



中国研究生创新实践系列大赛  
“华为杯”第二十一届中国研究生  
数学建模竞赛

学 校

武汉大学

---

参赛队号

24104860134

---

队员姓名

1. 罗宏昆
  2. 杨凯淳
  3. 郑丹
-

# 中国研究生创新实践系列大赛

# “华为杯”第二十一届中国研究生

# 数学建模竞赛

题 目： X 射线脉冲星光子到达时间建模与仿真

---

## 摘 要：

脉冲星是宇宙中的中子星，在旋转的过程中。它不断地向外辐射出具有周期性的光子。脉冲光子是连续的且极其稳定。这能够让宇宙中的卫星或者其他天体实现高精度定位与授时。这对航空航天以及深空探测具有重要的意义。

**针对问题一：**首先根据轨道六根数的几何关系进行位置建模，由此可以准确计算出卫星位置。卫星的精确位置向量为 **(1274.91341509km, -1848.85196499km, 6507.26265528km)**。根据卫星的速度是位置矢量的导数关系，进而能够求解出卫星速度向量为 **(-6.21732353km/s, 3.75137234km/s, 2.2702798km/s)**。在验证计算的准确性上，主要利用牛顿万有引力定律与角动量守恒验证计算结果的合理性。

**针对问题二：**首先需要通过给定的附件 de200.bsp 获取地球质心在太阳系质量坐标系下的坐标，然后根据第一问的卫星位置求解结果与坐标系转换模型得到卫星在太阳系质心坐标系下的坐标。由于脉冲星光子是平行的，计算卫星与太阳系质心距离在平行光上的投影来构建时间差模型。最后求解得到的时间差为 **490.5883(s)**。

**针对问题三：**主要考虑 Roemer 延迟，Einstein 延迟，Shapiro 延迟三个部分。Roemer 延迟考量几何传播时间延迟与脉冲星自行，延迟时间为 **-2.605e-5(s)**，Einstein 延迟包括狭义相对论的延迟，延迟时间为 **-4.054(s)**。Shapiro 延迟是根据光子传播路径进行建模，延迟时间为 **-479.525(s)**。总延迟时间为 **-483.578(s)**。

**针对问题四：**在模拟光子序列方面，考虑将 10s 时间微分成 10000 份，时间间隔 0.001s。首先通过标准轮廓值，建立分段混合高斯模型拟合标准轮廓曲线。将混合高斯模型泰特展开用于近似求解光子到达速率。进而模拟出光子到达的累计函数。根据泊松分布的数学期望仿真光子每一个时刻到达的数目。仿真结果如图7.42和图7.43。在模拟脉冲轮廓方面，采用时间微分的思想仿真光子速率。光子速率进行分段折叠可以仿真得到脉冲轮廓。脉冲轮廓仿真结果如图图7.11。在提高仿真精度方面，考虑将混合高斯模型进行二阶展开，这样精准地仿真出光子速率，光子累计函数，脉冲轮廓。精细化仿真结果如图7.13。

关键词：脉冲星 时延模型 光子序列仿真 混合高斯模型

## 目录

<b>1 问题背景与重述</b>	<b>4</b>
1.1 问题背景	4
1.2 问题重述	5
<b>2 模型的假设</b>	<b>6</b>
<b>3 符号说明</b>	<b>6</b>
<b>4 卫星位置与速度建模</b>	<b>7</b>
4.1 问题一分析	7
4.2 轨道根数-卫星定位模型的建立	8
4.3 卫星位置速度求解	10
<b>5 脉冲星光子传播时间差建模与求解</b>	<b>11</b>
5.1 问题二分析	11
5.2 建立卫星-太阳系质心光子传输时差模型	12
5.3 光子传输时差求解	14
<b>6 脉冲传输时延建模与求解</b>	<b>15</b>
6.1 问题三分析	15
6.2 脉冲时间延迟模型的建立	16
6.2.1 爱因斯坦 (Einstein) 延迟 [1]	17
6.2.2 罗默尔 (Roemer) 延迟	19
6.2.3 夏皮罗 (Shapiro) 延迟	20
6.3 脉冲传输时延精确求解	21
<b>7 光子到达时刻模型仿真与求解</b>	<b>23</b>
7.1 问题四整体分析	23
7.2 问题四逐步求解	23
7.2.1 混合高斯模型仿真光子序列	23
7.2.2 仿真算法流程	26
7.2.3 光子序列仿真结果	27
7.2.4 脉冲轮廓折叠仿真	27

7.2.5 脉冲轮廓还原模型 . . . . .	28
7.2.6 脉冲轮廓仿真结果与误差 . . . . .	28
7.2.7 仿真精度优化分析 . . . . .	29
7.2.8 精细化仿真模型 . . . . .	29
7.2.9 精细化仿真结果 . . . . .	30
<b>参考文献 . . . . .</b>	<b>30</b>
<b>附录 A MATLAB 源程序 . . . . .</b>	<b>32</b>
A.1 第 4 问程序 . . . . .	32
<b>附录 B Python 源程序 . . . . .</b>	<b>38</b>
B.1 第 1 问程序 . . . . .	38
B.2 第 2 问程序 . . . . .	40
B.3 第 3 问程序 . . . . .	41

# 1 问题背景与重述

## 1.1 问题背景

脉冲星是一种快速自旋的中子星 [2, 3]，如图1.1在它的磁极处产生电磁辐射束，并且脉冲星周期性地向外辐射各频段的电磁脉冲信号。脉冲星的发现是 20 世纪 60 年代天文学领域的重大突破。研究表明，脉冲星是大质量恒星演化、坍缩和超新星爆发后的残骸。这些天体拥有极端的物理条件特征，包括超高温度、超高压力、超高密度、超强磁场以及超强电场等。脉冲星最显著的特点是其极为稳定的自转周期，这种精确性使得脉冲星被誉为自然界中最精准的“天文钟”。得益于这一特性，脉冲星在多个重要的科学的研究和工程应用领域，如脉冲星深空导航、时间频率计量、引力波探测等展现了巨大的价值。

利用脉冲星的 X 射线频段辐射信号可实现近地及深空中航天器进行自主导航定位。X 射线脉冲星导航系统（XPNAV）通过脉冲星信号的到达时间差实现航天器的定位和导航，可以为大气层外航天器定位导航，与 GPS 为中低轨卫星导航的情况基本相同，光子到达时间的转换是利用 X 射线脉冲星进行导航的基础，其转换精度决定了导航精度。2016 年我国发射了 XPNAV-1 卫星并且成功观测了蟹状星云中的 Crab 脉冲星。该卫星的成功发射验证了脉冲星导航技术在深空探测中的可行性。脉冲星信号在传播过程中会受到多种因素的影响，例如真空环境下产生的 Roemer 时延、引力场引起的 Shapiro 时延、在强引力场传播中出现的引力红移、狭义相对论动钟变慢效应等，同时脉冲星在空间中的位置也会随时间变化，上述因素对脉冲信号到达时间的影响不可忽略。

由于地球大气会隔绝 X 射线，导致在地面上无法直接观测到 X 射线脉冲星的信号，对 X 射线脉冲星的观测必须在大气层之外的地球卫星或空间站上进行。然而在现阶段空间搭载实验成本高昂、技术难度较大。因此，在进行空间搭载实验之前，需要建立一个相对来说成本低、操作简单且信号处理辐射与实际情况相匹配的脉冲星仿真模型，在实测数据匮乏的情况下，通过脉冲星信号仿真也能进行深入研究。

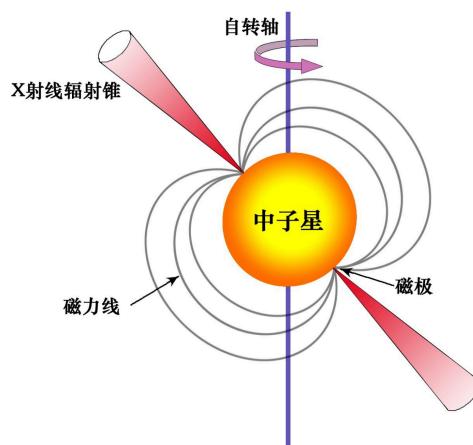


图 1.1 脉冲星信号辐射模型

## 1.2 问题重述

基于上述研究背景，本文需研究和解决一下问题：

**问题一：**问题 1 要求建立一个用以描述卫星的轨道根数与其在地心天球参考系 (Geocentric Celestial Reference System, GCRS) 中的三维位置 ( $X, Y, Z$ ) 和速度 ( $v_x, v_y, v_z$ ) 之间关系的数学模型。同时问题 1 提供 XPNAV-1 卫星在某一时刻的轨道根数，包括偏心率、角动量、轨道倾角、升交点赤经、近地点幅角和真近点角，根据这些给定的轨道参数计算该时刻卫星在 GCRS 中的三维空间位置坐标和速度。最后还要求对轨道参数的一致性和计算结果进行验证，从而为后续的脉冲星信号时延计算提供可靠的卫星位置数据。

**问题二：**问题 2 要求建立用于计算 X 射线光子到达卫星和太阳系质心之间的真空几何传播时延的数学模型。问题 2 中假设了脉冲星辐射的 X 射线光子信号传播形式为平行光，并且忽略太阳系天体自转以及扁率对光子信号的影响。问题 2 中给定光子到达卫星的时刻为 57062.0 约化儒略日 (MJD，基于地球自转的 TT 时间尺度)，需要根据问题 1 建立的模型，计算卫星在地心天球参考系转换到太阳系质心坐标系下的三维位置坐标，再计算脉冲星光子分别到达卫星和太阳系质心之间的传播路径时间差，从而得到真空几何传播时延。

**问题三：**问题 3 相比于问题 2 来说，需要考虑多种关键时延因素，建立一个更加精确的传播时延模型。需要考虑以下因素：理想真空环境中信号传播产生的几何传播时延，经过强引力场时弯曲路径造成的 Shapiro 时延、广义相对论中描述的引力红移时延和狭义相对论提到的动钟变慢效应，以及脉冲星自行（即相对于其他天体，脉冲星在天空中会随时间发生变位置变化）。在本问题中还给定了光子到达探测器的时刻为 58119.1651507519MJD，要求根据问题 1 中卫星的位置速度、附件 3 提供的太阳系天体位置信息 DE 系列历表和附件 4 给出的脉冲星位置参考历元和自行参数等信息，计算出脉冲星光子到达航天器与到达太阳系质心之间的精确时间差。

**问题四：**问题 4 要求：

1) 建立一个 X 射线脉冲星光子序列的仿真模型，并基于此模型对 Crab 脉冲星的光子序列进行仿真。同时本小问给定：仿真时间为 10 秒，背景光子流量密度为  $\lambda_b = 1.54 \text{ ph}/(\text{s} \cdot \text{cm}^2)$ ，Crab 脉冲星的光子流量密度  $\lambda_s = 10\lambda_b$ ，即背景光子流量的 10 倍。探测器的有效面积为 250 平方厘米

2) 根据附件 1, Crab 脉冲星的自转参数进行建模，生成相应的光子序列，并通过周期折叠技术，将观测数据折叠以得到脉冲轮廓，确保脉冲轮廓在相位区间  $[0, 1]$

3) 保持  $\lambda_b$  不变，为了更清晰地展现脉冲星的辐射特性，需要提出一种改进方法优化仿真精度，提高模型仿真的拟合度。

## 2 模型的假设

1. 假设干扰脉冲星信号传播过程的因素只有，Roemer 时延、Shapiro 时延、“引力红移”效应、“动钟变慢”效应、脉冲星自行，其他时延不考虑。
2. 假设脉冲星辐射的 X 射线光子信号传播形式为平行光且平行光方向与卫星到太阳系的连线共线；
3. 忽略太阳系天体自转以及扁率对脉冲星辐射的 X 射线光子信号的影响；
4. 假设探测器探测到的光子仅由背景光子和脉冲星光子两部分组成；
5. 针对问题三假设脉冲星为中心光源且单位方向向量为  $(1, 1, 1) / \sqrt{3}$ ；

## 3 符号说明

符号	意义
$e$	偏心率
$h$	角动量
$\Omega$	升交点赤经
$i$	轨道倾角
$\omega$	近地点幅角
$\theta$	真近点角
$p$	半通径
$\mu$	中心天体引力常数
$G$	万有引力常数 $(6.67430 \pm 0.00015) \times 10^{-11} \text{m}^3 \text{kg}^{-1} \text{s}^{-2}$
$d_{\parallel}$	投影长度
$\hat{n}$	光线方向的单位向量
$c$	光速
$\Delta_{\odot}$	光子到达时刻转换到 SSB 产生的总延迟量
$\Delta_{E\odot}$	爱因斯坦 (Einstein) 延迟 (包括引力红移效应和狭义相对论效应，对应题目中的引力红移时延和动钟变慢效应)
$\Delta_{R\odot}$	罗默尔 (Roemer) 延迟
$\Delta_{S\odot}$	夏皮罗 (Shapiro) 延迟
$h(t_i)$	$t_i$ 时刻脉冲轮廓归一化的仿真值

## 4 卫星位置与速度建模

### 4.1 问题一分析

问题1要求根据给定的轨道六根数,建立卫星轨道根数与其在地心天球参考系(GCRS)中的三维位置和速度关系的数学模型,并计算具体时刻卫星在GCRS的三维位置和速度。对于本问,可以先根据已知的轨道根数,计算出卫星轨道半径,然后确定卫星在轨道平面中的位置,进一步将轨道平面位置转换为三维位置,最后求出卫星速度。本问主要有以下难点:

**难点一: 轨道根数与三维位置速度之间的转换:** 将轨道六根数(偏心率 $e$ 、轨道倾角 $i$ 、升交点赤经 $\Omega$ 、近地点幅角 $\omega$ 、真近点角 $\theta$ 、以及角动量 $h$ )转换为三维位置和速度矢量,是一个复杂的数学过程。轨道根数描述的是轨道几何和相对位置,而三维位置和速度是具体的矢量,需要通过一系列几何变换和旋转矩阵来完成。这种转换涉及多个坐标系之间的映射。

