

“FAST” 主动反射面的形状调节

摘要

针对问题一，首先建立起三维空间直角坐标系，确定标准状态下抛物面的方程。引入关于仰角和方位角的旋转矩阵，建立起理想抛物面模型。在题中所给的条件下，焦点和焦距固定时，理想抛物面上的点与基准球面的径向距离会出现大于 $0.6m$ 的情况。所以，需要考虑反射面板调节因素，建立起以焦距为决策变量，阴影部分面积最小为目标函数的单目标优化模型。采用数值求解法对模型进行求解，以步长为 7.8×10^{-4} 对决策变量进行遍历。求得当焦距为 $140.3548m$ ，抛物线的顶点坐标为 $(0, 0, -300.7684)$ 时，目标函数达到最小值为 17.0549 。此时，抛物面上的点到基准球面的最大径向距离小于 $0.6m$ 。

针对问题二，通过问题一中求出的旋转矩阵，确定该问中的理想抛物面，得出该抛物面的顶点坐标为 $(-49.369, -36.926, -294.312)$ 。为建立反射面调节模型，先选定在理想抛物面口径范围内所有的主索节点；通过促动器的上下端点确定主索节点径向变化的单位向量。建立起以主索节点径向伸缩量为决策变量，调节后反射板与抛物面围成的径向体积之和最小为目标函数的单目标优化模型。对于模型求解，先将积分离散化，目标函数转化为求反射面上的点到抛物面的径向距离最短。再选取反射面板上的 7 个特征点，用以代表径向距离，表示工作抛物面与理想抛物面的贴合程度。接着将反射面板调整至较贴合的初始状态，然后同步推动所有主索节点，最后得到伸缩量矩阵、调整后节点坐标和抛物面顶点坐标，分析结果并放入 result.xlsx 中。经过计算，贴合后再度调节伸缩量为 **0.055** 时，得到平均径向距离的最小值为 **0.0087m**。

针对问题三，在问题二的调整方案下，首先将抛物面口径内所有主索节点，通过旋转矩阵变换到方位角为 0° 、仰角为 90° 的位置。然后确定一个与 z 轴垂直，过理想抛物面焦点的平面，建立起该平面上入射信号与反射信号的映射关系。通过定义函数并进行积分计算从而求得调节后馈源舱及基准反射球面的接收比。模型求解时，在平面内以 3 为步长， $[-150, 150]$ 为遍历范围对函数进行离散求和，利用海伦公式判断入射点是否在反射面板内，最终求得调节后馈源舱的接收比约为 **2%**，基准反射球面的接收比约为 **15.5%**。

关键词：旋转矩阵 单目标优化模型 一型曲线积分 空间向量 径向最短距离

一、问题重述

1.1 问题背景

天文设备对人类探索宇宙具有重要意义，由我国自主设计研发的 500 米口径球面射电望远镜——中国天眼(FAST)，其单口径之大和灵敏度之高皆为世界之最。FAST 主要用于探索宇宙起源和演化^[1]，大幅拓展了人类的视野；同时，也极大促进了我国科学研究事业的发展，是中华民族伟大复兴征程上的重要工程。

FAST 的主要组成部分有主动反射面、馈源舱以及支承系统等。调节主动反射面的形状使其成为工作抛物面，当工作抛物面越接近理想抛物面时，馈源舱获得天体电磁波经反射面反射后的接收效果越佳。因而如何根据反射面调节因素确定理想抛物面，并使工作抛物面尽可能接近理想抛物面，具有重要的科研实际意义。

1.2 需要解决的问题

1、结合考虑反射面板调节因素，确定一个天体 S 在基准球面正上方，即当 $\alpha=0^\circ$, $\beta=90^\circ$ 时的理想抛物面。

2、确定天体 S 位于 $\alpha=36.795^\circ$, $\beta=78.169^\circ$ 时的理想抛物面。建立反射面板调节模型，使反射面尽量贴近该理想抛物面，并将相应的结果保存在“result.xlsx”文件中。

3、根据第 2 问的反射面调节方案，计算调节后馈源舱的接收比，并与基准反射球面的接收比作比较。

二、问题分析

2.1 问题一的分析

问题一要求确定一个天体在标准状态（即 $\alpha=0^\circ$, $\beta=90^\circ$ ）时的理想抛物面，并且需要考虑反射面板的调节因素。首先，对于空间几何问题，需要建立空间直角坐标系，确定各点的位置；然后，根据抛物面上的点与确定的焦点坐标以及抛物面的焦距参数，可以确定抛物线在标准状态下的方程。由于天体位置的变动，抛物线会发生转动，根据方位角 α 和仰角 β ，引入旋转矩阵，用以表示抛物面的转动。求得旋转后抛物面的方程，代入题中给出的角度，求出该理想抛物面含参数的方程。

由于需要结合反射面板的调节因素，所以需要在满足约束条件的情况下，使该理想抛物面尽可能贴近基准球面。因而，建立起关于理想抛物线方程的优化模型。最后，遍历求解，确定该理想抛物面的位置及形状。

2.2 问题二的分析

问题二要求确定天体角度发生旋转时的理想抛物面，根据第一问中的旋转矩阵，只需将问题一的最佳理想抛物面旋转，即可确定本问的最佳理想抛物面。对于建立反射面调节模型，首先需要选定在 300m 口径范围内的所有主索节点；然后，分析促动器的伸缩，对主索节点移动的影响。从而，建立起以主索节点径向移动距离为决策变量，每块反射板与抛物面围成的径向体积之和最小为目标函数的反射面调节优化模型。关于模型求解，考虑对积分离散化、取特征值及遍历迭代等方法，求解出最优解。

2.3 问题三的分析

问题三要求在问题二反射面调节方案的基础上，计算调节后馈源舱的接收比，同时与基准反射球面的接收比作比较。由于问题二中的理想抛物面存在一定的倾斜角度，所以为方便分析计算，考虑将问题二中调整好的反射板旋转到基准位置。然后，在 XOY 平面内选择任意一点，作出入射信号和反射信号。根据光学性质和坐标位置，计算各点、

各直线之间的关系，求解出反射信号与馈源舱接收信号区域的交点位置。馈源舱的接收比可用馈源舱有效接收范围内反射信号的区域面积与 $300m$ 口径的面积之比。关于模型求解首先需要验算信号入射后是否落在合理的反射板上；然后，根据约束条件，对平面内的点进行遍历，找出反射至馈源舱的入射点，计算接收比，并与基准反射球面的接收比进行比较。

三、模型假设

1. 基准状态下，所有主索节点均位于基准球面上；
2. 电磁波信号及反射信号均视为直线传播；
3. 馈源舱接受平面的中心只能在与基准球面同心的一个球面上移动；
4. 理想抛物面与工作抛物面的焦点相同。
5. 下拉锁与促动器为刚性连接，不发生相对位移。

四、符号说明

符号	说明
a	抛物面焦距
β	天体的仰角
α	天体的方位角
R	基准球的半径
Q_i	第 i 个主索节点
d	主索节点的径向移动距离
(x, y, z)	抛物面上点的坐标
A_1	关于 α 的旋转矩阵
A_2	关于 β 的旋转矩阵
δ	馈源舱的接收比
P	抛物面的焦点

注：其他符号将在下文给出具体说明。

五、问题一模型的建立与求解

5.1 理想抛物面模型

1. 建立坐标系

为方便描述抛物面的位置变化，如图 1(a)所示，本文首先将理想抛物面的顶点设为坐标原点 O ，竖直向上为正方向 z 轴，建立三维直角坐标系；同时，其在平面 XOZ 上的投影如图 1(b)所示：

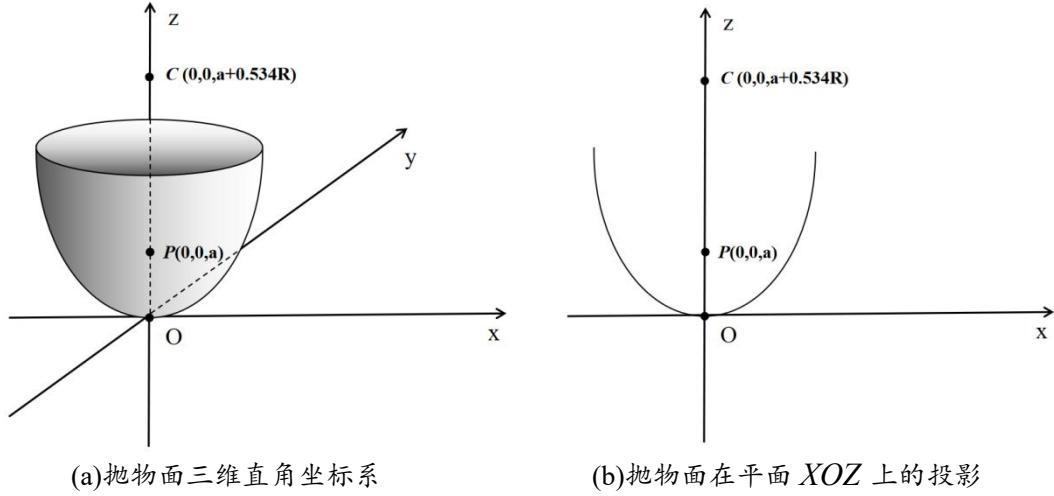


图 1：坐标系建立示意图

2. 抛物面的方程

①. 标准状态

定义：当抛物面顶点位于原点 O ，焦点 P 坐标为 $(0,0,a)$ 时，为标准状态抛物面。

根据题中所给的两同心球面的半径差为 $F=0.466R$ ，可以求出基准球面的球心 C 位于 $(0,0,a+0.534R)$ ，此时抛物面处于标准状态。

根据上述坐标，可以求出标准状态下的抛物面方程：

$$z = \frac{1}{4a}(x^2 + y^2) \quad (1)$$

②. 引入方位角 α 和仰角 β

对于仰角 β ：确定 y 轴为旋转轴，方向符合右手系，即在 XOZ 平面内逆时针旋转。则引入旋转矩阵 A_1 ，表示标准状态下的抛物面绕 y 轴旋转的角度：

$$A_1 = \begin{bmatrix} \cos(\frac{\pi}{2} - \beta) & 0 & \sin(\frac{\pi}{2} - \beta) \\ 0 & 1 & 0 \\ -\sin(\frac{\pi}{2} - \beta) & 0 & \cos(\frac{\pi}{2} - \beta) \end{bmatrix} \quad (2)$$

