮波频率不同而不同。

$$\begin{cases}(\Delta\bar{C}_{20})_{DT_1} = \dfrac{1}{\sqrt{5}}k_2\displaystyle\sum_{j=1}^{2}\frac{GM_j}{GE}\left(\frac{a_e}{r_j}\right)^3 P_{20}(\sin\phi_j) - <\Delta\bar{C}_{20}> \\ (\Delta\bar{C}_{21})_{DT_1} + i(\Delta\bar{S}_{21})_{DT_1} = \dfrac{1}{3}\sqrt{\dfrac{3}{5}}k_2\displaystyle\sum_{j=1}^{2}\frac{GM_j}{GE}\left(\frac{a_e}{r_j}\right)^3 P_{21}(\sin\phi_j)e^{i\lambda_j} \\ (\Delta\bar{C}_{22})_{DT_1} + i(\Delta\bar{S}_{22})_{DT_1} = \dfrac{1}{12}\sqrt{\dfrac{12}{5}}k_2\displaystyle\sum_{j=1}^{2}\frac{GM_j}{GE}\left(\frac{a_e}{r_j}\right) P_{22}(\sin\phi_j)e^{i2\lambda_j}\end{cases}$$

$$\begin{cases}(\Delta\bar{C}_{21})_{DT_2} + i(\Delta\bar{S}_{21})_{DT_2} = \displaystyle\sum_{S(2,1)} A_1\delta\ K_S H_S(\sin\theta_S + i\cos\theta_S) \\ (\Delta\bar{C}_{22})_{DT_2} + i(\Delta\bar{S}_{22})_{DT_2} = \displaystyle\sum_{S(2,2)} A_2\delta\ K_S H_S(\cos\theta_S - i\sin\theta_S)\end{cases}$$

海潮摄动

日月引力⇒海潮⇒ $\left\{\begin{array}{l}\text{海水负荷变化}\\ \text{负荷变化引起形变}\end{array}\right\}$ ⇒引力位变化

海潮模型以Schwiderski (1983)模型为基础的，它的动力学效应也可以通过对引力场系数的修正来体现的。

$$\Delta\bar{C}_{lm} - i\Delta\bar{S}_{lm} = F_{lm}\sum_{s(l,m)}\sum_{+}^{-}(C_{snm}^{\pm} \mp iS_{snm}^{\pm})e^{\pm i\theta_s}$$

$$F_{lm} = \frac{4\pi a_E^2\rho_w}{M_E}\sqrt{\frac{(l+m)!}{(2-\delta_{0m})(2l+1)(l-m)!}}\left(\frac{1+k_l'}{2l+1}\right)$$

$$\begin{cases}\Delta\bar{C}_{lm} = F_{lm}\sum_{s(l,m)}((C_{slm}^{+} + C_{slm}^{-})\cos\theta_s + (S_{slm}^{+} + S_{slm}^{-})\sin\theta_s)\\ \Delta\bar{S}_{lm} = F_{lm}\sum_{s(l,m)}((S_{slm}^{+} - S_{slm}^{-})\cos\theta_s - (C_{slm}^{+} - C_{slm}^{-})\sin\theta_s)\end{cases}$$

大气潮

大气潮起因——引力源、热源（主要），这也是主大气潮具有半太阳日的周期并比月潮大15倍左右的原因。据Lambeck等人（1974年）研究，大气潮中仅S_2波是主要的，而且S_2大气潮汐摄动的效应相当于固体潮效应的2.5%。

$$(\Delta\bar{C}_{nm})_{AT} = \sum_{\mu(n,m)} F_{nm}\left[(C_{\mu nm}^{A+} + C_{\mu nm}^{A-})\cos(\bar{n}_{\mu}\cdot\bar{\beta}) + (S_{\mu nm}^{A+} + S_{\mu nm}^{A-})\sin(\bar{n}_{\mu}\cdot\bar{\beta})\right]$$

$$(\Delta\bar{S}_{nm})_{AT} = \sum_{\mu(n,m)} F_{nm}\left[(S_{\mu nm}^{A+} + S_{\mu nm}^{A-})\cos(\bar{n}_{\mu}\cdot\bar{\beta}) - (C_{\mu nm}^{A+} - C_{\mu nm}^{A-})\sin(\bar{n}_{\mu}\cdot\bar{\beta})\right]$$

$$\bar{C}_{S_2 22}^{A+} = 0.344mb \qquad C_{S_2 22}^{A+} = \frac{\bar{C}_{S_2 22}^{A+}\sin\varepsilon_{S_2 22}^{A+}}{g\ \rho_w}$$

$$\varepsilon_{S_2 22}^{A+} = 158^0 \qquad S_{S_2 22}^{A+} = \frac{\bar{C}_{S_2 22}^{A+}\cos\varepsilon_{S_2 22}^{A+}}{g\ \rho_w}$$