**难点二: 验证参数一致性和结果的正确性:** 完成位置和速度的计算后,为了保证轨道根数一致性以及据其推算位置速度的结果准确性,需要进行验证。轨道参数单位统一是计算前提,例如,角动量可能以 $km^2/s$ 作为单位,而轨道倾角单位可能是度。其次轨道参数必须满足物理条件约束,对于椭圆轨道中的偏心率 $e$ 应满足 $0 \leq e < 1$ ; 轨道倾角通常为 $0^\circ \leq i \leq 180^\circ$ ; 真近点角的范围为 $0^\circ \leq \theta \leq 360^\circ$ ,已知条件中的这些参数经过物理合理性检查后,均在合理范围内。轨道力学中角动量是天体围绕中心天体运动的守恒量,也存在物理约束条件。

最后,通过轨道根数求解的位置和速度矢量,必须符合万有引力定律和角动量守恒定律。通过给定数值计算结果,如何验证轨道参数一致性和结果的正确性需要特别关注。

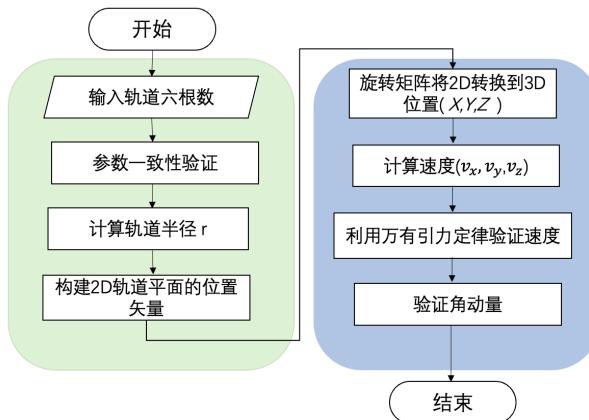


图 4.2 问题一分析流程图

## 4.2 轨道根数-卫星定位模型的建立

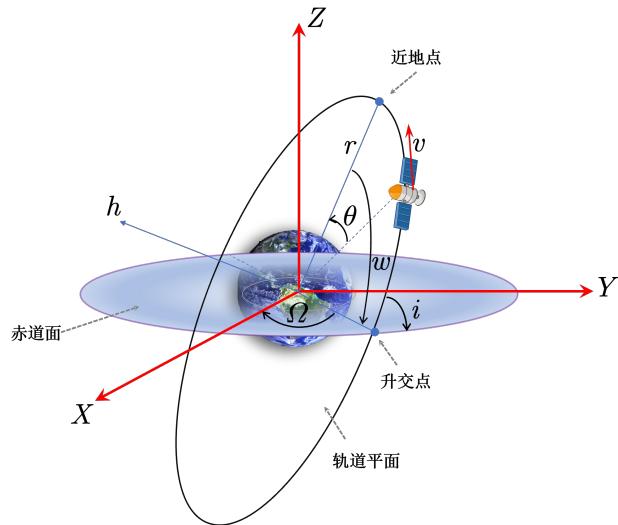


图 4.3 轨道六根数示意图

针对难点一，轨道根数要变换为地心天体惯性系，经过三次方向余弦矩阵变换即可。第一次旋转目的是让坐标系的 x 轴与升交线对齐，轨道平面需要绕参考坐标系的 z 轴旋转  $-\Omega$ ，第二次旋转保证轨道平面正法向与 z 轴重合，轨道平面绕 x 轴旋转  $-i$ ，第三次旋转将离心率矢量与 x 轴对齐，绕 z 轴旋转  $-\omega$ 。在这种变化下，物体的坐标可以表示为  $[r\cos\theta, r\sin\theta, 0]'$ 。

轨道力学中，角动量  $\mathbf{h}$  与半通径  $\mathbf{p}$  的关系公式表示为：

$$p = \frac{h^2}{\mu(1 - e^2)} \quad (4.1)$$

通过一系列旋转矩阵将轨道平面中的二维位置转换到惯性参考系中的三维位置矢量，位置矢量的计算公式：

$$\hat{r} = R_3(-\Omega)R_1(-i)R_3(-\omega) \begin{bmatrix} r \cos \theta \\ r \sin \theta \\ 0 \end{bmatrix} \quad (4.2)$$

其中，轨道半径  $r$  公式为：

$$r = \frac{p}{1 + e \cos \theta} \quad (4.3)$$

完整的展开形式为：

$$\hat{r} = \frac{p}{1 + e \cos \theta} \begin{bmatrix} \cos \Omega \cos(\omega + \theta) - \sin \Omega \sin(\omega + \theta) \cos i \\ \sin \Omega \cos(\omega + \theta) + \cos \Omega \sin(\omega + \theta) \cos i \\ \sin(\omega + \theta) \sin i \end{bmatrix} \quad (4.4)$$

速度矢量的公式为：

$$\hat{v} = \frac{d\hat{r}}{d\theta} \frac{d\theta}{dt} \quad (4.5)$$

其中，真近点角的微分为：

$$\frac{d\theta}{dt} = \frac{\sqrt{\mu}}{p^{3/2}} (1 + e \cos \theta)^2 \quad (4.6)$$

其中  $\mu$  是中心天体引力常数

最后得到速度公式：

$$\hat{v} = \sqrt{\frac{\mu}{p}} \begin{bmatrix} -\cos \Omega (\sin(\omega + \theta) + e \sin \omega) - \sin \Omega (\cos(\omega + \theta) + e \cos \omega) \cos i \\ -\sin \Omega (\sin(\omega + \theta) + e \sin \omega) + \cos \Omega (\cos(\omega + \theta) + e \cos \omega) \cos i \\ (\cos(\omega + \theta) + e \cos \omega) \sin i \end{bmatrix} \quad (4.7)$$

针对难点二，首先对轨道速度进行验证。万有引力定律在轨道验证中是关键的物理基础，它有助于确保卫星的速度和角动量等物理量与轨道参数相符。通过该定律，可以检验计算结果是否符合轨道力学的基本法则，进而保证轨道六根数的计算和结果保持一致性。万有引力定律表明，任意两个质点之间在连心线上存在相互吸引的力。这种引力的大小与它们质量的乘积成正比，并与它们之间距离的平方成反比，同时不受两物体的化学成分或它们之间介质类型的影响。

万有引力定律表示为：

$$F = G \frac{m_1 m_2}{r^2} \quad (4.8)$$

其中：

$F$  是引力大小

$G$  是引力常数

$r$  是两物体中心之间的距离

在轨道力学中，物体的速度与其轨道性质密切相关。通过万有引力定律，可以进一步推导出轨道速度的表达式。例如，对于绕地球做圆周运动的卫星， $v$  可由以下公式计算：

$$v = \sqrt{\frac{GM}{r}} \quad (4.9)$$

其中：

$M$  是地球的质量

$r$  是卫星到地球中心的距离

通过观测卫星的轨道半径  $r$  和已知地球质量  $M$ , 可以计算出卫星的理论轨道速度  $v$ 。实际观测到的速度与理论值的一致性验证了万有引力定律的正确性。

其次是对角动量的验证。角动量  $h$  是描述物体旋转状态的重要物理量, 对于绕中心天体运动的天体, 其角动量可以表示为:

$$h = \mathbf{r} \times \mathbf{v} \quad (4.10)$$

在平面运动中, 角动量的大小为:

$$h = rv_{\perp} \quad (4.11)$$

根据万有引力定律和守恒定律, 在没有外力矩作用下, 系统的总角动量保持不变。因此我们可以通过计算角动量从而对结果进行验证

### 4.3 卫星位置速度求解

表 4.1 卫星在 GCRS 中三维位置计算结果

	X	Y	Z
位置坐标 (km)	1274.91341509	-1848.85196499	6507.26265528

表 4.2 卫星在 GCRS 中速度计算结果

	$v_x$	$v_y$	$v_z$
速度分量 (km/s)	-6.21732353	3.75137234	2.2702798
总速度 (km/s):	7.608027120380198		

计算得到的总速度 ( $km/s$ ): 7.608027120380198, 万有引力定律验证的轨道速度 ( $km/s$ ): 7.615831511634537。

速度差异为:

$$\Delta v = v_{\text{已知}} - v_{\text{计算}} = 7.615831511634537 - 7.608027120380198 = 0.007804391254339 \text{ km/s}$$

#### 计算速度相对误差

$$\text{相对误差} = \frac{\Delta v}{v_{\text{已知}}} = \frac{0.007804391254339}{7.615831511634537} \approx 0.001024626576034 \quad (\text{即 } 0.102\% \text{ 的误差})$$

误差非常小，表明我们通过计算和万有引力定律验证的速度结果具有高度一致性，计算结果相对准确

计算得到的角动量 ( $km^2/s$ ): 52269.89607221833 与已知条件给定的角动量 52330.8462

**计算角动量差异为**

$$\Delta h = h_{\text{理论}} - h_{\text{计算}} = 52330.8462 - 52269.89607221833 = 60.95012778167 \text{ km}^2/\text{s}$$

角动量相对误差为：

$$\text{相对误差} = \frac{\Delta h}{h_{\text{已知}}} = \frac{60.95012778167}{52330.8462} \approx 0.001164846621837 \quad (\text{即 } 0.116\% \text{ 的误差})$$

这种误差也非常小，表明角动量计算与已知条件给定的角动量高度一致。

综上，速度和角动量的计算结果与已知条件非常接近，可以认为我们的计算结果准确性较高

## 5 脉冲星光子传播时间差建模与求解

### 5.1 问题二分析

问题 2 需要我们建立一个真空几何传播时延模型，测量脉冲星光子到达卫星与太阳系质心之间传播时延差值。通过对脉冲信号到达两者时间，可以实现航天器的定位和导航。为了保证在同一坐标系中讨论，首先根据问题 1 中给出的卫星在地心天球参考系 (GCRS) 中的位置坐标，进一步将其转换为卫星在太阳系质心坐标系中的位置。题目已经假设脉冲星光子信号传播方式为平行传播，并且在本小问中不考虑太阳系中天体自转和扁率造成的影响。有关卫星、地球、太阳系质心、脉冲星光子方向的几何关系如图 5.4

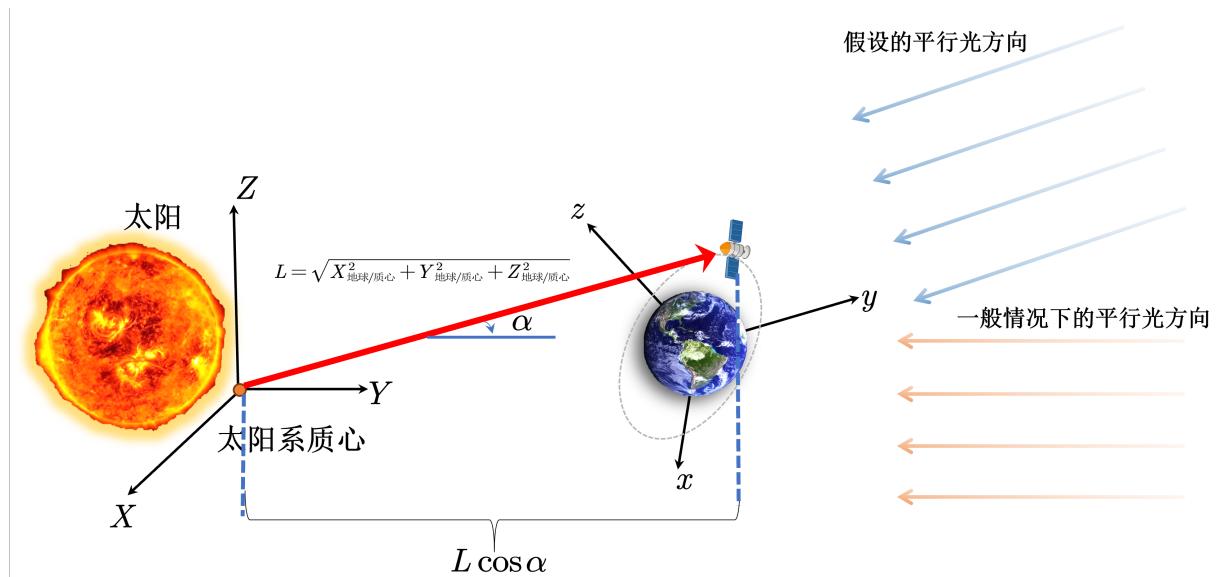


图 5.4 卫星位置坐标转换示意图

根据坐标原点的不同，ICRS 可以分为两类：太阳系质心天球参考系（BCRS）和地心天球参考系（GCRS）。BCRS 以太阳系质心坐标原点，主要用于行星轨道计算和星表编制；而 GCRS 以地心作为坐标原点的天文学坐标系，主要用于计算卫星轨道和编制卫星星历。为了得到卫星在太阳质心坐标系中的位置，我们需要先从 JPL 星历文件中读取地球在太阳质心坐标系中以太阳为中心的三维向量。卫星相对于地球的位置，加上地球相对于太阳质心的位置，即可得到参考系变换后的位置坐标。

脉冲星发出的光子在传播到卫星和太阳系质心的过程中，距离差异导致光子到达时间不同，从而产生时间延迟，计算过程中同时也要注意到光线方向与卫星位置夹角带来的影响。具体解决流程图如图5.5。

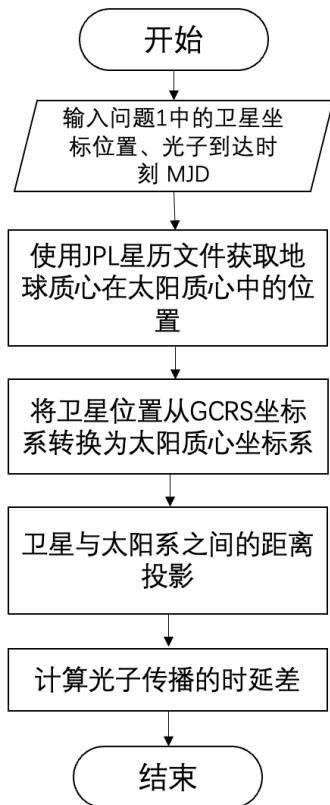


图 5.5 问题 2 思路流程图

## 5.2 建立卫星-太阳系质心光子传输时差模型

为了计算光子传播时延，首先需要将卫星的位置从 GCRS 转换到 BCRS，卫星在太阳质心坐标系中的位置等于它在 GCRS 中的位置加上地球质心在太阳质心坐标系中的位置。