对于方位角 α ：确定 z 轴为旋转轴，方向同样符合右手系，在 XOY 平面内逆时针旋转。同样，引入旋转矩阵 A_2 ，表示绕 y 轴旋转后的抛物面需要绕 z 轴旋转的角度：

$$A_2 = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

综上所述，在绕 y 轴旋转仰角 β 和绕 z 轴旋转方位角 α 后抛物面方程为：

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = A_2 \cdot (A_1 \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}) \quad (4)$$

其中， (x, y, z) 表示标准状态下的抛物面上点的坐标。

同样地，旋转后该抛物面的焦点坐标可以表示为：

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = A_2 \cdot (A_1 \cdot \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix}) \quad (5)$$

旋转后的球心坐标可以表示为：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = A_2 \cdot (A_1 \cdot \begin{bmatrix} 0 \\ 0 \\ a + 0.534R \end{bmatrix}) \quad (6)$$

其中， a 表示焦点的纵坐标值， R 表示球的半径。

③. 平移抛物面

根据题意，题目要求以基准球面的球心为坐标原点，且抛物面与坐标原点的位置关系如下图所示：

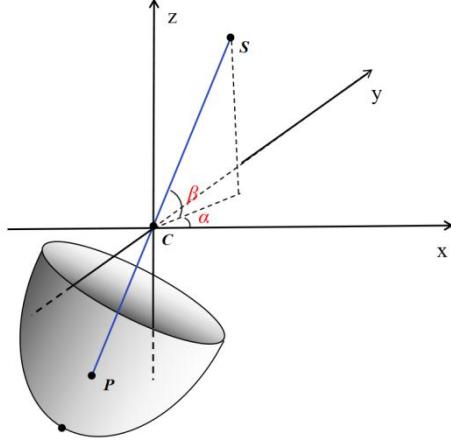


图 2：以球心为原点的三维坐标系

由上图 2 可知，当球心为坐标原点时，由标准状态旋转后的抛物面需要进行平移才能满足上述位置关系。而抛物面需要平移的距离等于未平移前抛物面的坐标减去未平移前球心的坐标：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (7)$$

综上所述，经过旋转和平移后抛物面的坐标可以表示为：

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = A_2 \cdot (A_1 \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}) - A_2 \cdot (A_1 \cdot \begin{bmatrix} 0 \\ 0 \\ a + 0.534R \end{bmatrix}) \quad (8)$$

其中， A_1 表示抛物面关于仰角 β 的旋转矩阵， A_2 表示抛物面关于方位角 α 的旋转矩阵；

(x, y, z) 表示标准状态下的抛物线上点的坐标， $z = \frac{1}{4a}(x^2 + y^2)$ ； a 表示焦点的纵坐标值， R 表示球的半径。

根据问题一的要求，当待观测天体 S 位于基准球面正上方，即 $\alpha=0^\circ$, $\beta=90^\circ$ 时，可得旋转矩阵 A_1 和 A_2 为：

$$A_1 = A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

将式(9)代入式(8)可以求出标准状态下抛物面的方程：

$$z = \frac{1}{4a}(x^2 + y^2) - (a + 0.534R) \quad (10)$$

其中， a 表示焦距， R 表示球的半径。

在理想抛物面焦点确定而焦距不确定的情况下，抛物面存在多种可能。假设当焦点固定在 P 点，而抛物线顶点位于球面上，即焦距固定 $a=0.466R$ 时，该抛物线方程为：

$$z = 1.786 \times 10^{-3}(x^2 + y^2) - R$$

此时，如下图所示，该理想抛物线上的点到基准球面的径向距离大于 0.6。

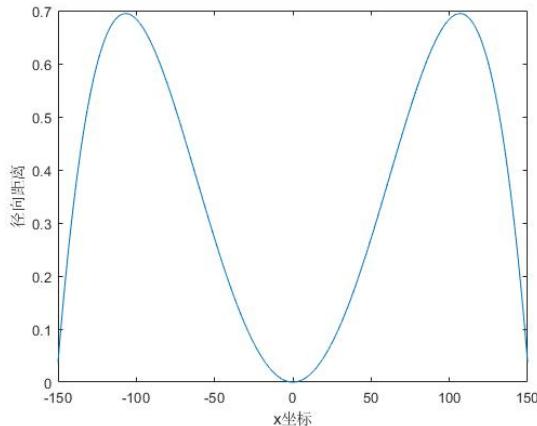


图 3：抛物线上点到原点距离

所以，结合题意需要考虑反射面板调节因素。由于促动器的伸缩存在一定的范围，主索节点的变化幅度也存在相应的限制，因此，在确定理想抛物面时，应当使该理想抛物面与基准球面尽可能贴近，使其尽量满足实际反射面板的调节因素。从而理想抛物线需要进行优化。

5.2 理想抛物面模型的优化

由于该理想抛物面为标准状态，其对称轴为经过球心 C 的 z 轴，故该抛物面可看作为抛物线绕 z 轴旋转形成的几何体。则沿对称轴 z 的任意平面与抛物面所截成的截面，各截面的形状和面积均相同。

取平面 XOZ ，截取该理想抛物面的平面图，如下所示：

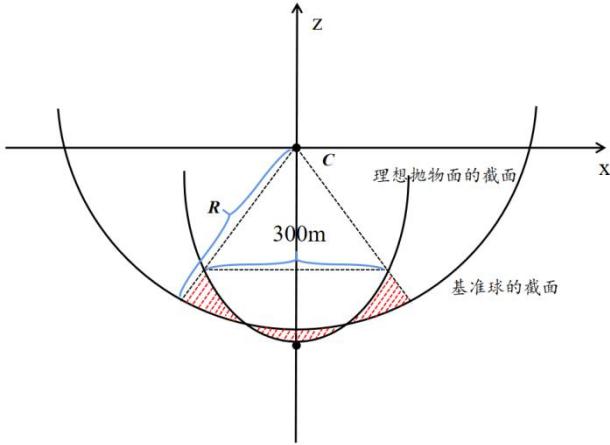


图 4：理想抛物面和基准球面的截面

当平面沿对称轴截取该抛物面时，截取到的抛物面的面积最大，且该抛物面的截取面积与基准球面的截取面积，在口径 $300m$ 的范围内形成的阴影部分面积(如图3)也最大。

当理想抛物面的焦点固定，而焦距不确定时，为使工作抛物面越贴近理想抛物面，该理想抛物面应尽可能与球面贴近，可以用抛物面上的点到球面的径向距离之和最短来描述，即在口径 $300m$ 的范围内抛物线与基准球面形成的阴影部分面积最小，根据第一型曲线积分公式，建立关于理想抛物面方程的**目标函数**：

$$\min \int_L (\sqrt{x^2 + z^2} - R)^2 ds \quad (11)$$

其中，在截面 XOZ ， $z = f(x) = ax^2 - 0.534R - \frac{1}{4a}$ 。

由于抛物面的口径为 $300m$ ，所以该目标函数可以表示为：

$$\min \int_{-150}^{150} (\sqrt{x^2 + f^2(x)} - R)^2 \sqrt{1 + f'^2(x)} dx \quad (12)$$

根据题目要求，促动器径向伸缩范围为 $\pm 0.6m$ ，即抛物面上的点到基准球面的径向距离必须满足 $< 0.6m$ 。在截面 XOZ 中，该约束条件可以表示为：

$$|\sqrt{x^2 + z^2} - R| \leq 0.6 \quad (13)$$

其中， (x, z) 表示在截面 XOZ 中的抛物面的点的坐标； R 表示基准球的半径。

已知抛物面焦点在与基准球面同心的一个球面上移动，所以焦距的大小必定小于基准球面的半径，即：

$$0 < a < R \quad (14)$$

综上所述，当待观测天体 S 位于基准球面正上方，即 $\alpha=0^\circ$, $\beta=90^\circ$ 时，建立起以焦距 a 为决策变量，阴影部分面积最小为目标函数，抛物面上的点到基准球面的径向距离等为约束条件的单目标优化模型：

目标函数： $\min \int_{-150}^{150} (\sqrt{x^2 + f^2(x)} - R)^2 \sqrt{1 + f'^2(x)} dx$

决策变量： a

约束条件:

$$s.t. \begin{cases} \left| \sqrt{x^2 + z^2} - R \right| \leq 0.6 \\ 0 < a < R \\ z = f(x) = ax^2 - 0.534R - \frac{1}{4a} \end{cases}$$

5.3 理想抛物面优化模型的求解

由题意可得, 基准态下, 所有主索节点均位于基准球面上。所以, 根据附件 1 中主索节点的坐标数据可知, 绝大部分主索节点到基准球圆心的距离在 300.4m 上下小范围内波动, 因此, 我们可以近似认为, 可以求出基准球的半径 $R=300.4m$ 。

针对目标函数, 由于无法求出其解析解, 本文采用数值求解法。在 Matlab 中对决策变量 a 进行遍历。已知约束条件 $a \in (0, R)$, 当 $a=0.466R=139.9864$ 时, 抛物面的顶点位于球面上。以此值为 a 的初始值, 步长为 7.8×10^{-4} , 在初始值的左右进行遍历。

找出所有满足约束条件 $\left| \sqrt{x^2 + z^2} - R \right| \leq 0.6$ 的 a 值, 并比较在满足约束条件下, 目标函数的大小。受篇幅限部分结果如下表所示:

表 1: 求解结果表

顶点到圆心距离	最大径向距离	目标函数	a
300.7999	0.4008	17.4406	140.3863
300.7921	0.3921	17.2802	140.3785
300.7842	0.3842	17.1624	140.3706
300.7763	0.3763	17.0873	140.3627
300.7684	0.3684	17.0549	140.3548
300.7605	0.3605	17.0652	140.3469
300.7527	0.3527	17.1181	140.3391
300.7448	0.3448	17.2137	140.3312
300.7369	0.3369	17.3519	140.3233

由上表可知当焦距 $a=140.3548$ 时, 目标函数达到最小值 17.0549, 此时最大的径向距离为 0.3684, 抛物线的顶点坐标为 $(0, 0, -300.7684)$ 。

在满足约束条件的求解结果中, 以顶点到球心的距离为横坐标, 阴影部分面积为纵坐标, 绘制曲线; 同时, 求出最佳理想抛物面下的各点径向距离, 结果如下图所示:

