

# "FAST"工作抛物面的优化设计

## 摘要

本文基于 FAST 的工作原理,通过机理分析、坐标变换、非线性最小二乘优化等方法,建立了反射面板调节优化模型,并利用 BFGS 算法、蒙特卡洛积分算法等算法,对不同条件下反射光线吸收比率进行了研究。

问题一中,首先基于固定的仰角 $\beta$ 、观测目标 $S$ 、圆心 $C$ 和焦点 $P$ ,利用旋转抛物面的中心对称性,选取焦距作为自由度控制变量,构建在极坐标系下开口竖直向上的二维抛物线方程,得到不同偏转角度下原点到抛物线的距离,进而导出三维下的旋转抛物面方程。其次,以焦距为决策变量,将口径 300 米的抛物面作为积分域,将理想抛物面到原点的距离与基准球面半径差值平方作为被积函数进行积分作为最小化目标函数,建立了确定理想抛物面的优化模型。最后,使用二分法求得目标函数导函数在定义区间上的零点,得到理想抛物面焦距的精确值为 280.854,误差平方积分的最小值为 10.112。此时对应理想抛物面的解析式为 $z = (x + y)^2 / 561.708 - 300.841$ 。

问题二中,首先利用球坐标下不同轴线方向抛物面的旋转不变性,在原坐标系和问题一的坐标系之间建立了双向可逆的变换关系,得到了不同方位角下理想抛物面到原点的距离。其次,以主索节点的工作坐标和促动器的伸缩长度为决策变量,以积分域覆盖的主索节点到原点的距离与理想抛物面到原点的距离之差的平方和为最小化目标函数,分别考虑下拉索长度固定、相邻节点的距离变化幅度不超过 0.07%、促动器的伸缩范围在  $\pm 0.6\text{m}$  为约束条件,建立反射面板调节优化模型。最后,使用拉格朗日乘子法和 BFGS 算法进行求解,得到误差平方在抛物面口径上的积分的最小值为  $5.1353 \times 10^{-9}$ ,理想抛物线的顶点坐标为 $(-49.392, -36.943, -294.450)$ ,调节后反射面 300 米口径内的主索节点编号、位置坐标、各促动器的伸缩量等结果见文件 result.xlsx。

问题三中,首先通过旋转变换,将反射问题的倾斜入射光线转化为垂直入射光线。其次,通过求解线性方程组,依次确定出入射光线与三角形面板的相交判定式、交点坐标、三角形面板的法线向量,并利用光线垂直入射的性质,使用法线向量简化计算得到出射光线的方向角。再次,通过联立射线方程与馈源舱所在的目标高度,得出射线方程的步长和出射光到达目标高度时的坐标,并与馈源舱的有效区域进行比对,作为入射光线是否被有效接收的判定式。最后,将 300 米口径内实际接收区域作为积分域,将入射光线有效接收判别式作为被积函数,使用蒙特卡洛算法进行积分,得到有效接收的光源面积,并计算出调节前接收比 0.811%,调节后接收比 1.103%,提升了 36%,调节工作抛物面的过程使有效光源的光斑可以尽量完整地出现在每个三角形面板内(图 11)。

本文的特色在于将机理分析与非线性最小二乘优化相结合,并灵活采用二分法、BFGS 算法和蒙特卡洛积分算法进行求解,在大规模、高维且带有非线性约束的求解中仍然以接近二阶的速度收敛至最优解,为 FAST 在不同情况下的调节与设计提供了参考依据。

**关键词:** 坐标旋转、非线性最小二乘、BFGS 算法、蒙特卡洛积分



## 一、问题重述

### 1.1 问题背景

中国天眼（FAST）由主动反射面、馈源舱及其他系统组成，其中主动反射面是由主索网、反射面板、下拉索、促动器及支承结构等构成的可调节球面。主索网由柔性主索按短程线三角网格方式构成，每个主索节点连接一根下端与固定在地表的促动器连接的下拉索。促动器沿基准球面径向安装，可沿径向伸缩。