地球自转形变附加摄动

地球自转离心力⇒形变⇒地球体积和密度分布的改变将引入一个附加位⇒对卫星运动产生附加摄动

$$\begin{cases} (\Delta\bar{C}_{20})_{R0} = -\dfrac{1}{\sqrt{5}}\dfrac{2a_e^3}{3GE}k_2 m_3 \Omega^2 \\ (\Delta\bar{C}_{21})_{R0} = -\dfrac{1}{\sqrt{15}}\dfrac{a_e^3}{GE}k_2 m_1 \Omega^2 \\ (\Delta\bar{S}_{21})_{R0} = -\dfrac{1}{\sqrt{15}}\dfrac{a_e^3}{GE}k_2 m_2 \Omega^2 \end{cases}$$

$$m_1 = x_p,\ \ m_2 = -y_p,\ \ m_3 = -\frac{D}{86400000}$$

非球形引力位与潮汐总加速度

$$U=\frac{GE}{r}\sum_{n=2}^{N}\sum_{m=0}^{n}\left(\frac{a_e}{r}\right)^n\overline{P}_{nm}(\sin\phi)(\overline{C}_{nm}^{*}\cos m\lambda+\overline{S}_{nm}^{*}\sin m\lambda)$$

$$\begin{cases}\overline{C}_{nm}^{*}=\overline{C}_{nm}+(\Delta\overline{C}_{nm})_{DT}+(\Delta\overline{C}_{nm})_{OT}+(\Delta\overline{C}_{nm})_{AT}+(\Delta\overline{C}_{nm})_{R0}\\ \overline{S}_{nm}^{*}=\overline{S}_{nm}+(\Delta\overline{S}_{nm})_{DT}+(\Delta\overline{S}_{nm})_{OT}+(\Delta\overline{S}_{nm})_{AT}+(\Delta\overline{S}_{nm})_{R0}\end{cases}$$

$$\vec{A}_{NS}+\vec{A}_{DT}+\vec{A}_{OT}+\vec{A}_{AT}+\vec{A}_{R0}=(HG)^T\left(\frac{\partial U}{\partial\vec{r}(x,y,z)}\right)^T$$

HG矩阵见时间与空间部分

天球与地球坐标系转换回顾

$$\boldsymbol{r}_{\mathrm{GCRS}} = \mathbf{BP}(t)\mathbf{N}(t)\mathbf{S}(t)\mathbf{W}(t)\boldsymbol{r}_{\mathrm{ITRS}}$$

$$\boldsymbol{r}_{\mathrm{ITRS}} = [\mathbf{W}(t)]^{\mathrm{T}}[\mathbf{S}(t)]^{\mathrm{T}}[\mathbf{N}(t)]^{\mathrm{T}}[\mathbf{P}(t)]^{\mathrm{T}}[\mathbf{B}]^{\mathrm{T}}\boldsymbol{r}_{\mathrm{GCRS}}$$

$$\begin{aligned}\boldsymbol{v}_{\mathrm{ITRS}} &= [\mathbf{W}]^{\mathrm{T}}\left\{[\mathbf{S}]^{\mathrm{T}}[\mathbf{BPN}]^{\mathrm{T}}\boldsymbol{v}_{\mathrm{GCRS}} - \boldsymbol{\omega}_{E} \times \boldsymbol{r}_{\mathrm{ECEFw/oPM}}\right\} \\ &= [\mathbf{W}]^{\mathrm{T}}[\mathbf{S}]^{\mathrm{T}}[\mathbf{BPN}]^{\mathrm{T}}\boldsymbol{v}_{\mathrm{GCRS}} + [\mathbf{W}]^{\mathrm{T}}[\mathbf{S}']^{\mathrm{T}}[\mathbf{BPN}]^{\mathrm{T}}\boldsymbol{r}_{\mathrm{GCRS}}\end{aligned}$$

$$\begin{aligned}\boldsymbol{v}_{\mathrm{GCRS}} &= [\mathbf{BPN}]\,[\mathbf{S}]\,\{[\mathbf{W}]\boldsymbol{v}_{\mathrm{ITRS}} + \boldsymbol{\omega}_{E} \times \boldsymbol{r}_{\mathrm{ECEFw/oPM}}\} \\ &= [\mathbf{BPN}]\,[\mathbf{S}]\,[\mathbf{W}]\,\boldsymbol{v}_{\mathrm{ITRS}} + [\mathbf{BPN}][\mathbf{S}'][\mathbf{W}]\boldsymbol{r}_{\mathrm{GCRS}}\end{aligned}$$