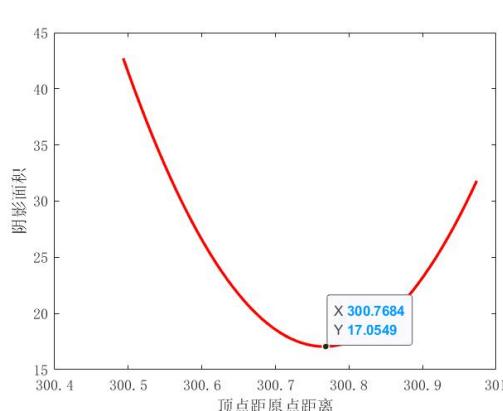


图 5: 顶点位置与阴影面积关系

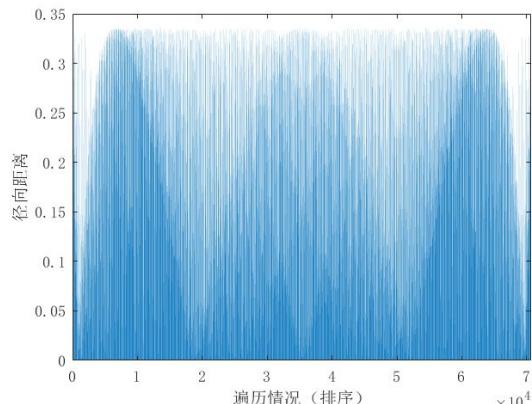


图 6: 径向距离结果

由图 6 可知, 阴影部分面积随顶点位置先逐渐减小, 在顶点坐标为 $(0, 0, -300.7684)$ 取到最小值, 然后又逐渐增大, 与实际相符。图 6 显示, 最佳理想抛物面各点到基准球面的径向距离均满足约束条件, 在 0.6 范围内, 符合题意。

5.4 问题一的结果

当待观测天体 S 位于基准球面正上方，即 $\alpha=0^\circ$, $\beta=90^\circ$ 时，结合考虑反射面板调节因素，得出理想曲面的顶点坐标为 $(0, 0, -300.7684)$ ，焦距为 140.3548，解析式为：

$$z = 1.7812 \times 10^{-3} (x^2 + y^2) - 20.0588$$

三维图如下所示：

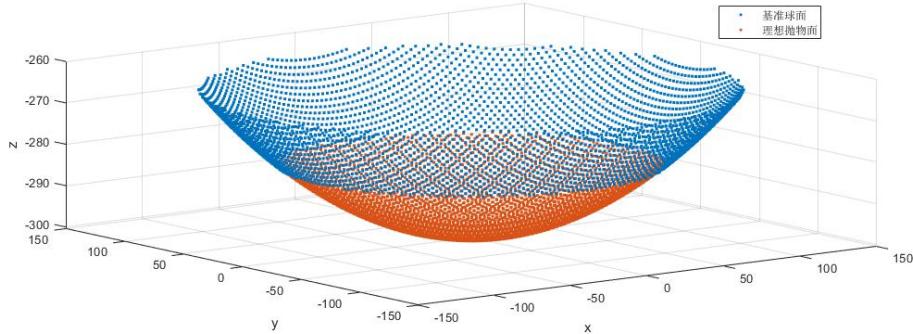


图 7：理想抛物面和基准球面的三维示意图

六、问题二的建模与求解

6.1 确定理想抛物面

问题一已经求出天体 S 位于基准球面正上方时的最佳理想抛物面，由于抛物面的焦点始终固定且抛物面绕着球心转动，抛物面的焦面与基准球面共球心。所以该抛物面无论如何旋转，在任何角度或方向上沿对称轴所截后形成的抛物面与基准球面的阴影部分面积始终最小，抛物面上的点到球面的径向距离最短。

所以，本问中只需将问题一求出的最佳理想抛物面旋转至问题二所需位置，即可确定本问的最佳理想抛物面。

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A_2 \cdot (A_1 \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}) \quad (15)$$

其中， A_1 表示抛物面关于仰角 β 的旋转矩阵， A_2 表示抛物面关于方位角 α 的旋转矩阵； (x_1, y_1, z_1) 表示问题一求出的最佳理想抛物面上的点的坐标。

将本题中的方位角 $\alpha=36.795^\circ$ ，仰角 $\beta=78.169^\circ$ 分别代入旋转矩阵中，求出理想抛物面的顶点为 $(-49.369, -36.926, -294.312)$ ，该理想抛物面的三维示意图如下所示：

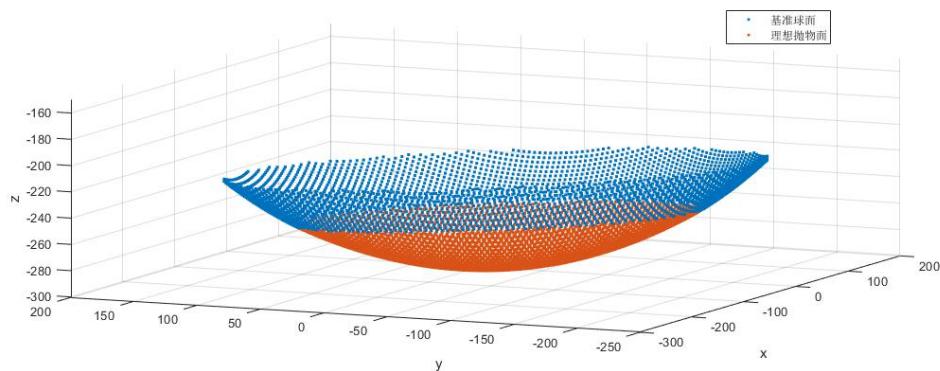


图 8：理想抛物面和基准球面的三维示意图

6.2 反射面调节模型的建立

1. 主索节点的选定

在口径为 $300m$ 的抛物面中，当主索节点 Q 位于抛物面口径边缘上时，其离抛物面顶点的距离最远，其位置关系如下图所示：

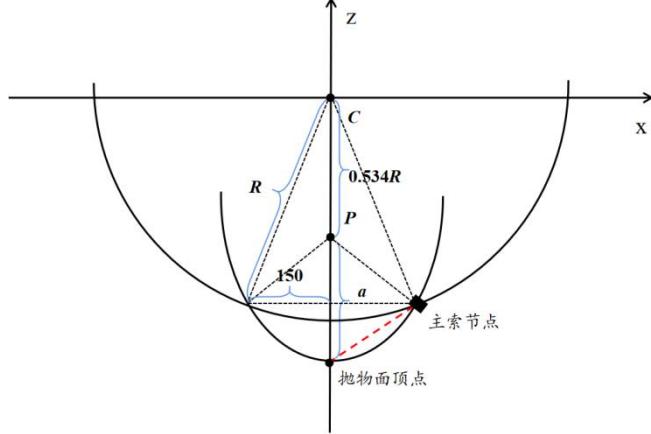


图 9：主索节点与顶点的位置示意图

由上图 7 可知，口径内最远主索节点与抛物线顶点的距离 D 可以表示为：

$$D^2 = (0.534R + a - \sqrt{R^2 - 150^2})^2 + 150^2 \quad (16)$$

其中， R 表示基准球的半径， a 表示抛物面的焦距。

所以，根据最远主索节点与抛物线顶点的距离，抛物面口径内所包含的主索节点坐标 (x_j, y_j, z_j) 均需满足：

$$(x_j - x_0)^2 + (y_j - y_0)^2 + (z_j - z_0)^2 \leq D^2 \quad (17)$$

其中， (x_0, y_0, z_0) 表示抛物面顶点的坐标； j 表示主索节点的编号， $j=1,2,3,\dots,2226$ 。

2. 促动器的伸缩

假设下拉索与促动器为刚性连接，不发生相对位移。已知下拉索长度固定，促动器沿基准球面径向安装。所以，促动器顶端通过径向伸缩下拉锁，使主索节点沿促动器上下端点所在的直线方向移动，即促动器上下端点与主索节点共线。

设促动器的下端点为 A ，上端点为 B ，主索节点为 Q ，促动器的伸缩量为 d ，则主索节点的移动可以用向量表示为：

$$\vec{Q}_j - \vec{Q}'_j = \overrightarrow{Q_j Q'_j} \quad (18)$$

其中， \vec{Q}_j 表示第 j 个主索节点， \vec{Q}'_j 表示第 j 个主索节点移动后的点； $j=1,2,3,\dots,2226$ 。

由于主索节点的移动方向即为促动器上下端点 $A_j B_j$ 所在直线的方向，则向量 $\overrightarrow{A_j B_j} \parallel \overrightarrow{Q_j Q'_j}$ ，且 $\overrightarrow{Q_j Q'_j}$ 可表示为：

$$\overrightarrow{Q_j Q'_j} = d_j \cdot \frac{\overrightarrow{A_j B_j}}{\| \overrightarrow{A_j B_j} \|} \quad (19)$$

其中， d_j 表示第 j 个主索节点沿径向的伸缩量； $\frac{\overrightarrow{A_j B_j}}{\| \overrightarrow{A_j B_j} \|}$ 表示直线 $A_j B_j$ 方向上的单位向量。

3.反射面调节优化模型

取调节后第 i 块反射面上的 3 个主索节点记为 Q_{i1} , Q_{i2} 和 Q_{i3} 。已知该三点的坐标, $Q_{i1}(x_{i1}, y_{i1}, z_{i1})$, $Q_{i2}(x_{i2}, y_{i2}, z_{i2})$, $Q_{i3}(x_{i3}, y_{i3}, z_{i3})$, 则可以确定该反射面的方程为:

$$A_i x + B_i y + C_i z + D_i = 0 \quad (20)$$

其中,

$$\begin{cases} A_i = (y_{i3} - y_{i1}) \cdot (z_{i3} - z_{i1}) - (z_{i2} - z_{i1}) \cdot (y_{i3} - y_{i1}) \\ B_i = (x_{i3} - x_{i1}) \cdot (z_{i2} - z_{i1}) - (x_{i2} - x_{i1}) \cdot (z_{i3} - z_{i1}) \\ C_i = (x_{i2} - x_{i1}) \cdot (y_{i3} - y_{i1}) - (x_{i3} - x_{i1}) \cdot (y_{i2} - y_{i1}) \\ D_i = -(A_i \cdot x_1 + B_i \cdot y_1 + C_i \cdot z_1) \end{cases}$$