卫星在太阳质心坐标系中的位置与在 GCRS 中的位置之间的关系可以用平移变换表示为：

$$\mathbf{r}_{\text{卫星}/\text{SSB}} = \mathbf{r}_{\text{卫星}/\text{GCRS}} + \mathbf{r}_{\text{地心}/\text{SSB}} \quad (5.12)$$

其中：

$\mathbf{r}_{\text{卫星}/\text{GCRS}}$ ：卫星在 GCRS 中的位置矢量问题 1 中的结果)

$\mathbf{r}_{\text{地心}/\text{SSB}}$ ：地心在太阳质心坐标系中的位置矢量（由 JPL 星历文件中获得）

$\mathbf{r}_{\text{卫星}/\text{SSB}}$ ：卫星在太阳质心坐标系中的位置矢量

这个公式表示，物体在太阳质心坐标系中的位置等于它在地心天体参考系中的位置加上地球在太阳质心坐标系中的位置。

如果将上述矢量展开为三个分量（在  $x$ 、 $y$ 、 $z$  方向），则可以表示为：

$$\begin{aligned}x_{\text{卫星}/\text{SSB}} &= x_{\text{卫星}/\text{GCRS}} + x_{\text{地心}/\text{SSB}} \\y_{\text{卫星}/\text{SSB}} &= y_{\text{卫星}/\text{GCRS}} + y_{\text{地心}/\text{SSB}} \\z_{\text{卫星}/\text{SSB}} &= z_{\text{卫星}/\text{GCRS}} + z_{\text{地心}/\text{SSB}}\end{aligned}\quad (5.13)$$

即在每个坐标轴上的坐标都需要加上地球在太阳质心坐标系中的相应分量。

光子在理想真空环境下光速传播，因此传播时间差  $\Delta t$  为光子从脉冲星分别到达卫星和太阳质心的距离除以光速的差值，同时值得注意的是，即使已经假设脉冲星辐射光子信号平行光传播，平行光和卫星之间的夹角仍然不可忽略，特别是在高精度定位和导航应用中，需要考虑减少因位置方向差异带来的误差，使得结果更精确。根据夹角可以计算出位置向量在光线方向上的投影长度：

设卫星-质心-脉冲星夹角为  $\alpha$

$$\cos \alpha = \frac{\mathbf{r}_{\text{卫星}} \cdot \hat{\mathbf{n}}}{\|\mathbf{r}_{\text{卫星}}\|} \quad (5.14)$$

计算投影长度  $d_{\parallel}$

$$d_{\parallel} = \mathbf{r}_{\text{卫星}} \cdot \hat{\mathbf{n}} = \|\mathbf{r}_{\text{卫星}}\| \cos \alpha \quad (5.15)$$

因此可得传播时间差  $\Delta t$  公式：

$$\Delta t = \frac{d_{\parallel}}{c} = \frac{\mathbf{r}_{\text{卫星}} \cdot \hat{\mathbf{n}}}{c} = \frac{\|\mathbf{r}_{\text{卫星}}\| \cos \alpha}{c} \quad (5.16)$$

其中：

$\hat{\mathbf{n}}$ ：光线方向的单位向量。

$\mathbf{r}_{\text{卫星}} = (X, Y, Z)$ ：卫星在太阳系质心坐标系中的位置向量

$\|\cdot\|$ ：向量的欧几里得范数，表示向量的长度或两点之间的距离。

$c$ ：光速

特别地，

当夹角  $\alpha$  为 0 时：

$$d_{\parallel} = \mathbf{r}_{\text{卫星}} \cdot \hat{\mathbf{n}} = \|\mathbf{r}_{\text{卫星}}\| = \sqrt{X^2 + Y^2 + Z^2} \quad (5.17)$$

$$\Delta t = \frac{d_{\parallel}}{c} = \frac{\mathbf{r}_{\text{卫星}} \cdot \hat{\mathbf{n}}}{c} = \frac{\|\mathbf{r}_{\text{卫星}}\|}{c} \quad (5.18)$$

通过计算投影长度  $d_{\parallel}$ , 我们能够量化卫星在光线方向上的位置, 得到更精确的传播时延。

### 5.3 光子传输时差求解

将问题 1 中计算得到的卫星在 GCRS 中的位置转换到 BCRS, 太阳系质心的坐标可以从本题中所提供的历表 (de200.bsp) 中获得, de200.bsp 记录了太阳系天体的位置数据, 根据表格5.3可以知道地球质心代码为 3, 太阳系质心代码为 0. 这个是获取地球质心时需要特别注意的地方。通过“打开附件 3(de200.bsp)”的 python 包 jplephem.spk 读取附件 3-de200, 代入上述公式, 计算脉冲星光子到达太阳系质心和到达卫星的传播时间差

表 5.3 附件 3-de200 数据展示

序号	行星质心	太阳系质心类别	日期范围
0	太阳系质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
1	水星质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
2	金星质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
3	地球质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
4	火星质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
5	木星质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
6	土星质心	第 2 类太阳系质心	1899-07-29 至 2053-10-09
10	太阳	第 2 类太阳系质心	1899-07-29 至 2053-10-09

在计算传播时间差的时候, 由于脉冲星光子与卫星到太阳系质量心的距离夹角未知。因此为简化问题, 假设这两个方向向量共线。因此, 卫星到太阳系质心的投影距离就是两个坐标的欧式距离, 同样也是卫星坐标向量的模长。最后转化坐标系后的卫星坐标如5.4。

表 5.4 问题 2 卫星从 GCRS 转换到 BCRS 三维位置结果

	X	Y	Z
GCRS 坐标 (km)	1274.91341509	-1848.85196499	6507.26265528
BCRS 坐标 (km)	-1.12094190e+08	8.75121480e+07	3.79151058e+07

为了验证计算结果的准确性, 我们还验证了卫星坐标转换后坐标的准确性。具体体现在求得的卫星与太阳系质心的距离与实际的卫星到太阳质心的距离差别很小。这是由于太

阳质量很大，太阳系质心在太阳表面，因此求解结果相近更加验证了计算结果的准确性。本题计算出的地球质心距离太阳系质心的距离(km)为147176896.18694443。实际上地球到太阳的距离约为149597870.7。计算误差大约为1.3%。脉冲星光子到达卫星与太阳系质心的传播路径时间延迟(s)为490.58834124534496。

## 6 脉冲传输时延建模与求解

### 6.1 问题三分析

如果忽略所有相对论效应，脉冲信号的传播延时仅为纯粹的几何延时，即传播距离除以真空光速。然而，在大尺度空间中的信号传播需要考虑相对论性[1]的修正。因此问题3的相比于问题2，对延时模型要求更严格，以便更准确地描述脉冲信号的传播过程。问题3中提到的需要考虑的因素有几何传播时延、Shapiro时延、引力红移效应、狭义相对论动钟变慢效应，以及脉冲星自行等因素对传播时间。在计算时，可以使用DE系列历表提供的太阳系天体位置信息进行高精度计算，最终输出时延差。天体相关的几何关系示意图如图6.6。

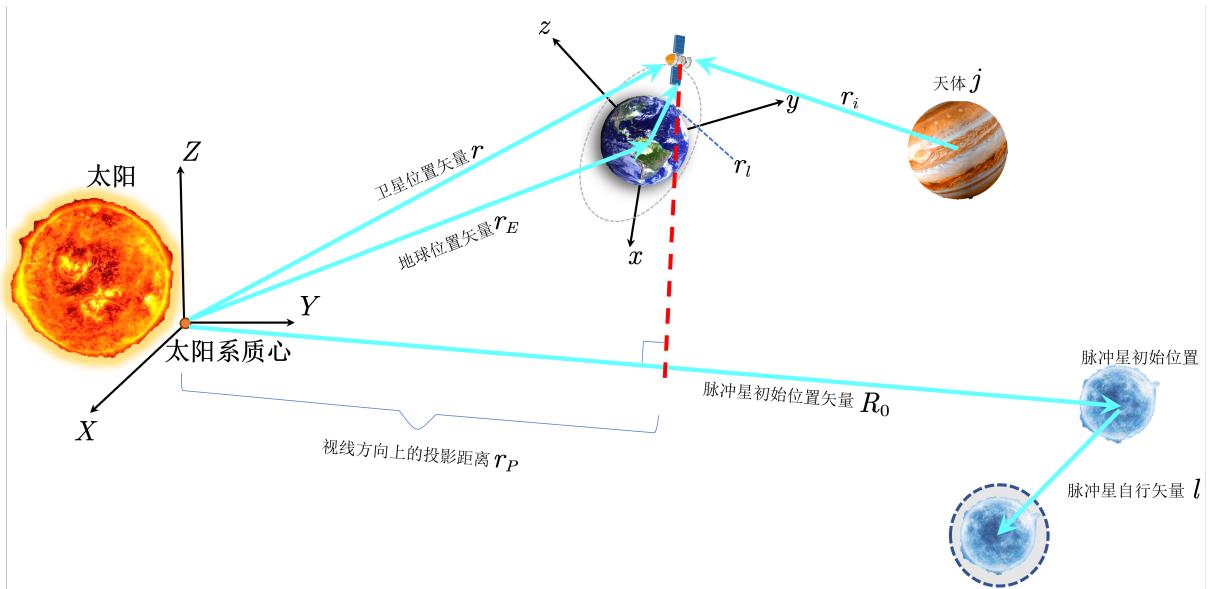


图 6.6 卫星与太阳系天体和脉冲星的几何关系示意

延时计算与光子时间校正是至关重要的步骤。由于航天器和地球的运动以及其他相对论效应的影响，探测器记录的光子到达时间会掩盖原有的周期性。因此，必须对这些因素进行修正，以恢复信号的周期特性，最终实现光子历元的折叠。

## 6.2 脉冲时间延迟模型的建立

我们希望通过建立一个时延模型 [4], 来精确计算脉冲星光子到达航天器与太阳系质心之间的时延。该模型的目的是用于校正观测数据, 从而获得精确的脉冲星到达时间 (TOA)。在此问题中, 已知观测时刻为 MJD 58119.1651507519, 我们将使用这一时刻结合卫星位置、速度、脉冲星位置与自行参数来计算这些时延项。时间延迟计算模型示意图如图6.7。

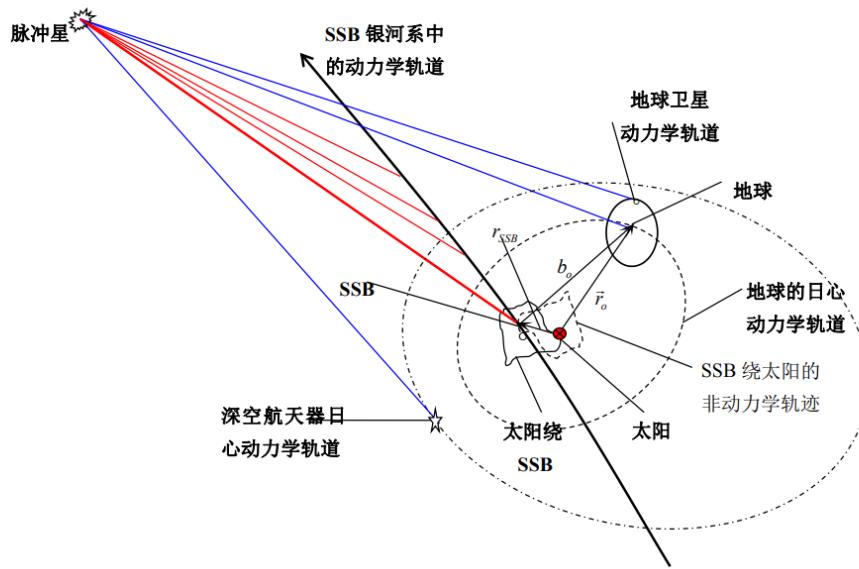


图 6.7 脉冲星信号到达太阳系质心及地球轨道示意图

观测单个脉冲星时, 问题 2 中获取的在太阳质心坐标系下的位置速度以及问题 1 中获取的地心坐标系下的位置, 可以进一步修正和计算光子到达时刻的延迟, 光子到达时间进行转换计算公式为:

$$\Delta_{\odot} = \Delta_{E\odot} + \Delta_{R\odot} + \Delta_{S\odot} \quad (6.19)$$

其中:

$\Delta_{\odot}$  表示光子到达时刻转换到 SSB 产生的总延迟量;

$\Delta_{E\odot}$  表示爱因斯坦 (Einstein) 延迟, 反映时间尺度变换所导致的延迟;

$\Delta_{R\odot}$  表示罗默尔 (Roemer) 延迟, 代表光子从脉冲星到达地球所需的时间延迟, 由观测位置和光速的影响决定

$\Delta_{S\odot}$  表示夏皮罗 (Shapiro) 延迟, 因时空弯曲引起的延迟, 暂不考虑除太阳外的一阶以上部分;

### 6.2.1 爱因斯坦 (Einstein) 延迟 [1]

$$\Delta_{E\odot} = \Delta_{\text{TAI-UTC}} + \Delta_{\text{TT-TAI}} + \Delta_{\text{TCB-TT}} + \Delta_{\text{TDB-TCB}} \quad (6.20)$$

Einstein 延迟 (Einstein Delay) 描述了由于引力场和天体运动的相对论效应引起的时间延迟, 包括了问题 3 中涉及到的两个主要效应, 它的计算主要涉及广义相对论中的引力红移以及狭义相对论的动钟变慢效应:

1) 引力红移效应: 由广义相对论引起, 光子在强引力场中会表现为红移, 即光子在靠近大质量天体 (如太阳) 的引力场中传播时, 其频率变小, 时间变慢。这导致光子传播的路径上产生时间延迟。

2) 动钟变慢 (狭义相对论效应): 也就是运动的钟相对于静止的钟走得更慢, 这与参考系的相对速度有关。简单来说, 当卫星在绕太阳或其他天体运动时, 动钟变慢效应会导致观测者的时钟相对质心 (SSB) 的时间变慢, 造成光子到达时刻的延迟。