两个典型引力场模型

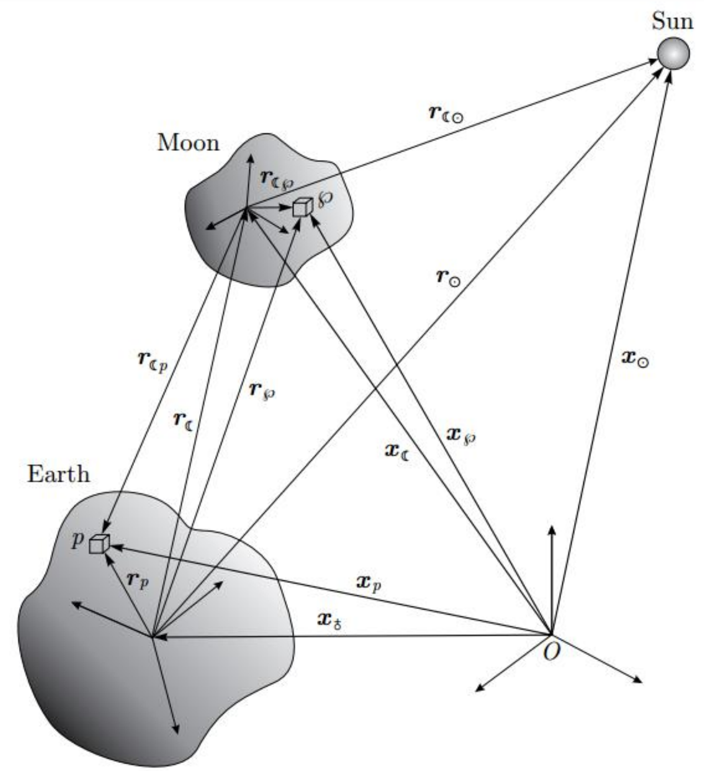
- EGM2008地球重力场模型是由NGA发布的全球超高阶重力场模型，它以PGM2007B 为参考，综合利用GRACE卫星重力数据、全球5’×5’重力异常数据、TOPEX 卫星测高数据、地形数据、地面重力数据。该地球重力场模型研制周期为4年，球谐系数的阶扩展至2190，阶次完全至2159，经过大量的测试与评估结果表明，2190阶的EGM2008地球重力场模型比其他模型的精度有了大幅度的提高，是迄今为止分辨率最高、精度最好、阶次最多的全球重力场模型之一。
- EIGEN-6C4重力场模型是由GFZ在2014 年11月发布的最新2190阶次重力场模型，成为继EIGEN-6C、EIGEN-6C2、EIGEN-6C3sata后，发布的又一超高阶重力场模型。该模型采用卫星重力数据（GOCE、GRACE、SLR）、地面重力数据与卫星测高数据解算而成，最高阶可达2190（前50阶含有时变参数）。与之前发布的模型相比，该模型在中长波段的精度有较大的提升，也是目前较优的一个超高阶地球重力场模型。

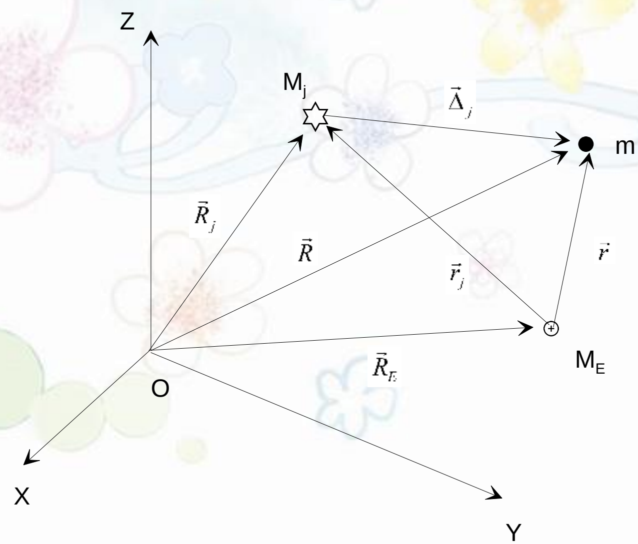
第三体引力

$$\ddot{\vec{r}} = -\frac{GM_E}{r^2}\cdot\frac{\vec{r}}{r} - \frac{GM_j}{\Delta_j^2}\cdot\frac{\vec{\Delta}_j}{\Delta_j} - \frac{GM_j}{r_j^2}\cdot\frac{\vec{r}_j}{r_j}$$

$$= -\frac{GM_E}{r^2}\cdot\frac{\vec{r}}{r} - GM_j\left(\frac{\vec{r}_j}{r_j^3}+\frac{\vec{\Delta}_j}{\Delta_j^3}\right)$$

$$\ddot{\vec{r}}_N = -\sum_{j=1}^{N} GM_j\left(\frac{\vec{r}_j}{r_j^3}+\frac{\vec{\Delta}_j}{\Delta_j^3}\right)$$





JPL行星历表

- JPL 星历是由美国喷气推进实验室研制，目前是为太空导航，行星探测以及精密天文观测的分析和归算提供精密数据，目前JPL 的主要星历有DE200、DE403、DE405、DE430。

如DE405，覆盖了从1600年到2170年大约600年时间段。所有星历都基于各自运动方程进行严格数值积分。除了月球、行星、和太阳的点质量相互作用外，部分小行星的摄动和运动方程的相对论后牛顿修正也要考虑。另外，日月扭矩对地球形状的影响，以及地球和太阳扭矩对月球形状的影响都精细了考虑

Mass parameters from DE421 expressed as ratios and as TDB-compatible values.