在第 i 块反射面内取一点 $M(x, y, z)$, 其沿径向伸缩与抛物面交于 $N(x', y', z')$, 则 M 点的径向距离可表示为:

$$|MN| = d(x, y, z) \quad (21)$$

其中, $(x, y, z) \in$ 第 i 块反射面板。

将第 i 块反射面上的所有点到抛物面上点的径向距离进行积分, 以每块反射板与抛物面围成的径向体积之和最小为目标函数:

$$\min \sum_{i=1}^n \left(\iint_{S_i} d(x, y, z) ds \right) \quad (22)$$

由于促动器的伸缩范围在 $\pm 0.6m$, 所以主索节点径向伸缩量 d_j 必须在 $\pm 0.6m$ 范围内:

$$|d_j| \leq 0.6 \quad (23)$$

根据题意, 主索节点调节后, 相邻节点之间的距离可能会发生微小变化, 变化幅度不超过 0.07%, 定义 l_{ij} 表示第 i 块反射板上第 j 个主索节点与相邻节点的距离:

$$\frac{|l_{ij} - l'_{ij}|}{l_{ij}} \leq 0.07\% \quad (24)$$

综上所述, 建立起以径向移动距离 d_j 为决策变量, 每块反射板与抛物面围成的径向体积之和最小为目标函数, 抛物面上的点到基准球面的径向距离等为约束条件的单目标优化模型:

目标函数:

$$\min \sum_{i=1}^n \left(\iint_{S_i} d(x, y, z) ds \right)$$

决策变量:

$$d_j$$

约束条件:

$$s.t. \begin{cases} |d_j| \leq 0.6 \\ \frac{|l_{ij} - l'_{ij}|}{l_{ij}} \leq 0.07\% \\ (x_j - x_0)^2 + (y_j - y_0)^2 + (z_j - z_0)^2 \leq D^2 \end{cases}$$

6.2 反射面调节优化模型的求解

1. 积分离散化

在反射面调节的优化模型中，目标函数为 $\min \sum_{i=1}^n (\iint_{S_i} d(x, y, z) ds)$ 。其中包含连续积分的部分，本文采用离散形式对其进行数值求解，对每个离散的体积，在求解过程中遍历的底面积 S 均相等。因此，本文的目标函数可转为求反射面上的点到理想抛物面的径向距离最短，即：

$$\min \sum_{i=1}^n \sum_{j=1}^m d_{ij} \quad (25)$$

其中， d_{ij} 表示第 i 块反射面上第 j 个点到理想抛物面的径向距离。

2. 取特征点

为简化计算，在每一块反射面上取 7 个特征点，用以代表整块平面到抛物面的径向距离。该 7 个特征点分别为 3 个主索节点、3 个相邻主索节点连线的中点及三角反射平面的重心。

3. 算法流程

Step1. 调整至较贴合初始状态

计算第 j 个主索节点 Q_j 沿径向变化至理想抛物面所需要的伸缩量 $d_j^{(1)}$ ，变化后的主索节点记作 Q'_j 。得到伸缩矩阵，使促动器进行伸缩调整。

Step2. 同步推动所有主索节点

由于所需优化的主索节点数量较多，难以同时优化，并且同步推动主索节点可以使得经过上一步调整至主索节点贴合理想抛物线的情况得到保留，使抛物面整体形态更稳定。此时遍历所有可能情况，求取最佳情况。

Step3. 观察结果

得到伸缩量矩阵、调整后节点坐标和抛物面顶点坐标，分析结果并放入附件 4 中。

6.3 问题二的结果

根据上述算法流程，调整后的伸缩量与平均径向距离的结果如下图所示：

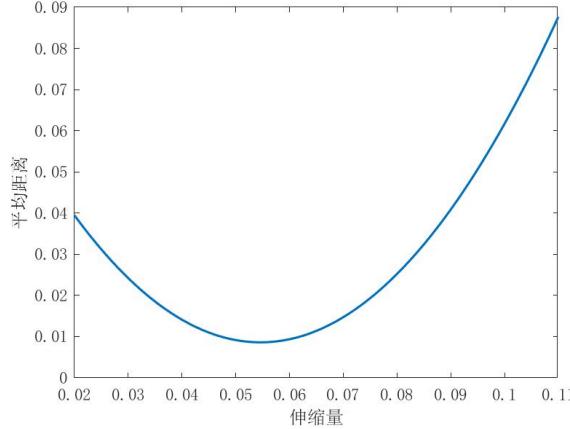


图 10：调节后伸缩量与平均径向距离的关系图

经过计算，贴合后再度调节伸缩量为 **0.055** 时，得到平均径向距离的最小值为 **0.0087m**。

七、问题三的建模与求解

7.1 电磁波入射与反射模型

为了方便分析与计算，将问题二中工作抛物面口径内所有主索节点的坐标乘以旋转矩阵的逆矩阵：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A_1^{-1} \cdot (A_2^{-1} \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}) \quad (26)$$

其中， (x', y', z') 表示问题二中主索节点的坐标。

旋转后抛物面的形状如下图所示：

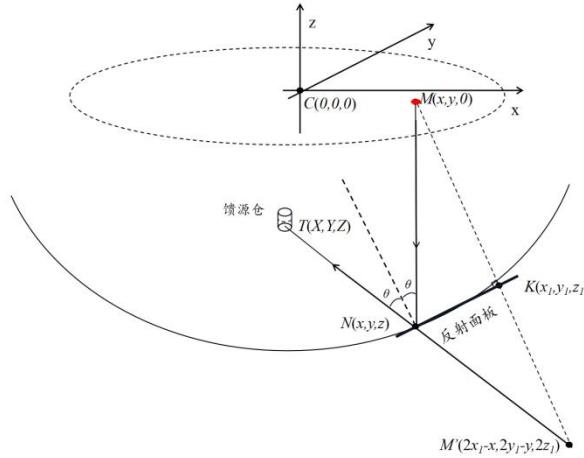


图 11：入射信号与反射信号示意图

在平面 XOY 内确定一个圆，圆方程满足条件：

$$\begin{cases} x^2 + y^2 \leq 150 \\ z = 0 \end{cases} \quad (27)$$

在圆内随机确定一点 $M(x, y, 0)$, 直线 M 以竖直方向入射反射面并交点于 $N(x, y, z)$ 。设平面 N 的方程:

$$Ax + By + Cz + D = 0 \quad (28)$$

其中, $z = -\frac{Ax + By + D}{C}$ 。

点 M 关于平面 N 的对称点为 M' , 直线 MM' 与平面 N 相交于 $K(x_1, y_1, z_1)$ 。设该平面的法向量 $\vec{\tau} = (A, B, C)$, 则 $\overrightarrow{MK} \parallel \vec{\tau}$, 可得:

$$\frac{x_1 - x}{A} = \frac{y_1 - y}{B} = \frac{z_1 - z}{C} = t \quad (29)$$

联立式(27)(28), 可得:

$$t = -\frac{Ax + By + D}{A^2 + B^2 + C^2} \quad (30)$$

设反射光线 $M'N$ 经过馈源舱接受平面, 交于点 $T(X, Y, Z)$ 。由于 \overrightarrow{TN} 与 $\overrightarrow{NM'}$ 共线, 可得:

$$\frac{X - x}{2(x_1 - x)} = \frac{Y - y}{2(y_1 - y)} = \frac{Z - z}{2z_1 - z} \quad (31)$$

解得:

$$\begin{cases} X = -2At \cdot \frac{0.534R + z}{2ct - z} + x \\ Y = -2Bt \cdot \frac{0.534R + z}{2ct - z} + y \\ Z = -0.534R \end{cases} \quad (32)$$

7.2 馈源舱的接收比

定义函数 I , 表示反射光线是否经过馈源舱:

$$I(x, y) = \begin{cases} 1, & x^2 + y^2 \leq \frac{1}{4} \\ 0, & x^2 + y^2 > \frac{1}{4} \end{cases} \quad (33)$$

由题意可得, 接收比 δ 等于馈源舱有效区域接收到的反射信号与 300m 口径内反射面的反射信号之比, 所以接收比可表示为馈源舱有效区域内接收到的反射信号的面积与 300m 口径的面积之比:

$$\delta = \frac{\iint_D I(x, y) dx dy}{\pi R^2} \quad (34)$$

其中, $R=150m$ 。

7.3 问题三模型求解

在圆面上随机选取一点, 以 [-150, 150] 为搜索范围, 取步长为 3 进行遍历。按照上述

建模过程，当信号入射时，首先需要判断其是否在反射面板内。我们采用海伦公式 $S=\sqrt{p(p-a)(p-b)(p-c)}$ ，利用反射面板主索节点的坐标，计算三边的距离及面积。再根据入射点将三角形分成 3 个小三角形，检验这三个小三角形面积之和是否等于上一步计算的面积。若两次计算的三角形面积相等，说明入射点位于该平面内。

当平面确定时，可以确定法线和入射角，则该信号的反射便可以表达。综上所述，依次选取了 484 个点，结果如下所示：

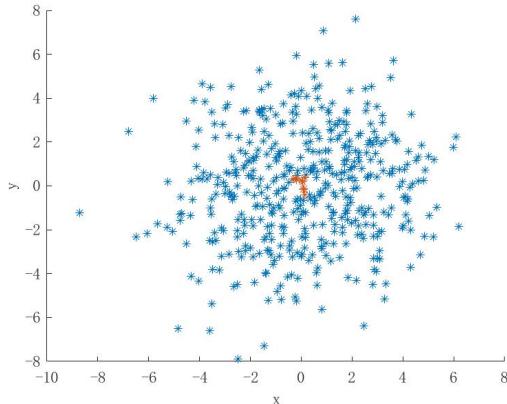


图 12：调节后反射面 484 个点的反射情况

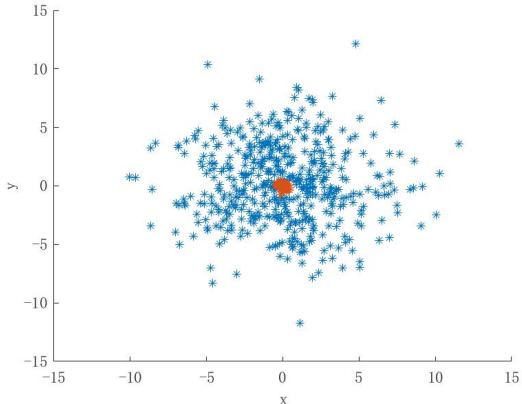


图 13：基准球面 600 个点的反射情况

如上图所示，调节后反射面反射的 484 个点中，有 9 个点投射到馈源舱的有效区域，**投射比约为 2%**；基准球面反射的 600 个点中，有 93 个点投射到馈源舱的有效区域，**投射比约为 15.5%**。