主动反射面有两个状态：基准态时反射面为半径约 300 米、口径为 500 米的球面；工作态时反射面为一个 300 米口径的近似旋转抛物面。馈源舱接收中心在与基准球面同心、半径差为  $\Delta R$  的一个球面上移动。当观测某个方向的目标  $S$  时，馈源舱接收中心被移动到直线  $SC$  与焦面的交点  $P$  处，调节基准球面上的反射板形成以直线  $SC$  为对称轴、以  $P$  为焦点的近似旋转抛物面，从而将来自  $S$  的平行电磁波汇聚到有效区域。

### 1.2 问题提出

在反射面板调节约束下，确定一个理想抛物面，调节促动器，使工作抛物面尽量贴近理想抛物面，以获得反射后的最佳接收效果。建立模型解决以下问题：

- 问题一：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，确定理想抛物面。
- 问题二：当待观测天体  $S$  位于  $\alpha = 36.765^\circ$ ,  $\beta = 78.169^\circ$  时，确定理想抛物面。建立反射面调节模型，调节相关促动器的伸缩量，使反射面尽量贴近该理想抛物面。
- 问题三：基于问题二方案，计算调节后馈源舱接收比，与基准球面接收比作比较。

## 二、问题假设

1. 假设光线在反射面不存在二次反射。
2. 假设不考虑板间间隙给反射带来的影响。
3. 假设电磁波在大气中的传播为直线传播。
4. 假设反射面板不会发生弯曲，始终维持一个平面。
5. 假设电磁波的反射为全反射，不考虑反射时的损耗。
6. 假设主索节点调节后，相邻节点之间的距离会发生微小变化，变化幅度不超过 0.07%。





### 三、符号说明

序号	符号	意义	单位
1	$L_f$	抛物线及旋转抛物面的焦距	$m$
2	$L_\omega$	从原点出发到理想抛物面的距离	$m$
3	$L_z(i)$	序号为 $i$ 的促动器的伸缩量	$m$
4	$L_s(i)$	序号为 $i$ 的下拉索的长度	$m$
5	$\mathcal{I}$	所有主索节点的下标集合	——
6	$\mathcal{J}$	一块反射面板所用到的下标集合	——
7	$E$	所有三角形的所有边的集合	——
8	$(x_0(i), y_0(i), z_0(i))$	序号为 $i$ 的主索节点的基准坐标	——
9	$(x(i), y(i), z(i))$	序号为 $i$ 的主索节点的工作坐标	——
10	$(x^-(i), y^-(i), z^-(i))$	序号为 $i$ 的促动器底端的坐标	——
11	$(x^+(i), y^+(i), z^+(i))$	序号为 $i$ 的促动器顶端的坐标	——
12	$(x^o(i), y^o(i), z^o(i))$	序号为 $i$ 的促动器顶端基准状态的坐标	——
13	$(x_w(i), y_w(i), z_w(i))$	序号为 $i$ 的主索节点旋转后的工作坐标	——

### 四、问题分析

#### 4.1 问题一的分析

问题一要求当 $\alpha = 0^\circ$ ,  $\beta = 90^\circ$ 时, 结合反射面板调节因素, 确定理想抛物面。

基于 $\beta = \pi/2$ , 且观测目标 $S$ 、圆心 $C$ 和焦点 $P$ 固定, 确定所求剖面抛物线的自由度只有 1, 注意到旋转抛物面的中心对称性, 故所求抛物面是其剖面上的抛物线的复合。故可利用焦距作最优化控制变量, 进而构建在直角坐标系下开口竖直向上的二维抛物线方程, 将其代入极坐标进行变换, 可导出旋转抛物面的方程。最后对理想抛物面到原点的距离与基准球面半径的差值平方进行积分, 并进行最小二乘最优化求解, 使得基准球面与最终确定的抛物面之间的总调节距离的平方和最小, 从而求出理想抛物面。

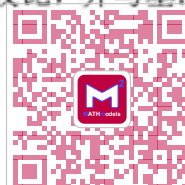
#### 4.2 问题二的分析

问题二要求当 $\alpha = 36.795^\circ$ ,  $\beta = 78.169^\circ$ 时, 确定理想抛物面。并建立反射面板调节模型, 调节相关促动器的伸缩量, 使反射面尽量贴近该理想抛物面。