	$GM_{\odot}/GM_i$	$GM_i/\mathrm{km}^3\mathrm{s}^{-2}$
Mercury	6023597.400017	22032.090000
Venus	408523.718655	324858.592000
Earth	332946.048166	398600.436233
Moon	27068703.185436	4902.800076
Mars	3098703.590267	42828.375214
Jupiter	1047.348625	126712764.800000
Saturn	3497.901768	37940585.200000
Uranus	22902.981613	5794548.600000
Neptune	19412.237346	6836535.000000
Pluto	135836683.767599	977.000000

	$GM_{\oplus}/GM_i$
Earth-Moon mass ratio	81.3005690699

DE历表切比雪夫多项式逼近

	水星	金星	地月系质心	火星	木星	土星	天王星	海王星	冥王星	月球相对地心	太阳	章动
编号	1	2	3	4	5	6	7	8	9	10	11	12
I	4	1	2	1	1	1	1	1	1	8	1	4
N	12	12	15	10	9	8	8	6	6	12	15	10

$$\begin{cases} T_1(\tau)=1 \\ T_2(\tau)=\tau \\ T_i(\tau)=2\tau T_{i-1}(\tau)-T_{i-2}(\tau), i\geq 3 \end{cases}$$

$$\begin{cases} T_1'(\tau)=0 \\ T_2'(\tau)=1 \\ T_i'(\tau)=2T_{i-1}(\tau)+2\tau T_{i-1}'-T_{i-2}'(\tau), i\geq 3 \end{cases}$$

$$\tau=\frac{2(t-t_0)}{\Delta t}-1$$

拟合出系数后，如果需要计算速度，则需要把速度量纲的分母项还原为原量纲。如，原始位置速度单位为km，拟合数据区间为 。拟合出的速度分母量纲为无单位量纲。可以通过以下式子还原为 m/s量纲。

$$v=\frac{v_{fit}\times 1000}{60\times\Delta t\times 2}$$

切比雪夫多项式计算

```fortran
subroutine basecheby(bv,t,N,bvel)
!-------------------------------subroutine  comment
!  Version   :  V1.0
!  Coded by  :  syz
!  Date      :2010.07.09
!--------------------------------------------------
!  Purpose     :  切比雪夫多项式基函数!
!------------------------------
!  Input  parameters  :
!       1.    t 自变量
!       2.    m 切比雪夫多项式阶数
!  Output parameters  :
!       1.    bv ------ 切比雪夫基向量
!       2.    bvel ------ 速度基向量（与切比雪夫向量维数相同）
!-------------------------------------------------
implicit none
integer         ::N
real*8          ::bv(N+1) ,t , bvel(N+1)
integer         ::i
bv(1) = 1d0
bv(2) = t
do i=3,N+1
    bv(i) = 2d0* t * bv(i-1)  - bv(i-2)
end do
!位置基向量计算完毕
!--------------------
bvel(1) = 0d0
bvel(2) = 1d0
do i=3, N+1
    bvel(i) = 2d0 * bv(i-1) + 2d0 *t * bvel(i-1) - bvel(i-2)
end do
!速度基向量计算完毕
end subroutine basecheby
```

```fortran
subroutine chebyval(p,N,x,fx,vel)
!-----------------------------------------------------
! Purpose : 2019-05-08 10:11                (Created)
!    polyval(p,x) returns the value of a polynomial
!    of degree n evaluated at x.
!    The input argument p is a vector of
!    length n+1 whose elements are the coefficients in
!    powers of the polynomial to be evaluated.
!    fx = p(1) + p(2)x + p(3)x*x + ....
!-----------------------------------------------------
! Input Parameters    :
!    p  多项式系数
!    N  多项式阶数
!    x  自变量
! Output Parameters   :
!    fx  应变量    !
!-----------------------------------------------------
! Author       : Song Yezhi      <song.yz@foxmail.com>
! Copyrigt (C) : Shanghai Astronomical Observatory,CAS
!              (All rights reserved, 2019)
!----------------------------------------------------
implicit none
integer          :: N ,i
real*8           :: p(N+1),x ,fx ,H(N+1) ,vel, Hvel(N+1)
!-----------------------------------------------------
call basecheby(H,x,N,Hvel)
!获取切比雪夫多项式及其导数的基向量
fx = p(1) * H(1)     ! p(1) *1
vel = p(1) * Hvel(1) ! p(1) *0
do i=2,N+1
    fx  = fx + p(i) *H(i)
    vel = vel+ p(i)* Hvel(i)
end do
end subroutine chebyval
```