比较两者的投射比，存在较明显的差异，分析原因，我们认为反射面调节优化模型存在一定的欠缺，使得在调节后不如基准球面的投射比，这也侧面反映了“中国天眼”FAST 主动反射面在设计时精细程度之高。对于本科生而言，在规定时间内难以做到比较合理的答案。

八、模型评价

8.1 模型的优点

- 1、采用连续性模型建模，具有较好的普适性。
- 2、问题一模型经过多个初步模型的比较后最终筛选而成，具有相对的稳健性与严密性。
- 3、问题一模型引入旋转矩阵，能方便表示抛物面上的点旋转后的坐标。
- 4、问题一、二、三模型均通过离散化求解，问题二在计算时选取了具有代表性的特殊点，减小了计算量。

8.2 模型的改进

- 遍历精度改进

在第三问的模型中，我们在馈源仓的有效区域内划分了 484 个小网格，得到的有效接收比约为，如果我们进一步增加遍历精度，则理论上得到有效接收比会更小，其原理可解释如下：

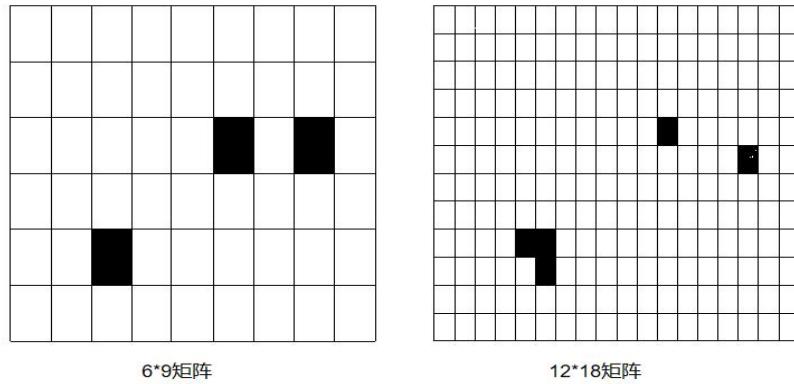


图 14：不同精细程度网格划分接收示意图

对于将区域划分为 6×9 个网格的情况，从图中可以看出，一共有 3 块网格的入射光线能够被馈源仓接收，接收比为 $3 / (6 \times 9) = 5.56\%$ ，而对于将区域划分为 12×18 个网格的情况，一共有 5 块网格的入射光线能够被馈源仓接收，接收比为 $5 / (12 \times 18) = 2.31\%$ ，这意味着，在从精细步长到粗糙步长的变化过程中，一些原本不能反射到馈源仓的网格位置被一并归入能被馈源仓接收的入射光网格中，从而使得实际接收比大于理论接收比。因此，在问题三的模型基础上，如果适当提高模型求解时遍历的精度，可以得到更小并且更接近于实际的接收比。

九、参考文献

- [1] 张承民,崔翔翰,杨佚沿,王德华.从阿雷西博到中国天眼射电望远镜:工业革命的接力[J].科技导报,2021,39(11):9-15.
- [2] 钱宏亮. FAST 主动反射面支承结构理论与试验研究[D].哈尔滨工业大学,2007.

十、附录

```
%% 附件数据分析
clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
data1 = readmatrix('附件 1.csv','OutputType','string');
data2 = readmatrix('附件 2.csv','OutputType','string');
data3 = readmatrix('附件 3.csv','OutputType','string');
dots_loc_str = data1(:,2:4);
c_d_q_str = data2(:,2:7);
tri_dots_num = data3(2:end,:);
for i = 1:size(c_d_q_str,2)
    c_d_q(:,i) = str2num(char(c_d_q_str(:,i))); % 上端点在前, 下端点在后
end
for i = 1:size(dots_loc_str,2)
    dots_loc(:,i) = str2num(char(dots_loc_str(:,i)));
end
dots_num = data1(:,1);
clearvars -except tri_dots_num c_d_q dots_loc dots_num
%% 真实的 R
clc,close all
R_real = (sum((dots_loc).^(2,2)).^(1/2)); % 300.4!
%% 下拉索夹角 (10 个异常值)
clc,close all
v1 = [c_d_q(:,1),c_d_q(:,2),c_d_q(:,3)] - [c_d_q(:,4),c_d_q(:,5),c_d_q(:,6)];
v2 = [c_d_q(:,1),c_d_q(:,2),c_d_q(:,3)] - [dots_loc(:,1),dots_loc(:,2),dots_loc(:,3)];
theta = acos(sum(v1.*v2,2)./(sum(v1.^2,2).^(1/2).*sum(v2.^2,2).^(1/2)));
theta_sort = sort(theta);
dots_num(find(theta>10^-3))
%% 促动器夹角
clc,close all
v1 = [c_d_q(:,1),c_d_q(:,2),c_d_q(:,3)] - [c_d_q(:,4),c_d_q(:,5),c_d_q(:,6)];
v2 = [c_d_q(:,1),c_d_q(:,2),c_d_q(:,3)] - [0,0,0];
theta = acos(sum(v1.*v2,2)./(sum(v1.^2,2).^(1/2).*sum(v2.^2,2).^(1/2)));
theta_sort = sort(theta);
dots_num(find(theta>10^-3))
%% 化方程
clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
syms alpha beta i j a R
M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角
M1 * (M2 * [i; j; a*(i^2+j^2)]) - M1 * (M2 * [0; 0; R])
M2_2 = [sin(beta),0,cos(beta);0,1,0;-cos(beta),0,sin(beta)]; % 仰角
```

```

M1 * (M2_2 * [i; j; a*(i^2+j^2)]) - M1 * (M2_2 * [0; 0; R])
%% 逆
alpha = 36.795/180*pi;
beta = 78.169/180*pi;
M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角
a = [];
robact = (M2^-1 * (M1^-1 * a'))';

```

```

%% 问题一代码
%% 优化焦距
clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
R = 300.4;
minimum = inf;
%a0 = 1/(4*(0.466*R)); %0.0018 % 焦距的四分之一, 实际 139.9864
for a = 0.001778 : 10^-7 : 0.00179
    k = 1;
    for i = -150:150
        for j = -150:150
            if i^2+j^2 <= 150^2
                %A(k,:) = [i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)];
                D(k,1) = abs(sum(([i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)]).^2)^(1/2) - R);
                k = k+1;
            end
        end
    end
    max_d = max(D);
    if max_d < minimum
        minimum = max_d;
        %     sum_d = sum(D);
        %     if sum_d < minimum
        %         minimum = sum_d;
        opti_a = a;
    end
end
% 焦距结果
clc,clear D
minimum
a = opti_a;
k = 1;
for i = -150:150
    for j = -150:150

```

```

if i^2+j^2 <= 150^2
    x = i;
    y = j;
    z = a*(x^2 + y^2);
    A(k,:) = [i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)];
    D(k,1) = abs(sum(([i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)]) .^2) ^ (1/2) - R);
    k = k+1;
end
end
end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.')
figure
bar(sort(D))
figure
bar(D)
R * (1 - 0.466) + 1 / (4*a)
length(D(D>0.6))/length(D)

```

```

%% 优化焦点
clc,clear,close all
cd 'E:\% Documents\用户_2021\cumcm2021A'
minimum = inf;
for R = 299:0.01:301
    a = 1/(4*(0.466*R)); % 焦距的四分之一
    k = 1;
    for i = -150:150
        for j = -150:150
            if i^2+j^2 <= 150^2
                x = i;
                y = j;
                z = a*(x^2 + y^2);
                A(k,:) = [x,y,z] - [0,0,R];
                D(k,1) = abs(sum(([x,y,z] - [0,0,R]) .^2) ^ (1/2) - 300);
                k = k+1;
            end
        end
    end
    max_d = max(D);
    if max_d < minimum
        minimum = max_d;
    end
end

```

```

opti_R = R;
end
end
% 焦距结果
clc,clear D
R = opti_R;
k = 1;
for i = -150:150
    for j = -150:150
        if i^2+j^2 <= 150^2
            x = i;
            y = j;
            z = a*(x^2 + y^2);
            A(k,:) = [x,y,z] - [0,0,R];
            D(k,1) = abs(sum(([x,y,z] - [0,0,R]) .^2)^(1/2) - 300);
            num(k) = i^2+j^2;
            k = k+1;
        end
    end
end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.')
figure
bar(sort(D))
figure
bar(D)
x
length(D(D>0.6))/length(D)

%% 优化两个
clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
minimum = inf;
for R = 299.9:0.02:301
    for a = 0.8*(1/(4*(0.466*R))):(1/(4*(0.466*R)))/100:1.2*(1/(4*(0.466*R)))%1/(4*(0.466*R)); % 焦距的四分之一
        k = 1;
        for i = -150:150
            for j = -150:150
                if i^2+j^2 <= 150^2
                    x = i;
                    y = j;
                    z = a*(x^2 + y^2);
                    A(k,:) = [x,y,z] - [0,0,R];
                end
            end
        end
    end
end

```

```

D(k,1) = abs(sum(([x,y,z] - [0,0,R]) .^2)^(1/2) - 300);
k = k+1;
end
end
end

max_d = max(D);
if max_d < minimum
    minimum = max_d;
    opti_a = a;
    opti_R = R;
end

end
end
% 结果
clc,clear D
R = opti_R;
a = opti_a;
k = 1;
for i = -150:150
    for j = -150:150
        if i^2+j^2 <= 150^2
            x = i;
            y = j;
            z = a*(x^2 + y^2);
            A(k,:) = [x,y,z] - [0,0,R];
            D(k,1) = abs(sum(([x,y,z] - [0,0,R]) .^2)^(1/2) - 300);
            k = k+1;
        end
    end
end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.')
figure
bar(sort(D))
figure
bar(D)
length(D(D>0.6))/length(D)

%% 问题二代码
clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
data1 = readmatrix('附件 1.csv','OutputType','string');