对  $\Delta_{E\odot}$  进一步解释:

$$\Delta_{\text{TAI-UTC}} = l \quad (6.21)$$

$$\Delta_{\text{TT-TAI}} = 32^s.184 + \delta \quad (6.22)$$

$$\Delta_{\text{TCB-TT}} = \frac{1}{c^2} (\vec{r}_{\text{obs/地心}} \cdot \vec{v}_E) + L_G \Delta t_{\text{TT}} + \frac{1}{c^2} \int_{t_0}^t \left[ U_{\oplus} + \frac{v_E^2}{2} + \Delta L_c^{(\text{PN})} + \Delta L_c^{(\text{A})} \right] dt \quad (6.23)$$

$$\Delta_{\text{TDB-TCB}} = -(L_c + L_G) \Delta t_{\text{TCB}} \quad (6.24)$$

其中

$l$  表示观测期间对应的闰秒数;

$\delta$  表示观测期间对应的 TT 时间尺度与 TAI 时间尺度的小量差;

$\vec{v}_E$  表示地心相对于太阳系质心的运动速度;

$L_G$  表示地球旋转大地水准面上的地球重力势;

$L_c$  表示距离太阳一个天文单位处的太阳引力势;

$\Delta t_{\text{TT}}$ 、 $\Delta t_{\text{TCB}}$  分别表示在 TT、TCB 时间尺度下, 由起算时刻走过的坐标时;

$U_{\oplus}$  代表太阳系内除地球之外, 卫星受到其他天体的引力势;

$\Delta L_c^{(\text{PN})}$  表示高阶相对论修正项;

$\Delta L_c^{(\text{A})}$  表示小行星修正项;

因为光子的到达时间是在地球上测得的 TT 时间（MJD 为 58119.1651507519），但我们通常需要将时间转换到一个全局的参照系（如太阳系质心 TCB）来进行分析，特别是当我们讨论宇宙尺度的现象时（如脉冲星的精确测量）。TDB 又比 TCB 略微更适合处理近似开普勒轨道的天体运动（如行星和小行星）。因此  $\Delta_{\text{TAI-UTC}}$  以及  $\Delta_{\text{TT-TAI}}$  不需要改正，而仅考虑  $\Delta_{\text{TCB-TT}}$  和  $\Delta_{\text{TDB-TCB}}$

在  $\Delta_{\text{TCB-TT}}$  中， $\frac{1}{c^2}(\vec{r}_{\text{obs/地心}} \cdot \vec{v}_E)$  描述了观测器位置 ( $\vec{r}_{\text{obs/地心}}$  与地球速度  $\vec{v}_E$ ) 的相对运动引起的爱因斯坦时间延迟。由于观测器与脉冲星之间存在相对速度，会导致时间在不同参照系中流逝的速率不同。其中地球速度  $\vec{v}_E$  可以通过附件 3-de200.bsp。

使用 *compute\_and\_differentiate* 函数来计算地球在 58119.165150751 的速度和位置，得出的速度为  $-1706.27257479 \text{ km/s}$ ,  $-1808.90304994 \text{ km/s}$ ,  $-784.19444061 \text{ km/s}$ 。

地球所在的引力势导致的时间膨胀效应，使得引力场中的钟运行得比远离引力场的钟更慢，产生了  $L_G \Delta_{\text{TT}}$ ，其中  $L_G$  是广义相对论中的引力红移因子。通常是一个已知常数，取值为  $1.550519768 \times 10^{-8}$ 。

在  $\Delta(TCB - TT)$  中仅考虑

$$\Delta(TCB - TT) = \int_{t_0}^t (U_\oplus + v_E^2) dt \quad (6.25)$$

因为广义相对论的高阶修正项  $\Delta L_c^{(PN)}$  以及额外的修正项  $\Delta L_c^{(A)}$  无法获取，且影响较小。

### **$\Delta(\text{TDB} - \text{TCB})$**

$$\Delta(TDB - TCB) = -(L_c + L_G) \Delta t_{\text{TCB}} \quad (6.26)$$

$L_c$  和  $L_G$  是时间尺度之间的校正因子：

1.  $L_c$  是太阳系质心时（TDB）与太阳质心时（TCB）之间的尺度因子。
2.  $L_G$  是广义相对论中的引力红移因子。

这一项描述了不同时间尺度之间由于引力效应和速度效应引起的时间差异。脉冲星信号的传播路径会受到太阳系质心参考系的影响，因此需要考虑这些校正项。

$\Delta t_{\text{TCB}}$  代表了 TCB 时间的变化，等于  $\Delta(TCB - TT)$ 。

所有星体的引力势可以通过轨道力学计算得到。可以使用类似牛顿引力公式进行近似计算：

$$U_\oplus = -\frac{GM_j m}{r_\oplus} \quad (6.27)$$

$M_j$  为天体的质量， $m$  卫星的质量可以假设为 1。 $r_\oplus$  是卫星到天体的距离， $G$  是万有引力常数， $6.67430 \times 10^{-11} \text{ N}\text{m}^2\text{kg}^{-2}$  单位为  $\text{m}^3\text{kg}^{-1}\text{s}^{-2}$ 。

$L_G$  是地球的重力势能。在地球表面上，通常是在人类地水准面（geoid）上的值。这个值反映了地球引力场的强度。 $r$  是距离地球中心的距离（通常是地球的平均半径 6371 公里，即地球引力场的强度距离）。

$$G = 6.67430 \times 10^{-11} \text{ m}^3 \cdot \text{kg}^{-1} \cdot \text{s}^{-2}, \quad M_j = 5.97219 \times 10^{24} \text{ kg}, \quad r_{\oplus} = 6.371 \times 10^6 \text{ m}$$

$$U = -\frac{6.67430 \times 10^{-11} \times 5.97219 \times 10^{24}}{6.371 \times 10^6} \approx -6.969290134e-10 \text{ m}^2 \text{s}^{-2}, \quad L_G = 6.969290134e-10.$$

$L_c$  是在距离太阳 1 天文单位 (AU, 约  $1.496 \times 10^{11}$  m) 处的引力势，同样代入公式 6.27，  
 $L_c \approx 1.48082686741 \times 10^{-8}$ 。

### 6.2.2 罗默尔 (Roemer) 延迟

$$\Delta_{R\odot} = -\frac{(\vec{r}_{\text{obs}} + \vec{r}_E) \cdot \vec{n}}{c} \quad (6.28)$$

其中：

$r_{\text{obs}}$  表示卫星相对于地心的矢量；

$r_E$  为地心相对于 SSB 的矢量；

$n$  为 Crab 脉冲星的方向矢量；

Roemer 延迟 [5, 6] 是由于光子传播路径中的几何差异导致的传播时间差异。要求脉冲星光子到达卫星相对于太阳系质心的几何延迟，首先需要获取卫星相对于太阳系质心的位置矢量，即卫星和太阳系质心之间的空间距离差异，这在问题 2 中已经获取。由于光子来源于脉冲星，我们还需要得到脉冲星在太阳质心坐标系下的位置向量  $\vec{n}$ 。然后计算卫星在脉冲星方向上的位置投影，就可以得到卫星在脉冲星方向上的相对距离差。

**脉冲星自行效应。**与第二问不同的是，在问题 3 中几何延迟传播时延中包含了脉冲星的自行效应。脉冲星处于运动状态，其位置受自行影响发生变化，因此脉冲星方向矢量  $\vec{n}$  并不是一个固定的值。脉冲星的自行包括两部分：赤经方向的自行  $\mu_a = -14.7 / \text{year}$ ，以及赤纬方向的自行  $\mu_\delta = 2.0 / \text{year}$ 。（在附件 4 中已经给出）

其中，对  $\vec{n}$  进一步解释：

$$\vec{n} = \vec{n}^0 + \mu_\perp(t_a - t_0) - \left( \frac{1}{2} |\mu_\perp|^2 \vec{n}^0 + \mu_\parallel \mu_\perp \right) (t_a - t_0)^2 \quad (6.29)$$

$\vec{n}^0$  表示初始历元时刻脉冲星质心在太阳系质心系中的方向矢量

$t_a$  表示脉冲星质心处收到脉冲信号的时刻；

$\mu$  是这段时间内脉冲星的自行平均速度；

$\mu_{\perp}$  是脉冲星在天球上的自行矢量。它由脉冲星赤经、赤纬方向的自行  $\mu_{\alpha}$  和  $\mu_{\delta}$  组成：

$$\mu_{\perp} = (\mu_{\alpha} \cos \delta, \mu_{\delta})$$

$\mu_{\parallel}$  是脉冲星在观测方向上的速度，通常认为其数值较小。

$n^0$  为脉冲星在某个参考时间时的方向，是脉冲星在固定时刻的位置，从天文测量得到，通常可以从星表中查到。但由于本节的时刻是假设的，因此我们假设初始时刻的方向为 (1, 1, 1)。 $\mu_{\perp}(t_a - t_0)$  反映了脉冲星在天球上由于自行移动而引起的方向变化。随着时间的推移，脉冲星的实际位置与其初始位置会发生偏移。自行的单位是角度（通常是毫角秒/年），需要转换为弧度/秒，因此时间差  $(t_a - t_0)$  是以秒为单位计算的。这个二次项是为了更精确地考虑脉冲星自行的二阶效应，尤其在时间差  $(t_a - t_0)$  较大的情况下。这部分包括了脉冲星在视向  $\mu_{\parallel}$  上的运动及自行加速度。 $t_a$  在题目中已经给出为 MJD 58119.1651507519，而  $t_0$  是脉冲星方向的参考历元时间，在附件 4 中已经给出为 57715.000000295。

需要通过上述公式精确计算脉冲星自行影响。如果不考虑自行效应，计算出的脉冲星到达时间将会有偏差。譬如我们将光子到达卫星的时刻  $t_a$  直接设为与问题 2 一致为 57062.0，我们发现 Roemer 延迟为 314.71707790055365s，这表明自行的延迟影响达到  $490 - 314 = 176$ s，影响非常大。

### 6.2.3 夏皮罗 (Shapiro) 延迟

Shapiro 延迟是由于光信号经过大质量天体附近时，受到其引力场影响而导致的时间延迟。这种延迟效应由广义相对论预测，它表明引力场会弯曲时空，从而使光信号的传播时间延长。

$$\Delta_{S\odot} = - \sum_j \frac{2Gm_j}{c^3} \ln |r_j(1 - \cos \psi_j)| + \Delta_{S\odot 2} \quad (6.30)$$

其中：

$j$  表示太阳系内第  $j$  个大天体；

$\psi_j$  表示 Crab 脉冲星-卫星-天体  $j$  的夹角；

$\Delta_{S\odot 2}$  表示太阳的二阶 Shapiro 延迟；

本公式中，Shapiro 延迟  $\Delta_{S\odot}$  包含了与夹角  $\psi_j$ ，这个夹角表示 Crab 脉冲星-卫星-天体  $j$  之间的角度。我们采用基于的向量点积来计算这个夹角。

$$\vec{r}_1 \cdot \vec{r}_2 = r_1 r_2 \cos \psi_j$$

其中，

$r_1$  因此卫星到脉冲星的方向向量；

$r_2$  卫星到大天体  $j$  的方向向量；

从点积公式中，解出  $\psi_j$ :

$$\psi_j = \arccos \left( \frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1||\vec{r}_2|} \right)$$

由于脉冲星非常远，方向可以被近似认为是固定的，卫星到脉冲星的方向向量可以近似为  $\vec{r}_2 = -\vec{n}$ ， $\vec{r}_1$  是大天体位置和卫星位置之间的差， $\vec{r}_1 = \vec{a} - \vec{b}$  表示卫星相对于大天体的位置，由此计算出脉冲星、卫星和天体之间的夹角  $\psi_j$ ，然后代入 Shapiro 延迟公式进行计算。

一旦得到了夹角  $\psi_j$ ，可以代入 Shapiro 延迟公式的第一部分，并进一步结合  $\Delta_{S\odot 2}$  来计算完整的 Shapiro 延迟。

$\frac{2Gm_j}{c^3}$  表示天体  $j$  的质量  $m_j$  对光子路径的引力效应。天体质量越大，对光子产生的时延效应越显著。 $\ln |r_j|$  表示天体相对于脉冲星和卫星的距离，距离越近，延迟效应越显著。 $r_{j/SSB}$  为天体相对于 SSB 的位置，可以从附件 3-de200.bsp 中获取。

对  $\Delta_{S\odot 2}$  进一步展开:

$$\Delta_{S\odot 2} = \frac{4G^2 m_\odot^2}{c^5 |r| \tan \psi \sin \psi} \quad (6.31)$$

要精确计算 Shapiro 延迟，需要：1. 获取太阳系内主要天体的质量和相对位置（可以从天体历和天文数据库中查找）。2. 通过轨道力学和向量运算计算天体间的夹角。3. 在长期观测中，考虑脉冲星的自行对其方向矢量的影响。

通过这种方式，可以结合已知的天体位置和向量关系来计算公式中的夹角  $\psi_j$ ，最终得到 Shapiro 延迟。

### 6.3 脉冲传输时延精确求解

在求解的过程中，我们需要用到相关的太阳系质量数据如下表6.5。求解的结果如表6.7

表 6.5 太阳系各行星的质心相关数据

序号	行星质心	质量 (kg)	太阳系质心类别	日期范围
1	水星质心	$0.330 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
2	金星质心	$4.87 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
3	地球质心	$5.97 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
4	火星质心	$0.642 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
5	木星质心	$1898 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
6	土星质心	$568 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
7	天王星质心	$868 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09
8	海王星质心	$102 \times 10^{24}$	2 型太阳系质心	1899-07-29 至 2053-10-09
9	冥王星质心	$0.0130 \times 10^{24}$	2 型太阳系质心	1899-07-29 至 2053-10-09
10	太阳	$1.989 \times 10^{24}$	第 2 类太阳系质心	1899-07-29 至 2053-10-09