生成切比雪夫系数

```
program main
!-----------------------------------------
!  Purpose   :
!  Author    :  Song Yezhi      <song.yz@
!  Versions and Changes   :
!       V1.0 ----2019-05-08 09:16:37!
!-----------------------------------------
!  Notes     :
!     the coefficients for a polynomial p(
!     n that is a best fit (in a least-squ
!     The coefficients in p are in powers
!-----------------------------------------
!  Copyrigt (C)  :
!        Center for Astro-geodynamics
!        Shanghai Astronomical Observatory
!        Chinese Academy of Sciences
!-----------------------------------------
implicit none
integer                           :: N, r
character*8                       :: strN
real*8                            :: tspa
narg = iargc()
!get the number of args input
if (narg<2) then
   write(*,*) 'usage:'
   write(*,*) 'main.app  N  tspan'
   write(*,*) ' N  is the order of the pol
   write(*,*) ' tspan is time span for dat
else
  call getarg(1,strN)
  call getarg(2,strSpan)
  read(strN,*) N
  read(strSpan,*) tspan
  call ephfit_shell(N,tspan)
end if
end program main
```

```
subroutine ephfit_shell(N,tspan)
!---------------------------------------------
! Purpose : 2019-06-05 17:42                (Creat
!---------------------------------------------
! Author      : Song Yezhi      <song.yz@foxmail.c
! Copyrigt (C) : Shanghai Astronomical Observatory,
!               (All rights reserved, 2019)
!---------------------------------------------
implicit none
integer                 :: N,i
real*8                  :: px(N+1),py(N+1),pz(N+1)
real*8                  :: tspan
!---------------------------------------------
open(unit=12,file='coe.txt')
write(12,'(A)') 'order of chebyshev polynomial  and
write(12,'(I5,F18.10)')  N, tspan
call ephfit(N,tspan)
end subroutine ephfit_shell
```

生成切比雪夫系数

```fortran
subroutine ephfit(N,tspan)
!---------------------------------------------------------
! Purpose : 2019-05-08 09:50                 (Created)
!---------------------------------------------------------
! Input Parameters    :
!      M ------ dimention of observation data
!      N ------ order of polynomial
!
! Output Parameters   :
!      px ---- coefficients of x
!      py ---- coefficients of y
!      pz ---- coefficients of z
!---------------------------------------------------------
! Author       : Song Yezhi      <song.yz@foxmail.com>
! Copyrigt (C) : Shanghai Astronomical Observatory,CAS
!               (All rights reserved, 2019)
!---------------------------------------------------------

implicit none
integer              ::i,ioerr,N, M ,iter
real*8               :: ATAx(N+1,N+1),ATbx(N+1),px(N+1),covx(N+1,N+1),Hx(N+1)
real*8               :: ATAy(N+1,N+1),ATby(N+1),py(N+1),covy(N+1,N+1),Hy(N+1)
real*8               :: ATAz(N+1,N+1),ATbz(N+1),pz(N+1),covz(N+1,N+1),Hz(N+1)
integer              :: year,mon,day,hour,minu ,day0,hour0,minu0
real*8               :: sec,sec0, x, y ,z  ,t, tspan
open(unit=11,file="xyz_ECF")
open(unit=17,file="tmp")
ATAx = 0D0
ATbx = 0d0
ATAy = 0D0
ATby = 0d0
ATAz = 0D0
ATbz = 0d0
iter = 0
do
   read(11,*,iostat=ioerr) year,mon,day0,hour0,minu0,sec0 , x, y, z
   if(ioerr/=0) exit
   backspace(11)
   iter = iter+1
   write(12, '(A3,i5)') 'coe',iter
   write(17, '(A3,i5)') 'coe',iter
   do
           read(11,*,iostat=ioerr) year,mon,day,hour,minu,sec , x, y, z
           if(ioerr/=0) exit
           t= (day-day0)*24*60 + (hour-hour0)*60 +  (minu - minu0)  + (sec - sec0)/60d0
           if (t>=tspan) then
                   backspace(11)
                   exit
           end if
           t = 2d0*(t-0d0)/tspan - 1d0
           write(17,'(F15.8,I7,4I4,f10.3,3f20.3)') t, year,mon,day,hour,minu,sec, x, y, z
           x = x/1d3
           y = y/1d3
           z = z/1d3
           call basecheby(Hx,t,N)
           call add_NEQ(ATAx,ATbx,x ,Hx,N+1)
           call basecheby(Hy,t,N)
           call add_NEQ(ATAy,ATby,y ,Hy,N+1)
           call basecheby(Hz,t,N)
           call add_NEQ(ATAz,ATbz,z ,Hz,N+1)
   end do
   call chol_eq(ATAx,ATbx,Px,covx,N+1)
   call chol_eq(ATAy,ATby,Py,covy,N+1)
   call chol_eq(ATAz,ATbz,Pz,covz,N+1)
   ATAx = 0D0
   ATbx = 0d0
   ATAy = 0D0
   ATby = 0d0
   ATAz = 0D0
   ATbz = 0d0
   write(12,'(<N+1>(F25.12,/))') (px(i),i=1,N+1)
   write(12,'(<N+1>(F25.12,/))') (py(i),i=1,N+1)
   write(12,'(<N+1>(F25.12,/))') (pz(i),i=1,N+1)
end do
end subroutine ephfit
```