```

```

data2 = readmatrix('附件 2.csv','OutputType','string');
data3 = readmatrix('附件 3.csv','OutputType','string');
dots_loc_str = data1(:,2:4);
c_d_q_str = data2(:,2:7);
tri_dots_num = data3(2:end,:);
for i = 1:size(c_d_q_str,2)
    c_d_q(:,i) = str2num(char(c_d_q_str(:,i))); % 上端点在前, 下端点在后
end
for i = 1:size(dots_loc_str,2)
    dots_loc(:,i) = str2num(char(dots_loc_str(:,i)));
end
dots_num = data1(:,1);
clearvars -except tri_dots_num c_d_q dots_loc dots_num

%% 问题二
%% 促动器伸缩: 单位向量
clc,close all
v = [c_d_q(:,4),c_d_q(:,5),c_d_q(:,6)] - [c_d_q(:,1),c_d_q(:,2),c_d_q(:,3)];
v_length = sum(v.^2,2).^(1/2);
v_i = v./v_length;
% figure
% scatter3(v_i(:,1),v_i(:,2),v_i(:,3),'*')

%% 确定主索节点
clc,close all
R = 300.4;
P1 = [-49.3693, -36.9262, -294.3121]; % 问题二抛物面顶点
R1 = (150^2 + (300.76 - (R^2 - 150^2)^(1/2))^2)^(1/2);
choose = [];choose_num = [];choose_v_i = [];
for i = 1:size(dots_loc,1)
    if (dots_loc(i,1)-P1(1))^2 + (dots_loc(i,2)-P1(2))^2 + (dots_loc(i,3)-P1(3))^2 < R1^2
        choose = [choose; dots_loc(i,:)];
        choose_num = [choose_num; dots_num(i,:)];
        choose_v_i = [choose_v_i; v_i(i,:)];
    end
end
% %%
% figure
% scatter3(choose(:,1),choose(:,2),choose(:,3),'!')
% axis([0 1000 95 105 0 100])

%% 伸缩量 s_s_l 数值解
clc,close all
R = 300.7; % 此处的 R 指顶点与圆心距离

```

```

a = 1/(4*(R - 0.534 * 300.4));
alpha = 36.795/180*pi;
beta = 78.169/180*pi;
M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角
delta_d = 0.001;
for i = 1 :size(choose,1)
    k = 1;
    for d = -0.6 : delta_d : 0.6
        move = choose(i,:)+ d .* choose_v_i(i,:);
        robact = (M2^-1 * (M1^-1 * move'));
        delta_z_2(k) = (robact(1)*sin(beta - pi/2) - R*cos(beta - pi/2) ...
            + a*cos(beta - pi/2)*(robact(1)^2 + robact(2)^2) - move(3))^2;
        k = k+1;
    end
    s_s_l(i) = -0.6 + (find(delta_z_2==min(delta_z_2))-1) * delta_d;
end
figure
bar(s_s_l)
%% 移动后点位置
clc,close all
choose_move = [];
for i = 1 : size(s_s_l,2)
    choose_move(i,:) = choose(i,:) + (s_s_l(i)-0.055) .* choose_v_i(i,:);
end
%% 筛出三角形、坐标 (慢)
A = zeros(size(tri_dots_num));
B = [];
for i = 1 : size(tri_dots_num,1)
    for j = 1 : size(tri_dots_num,2)
        for k = 1 : size(choose_num,1)
            if tri_dots_num(i,j) == choose_num(k)
                A(i,j) = 1;
                B(i,3*j-2:3*j) = choose_move(k,:);
            end
        end
    end
end
choose_tri = []; choose_tri_move_loc = [];
sum_a = sum(A,2);
for i = 1:size(A,1)
    if sum_a(i) == 3
        choose_tri = [choose_tri; tri_dots_num(i,:)];
        choose_tri_move_loc = [choose_tri_move_loc; B(i,:)];
    end
end

```

```

    end
end

%% 计算重心
z_d = [];
for i = 1:size(choose_tri_move_loc,1)
    for j = 1:3
        z_d(i,j) = ((choose_tri_move_loc(i,j)) + choose_tri_move_loc(i,j+3)) +
choose_tri_move_loc(i,j+6))/3;
    end
end

%% 计算中点距离
clc,close all
for i = 1:size(z_d,1)
    z_d_i(i,:) = z_d(i,:)/sum(z_d(i,:).^2)^(1/2); % 单位向量
end
R = 300.7; % 此处的 R 指顶点与圆心距离
a = 1/(4*(R - 0.534 * 300.4));
alpha = 36.795/180*pi;
beta = 78.169/180*pi;
M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角
delta_d = 0.001;
for i = 1 :size(z_d,1)
    k = 1;
    for d = -0.6 : delta_d : 0.6
        move = z_d(i,:) + d .* z_d_i(i,:);
        roback = (M2^-1 * (M1^-1 * move)');
        delta_z_2(k) = (roback(1)*sin(beta - pi/2) - R*cos(beta - pi/2) ...
        + a*cos(beta - pi/2)*(roback(1)^2 + roback(2)^2) - move(3))^2;
        k = k+1;
    end
    z_d_d(i) = -0.6 + (find(delta_z_2==min(delta_z_2)) - 1) * delta_d;
end
z_d_d = abs(z_d_d);
figure
bar(z_d_d)

%% 计算顶点中点
z_b1_d = []; z_b2_d = []; z_b3_d = [];
for i = 1:size(choose_tri_move_loc,1)
    for j = 1:3
        z_b1_d(i,j) = (choose_tri_move_loc(i,j) + choose_tri_move_loc(i,j+3))/2;
    end
    for j = 1:3

```

```

z_b2_d(i,j) = (choose_tri_move_loc(i,j) + choose_tri_move_loc(i,j+6))/2;
end
for j = 1:3
    z_b3_d(i,j) = (choose_tri_move_loc(i,j+3) + choose_tri_move_loc(i,j+6))/2;
end
end

%% 计算顶点中点距离
clc,close all
z_b1_d_d = jisuan(z_b1_d);
z_b2_d_d = jisuan(z_b2_d);
z_b3_d_d = jisuan(z_b3_d);

z_all_d = [z_d_d, z_b1_d_d, z_b2_d_d, z_b3_d_d];
figure
bar(z_all_d)
sum(z_all_d)/size(z_all_d,2)

save('ans_tui.mat','choose_move','s_s_l')

%% function
function z_d_d = jisuan(z_d)
    clc,close all
    for i = 1:size(z_d,1)
        z_d_i(i,:) = z_d(i,:)./sum(z_d(i,:).^2)^(1/2); % 单位向量
    end
    R = 300.7; % 此处的 R 指顶点与圆心距离
    a = 1/(4*(R - 0.534 * 300.4));
    alpha = 36.795/180*pi;
    beta = 78.169/180*pi;
    M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
    M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角
    delta_d = 0.001;
    for i = 1 :size(z_d,1)
        k = 1;
        for d = -0.6 : delta_d : 0.6
            move = z_d(i,:) + d .* z_d_i(i,:);
            roback = (M2^-1 * (M1^-1 * move'))';
            delta_z_2(k) = (roback(1)*sin(beta - pi/2) - R*cos(beta - pi/2) ...
                + a*cos(beta - pi/2)*(roback(1)^2 + roback(2)^2) - move(3))^2;
            k = k+1;
        end
        z_d_d(i) = -0.6 + (find(delta_z_2==min(delta_z_2)) - 1) * delta_d;
    end
end

```

```

z_d_d = abs(z_d_d);
end

%% 可以跑, 3min 一轮
clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
data1 = readmatrix('附件 1.csv','OutputType','string');
data2 = readmatrix('附件 2.csv','OutputType','string');
data3 = readmatrix('附件 3.csv','OutputType','string');
dots_loc_str = data1(:,2:4);
c_d_q_str = data2(:,2:7);
tri_dots_num = data3(2:end,:);
for i = 1:size(c_d_q_str,2)
    c_d_q(:,i) = str2num(char(c_d_q_str(:,i))); % 上端点在前, 下端点在后
end
for i = 1:size(dots_loc_str,2)
    dots_loc(:,i) = str2num(char(dots_loc_str(:,i)));
end
dots_num = data1(:,1);
clearvars -except tri_dots_num c_d_q dots_loc dots_num
% 促动器伸缩: 单位向量
v = [c_d_q(:,4),c_d_q(:,5),c_d_q(:,6)] - [c_d_q(:,1),c_d_q(:,2),c_d_q(:,3)];
v_length = sum(v.^2,2).^(1/2);
v_i = v./v_length;
% 确定主索节点
R = 300.4;
P1 = [-49.3693, -36.9262, -294.3121]; % 问题二抛物面顶点
R1 = (150^2 + (300.76 - (R^2 - 150^2)^(1/2))^2)^(1/2);
chose = []; chose_num = []; chose_v_i = [];
for i = 1:size(dots_loc,1)
    if (dots_loc(i,1)-P1(1))^2 + (dots_loc(i,2)-P1(2))^2 + (dots_loc(i,3)-P1(3))^2 < R1^2
        chose = [chose; dots_loc(i,:)];
        chose_num = [chose_num; dots_num(i,:)];
        chose_v_i = [chose_v_i; v_i(i,:)];
    end
end
% 筛出三角形、坐标 (慢: 13s)
A = zeros(size(tri_dots_num));
for i = 1 : size(tri_dots_num,1)
    for j = 1 : size(tri_dots_num,2)
        for k = 1 : size(chose_num,1)
            if tri_dots_num(i,j) == chose_num(k)
                A(i,j) = 1;
            end
        end
    end
end

```

```

        end
    end
end
chose_tri = [];
sum_a = sum(A,2);
for i = 1:size(A,1)
    if sum_a(i) == 3
        chose_tri = [chose_tri; tri_dots_num(i,:)];
    end
end

%% -----
%% 记录移动值
%% (循环时可以将临近点名称信息录入元胞, 加快速度)
for iii = 1:3
clc,close all
tic
for ii = 1:size(chose_num,1)
    % 寻找相邻三角形
    A = [];
    for i = 1:size(chose_tri,1)
        for j = 1:size(chose_tri,2)
            if chose_num(ii) == chose_tri(i,j)
                A(i,j) = 1;
            end
        end
    end
    x_1_d = [];
    sum_a = sum(A,2);
    for i = 1:size(A,1)
        if sum_a(i) == 1
            x_1_d = [x_1_d; chose_tri(i,:)];
        end
    end
    if size(x_1_d,1) < 2
        disp('相邻三角形小于 2 个, 舍去本次情况')
        continue
    end
    % 相邻点坐标
    x_1_d_loc = [];
    for i = 1:size(x_1_d,1)
        for j = 1:size(x_1_d,2)
            for k = 1 : size(chose_num,1)