注 1：质量数据来源于 NASA Planetary Fact Sheets

表 6.6 太阳系各行星相对于 SSB 位置坐标

序号	星体	X (km)	Y (km)	Z (km)
1	水星	-5.61955320e+10	-1.69744127e+09	6.76388336e+09
2	金星	1.01542674e+11	3.77342182e+10	1.05593063e+10
3	地球	-1.12094190e+11	8.75121480e+10	-3.79151058e+07
4	火星	2.05324588e+11	4.45319981e+10	1.48735611e+10
5	木星	-5.88594233e+11	-4.89123009e+11	-2.23968694e+11
6	土星	-7.82894680e+11	-1.18328239e+12	-4.55050784e+11
7	天王星	2.88195839e+12	7.53618911e+11	2.89303288e+11
8	海王星	4.12546616e+12	-1.58657980e+12	-7.52103507e+11
9	冥王星	1.12517916e+12	-4.44559668e+12	-1.72916923e+12
10	太阳	4.46720405e+08	-7.97511235e+07	-5.75474969e+07

表 6.7 问题三最终计算结果

	Einstein 延迟	Roemer 延迟	Shapiro 延迟
三种时延 (s)	-4.054	-2.605e-05	-479.525
总时延 (s):	-483.579		

## 7 光子到达时刻模型仿真与求解

### 7.1 问题四整体分析

问题四关键在通过标准轮廓建立模拟光子到达速率函数，进而模拟光子序列。通过模拟的光子序列反求光子到达速度，折叠形成脉冲轮廓。精细化的过程就是更加精准地通过标准轮廓模拟出光子到达速率。因此可以考虑将函数拟合函数或者插值函数更加精准地线性化。

从题目已知条件，背景光子是随机的，到达探测器的时间服从均匀泊松分布 [? ]，而脉冲星发出的 X 射线光子信号虽然具有周期性，但是在传播中会逐渐衰减，抵达探测器的光子流已经变得非常微弱，可建模为非齐次泊松过程。因此可以使用泊松分布的期望求得微小时间段内到达卫星光子数量的期望模拟光子到达的时间序列。

问题四第一问需要通过仿真来得到卫星接受到脉冲星光子的时间序列。从另一个角度来说，我们需要仿真出 10 秒内每一个时刻所卫星所接受到光子的数量。由于 10 秒是一个连续的时间，因此我们考虑将时间进行微分。将 10 秒时间微分成 10000 个时间段。每个时间段的具体时间设定为 0.001s。针对每个时间段内的光子，我们认为在极狭小的时间段  $(ta, tb)$  的平均光子数也就是  $ta$  或者  $tb$  时刻的瞬时光子数据。为了得到  $(ta, tb)$  的平均光子数目也就是  $(ta, tb)$  时间段内，光子到达的泊松分布的数学期望。泊松分布的数学期望很好求解。因此转化为了模拟出  $(ta, tb)$  时间段内的光子累计。这个量是一个积分量，主要于光子到达卫星的速率  $\lambda$  有关。这个值又由标准脉冲轮廓归一化值的影响。于是本文通过拟合或者插值的方式将标准脉冲轮廓拟合或插值出来就解决了问题。最后将所得到的光子序列进行折叠来凸显脉冲峰值。仿真精度提高，我们直接使用更加精确地高阶泰勒 [7, 8] 展开，这样对标准轮廓拟合函数进行线性化。

### 7.2 问题四逐步求解

#### 7.2.1 混合高斯模型仿真光子序列

对于脉冲星标准轮廓，我们采用分段混合高斯模型来拟合，其形式为：

$$h(\phi) = \begin{cases} a_1 e^{\frac{(\phi-b_1)^2}{c_1}} + a_2 e^{\frac{(\phi-b_2)^2}{c_2 c_3}}, & 0 \leq \phi \leq 0.4 \\ a_1 e^{\frac{(\phi-b_1)^2}{c_1 c_4}} + a_2 e^{\frac{(\phi-b_2)^2}{c_2^2}} + a_3 e^{\frac{(\phi-b_3)^2}{c_3 c_4}}, & \phi > 0.4 \end{cases} \quad (7.32)$$

其中公式中涉及到的各项参数通过最小二乘法得到后如下：

当  $0 \leq \phi \leq 0.4$ : a1=0.0086; a2=5.1585; b1=0.2119; b2=1.3552; c1=0.0099; c2=-2.7490; c3=8.7100;

当  $0.4 \leq \phi \leq 1$ : a1=-0.1398; a2=0.0003; a3=0.0093; b1=0.9618; b2=-1.4824; b3=0.6865; c1=-0.0466; c2=0.0075; c3=0.0502; c4=2.4411;

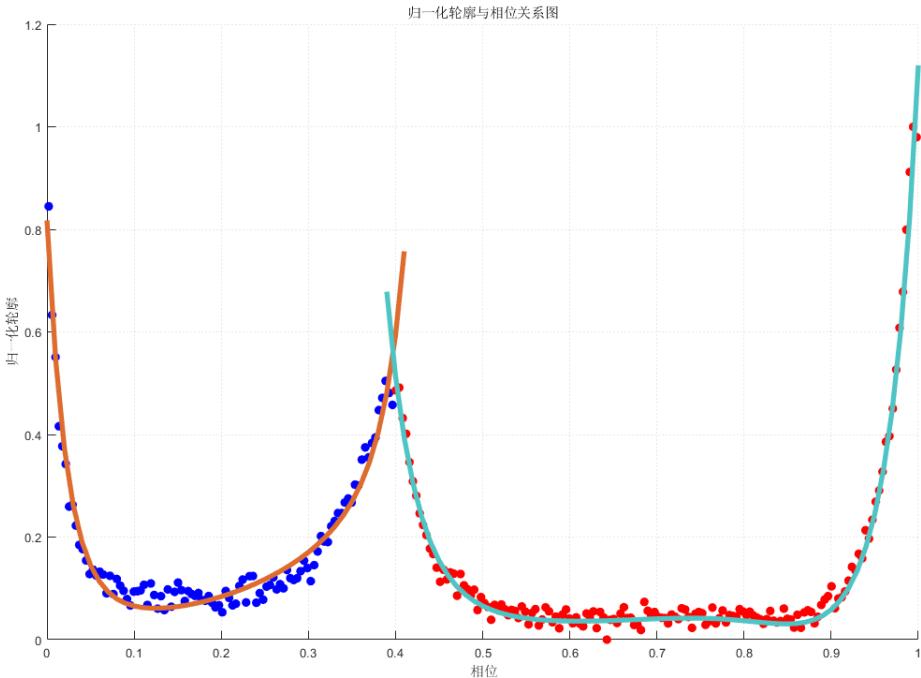


图 7.8 混合高斯模型与标准脉冲轮廓可视化

用  $\lambda(t) \geq 0$  表示光子到达的速率函数，则在观测时间段  $(s, e)$  内接收到的光子数  $N_{t_a, t_b}$  服从参数为  $\Lambda = \int_0^t \lambda(s) ds$  的泊松分布，即

$$P(k; t_a, t_b) = \frac{\exp(-(\Lambda(t_b) - \Lambda(t_a))) (\Lambda(t_b) - \Lambda(t_a))^k}{k!} \quad (7.33)$$

速率函数  $\lambda(t)$  可表示为：

$$\lambda(t) = \lambda(\phi(t)) = \lambda_b + \lambda_s h(\phi(t)) \quad (7.34)$$

其中：

$\lambda_b \geq 0$ ,  $\lambda_s \geq 0$  分别为 X 射线信号光子和背景光子到达探测器的平均速率；

$h(\phi)$  为  $\phi \in [0, 1]$  上的脉冲星标准轮廓, 满足  $\min h(\phi) = 0, \int_0^1 h(\phi)d\phi = 1$ ,  $h(\phi)$  可以扩展为  $(-\infty, +\infty)$  上的周期函数, 即  $h(\phi + n) = h(\phi), n$  为整数  
 $\phi(t)$  为 SSB 处的脉冲星相位预测模型, 代表  $t$  时刻 SSB 处的脉冲相位

$\phi(t)$  得泰勒级数展开:

$$\phi(t) = \phi_0 + f(t - t_0) + \frac{\dot{f}}{2}(t - t_0)^2 + \frac{\ddot{f}}{6}(t - t_0)^3 \quad (7.35)$$

其中:

- $t_0$  为相位模型的参考起始时刻
- $\phi_0$  为在 SSB 处在  $t_0$  时刻的脉冲相位
- $f, \dot{f}, \ddot{f}$  分别为脉冲星的自转频率、自转频率一阶项和二阶项, 但是二阶项未给, 因此本文直接考虑一阶项。

针对速率函数有, 当  $t_a, t_b$  足够小时, 进行泰勒展开:

$$\lambda(t_b) = \lambda_b + \lambda_s \times \left( h(\phi'(t_a)) + \frac{\partial h}{\partial \phi'} \cdot dt \right) \quad (7.36)$$

时间间隔  $dt = 0.01\text{s}$ , 于是:

$$\Lambda(t_b) = \lambda_b + \lambda_s \times \left( h(\phi'(t_a)) + \frac{\partial h}{\partial \phi'(t_a)} \cdot dt \right) \quad (7.37)$$

$$\Lambda(t_a) = \lambda_b + \lambda_s \times (h(\phi'(t_a))) \quad (7.38)$$

$$\Lambda(t_b) - \Lambda(t_a) = \lambda_s \times \frac{\partial h}{\partial \phi(t_a)} \cdot dt \quad (7.39)$$

$(t_a, t_b)$  时间段内, 间隔足够小, 间隔等小, 可认为是  $t_b$  瞬时的光子数

$$E(k) = \int kP(k, t_a, t_b) = \Lambda(t_b) - \Lambda(t_a) = \lambda_s \times \frac{\partial h}{\partial \phi(t_a)} \cdot dt \quad (7.40)$$

$\lambda(t) = \lambda_b + \lambda_s h(\phi(t))$ , 有关  $\phi(t)$  函数可知:  $\phi(t) = \phi(t_0) + \nu(t - t_0) + \frac{1}{2}\dot{\nu}(t - t_0)^2$

令  $\phi(t_0) = 0$ , 也就是初始相位为 0,  $t_0 = 0$  为初始时刻。, 仿真相位过程中, 要保证相位小于 1, 因此需要每次计算的相位取 mod 1 运算, 也就是要对 1 取余。

$$\begin{cases} \frac{\partial h_1}{\partial \phi} = \frac{2a_1(\phi-b_1)}{c_1} \cdot e^{\frac{(\phi-b_1)^2}{c_1}} + \frac{2a_2(\phi-b_2)}{c_2 c_3} \cdot e^{\frac{(\phi-b_2)^2}{c_2 c_3}} \\ \frac{\partial h_2}{\partial \phi} = \frac{2a_1(\phi-b_1)^2}{c_1 c_4} \cdot e^{\frac{(\phi-b_1)^2}{c_1^2}} + \frac{2(\phi-b_2)}{c_2^2} \cdot e^{\frac{(\phi-b_2)^2}{c_2^2}} + \frac{2a_3(\phi-b_3)}{c_3 c_4} \cdot e^{\frac{(\phi-b_3)^2}{c_3 c_4}} \end{cases} \quad (7.41)$$

## 7.2.2 仿真算法流程

---

### 算法 1 相位值和光子数计算算法

---

**输入:** 时间分辨率  $dt$ , 脉冲星旋转频率  $v$ , 频率一阶导数  $v_{dot}$ , 蟹脉冲星光子通量密度  $\lambda_s$ , 探测器有效面积  $A$

**输出:** 相位值  $\phi\_values$ , 光子数  $E\_vector$

```
1: 初始化相位数组  $\phi\_values$  和光子数数组  $E\_vector$ 
2: 设置初始相位  $\phi\_ta = 0$ 
3:  $\phi\_values(1, 1) = 0$ 
   计算相位
4: for  $i = 1$  to  $N$  do
5:   计算当前时间  $ta = i \cdot dt$  和下一个时间  $tb = (i + 1) \cdot dt$ 
6:   计算相位值  $\phi\_temp = \phi\_ta + v \cdot (tb - ta) + 0.5 \cdot v_{dot} \cdot (tb - ta)^2$ 
7:   更新相位数组  $\phi\_values(1, i) = \phi\_ta$ 
8:   更新  $\phi\_ta = \phi\_temp$ 
9: for  $i = 1$  to  $N$  do
10:   计算当前时间  $ta = i \cdot dt$  和下一个时间  $tb = (i + 1) \cdot dt$ 
11:   获取相位值  $\phi\_ta = normalized\_phi(i)$  和  $\phi\_tb = normalized\_phi(i + 1)$ 
     分段计算导数, 用于模拟光子速率
12:   if  $\phi\_ta \leq 0.4$  then
13:     计算  $dao = Dao\_Gaussian1(\phi\_ta)$ 
14:   else
15:     计算  $dao = Dao\_Gaussian2(\phi\_ta)$ 
     根据泊松分布的数学期望计算每个时刻的光子数
16:   计算光子数  $E = \lambda_s \cdot dao \cdot dt \cdot A$ 
17:   更新光子数数组  $E\_vector(1, i) = E$ 
18: 计算光子数的绝对值  $abs\_E = abs(E\_vector)$  返回相位值, 光子数 (泊松分布数学期望)
19: return  $\phi\_values, E\_vector, abs\_E$ 
```