生成切比雪夫系数

```fortran
subroutine basecheby(bv,t,N)
!-----------------------------------subroutine  comme
!  Version   :  V1.0
!  Coded by  :  syz
!  Date      :2010.07.09
!--------------------------------------------------------
!  Purpose     :  任意阶多项式基函数
!--------------------------------------------------------
implicit none
integer         ::N
real*8          ::bv(N+1) ,t
integer         ::i
bv(1) = 1d0
bv(2) = t
do i=3,N+1
    bv(i) = 2d0* t * bv(i-1)  - bv(i-2)
end do
end subroutine basecheby
```

```fortran
subroutine covariance(L,D,P,N)
!----------------------------------------------------------------
! Purpose : 2019-04-15 13:08                 (Created)
!   根据LDL'分解计算协方差矩阵
!----------------------------------------------------------------
! Input Parameters    :
!    L    ------
!    D    ----  对角线元素
!    N    ----  矩阵维数
! Output Parameters   :
!----------------------------------------------------------------
! Author       : Song Yezhi      <song.yz@foxmail.com>
! Copyrigt (C)  : Shanghai Astronomical Observatory,CAS
!                (All rights reserved, 2019)
!----------------------------------------------------------------
implicit none
integer     :: N
real*8      :: L(N,N),D(N),P(N,N)
!----------------------------------------------------------------
integer     :: i
real*8      :: invL(N,N),invLT(N,N)
!----------------------------------------------------------------
call inv_dtri(L,invL,N)
invLT = transpose(invL)
do i =1,N
    invLT(:,i)=invLT(:,i)/D(i)
end do
P = matmul(invLT,invL)
end subroutine covariance
```

```fortran
subroutine chol_eq(A,b,x,P,N)
!---------------------------------subroutine  comment
!  Version   :  V1.0
!  Coded by  :  song.yz
!  Date      :  2019.04.15
!----------------------------------------------------------
!  Purpose     : 对称正定方程解算（如法方程）!
!  Post Script :
!       1.    修正的不开平方Cholesky分解法
!----------------------------------------------------------
!  Input  parameters  :
!       1.  A(N,N)对称正定系数矩阵
!       2.  b(N)右向量
!       3.  N--方程维数
!  Output parameters  :
!       1.  x-计算结果
!       2.  P ----协方差 （由于A已经为对称正定矩阵<法矩阵>，因
!----------------------------------------------------------
!  Copyrigt  :
!        Center for Astro-geodynamics
!        Shanghai Astronomical Observatory
!        Chinese Academy of Sciences
!----------------------------------------------------------
implicit real*8(a-z)
integer::N
real*8::A(N,N),b(n),x(n),P(N,N)
!---------------------------------subroutine variable
real*8::L(N,N),d(n)
integer::i,k
real*8::y(N)
call chol_rf(A,L,d,P,n)
!以上已经得到Cholesky分解
y(1)=b(1)
do i=2,n
   tmp1=0d0
   do k=1,i-1
      tmp1=tmp1+L(i,k)*y(k)
   end do
   y(i)=b(i)-tmp1
end do
x(n)=y(n)/d(n)
do i=n-1,1,-1
  tmp1=0d0
  do k=i+1,n
  tmp1=tmp1+L(k,i)*x(k)
  end do
  x(i)=y(i)/d(i)-tmp1
end do
end subroutine chol_eq
```