```

```

        if x_1_d(i,j) == chose_num(k)
            x_1_d_loc(i,3*j-2:3*j) = chose(k,:);
        end
    end
end

% 计算中点
z_d = [];
for i = 1:size(x_1_d_loc,1)
    for j = 1:3
        z_d(i,j) = (x_1_d_loc(i,j) + x_1_d_loc(i,j+3) + x_1_d_loc(i,j+6))/3;
    end
end

% 计算边中点
z_b1_d = []; z_b2_d = []; z_b3_d = [];
for i = 1:size(x_1_d_loc,1)
    for j = 1:3
        z_b1_d(i,j) = (x_1_d_loc(i,j) + x_1_d_loc(i,j+3))/2;
    end
    for j = 1:3
        z_b2_d(i,j) = (x_1_d_loc(i,j) + x_1_d_loc(i,j+6))/2;
    end
    for j = 1:3
        z_b3_d(i,j) = (x_1_d_loc(i,j+3) + x_1_d_loc(i,j+6))/2;
    end
end

% 各点距离
z_all = [];
z_all = [chose(ii,:); z_d; z_b1_d; z_b2_d; z_b3_d];
z_all_d = jisuan(z_all);
D_ii(ii) = sum(z_all_d)/size(z_all_d,2);
disp(['————>',num2str(100*ii/size(chose_num,1)),'%'])
toc
end

% 移动
chose_move = [];
for i = 1 : size(D_ii,2)
    chose_move(i,:) = chose(i,:) + D_ii(i).* chose_v_i(i,:);
end

% 求一次平均距离
% 筛出三角形、坐标 (慢: 13s)
A = zeros(size(tri_dots_num));
B = [];
for i = 1 : size(tri_dots_num,1)

```

```

for j = 1 : size(tri_dots_num,2)
    for k = 1 : size(choose_num,1)
        if tri_dots_num(i,j) == choose_num(k)
            A(i,j) = 1;
            B(i,3*j-2:3*j) = choose_move(k,:);
        end
    end
end
chose_tri = []; chose_tri_move_loc = [];
sum_a = sum(A,2);
for i = 1:size(A,1)
    if sum_a(i) == 3
        chose_tri = [chose_tri; tri_dots_num(i,:)];
        chose_tri_move_loc = [chose_tri_move_loc; B(i,:)];
    end
end
% 计算中点
z_d = [];
for i = 1:size(chose_tri_move_loc,1)
    for j = 1:3
        z_d(i,j) = (chose_tri_move_loc(i,j) + chose_tri_move_loc(i,j+3) + chose_tri_move_loc(i,j+6))/3;
    end
end
% 计算顶点中点
z_b1_d = []; z_b2_d = []; z_b3_d = [];
for i = 1:size(chose_tri_move_loc,1)
    for j = 1:3
        z_b1_d(i,j) = (chose_tri_move_loc(i,j) + chose_tri_move_loc(i,j+3))/2;
    end
    for j = 1:3
        z_b2_d(i,j) = (chose_tri_move_loc(i,j) + chose_tri_move_loc(i,j+6))/2;
    end
    for j = 1:3
        z_b3_d(i,j) = (chose_tri_move_loc(i,j+3) + chose_tri_move_loc(i,j+6))/2;
    end
end
% 计算顶点中点距离
z_all = [];
z_all = [z_d; z_b1_d; z_b2_d; z_b3_d];
z_all_d = jisuan(z_all);

DDD(iii) = sum(z_all_d)/size(z_all_d,2);
D_ii_s(iii,:) = D_ii;

```

```

z_all_d_s(iii,:) = z_all_d;
% 下一轮
chose = chose_move;
end

%% -----
%% 点到抛物面径向距离计算函数
function z_d_d = jisuan(z_d)
    for i = 1:size(z_d,1)
        z_d_i(i,:) = z_d(i,:)/sum(z_d(i,:).^2)^(1/2); % 单位向量
    end
    R = 300.7; % 此处的 R 指顶点与圆心距离
    a = 1/(4*(R - 0.534 * 300.4));
    alpha = 36.795/180*pi;
    beta = 78.169/180*pi;
    M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
    M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角
    delta_d = 0.001;
    for i = 1 :size(z_d,1)
        k = 1;
        for d = -0.5 : delta_d : 0.5
            move = z_d(i,:) + d .* z_d_i(i,:);
            robact = (M2^-1 * (M1^-1 * move)');
            delta_z_2(k) = (robact(1)*sin(beta - pi/2) - R*cos(beta - pi/2) ...
                + a*cos(beta - pi/2)*(robact(1)^2 + robact(2)^2) - move(3))^2;
            k = k+1;
        end
        z_d_d(i) = -0.6 + (find(delta_z_2==min(delta_z_2)) - 1) * delta_d;
    end
    z_d_d = abs(z_d_d);
end

```

```

clc,clear,close all
cd 'E:\% Documents\用户\m 文件_2021\cumcm2021A'
data1 = readmatrix('附件 1.csv','OutputType','string');
data2 = readmatrix('附件 2.csv','OutputType','string');
data3 = readmatrix('附件 3.csv','OutputType','string');
dots_loc_str = data1(:,2:4);
c_d_q_str = data2(:,2:7);
tri_dots_num = data3(2:end,:);
for i = 1:size(c_d_q_str,2)
    c_d_q(:,i) = str2num(char(c_d_q_str(:,i))); % 上端点在前，下端点在后

```

```

end
for i = 1:size(dots_loc_str,2)
    dots_loc(:,i) = str2num(char(dots_loc_str(:,i)));
end
dots_num = data1(:,1);
clearvars -except tri_dots_num c_d_q dots_loc dots_num
% 确定主索节点
R = 300.4;
P1 = [-49.3693, -36.9262, -294.3121]; % 问题二抛物面顶点
R1 = (150^2 + (300.76 - (R^2 - 150^2)^(1/2))^2)^(1/2);
chose = []; chose_num = [];
for i = 1:size(dots_loc,1)
    if (dots_loc(i,1)-P1(1))^2 + (dots_loc(i,2)-P1(2))^2 + (dots_loc(i,3)-P1(3))^2 < R1^2
        chose = [chose; dots_loc(i,:)];
        chose_num = [chose_num; dots_num(i,:)];
    end
end
clear P1 R1
% 筛出三角形、坐标 (慢: 13s)
A = zeros(size(tri_dots_num));
for i = 1 : size(tri_dots_num,1)
    for j = 1 : size(tri_dots_num,2)
        for k = 1 : size(chose_num,1)
            if tri_dots_num(i,j) == chose_num(k)
                A(i,j) = 1;
            end
        end
    end
end
chose_tri = [];
sum_a = sum(A,2);
for i = 1:size(A,1)
    if sum_a(i) == 3
        chose_tri = [chose_tri; tri_dots_num(i,:)];
    end
end
load ans_2.mat
load ans_tui.mat

% chose_move = cell2mat(answers(20));
alpha = 36.795/180*pi;
beta = 78.169/180*pi;
M1 = [cos(alpha),-sin(alpha),0;sin(alpha),cos(alpha),0;0,0,1]; % 方位角
M2 = [cos(pi/2-beta),0,sin(pi/2-beta);0,1,0;-sin(pi/2-beta),0,cos(pi/2-beta)]; % 仰角

```

```

for i = 1:size(chose_move,1)
    roback = (M2^-1 * (M1^-1 * chose_move(i,:)))';
    chose_move(i,:) = roback;
end
% figure
% scatter3(chose_move(:,1),choose_move(:,2),choose_move(:,3))
choose_tri_loc = [];
for i = 1:size(choose_num,1)
    for j = 1:size(choose_tri,1)
        for k = 1:size(choose_tri,2)
            if choose_tri(j,k) == choose_num(i)
                choose_tri_loc(j,3*k-2:3*k) = choose_move(i,:);
            end
        end
    end
end

%% 第三題
clc
steps = 100;
delta_xy = 300/steps;
k = 1;
for i0 = 1:steps
    for j0 = 1:steps
        x = -150 + (i0-1) * delta_xy;
        y = -150 + (j0-1) * delta_xy;
        if x^2 +y^2 <= 150^2
            for i = 1:size(choose_tri_loc,1)
                if hailun_logic([x,y],choose_tri_loc(i,1:2),...
                    choose_tri_loc(i,4:5),choose_tri_loc(i,7:8)) == 1
                    [xx, yy] = find_pq(x,y,choose_tri_loc(i,1:3),...
                        choose_tri_loc(i,4:6),choose_tri_loc(i,7:9));
                    flash(k,:) = [xx, yy];
                    k = k + 1;
                end
            end
        end
    end
end
ds = sum(flash.^2,2);
ds_sort = sort(ds);

%%

```

```

k = 1;j = 1;
for i = 1:size(flash,1)
    if ds(i) < 0.25
        A1(k,:) = flash(i,:);
        k = k+1;
    else
        B(j,:) = flash(i,:);
        j = j+1;
    end
end
figure
hold on
scatter(B(:,1),B(:,2),'*')
scatter(A1(:,1),A1(:,2),'*')

xlabel('x');
ylabel('y');
set(gca,'FontName','宋体')
set(gca,'FontSize',11);
set(gcf,'Position',[500 300 550 400]);
time_now = datestr(now,15);
name_pic = convertCharsToStrings(['投影',time_now(1:2),time_now(4:5)]);
saveas(gcf, name_pic, 'png')

%%
function n = hailun_logic(A0,A1,A2,A3)
    if hailun(A1,A2,A3) == (hailun(A0,A2,A3) + hailun(A0,A1,A3) + hailun(A0,A1,A2))
        n = 1;
    else
        n = 0;
    end
end

function s = hailun(A1,A2,A3)
    a = (sum((A1 - A2).^2))^(1/2);
    b = (sum((A2 - A3).^2))^(1/2);
    c = (sum((A1 - A3).^2))^(1/2);
    p = (a+b+c)/2;
    s = (p*(p-a)*(p-b)*(p-c))^(1/2);
end

function [xx,yy] = find_pq(x,y,A1,A2,A3)
    R = 300.4;
    A = (A2(2)-A1(2))*(A3(3)-A1(3)) - (A3(2)-A1(2))*(A2(3)-A1(3));