---

### 7.2.3 光子序列仿真结果

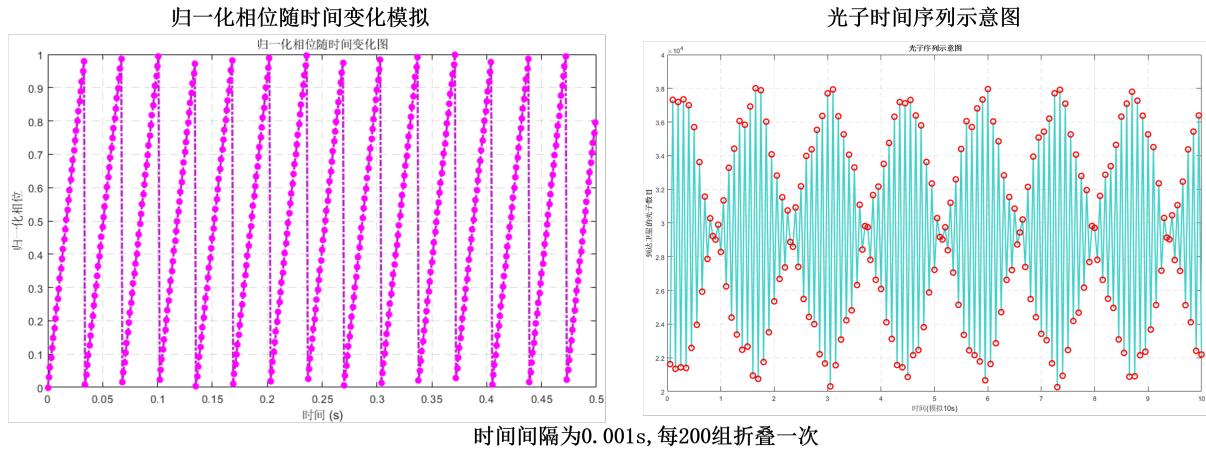


图 7.9 时间微分 0.001 且的 200 组一次折叠仿真

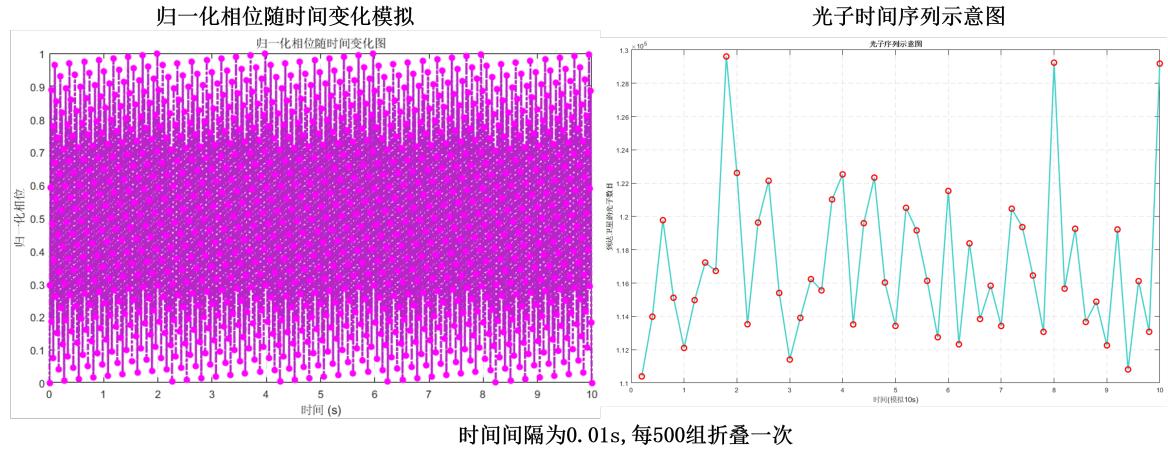


图 7.10 时间微分 0.01 且的 500 组一次折叠仿真

我们模拟了光子在两种情况下的光子序列，在 10 秒内，一种是 0.001 秒时间间隔，每 200 组折叠。一组是 0.01 秒内 500 组折叠。由于要扩大折叠次数，因此需要尽量增加时间间隔才能对比更加明显。从光子序列的仿真结果可以知道当折叠组数与时间间隔增加时，光子序列的模拟效果会变差。这是因为时间微分变大，极短时间内的光子到达数目更加不准确。另一方面，折叠次数增加能够尽可能增加信号的峰值。周期性下降，让信号更加明显。

### 7.2.4 脉冲轮廓折叠仿真

由于问题四第一问已经仿真得到了光子瞬时脉冲序列。我们可以根据上一时刻的光子量与当前时刻的光子量进行作差得到光子变化量。光子速率为光子在单位时间内的变化

量。于是可以直接用光子变化量与微小时间的比值来模拟光子速率。仿真脉冲轮廓可以由仿真的光子速率来表示。于是就完成了脉冲轮廓的仿真。

### 7.2.5 脉冲轮廓还原模型

由于前一小节已经模拟出了光子的时间序列，因此需要对模拟的光子序列在微分时间上求斜率得到光子速率。于是有公式7.42。

$$\lambda(t_i) = \frac{G(t_{i+1}) - G(t_i)}{dt} \quad (7.42)$$

其中  $dt$  为仿真时间间隔,  $G(t_i)$  为  $t_i$  时刻模拟的光子量。根据光子速率可以仿真得到脉冲轮廓归一化仿真结果如公式7.43

$$h(t_i) = \frac{\lambda(t_i) - \lambda_b}{\lambda_s} \quad (7.43)$$

其中,  $h(t_i)$  为  $t_i$  时刻脉冲轮廓归一化的仿真值。同时, 也可以获取  $t_i$  时刻对应的相位。这样就能仿真出脉冲轮廓随相位的函数变化。

### 7.2.6 脉冲轮廓仿真结果与误差

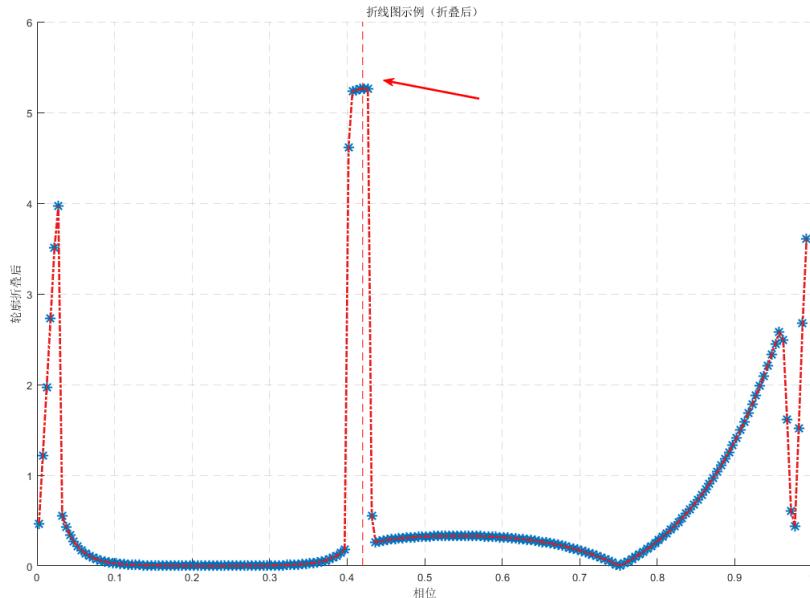


图 7.11 折线图示例（折叠后）

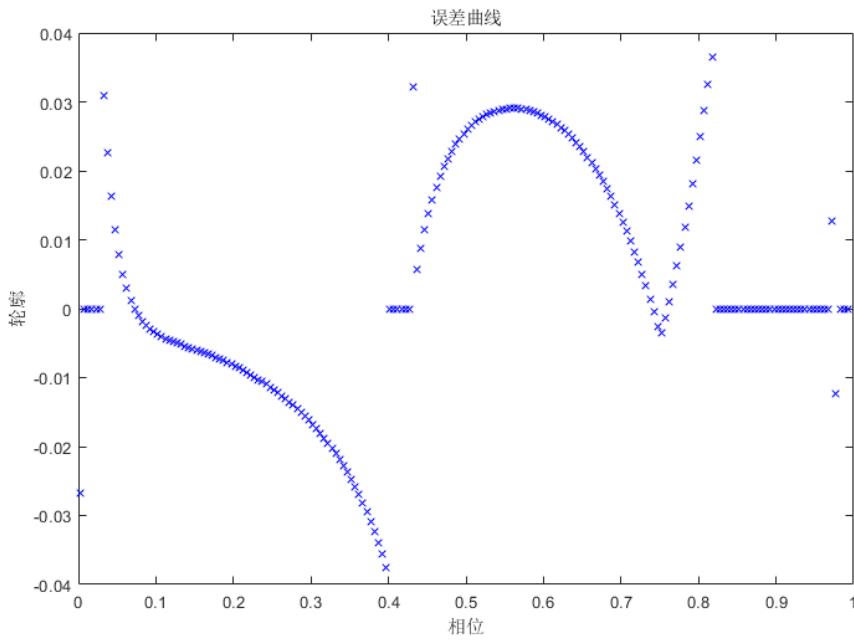


图 7.12 脉冲轮廓误差曲线

### 7.2.7 仿真精度优化分析

为了提高仿真精度，如果希望模拟更加精确到光子到达速率我们期刊如何能够让拟合函数更加精确，可以采用线性插值或者扩展泰勒展开项。同时，可以采用更小的仿真时间间隔。于是本文将两种方法进行融合。一方面。我们将脉冲轮廓模拟方法的融合混和高斯模型与线性插值进行加权平均进行融合。另一方面，我们将仿真模拟的时间间隔再缩小十倍，来提高仿真精度。

### 7.2.8 精细化仿真模型

由于需要更加精确地仿真结果，在保证子流量密度值不变。根据公式7.40可以知道，可以精细化偏导数来精确模拟仿真结果，由于在前问仿真脉冲轮廓的时候，我们只是保留了混合高斯模型泰特展开的一阶项。在精细化过程中，于是考虑保留到二阶项。

二次偏导求解进行偏导精细化建模

$$e^{\frac{(\phi-b_1)^2}{c_1}} = H(b_1, c_1) \quad (7.44)$$

$$e^{\frac{(\phi-b_2)^2}{c_2 c_3}} = H(b_2, c_2, c_3) \quad (7.45)$$

偏导数表达式

$$\frac{\partial'' h_1}{\partial \phi} = \left[ \frac{2a_1}{c_1} + \frac{4a_1(\phi - b_1)^2}{c_1^2} \right] H(b_1, c_1) + \left[ \frac{2a_2 + 4a_2(\phi - b_2)^2}{c_2 c_3} \right] H(b_2, c_2, c_3) \quad (7.46)$$

同理可得:  $\frac{\partial'' h_2}{\partial \phi}$

$$h = h_0 + \frac{\partial h}{\partial \phi} dt + \frac{\partial'' h}{\partial \phi} (dt)^2 \quad (7.47)$$

根据公式  $E(K)$  得到:

$$E(K) = \lambda_s \cdot \left( \frac{\partial h}{\partial \phi(t_a)} dt + \frac{\partial'' h}{\partial \phi} (dt)^2 \right) \quad (7.48)$$

公式7.48为精细化仿真具体公式。通过增加二次导数项，可以更加精准的模拟到达探测器的光子数量。

### 7.2.9 精细化仿真结果

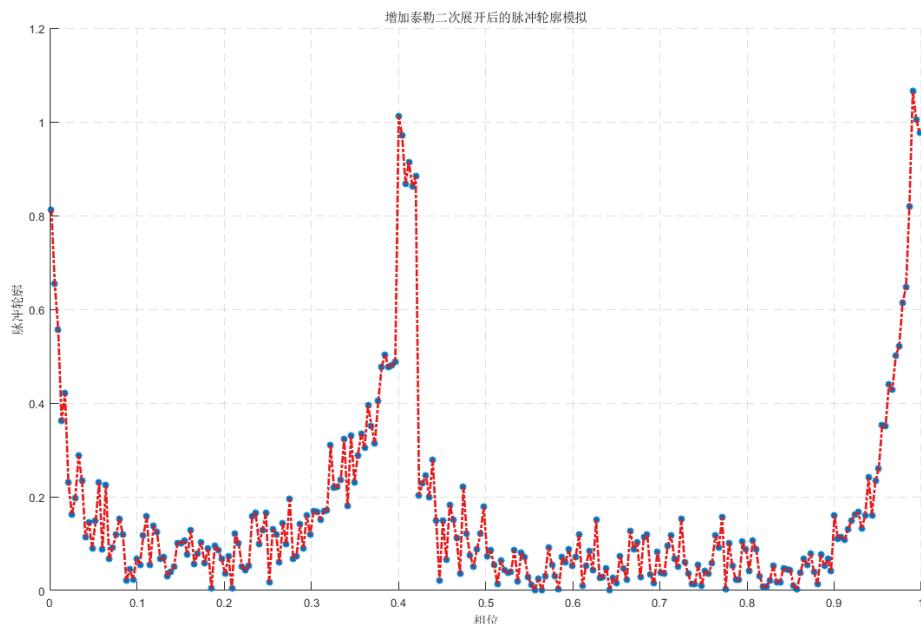


图 7.13 增加泰勒二次展开后的脉冲轮廓模拟

从图7.13中可以看出，在精细化仿真结果已经十分接近标准轮廓的形状了。这也证明了这个精细化仿真方法的有效性。

## 参考文献

- [1] 朱鸿旭, Crab 脉冲星 X 射线计时观测数据分析, 中国科学院大学 (中国科学院国家授时中心), 2019.
- [2] 薛梦凡, 李小平, 孙海峰, 刘兵, 方海燕, 沈利荣, 一种新的 X 射线脉冲星信号模拟方法, 2015.

- [3] 任红飞, 刘思伟, 姬剑锋, 周庆勇, X 射线脉冲星信号辐射的数值模拟方法, 2022.
- [4] 苏剑宇, 方海燕, 包为民, 等, 航天器处 X 射线脉冲星观测信号模拟方法, 物理学报, 71 (22):393-408, 2022.
- [5] 陈师, X 射线脉冲星计时模型参数估计方法研究, 西安电子科技大学, 2022.
- [6] 周庆勇, 杜兰, 一种脉冲星计时模型与其精度评定, 第一届中国卫星导航学术年会论文集 (中), 信息工程大学测绘学院;, 341-348, 2010.
- [7] 王辉, 温志刚, 王娜, 等, Crab 脉冲星巨脉冲辐射等待时间分布的非稳态泊松模拟, 天文学报, 65(02):105-112, 2024.
- [8] 周庆勇, 脉冲星计时数据的处理理论与方法研究, 战略支援部队信息工程大学, 2020.