生成切比雪夫系数

```fortran
subroutine polyval(p,N,x,fx)
!-------------------------------------------------------------
! Purpose : 2019-05-08 10:11                  (
!    polyval(p,x) returns the value of a polyn
!    The input argument p is a vector of lengt
!    powers of the polynomial to be evaluated.
!    fx = p(1) + p(2)x + p(3)x*x + ....
!-------------------------------------------------------------
! Input Parameters    :
!    p  多项式系数
!    N  多项式阶数
!    x  自变量
! Output Parameters   :
!    fx  应变量
!-------------------------------------------------------------
! Author       : Song Yezhi      <song.yz@foxm
! Copyrigt (C) : Shanghai Astronomical Observa
!               (All rights reserved, 2019)
!-------------------------------------------------------------
implicit none
integer          :: N ,i
real*8           :: p(N+1),x ,fx ,H(N+1)
!-------------------------------------------------------------
call basefunc(H,x,N+1)
fx =p(1)
do i=2,N+1
   fx =fx + p(i) *H(i)
end do
end subroutine polyval
```

```fortran
subroutine  add_neq(ATA,ATb,oc,H,N)
!---------------------------------------subroutine  comment
!  Purpose   :   2018-09-05 13:33          (Created)
!            add a new observation to normal equatior
!              (ATA+ H'H) x = ATb + H'oc
!------------------------------------------------------------
!  Input Parameters    :
!       ATA  -----  initial normal matrix
!       Atb  -----  initial right vector
!       oc ------   o-c  for liner equation
!       H -------   obs vector
!       N -------   the dimention of the matrix
!  Output Parameters   :
!      ATA -------  updated normal matrix
!      ATb --------  updated right vector
!------------------------------------------------------------
!  Author       : Song Yezhi     <song.yz@foxmail.co
!  Copyrigt (C) : Shanghai Astronomical Observatory,(
!               (All rights reserved, 2018)
!------------------------------------------------------------
implicit real*8(a-h,o-z)
real*8:: ATA(N,N),ATb(N) ,H(N)
do i = 1,N
    if (H(i)==0d0) cycle
    do j =1 ,N
      ATA(i,j) = ATA(i,j)+ H(i)*H(j)
    end do
    ATb(i)= ATb(i)+H(i)*oc
end do
end subroutine  add_neq
```

生成切比雪夫系数

```fortran
subroutine chol_rf(A,L,d,P,N)
!---------------------------------subrouti
!  Version   :  V1.0
!  Coded by  :  syz
!  Date      :  2019.04.15
!--------------------------------------------
!  Post Script :
!       1.  不用开平方的Cholesky分解
!--------------------------------------------
!  Input  parameters  :
!       1. A(N,N)--输入矩阵
!       2. N----矩阵维书
!  Output parameters  :
!       1.  L矩阵
!       2.  d---对角矩阵（用向量存储）
!       4.  P ---协方差矩阵（由于A已经为对
!--------------------------------------------
!  Copyrigt  :
!        Center for Astro-geodynamics
!        Shanghai Astronomical Observatory
!        Chinese Academy of Sciences
!--------------------------------------------
implicit real*8(a-z)
integer::N
real*8::A(n,n),L(n,n),d(n),P(N,N)
!---------------------------------subrouti
integer::i,j
real*8::  g(n,n)
!设置初值
L=0d0
d(1)=a(1,1)
do i=2,n
  do j=1,i-1
   !此层循环算g(i,j)
    tmp1=0d0
    do k=1,j-1
      tmp1=tmp1+g(i,k)*L(j,k)
    end do
    g(i,j)=a(i,j)-tmp1
  end do
  do j=1,i-1
  !此层循环算L(i,j)
    L(i,j)=g(i,j)/d(j)
  end do
  !以下算d(i)
  tmp1=0d0
  do k=1,i-1
   tmp1=tmp1+g(i,k)*L(i,k)
  end do
  d(i)=a(i,i)-tmp1
end do
!设置对角线元素
do i=1,N
   L(i,i)=1d0
end do
call covariance(L,D,P,N)
end subroutine chol_rf
```

太阳简化解析历表

$$
\begin{cases}
a = 149600000km \\
e = 0.016709 \\
i = 0.^{\circ}0000 \\
\Omega+\omega=282.^{\circ}9400 \\
M = 359.^{\circ}5256 + 35999.^{\circ}049T
\end{cases}
$$

$$
\vec{r}_{\odot} = R_x(-\varepsilon)\begin{pmatrix} r_{\odot}\cos\lambda_{\odot}\cos\beta_{\odot} \\ r_{\odot}\sin\lambda_{\odot}\cos\beta_{\odot} \\ r_{\odot}\sin\beta_{\odot} \end{pmatrix}
$$

$$
\varepsilon = 23^{\circ}.4329111
$$

$$
\begin{cases}
\lambda_{e} = \Omega+\omega+M + 6892''\sin M + 72''\sin 2M \\
r_{e} = (149.619 - 2.499\cos M - 0.021\cos 2M)\cdot 10^{6}\,km
\end{cases}
$$

月球简化解析历表

月球平黄经L_0、月球平近点角l、太阳平近点角l′、月球平升交点经度F、太阳平黄经和月球平黄经之间的差D。

$$
\begin{cases}
L_0 = 218^{\circ}.31617 + 481267^{\circ}.88088 \cdot T - 1^{\circ}.3972 \cdot T \\
l = 134^{\circ}.96292 + 477198^{\circ}.86753 \cdot T \\
l' = 357^{\circ}.52543 + 35999^{\circ}.04944 \cdot T \\
F = 93^{\circ}.27283 + 483202^{\circ}.01873 \cdot T \\
D = 297^{\circ}.85027 + 445267^{\circ}.11135 \cdot T
\end{cases}
$$