```

```

B = (A2(3)-A1(3))*(A3(1)-A1(1)) - (A3(3)-A1(3))*(A2(1)-A1(1));
C = (A2(1)-A1(1))*(A3(2)-A1(2)) - (A3(1)-A1(1))*(A2(2)-A1(2));
D = -(A * A1(1) + B * A1(2) + C * A1(3));
z = -(A*x + B*y + D)/C;
t = -(A*x + B*y + D)/(A^2 + B^2 + C^2);
xx = (-2) * (0.534 * R + z)/(2*C*t - z) * A * t + x;
yy = (-2) * (0.534 * R + z)/(2*C*t - z) * B * t + y;
End

%% 作图程序
%% 初始抛物线
clc,clear,close all
a = 1;
k = 1;
for i = -1:0.01:1
    for j = -1:0.01:1
        if i^2+j^2 <= 1
            x = i;
            y = j;
            z = a*(x^2 + y^2);
            A(k,:) = [x,y,z];
            k = k+1;
        end
    end
end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.');

%% 旋转后抛物线
clc,clear,close all
alpha = 0;
beta = 0;
a = 1;
k = 1;
for i = -1:0.01:1
    for j = -1:0.01:1
        if i^2+j^2 <= 1
            x = cos(alpha)*(i*cos(beta - pi/2) - a*sin(beta - pi/2)*(i^2 + j^2)) - j*sin(alpha);
            y = j*cos(alpha) + sin(alpha)*(i*cos(beta - pi/2) - a*sin(beta - pi/2)*(i^2 + j^2));
            z = i*sin(beta - pi/2) + a*cos(beta - pi/2)*(i^2 + j^2);
            A(k,:) = [x,y,z];
            k = k+1;
        end
    end
end

```

```

end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.')
hold on
scatter3(-(a*cos(alpha)*sin(beta - pi/2))/4, -(a*sin(alpha)*sin(beta - pi/2))/4, (a*cos(beta - pi/2))/4, '*')

%% 旋转+平移后抛物线
clc,clear,close all
alpha = 0;
beta = pi/2;
a = 1;
k = 1;
R = 10;
for i = -1:0.01:1
    for j = -1:0.01:1
        if i^2+j^2 <= 1
            x = cos(alpha)*(i*cos(beta - pi/2) - a*sin(beta - pi/2)*(i^2 + j^2)) - j*sin(alpha) +
R*cos(alpha)*sin(beta - pi/2);
            y = j*cos(alpha) + sin(alpha)*(i*cos(beta - pi/2) - a*sin(beta - pi/2)*(i^2 + j^2)) +
R*sin(alpha)*sin(beta - pi/2);
            z = i*sin(beta - pi/2) - R*cos(beta - pi/2) + a*cos(beta - pi/2)*(i^2 + j^2);
            A(k,:) = [x,y,z];
            k = k+1;
        end
    end
end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.')
hold on
scatter3(0,0,0)

%% 问题二的图
clc,clear,close all
alpha = 36.795/180*pi;
beta = 78.169/180*pi;
% alpha = 0/180*pi;
% beta = 90/180*pi;
R = 300.7;
a = 1/(4*(0.466*R));
k = 1;
for i = -150:4:150
    for j = -150:4:150
        if i^2+j^2 <= 150^2
            x = cos(alpha)*(i*cos(beta - pi/2) - a*sin(beta - pi/2)*(i^2 + j^2)) - j*sin(alpha) +

```

```

R*cos(alpha)*sin(beta - pi/2);
y = j*cos(alpha) + sin(alpha)*(i*cos(beta - pi/2) - a*sin(beta - pi/2)*(i^2 + j^2)) +
R*sin(alpha)*sin(beta - pi/2);
z = i*sin(beta - pi/2) - R*cos(beta - pi/2) + a*cos(beta - pi/2)*(i^2 + j^2);
A(k,:) = [x,y,z];
k = k+1;
end
end
end
figure
scatter3(A(:,1),A(:,2),A(:,3),'.')
axis([-200 200 -200 200 -310 -200])
hold on
scatter3(0,0,0,'*')
scatter3(dots_loc(:,1),dots_loc(:,2),dots_loc(:,3),'*')
scatter3(c_d_q(:,1),c_d_q(:,2),c_d_q(:,3),'.')
scatter3(c_d_q(:,4),c_d_q(:,5),c_d_q(:,6),'.')
hold off
%% 凹下图
clc,close all
R = 300.4;
A = [];
for a = 0.001778 : 10^-7 : 0.00179
    k = 1;
    s1 = 0;
    s2 = 0;
    delta_x = 0.01;
    for i = -150:150
        for j = -150:150
            if i^2+j^2 <= 150^2
                x = i;
                y = j;
                z = a*(x^2 + y^2);
                D(k,1) = abs(sum(([i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)]) .^2)^(1/2) - R);
                k = k+1;
            end
        end
    end
    if max(D) <= 0.6
        for i = -150 : delta_x : 150
            s1 = s1 + (a * i^2 - (1/(4 * a) + 0.534 * R) + (R^2 - i^2)^(1/2))^2 * delta_x;
            s2 = s2 + ((i^2 + (a * i^2 - 0.534 * R - 1/(4*a))^2)^(1/2) - R)^2 * (1 + (2 * a * i)^2)^(1/2) *
delta_x;
        end
    end
end

```

```

A = [A;R * (1 - 0.466) + 1 / (4*a),max(D),s1,s2,a];
D = [];
end
end
figure
plot(A(:,1),A(:,2))
figure
plot(A(:,1),A(:,3))
hold on
plot(A(:,1),A(:,4),'linewidth',2,'color','r')
xlabel('顶点距原点距离');
ylabel('阴影面积');
set(gca,'FontName','宋体')
set(gca,'FontSize',11);
set(gcf,'Position',[500 300 550 400]);
[row,col] = find(A==min(A(:,4)));
%text(A(row,1),A(row,col),strcat('最小值位于',num2str(A(row,1))));
time_now = datestr(now,15);
name_pic = convertCharsToStrings(['凹下图',time_now(1:2),time_now(4:5)]);
saveas(gcf, name_pic, 'png')
%% 画图
clc,close all
a = A(row,5);
k = 1;
for i = -150:150
    for j = -150:150
        if i^2+j^2 <= 150^2
            x = i;
            y = j;
            z = a*(x^2 + y^2);
            A2(k,:) = [i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)];
            D(k,1) = abs(sum(([i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)]) .^2)^(1/2) - R);
            k = k+1;
        end
    end
end
figure
scatter3(A2(:,1),A2(:,2),A2(:,3),'.')
figure
bar(sort(D))
xlabel('遍历情况（排序） ');
ylabel('径向距离');
set(gca,'FontName','宋体')
set(gca,'FontSize',11);

```

```

set(gcf,'Position',[500 300 550 400]);
[row,col] = find(A==min(A(:,4)));
%text(A(row,1),A(row,col),strcat('最小值位于',num2str(A(row,1))));
time_now = datestr(now,15);
name_pic = convertCharsToStrings(['距离 bar',time_now(1:2),time_now(4:5)]);
saveas(gcf, name_pic, 'png')
figure
bar(D)
R * (1 - 0.466) + 1 / (4*a)
length(D(D>0.6))/length(D)
%% 凹下图
clc,close all
R = 300.4;
A = [];
for a = 0.001778 : 10^-7 : 0.00179
    k = 1;
    s1 = 0;
    s2 = 0;
    delta_x = 0.01;
    for i = -150:150
        for j = -150:150
            if i^2+j^2 <= 150^2
                x = i;
                y = j;
                z = a*(x^2 + y^2);
                D(k,1) = abs(sum(([i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)]) .^2) ^ (1/2) - R);
                k = k+1;
            end
        end
    end
    if max(D) <= 0.6
        for i = -150 : delta_x : 150
            s1 = s1 + (a * i^2 - (1/(4 * a) + 0.534 * R) + (R^2 - i^2)^(1/2))^2 * delta_x;
            s2 = s2 + ((i^2 + (a * i^2 - 0.534 * R - 1/(4*a))^2)^(1/2) - R)^2 * (1 + (2 * a * i)^2)^(1/2) * delta_x;
        end
        A = [A;R * (1 - 0.466) + 1 / (4*a),max(D),s1,s2,a];
        D = [];
    end
end
figure
plot(A(:,1),A(:,2))
figure
plot(A(:,1),A(:,3))

```

```

hold on
plot(A(:,1),A(:,4),'linewidth',2,'color','r')
xlabel('顶点距原点距离');
ylabel('阴影面积');
set(gca,'FontName','宋体')
set(gca,'FontSize',11);
set(gcf,'Position',[500 300 550 400]);
[row,col] = find(A==min(A(:,4)));
%text(A(row,1),A(row,col),strcat('最小值位于',num2str(A(row,1)))); 
time_now = datestr(now,15);
name_pic = convertCharsToStrings(['凹下图',time_now(1:2),time_now(4:5)]);
saveas(gcf, name_pic, 'png')
%% 画图
clc,close all
a = A(row,5);
k = 1;
for i = -150:150
    for j = -150:150
        if i^2+j^2 <= 150^2
            x = i;
            y = j;
            z = a*(x^2 + y^2);
            A2(k,:) = [i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)];
            D(k,1) = abs(sum(([i,j,a*(i^2 + j^2)] - [0,0,0.534*R+1/(4*a)]) .^2) ^ (1/2) - R);
            k = k+1;
        end
    end
end
figure
scatter3(A2(:,1),A2(:,2),A2(:,3),'.');

figure
bar(sort(D))
xlabel('遍历情况（排序） ');
ylabel('径向距离');
set(gca,'FontName','宋体')
set(gca,'FontSize',11);
set(gcf,'Position',[500 300 550 400]);
[row,col] = find(A==min(A(:,4)));
%text(A(row,1),A(row,col),strcat('最小值位于',num2str(A(row,1)))); 
time_now = datestr(now,15);
name_pic = convertCharsToStrings(['距离 bar',time_now(1:2),time_now(4:5)]);
saveas(gcf, name_pic, 'png')
figure
bar(D)

```

$R * (1 - 0.466) + 1 / (4*a)$
$\text{length}(D(D > 0.6)) / \text{length}(D)$