## 附录 A MATLAB 源程序

### A.1 第 4 问程序

problem4.m

```
clc;
clear;

% 定义参数
T_obs = 10; % 观测时间 (秒)
lambda_b = 1.54; % 背景光子通量密度 (ph/(s*cm^2))
lambda_s = 10 * lambda_b; % 蟹脉冲星光子通量密度 (ph/(s*cm^2))
A = 250; % 探测器有效面积 (cm^2)

% 脉冲星旋转参数
v = 29.647854750036593; % 脉冲星旋转频率 (Hz)
v_dot = -368970.96e-15; % 频率一阶导数 (Hz/s)

% 定义相位函数
phi0 = 0; % 参考时间的相位
t0 = 0; % 参考时间

% 定义高斯脉冲配置文件函数
h1 = @(x) Gaussian1(x); % 确保该函数已定义
h2 = @(x) Gaussian2(x); % 确保该函数已定义

% 定义模拟次数
num_moni=50;

% 生成初始值为0的1*100的向量 (代表区间)
num_accumulate=zeros(1,100);

% 对于一次模拟有
% 定义时间分辨率和向量
dt = 0.001; % 时间分辨率 (秒)
t = 0:dt:T_obs; % 时间向量
N=T_obs/dt;

phi_values=zeros(1,N+1)
```

```

phi_ta=0;
phi_values(1,1)=0;
for i=1:N
    ta=i*dt;
    tb=(i+1)*dt;
    % 计算相位值
    phi_temp = phi_ta + v * (tb - ta) + 0.5 * v_dot * (tb - ta).^2;

    phi_values(1,i)=phi_ta;
    phi_ta=phi_temp;

end

normalized_phi = mod(phi_values, 1);

% 绘制归一化相位值
figure('Position', [100, 100, 800, 400]);
plot(t, normalized_phi, 'LineWidth', 2);
xlabel('时间 (s)');
ylabel('归一化相位');
title('归一化相位随时间变化');
grid on;

% 截取前 30 个元素
t_subset = t(1:500); % 截取 t 的前 30 个元素
normalized_phi_subset = normalized_phi(1:500); % 截取
    normalized_phi 的前 30 个元素

% 绘制截取后的归一化相位值
figure('Position', [100, 100, 800, 400]);
plot(t_subset, normalized_phi_subset, 'LineWidth', 2);
xlabel('时间 (s)');
ylabel('归一化相位');
title('前 30 个归一化相位随时间变化');
grid on; % 显示网格

E_vector=zeros(1,N);
%计算k时刻的光子数
for i=1:N

```

```

ta=i*dt;
tb=(i+1)*dt;
phi_ta=normalized_phi(i);
phi_tb=normalized_phi(i+1);
if phi_ta<=0.4
    dao.Dao_Gaussian1(phi_ta);
else
    dao.Dao_Gaussian2(phi_ta);
end
E=lambda_s*dao*dt*A;
E_vector(1,i)=E;
end
abs_E=abs(E_vector);

% 设置每组的大小
group_size = 50;

% 检查 abs_E 的长度是否可以整除 group_size
if mod(length(abs_E), group_size) ~= 0
    error('向量的长度不能被 %d 整除。', group_size);
end

% 将 abs_E 转换为矩阵形式，每列为一个组
reshaped_E = reshape(abs_E, group_size, []); % 每组 50 个元素

% 计算每组的和
folded_E = sum(reshaped_E, 1); % 对每列求和

% 显示折叠结果
disp('每组的和:');
disp(folded_E);

%模拟最小的时间间隔0.001
%折叠时间为0.5s
delta_t=dt*group_size
fo_x=delta_t:delta_t:10;
% 可视化折叠后的结果
figure; % 创建新图形窗口

```

```

plot(fo_x,folded_E, 'o-', 'LineWidth', 2, 'MarkerSize', 8); % 绘制
    折叠结果
xlabel('时间(模拟10s)'); % X轴标签
ylabel('到达卫星的光子数目'); % Y轴标签
title('光子序列示意图'); % 图形标题
grid on; % 显示网格

%接下来还原轮廓
[M_rows, M_cols] = size(abs_E);
V=zeros(1,M_cols-1)
for i=1:M_cols-1
    V(i)=(((abs_E(i+1)-abs_E(i))/dt)-lambda_b)/lambda_s;
end

V_x=normalized_phi(2:10/dt);
% 根据 V_x 的升序排列 v
[V_x_sorted, sort_idx] = sort(V_x);
V_sorted = V(sort_idx);

V_sorted=abs(V_sorted);

group_size = 50; % 每组的数据数量

% 确保 V_sorted 的长度是 group_size 的整数倍
num_groups = floor(length(V_sorted) / group_size);
V_sorted_grouped = reshape(V_sorted(1:num_groups * group_size),
    group_size, num_groups);

% 计算每组的平均值
V_sorted_mean = mean(V_sorted_grouped, 1);

% 对应的 V_x 进行折叠
V_x_sorted_grouped = reshape(V_x_sorted(1:num_groups * group_size),
    group_size, num_groups);
V_x_sorted_mean = mean(V_x_sorted_grouped, 1);

% 创建散点图
figure;

```

```

scatter(V_x_sorted_mean, V_sorted_mean, 100); % 100 为点的大小
title('折叠轮廓');
xlabel('相位');
ylabel('模拟轮廓');
hold on;

plot(V_x_sorted_mean, V_sorted_mean, '-o'); % '-o' 表示折线和点
title('折线图示例（折叠后）');
xlabel('相位');
ylabel('轮廓折叠后');

% 绘制一条垂直于 x 轴的线
x_line = 0.4; % x 轴上的位置
line([x_line, x_line], ylim, 'Color', 'r', 'LineStyle', '--'); % 红色虚线
% 绘制箭头
arrow_start = [x_line, 0]; % 箭头起点
arrow_end = [x_line, 0.5]; % 箭头终点
grid on;
function [result] = Gaussian1(x)
%高斯函数初始化
%轮廓函数拟合的混合高斯模型
a1= 0.0086;
a2=5.1585;
b1= 0.2119 ;
b2=1.3552;
c=8.7100;
c1=0.0099;
c2=-2.7490;
result=a1*exp((x-b1).*(x-b1)/c1)+a2*exp((x-b2).*(x-b2)/c2*c);
end
function [result] = Gaussian2(x)
%高斯函数初始化
%轮廓函数拟合的混合高斯模型
a1=-0.1398;
a2=0.0003;
a3=0.0093;
b1=0.9618;
b2=-1.4824;

```

```

b3=0.6865;
c=2.4411;
c1=-0.0466;
c2=0.0075;
c3=0.0502;
result=a1*exp((x-b1).* (x-b1)/c1*c)+a2*exp((x-b2).* (x-b2)/c2*c2)
+a3*exp((x-b3).* (x-b3)/c3*c);
end
function [result] = Dao_Gaussian1(x)
%高斯函数初始化
a1= 0.0086;
a2=5.1585;
b1= 0.2119 ;
b2=1.3552;
c=8.7100;
c1=0.0099;
c2=-2.7490;
result=a1*exp((x-b1)*(x-b1)/c1)*2*(x-b1)/c1+a2*exp((x-b2)*(x-b2)
)/c2*c)*2*(x-b2)/(c2*c);
end
function [result] = Dao_Gaussian2(x)
%高斯函数初始化
%轮廓函数拟合的混合高斯模型
a1=-0.1398;
a2=0.0003;
a3=0.0093;
b1=0.9618;
b2=-1.4824;
b3=0.6865;
c=2.4411;
c1=-0.0466;
c2=0.0075;
c3=0.0502;
result=a1*exp((x-b1)*(x-b1)/c1*c)*2*(x-b1)/(c1*c)+a2*exp((x-b2)
)*(x-b2)/c2*c2)*2*(x-b2)/(c2*c2)+a3*exp((x-b3)*(x-b3)/c3*c)
*2*(x-b1)/(c3*c);
end

```

## 附录 B Python 源程序

### B.1 第1问程序

problem1.py

```
import numpy as np
import pandas as pd

# 根据轨道六参数计算卫星的速度与位置
def keplerian_to_cartesian(a, e, i, Omega, omega, nu, mu
=398600.4418):
    """
    将开普勒轨道元素转换为笛卡尔坐标系中的位置和速度。
    参数：
        a      : 半长轴 (km)
        e      : 偏心率
        i      : 倾角 (弧度)
        Omega  : 升交点经度 (弧度)
        omega  : 近地点角距 (弧度)
        nu     : 真近点角 (弧度)
        mu     : 地球引力常数 (km^3/s^2)，默认值为地球的引力常数

    返回：
        r      : 位置向量 (km)
        v      : 速度向量 (km/s)
    """
    # 计算卫星的距离
    r = a * (1 - e**2) / (1 + e * np.cos(nu))

    # 计算位置向量
    x_orb = r * (np.cos(Omega) * np.cos(omega + nu) - np.sin(
        Omega) * np.sin(omega + nu) * np.cos(i))
    y_orb = r * (np.sin(Omega) * np.cos(omega + nu) + np.cos(
        Omega) * np.sin(omega + nu) * np.cos(i))
    z_orb = r * (np.sin(i) * np.sin(omega + nu))

    # 计算速度向量
    p = a * (1 - e**2) # 轨道参数
```

```

vx_orb = (mu / p)**0.5 * (-np.cos(Omega) * (np.sin(omega +
nu)+e*np.sin(nu)) - np.sin(Omega) * (np.cos(omega + nu)
+e*np.cos(nu)) * np.cos(i))
vy_orb = (mu / p)**0.5 * (-np.sin(Omega) * (np.sin(omega +
nu)+e*np.sin(nu)) + np.cos(Omega) * (np.cos(omega + nu)
+e*np.cos(nu)) * np.cos(i))
vz_orb = (mu / p)**0.5 * np.sin(i) * (np.cos(omega + nu)+e
*np.cos(nu))

return np.array([x_orb, y_orb, z_orb]), np.array([vx_orb,
vy_orb, vz_orb])

# 读取参数
file_path = './six_root.txt' # 替换为你的文件路径
data = pd.read_csv(file_path, delim_whitespace=True, header=
None)

# 计算半长轴，每一列代表什么意思？
# 离心率
# 特征高度
# 升交点经度
# 轨道倾角
# 近地点角距
# 真近点角
data_root=data.to_numpy()

print(data_root[0,1])

# 给定参数
e =data_root[0,0] # 偏心率
h =data_root[0,1] # 特征高度 (km^2/s)
mu=398600.4418 # 地球引力常数
# 计算半长轴
a = (h**2) / (mu * (1 - e**2))
print(a)

i = data_root[0,3] # 倾角
Omega = data_root[0,2] # 升交点经度
omega = data_root[0,4] # 近地点角距 (度转弧度)

```

```

nu = data_root[0, 5] # 真近点角 (度转弧度)

# 计算位置和速度
position, velocity = keplerian_to_cartesian(a, e, i, Omega,
                                              omega, nu)

print("位置 (km):", position)
print("速度 (km/s):", velocity)

# 接下来根据万有引力定律进行验证
total_v=np.sqrt(velocity[0]**2+velocity[1]**2+velocity[2]**2)
print("总速度 (km/s):", total_v)

# 万有引力常数 (m^3 kg^-1 s^-2)
G = 6.67259e-11

# 地球质量 (kg)
M = 5.972e24

# 轨道高度
H = a*(10**3) # 地球半径 (m), 先用轨道长半轴代替

# 计算轨道速度
v = np.sqrt(G * M / H)
V=v/1000 # 单位转换(换成km/s)
print("轨道速度 (km/s):", V)

# 验证角动量

```

## B.2 第2问程序

problem2.py

```

from jplephem.spk import SPK
import numpy as np
kernel = SPK.open("de200.bsp")
MJD=57062.0
# 获取约化儒略日
jd=MJD+2400000.5
jd= np.array([jd])

```

```

# 获取GCRS坐标系中卫星的位置
positio_of_satellite = np.array
([1274.91341509, -1848.85196499, 6507.26265528])
print('GCRS坐标系中卫星的位置:', positio_of_satellite)

# 获取MJD给定时刻地球质心在太阳质心坐标系中的位置
# 3代表地球质心，0代表太阳质心
position = kernel[0, 3].compute(jd)
# print("地球质心在太阳质心坐标系中的位置:", position)
pos=np.array(position[:, 0])
print('地球质心在太阳质心坐标系中的位置:', pos)

# 验证pos的单位
xxx=np.sqrt(pos[0]**2+pos[1]**2+pos[2]**2)
print('地球质心距离太阳系质心的距离:', xxx)

# 计算卫星在太阳系质心的位置
satellite_position = positio_of_satellite + pos
print('卫星在太阳系质心的位置', satellite_position)

# 计算距离
L=np.sqrt(satellite_position[0]**2+satellite_position[1]**2+
           satellite_position[2]**2)
# 计算时间延迟
c=3*10**5
delta_t=L/c
print('时间延迟(s):', delta_t)

```

### B.3 第3问程序

```

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from jplephem.spk import SPK
import numpy as np
def pulsar_position(ra_ref, dec_ref, distance, pch_pm, dt_ra,
                    dt_dec, mjd_obs):
    # 将赤经和赤纬转换为弧度
    ra_rad = np.deg2rad(ra_ref)

```

```

dec_rad = np.deg2rad(dec_ref)

# 计算时间差 (年)
delta_t = (mjd_obs - pch_pm) / 365.25 # 从参考历元到观测
    时刻的时间差，单位为年

# 自行的影响 (转换为弧度)
dt_ra_rad = np.deg2rad(dt_ra / (3600 * 1000)) # 赤经自行
    (弧度/年)
dt_dec_rad = np.deg2rad(dt_dec / (3600 * 1000)) # 赤纬自行
    (弧度/年)

# 计算新的赤经和赤纬
ra_new = ra_rad + dt_ra_rad * delta_t
dec_new = dec_rad + dt_dec_rad * delta_t

# 使用球坐标转换为笛卡尔坐标
x = distance * np.cos(dec_new) * np.cos(ra_new)
y = distance * np.cos(dec_new) * np.sin(ra_new)
z = distance * np.sin(dec_new)

# 返回脉冲星的空间位置 (km)
return np.array([x, y, z])
"""

第三题

"""

# 计算 Roemer 延迟
def calculate_time_difference(mjd_obs, mjd_ref):
    """ 计算MJD 的时间差，单位：年 """
    days_per_year = 365.25
    delta_mjd = mjd_obs - mjd_ref
    delta_years = delta_mjd / days_per_year
    return delta_years

def calculate_pulsar_direction_with_proper_motion(n0, dt_ra,
    dt_dec, t_a, t_0):
    """