月球简化历表

2000年黄道和春分点的月球黄经

$$\lambda_M = L_0 + 22640'' \cdot \sin(l) + 729'' \sin(2l)$$
$$-4589'' \cdot \sin(l-2D) + 2370'' \cdot \sin(2D)$$
$$-668'' \cdot \sin(l') - 412'' \cdot \sin(2F)$$
$$-212'' \cdot \sin(2l-2D) - 206'' \cdot \sin(l+l'-2D)$$
$$+192'' \cdot \sin(l+2D) - 165'' \cdot \sin(l'-2D)$$
$$+148'' \cdot \sin(l-l') - 125'' \cdot \sin(D)$$
$$-110'' \cdot \sin(l+l') - 55'' \cdot \sin(2F-2D)$$

月球纬度

$$\beta_M = 18520'' \cdot \sin(F + \lambda - L_0 + 412'' \cdot \sin 2F + 541'' \cdot \sin l')$$
$$-526'' \cdot \sin(F-2D) + 44'' \cdot \sin(l+F-2D)$$
$$-31'' \cdot \sin(l'+F-2D) - 25'' \cdot \sin(-2l+F)$$
$$+23'' \cdot \sin(l'+F-2D) + 21'' \cdot \sin(-l+F)$$
$$+11'' \cdot \sin(-l'+F-2D)$$

月球简化历表

月球的地心距

$$
\begin{aligned}
r_M = &(35800 - 20905\cos(l) - 3699\cos(2D - l) \\
&-2956\cos(2D) - 570\cos(2l) + 246\cos(2l - 2D) \\
&-205\cos(l' - 2D) - 171\cos(l + 2D) \\
&-152\cos(l + l' - 2D))
\end{aligned}
$$

黄道球坐标转化为赤道笛卡尔直角坐标

$$
\vec{r}_m = R_x(-\varepsilon)\begin{pmatrix} \mathrm{r}_M \cos\lambda_M \cos\beta_M \\ \mathrm{r}_M \sin\lambda_M \cos\beta_M \\ \mathrm{r}_M \sin\beta_M \end{pmatrix}
$$

其他大大行星简化公式

$$a = a_0 + \dot{a}\, t \quad \text{AU},$$

$$e = e_0 + \dot{e}\, t,$$

$$I = I_0 + (\dot{I}/3600)\, t \quad \text{degrees},$$

$$\varpi = \varpi_0 + (\dot{\varpi}/3600)\, t \quad \text{degrees},$$

$$\Omega = \Omega_0 + (\dot{\Omega}/3600)\, t \quad \text{degrees},$$

$$\lambda = \lambda_0 + (\dot{\lambda}/3600 + 360 N_r)\, t \quad \text{degrees},$$

The following tables give the orbital elements of the planets and their variations at the epoch of J2000 (JD 2451545.0) with espect to the mean ecliptic and equinox of J2000

其他大行星简化公式

Planet	a_0 (AU)	e_0	I_0 (°)	ϖ_0 (°)	Ω_0 (°)	λ_0 (°)
Mercury	0.38709893	0.20563069	7.00487	77.45645	48.33167	252.25084
Venus	0.72333199	0.00677323	3.39471	131.53298	76.68069	181.97973
Earth	1.00000011	0.01671022	0.00005	102.94719	348.73936	100.46435
Mars	1.52366231	0.09341233	1.85061	336.04084	49.57854	355.45332
Jupiter	5.20336301	0.04839266	1.30530	14.75385	100.55615	34.40438
Saturn	9.53707032	0.05415060	2.48446	92.43194	113.71504	49.94432
Uranus	19.19126393	0.04716771	0.76986	170.96424	74.22988	313.23218
Neptune	30.06896348	0.00858587	1.76917	44.97135	131.72169	304.88003
Pluto	39.48168677	0.24880766	17.14175	224.06676	110.30347	238.92881

其他大行星简化公式

Planet	$\dot{a}_0$	$\dot{e}_0$	$\dot{I}_0$	$\dot{\varpi}_0$	$\dot{\Omega}_0$	$\dot{\lambda}_0$	N_r
Mercury	66	2527	−23.51	573.57	−446.30	261628.29	415
Venus	92	−4938	−2.86	−108.80	−996.89	712136.06	162
Earth	−5	−3804	−46.94	1198.28	−18228.25	1293740.63	99
Mars	−7221	11902	−25.47	1560.78	−1020.19	217103.78	53
Jupiter	60737	−12880	−4.15	839.93	1217.17	557078.35	8
Saturn	−301530	−36762	6.11	−1948.89	−1591.05	513052.95	3
Uranus	152025	−19150	−2.09	1312.56	1681.40	246547.79	1
Neptune	−125196	2514	−3.64	−844.43	−151.25	786449.21	0
Pluto	−76912	6465	11.07	−132.25	−37.33	522747.90	0


Q&A!