```

根据赤经、赤纬自行计算脉冲星方向矢量的变化

参数：

n0: 初始方向矢量 (单位矢量)

dt\_ra: 赤经自行 (毫角秒/年)

dt\_dec: 赤纬自行 (毫角秒/年)

t\_a: MJD观测时刻

t\_0: MJD参考历元时刻

返回：

更新后的方向矢量 n

"""

# 将自行从毫角秒/年 转换为 弧度/年

arcsec\_to\_radians = 4.84814e-9

mu\_ra = dt\_ra \* arcsec\_to\_radians \* 3600 \* 24 \* 365.25 #

赤经自行

mu\_dec = dt\_dec \* arcsec\_to\_radians \* 3600 \* 24 \* 365.25 #

赤纬自行

# 计算观测时间与参考历元之间的时间差 (单位: s)

delta\_t = calculate\_time\_difference(t\_a, t\_0)

# 初始方向矢量 (n0可以根据脉冲星位置或假设为单位矢量)

# 这里只是占位符，需要实际脉冲星位置进行初始化

n = n0

# 赤经方向的变化

n\_ra\_change = mu\_ra \* delta\_t

# 赤纬方向的变化

n\_dec\_change = mu\_dec \* delta\_t

# 假设n0方向为 [n\_x, n\_y, n\_z]，使用近似的变化公式调整方向

n[0] += n\_ra\_change # 对应 x 轴 (赤经方向)

n[1] += n\_dec\_change # 对应 y 轴 (赤纬方向)

# 确保方向矢量归一化

n\_normalized = n / np.linalg.norm(n)

**return** n\_normalized

```

def roemer_delay(robs, rE, n):
    """
    计算 Roemer 延迟

    参数：
    robs: 卫星相对于地心的矢量 (单位: 米)
    rE: 地心相对于 SSB 的矢量 (单位: 米)
    n: 脉冲星的方向单位矢量

    返回：
    Roemer 延迟 (单位: 秒)

    """
    c = 3e5 # 光速, 单位: km/s

    # 计算 robs + rE 的和
    relative_position = np.array(robs) + np.array(rE)
    # 计算点积
    dot_product = np.dot(relative_position, n)

    # 计算 Roemer 延迟
    roemer_delay = -dot_product / c

    return roemer_delay


# 计算 Shapiro 延迟
def calculate_pulsar_satellite_body_angle(a, b, n):
    """
    计算脉冲星-卫星-大天体的夹角 (弧度)

    参数：
    a -- 大天体相对于太阳系质心的位置向量 (numpy array)
    b -- 卫星相对于太阳系质心的位置向量 (numpy array)
    n -- 脉冲星的方向向量 (单位向量) (numpy array)

    返回：
    theta -- 脉冲星-卫星-大天体的夹角 (弧度)

    """

```

```

# 计算卫星到大天体的相对位置向量 r1
r1 = a - b

# 计算卫星到脉冲星的方向向量 r2 (反向的n)
r2 = -n

# 归一化 r1 和 r2
r1_hat = r1 / np.linalg.norm(r1)
r2_hat = r2 / np.linalg.norm(r2)

# 计算 r1 和 r2 之间的点积
dot_product = np.dot(r1_hat, r2_hat)

# 使用 arccos 计算夹角 (弧度)
theta = np.arccos(dot_product)

return theta


def shapiro_delay(masses, positions, satellite_SSB, n):
    """
    计算 Shapiro 延迟

    参数：
    masses: 大天体的质量列表 (单位: kg)
    positions: 大天体的位置列表 (单位: km, 转换为米)
    satellite_SSB: 卫星与在太阳质心坐标系的坐标 (单位: km, 转
        换为米)
    n: 脉冲星方向单位矢量 (因为很远所以可等价于卫星相对脉冲星
        的单位矢量)
    rE: 地心相对于 SSB 的矢量 (单位: km, 转换为米)
    sun_mass: 太阳的质量 (单位: kg)

    返回：
    Shapiro 延迟 (单位: 秒)
    """

    G = 6.67430e-11 # 万有引力常数 (单位: m^3 kg^-1 s^-2)
    c = 299792458 # 光速 (单位: m/s)

```

```

# 将位置和距离从 km 转换为 m
positions_m = [np.array(pos) * 1e3 for pos in positions]
    # 转换为米
r_m = [np.sqrt(pos_m[0]**2 + pos_m[1]**2 + pos_m[2]**2)
    for pos_m in positions_m] # 转换为米
r_s = r_m[9] # 太阳位置
angle_sun = 0
# 计算 Shapiro 延迟
delta_S = 0.0
flag = 0 # 记录计算到了第几个星体
for m, position, r in zip(masses, positions_m, r_m):
    # 计算夹角
    angle = calculate_pulsar_satellite_body_angle(position,
        , satellite_SSB, n)
    # 计算每个天体的 Shapiro 延迟
    delta_S += -(2 * G * m / c ** 3) * np.log(np.abs(r)) *
        (1 - np.cos(angle))
    if flag == 9:
        angle_sun = angle

delta_S += (4 * G ** 2 * masses[9] ** 2 / (c ** 5 * np.abs
    (r_s))) * np.tan(angle_sun) * np.sin(angle_sun)

return delta_S

# 计算爱因斯坦延迟
def get_gravitational_potential(kernel_path, jd,
    position_satellite, body_ids):
    """
    获取其他天体的引力势修正项。
    参数：
    kernel_path : str
        SPICE 内核文件路径 (.bsp 文件)
    jd : float
        儒略日，用于计算天体位置
    position_satellite: numpy array
        卫星的位置向量 (单位: m)
    """

```

```

body_ids      : list of int
    其他天体的 SPICE ID 列表

    返回：
total_potential: float
    其他天体的引力势修正项总和 (单位: m2/s2)
"""

# 打开 SPICE 内核文件
kernel = SPK.open(kernel_path)

# 初始化引力势修正项
total_potential = 0.0

# 引力常数 G (m3 kg(-1) s(-2))
G = 6.67430e-11

# 计算每个天体的引力势修正项
for body_id in body_ids:
    position_celestial = kernel[0,body_id].compute(jd)
    distance_vector = position_celestial -
        position_satellite

    # 计算距离 (单位: m)
    distance = np.linalg.norm(distance_vector) * 1000

    # 计算引力势 (单位: m2/s2)，需定义一个质量字典
    mass = get_mass(body_id)  # 天体质量，单位 kg
    potential = - G * mass / distance # 引力势公式

    # 叠加引力势
    total_potential += potential

return total_potential

def get_mass(body_id):
    # 根据 SPICE ID 返回相应天体的质量 (单位: kg)
    mass_dict = {
        1: 3.3011e+23,  # 水星

```

```

2: 4.8675e+24,    # 金星
3: 5.97219e+24,   # 地球
4: 6.4171e+23,    # 火星
5: 1.8982e+27,    # 木星
6: 5.6834e+26,    # 土星
7: 8.6810e+25,    # 天王星
8: 1.024e+26,     # 海王星
9: 1.303e+22,     # 冥王星
10: 1.989e+30,    # 太阳
}

return mass_dict.get(body_id, 0)  # 默认返回0, 如果没有找到

```

# 计算爱因斯坦延迟的函数

```

def einstein_delay(mjd, pch_pm, satellite_position, ve, LG, Lc
, U_oplus):
    # 3. 计算  $\Delta TCB-TT = 1/c^2 * (robs * ve) + LG * \Delta t TT +$  高阶
    # 修正项积分
    Ve = ve / 1000
    # robs 是卫星相对于地心的矢量, ve 是地心速度
    robs_dot_ve = np.dot(satellite_position, ve)  # robs . ve,
    # 注意这部分单位要一致

    #  $1/c^2$  项
    delta_tcb_tt = (robs_dot_ve / c ** 2)
    # 引力势项 (LG)
    delta_tcb_tt += LG * (mjd - pch_pm)

    # 高阶修正项 (积分项)
    delta_tcb_tt += (1 / c ** 2) * (U_oplus + np.dot(Ve, Ve))
    * (mjd - pch_pm)

    # 4.  $\Delta TDB-TCB = -(Lc + LG) * \Delta t TCB$ 
    delta_tdb_tcb = -(Lc + LG) * delta_tcb_tt

    # 总爱因斯坦延迟  $\Delta E = \Delta TAI-UTC + \Delta TT-TAI + \Delta TCB-TT + \Delta TDB$ 
    # -TCB

```

```

delta_E_sun = delta_tcb_tt + delta_tdb_tcb

return delta_E_sun


file_path = './six_root.txt' # 替换为你的文件路径
data = pd.read_csv(file_path, delim_whitespace=True, header=None)
data_root=data.to_numpy()

print(data_root[0,1])

# 给定参数
e =data_root[0,0] # 偏心率
h =data_root[0,1] # 比角动量 (km^2/s)
mu=398600.4418 # 地球引力常数
# 计算半长轴
a = (h**2) / (mu * (1 - e**2))
print(a)

i = data_root[0,3] # 倾角
Omega = data_root[0,2] # 升交点经度
omega = data_root[0,4] # 近地点角距 (度转弧度)
nu = data_root[0,5] # 真近点角 (度转弧度)

# 计算位置和速度
position, velocity = keplerian_to_cartesian(a, h, e, i, Omega,
                                              omega, nu)
print("位置 (km):", position)
print("速度 (km/s):", velocity)

# 第一步获取地球在太阳系质心的位置
kernel_path = "E://2024华为杯正式比赛//1.2024年中国研究生数学
建模竞赛赛题//2024年中国研究生数学建模竞赛赛题//F题//附件//
附件3-de200.bsp"
MJD = 57062.0
JD_earth = MJD + 2400000.5
# jd = np.array([2457061.5, 2457062.5, 2457063.5, 2457064.5])

```

```

# 获取地球在SSB中的位置
position(ssb) = get_earth_position(ssb)(kernel_path, JD_earth,
0, 3)
print("太阳质心位置 (km):", position(ssb))

# 第二步基于卫星在GCRS和地球在太阳中的位置计算卫星在太阳系质心
# 的位置
L_SAT = np.sqrt(position[0]**2+position[1]**2+position[2]**2)
L_SSB = np.sqrt(position_ssbb[0]**2+position_ssbb[1]**2+
position_ssbb[2]**2)
satellite_SSB = position + position_ssbb
delta_L = np.sqrt((satellite_SSB[0])**2 + (satellite_SSB[1])**2
+ (satellite_SSB[2])**2)
print("卫星在太阳系质心坐标系中的位置 (km):", satellite_SSB)

'''



第三题
'''



G = 6.67430e-20
M_sun = 1.989e30
c = 3.0e5
mu = 3.986e5

# 示例调用
orbit_radius = 2.0e11 # 卫星轨道半径, 单位: m
time_interval = 86400 # 时间间隔, 单位: s
frequency = 1.4e9 # 观测频率, 单位: Hz

# 计算总时延
# 输入数据
mjd_obs = 58119.1651507519 # 探测器接收光子时刻
mjd_ref = 57715.000000295 # 脉冲星参考历元
dt_ra = -14.7 # 赤经自行 (毫角秒/年)
dt_dec = 2.0 # 赤纬自行 (毫角秒/年)

# 初始方向矢量 (占位符, 需根据实际情况调整)
n0 = np.array([1.0, 1.0, 1.0])

# 计算方向矢量更新

```

```

n_updated = calculate_pulsar_direction_with_proper_motion(n0,
    dt_ra, dt_dec, mjd_obs, mjd_ref)
delta_roemer = roemer_delay(position, position(ssb, n_updated))
print(f"roemer 时延: {delta_roemer} 秒")

# 第二步 Shapiro 时延计算 (缺少示 Crab 脉冲星-卫星-天体 j 的夹
角, 还有太阳的二阶 Shapiro 延迟)

masses = [3.3011e23, 4.8675e24, 5.97237e24, 6.4171e23, 1.8982
          e27, 5.6834e26, 8.6810e25, 1.02413e26, 1.303e22, 1.9885e30]
          # 太阳和地球的质量 (单位: kg)
position_planets = []
# 获取各大行星位置 planet_ids=10 为太阳 3 为地球
planet_ids = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for planet_id in planet_ids:
    # 获取行星相对于太阳系质心 (SSB) 的位置
    position_planet = get_earth_position(ssb(kernel_path,
        JD_earth, 0, planet_id))
    position_planets.append(position_planet)

# 计算 Shapiro 延迟
shapiro_delay_value = shapiro_delay(masses, position_planets,
    satellite_SSB, n_updated)

print(f"Shapiro 延迟: {shapiro_delay_value} 秒")

# 第三步计算爱因斯坦延迟
mjd = 58119.1651507519 # 光子到达探测器的时刻 (MJD)
pch_pm = 57715.000000295 # par 文件的参考历元
satellite_position = position # 卫星相对于地心的位置 (单位: m
)
# 获取地心相对于太阳系质心的速度
kernel = SPK.open(kernel_path)
# 通过内核文件获取地球在SSB坐标系中的位置和速度
_, velocity(ssb = kernel[0, 3].compute_and_differentiate(
    JD_earth))

ve = np.array(velocity(ssb)) # 地心相对于太阳系质心的速度 (单
位: km/s)

```

```
LG = 6.969290134e-10 # 地球旋转大地水准面上的地球重力势
Lc = 1.48082686741e-8 # 一个天文单位处太阳引力势
body_ids = [1, 2, 4, 5, 6, 7, 8, 9, 10] # 除地球外所有天体的
    SPICE ID
U_oplus = get_gravitational_potential(kernel_path, JD_earth,
    satllite_SSB, body_ids) # 其他天体的引力势修正项 (单位: m2 /
    s2)

# 计算爱因斯坦延迟
delay_einstein = einstein_delay(mjd, pch_pm,
    satellite_position, ve, LG, Lc, U_oplus)
print(f"爱因斯坦延迟: {delay_einstein} 秒")

delay_total = delta_roemer + shapiro_delay_value +
    delay_einstein
print(f"总延迟: {delay_total} 秒")
```