注意到球坐标下不同轴线方向的抛物面为各同向性这一性质, 本文考虑在原有空间坐标系和问题一已建立的坐标系之间建立交互的旋转变换关系, 使得问题二中可部分沿用问题一中所得有关理想抛物面的数学关系, 进而可以确定本问的目标函数和决策变量。然后引入下拉索固定长度、节点之间的伸缩变动率、促动器的伸缩量等约束, 再引入相邻节点的距离变化幅度的约束和促动器伸缩范围的约束, 求出相关促动器的伸缩量优化解。

#### 4.3 问题三的分析

问题三要求基于问题二的反射面调节方案, 计算调节后馈源舱的接收比, 并与基准



球面接收比作比较。

本文拟采用与问题二相同的坐标旋转方法，将入射的光线从斜射变为垂直于水平面入射。接着，本文拟通过构建以一块反射面板上三个主索节点坐标为基准的参数方程，利用参数方程判断入射光与任意一块反射面板的几何关系，从而确定交点坐标。然后，本文拟利用待定系数法确定反射平面的法向量，从而利用向量方向角的关系确定反射光的指向。然后，可以利用方程联立求得的反射光与接收平面的交点判断该光线是否被有效接收。最终，可将入射光的照射区域作为积分域，是否接收到光线的判别式作为被积函数进行积分，将积出的结果与照射面积做比值，从而计算出馈源舱的接收比。

## 五、问题一模型的建立与求解

### 5.1 问题一模型的建立

#### 5.1.1 直角坐标系下二维抛物线方程的构建

题目要求给出考虑反射面板调节因素下的理想抛物面。根据描述，FAST 的抛物面由二维抛物线旋转得来，因此本文首先基于  $\beta = \pi/2$ ，讨论在二维平面内，抛物线开口方向竖直向上的抛物线方程，再通过旋转确定三维的旋转抛物面。

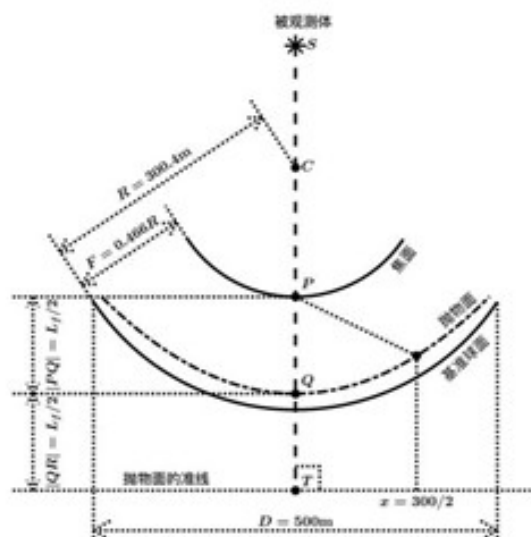


图 1: FAST 剖面示意图

本文基于题目附件 8 给出的空间直角坐标，以  $C$  为坐标原点，构建剖面图上的二维直角坐标系。则水平向右方向为  $x$  轴正向，竖直向上为  $z$  轴正向。图 1 为 FAST 的剖面示意图，其中抛物线的焦点为点  $P$ 。由于讨论的是开口方向竖直向上的抛物线，故其焦点  $P$  的坐标为：

$$(x_p, z_p) = (0, -(R - F)) \quad (5-1)$$

根据抛物线的几何性质，设该抛物线的焦距为  $L_f$ 。则焦点与顶点的距离为  $0.5L_f$ ，则该抛物线顶点  $Q$  坐标为：

$$(x_q, z_q) = (0, -(R - F) - 0.5L_f) \quad (5-2)$$

由于抛物线焦点与准线的距离为一倍焦距，故可得抛物线准线方程为：



$$z = -(R - F) - L_f \quad (5-3)$$

根据抛物线的几何性质，可知抛物线上的任意一点到准线的距离等于该点到焦点的距离。利用该性质构建抛物线方程：

$$(z - z_T)^2 = x^2 + (z - z_P)^2 \quad (5-4)$$

其中， $z_T = -(R - F) - L_f$  为点  $T$  的  $z$  轴坐标，由准线方程(5-3)得出；

$z_P = -(R - F)$  为点  $P$  的  $z$  轴坐标。

接着，将  $z_T$  和  $z_P$  的值代入式(5-4)，可得：

$$[z + (R - F) + L_f]^2 = x^2 + [z + (R - F)]^2 \quad (5-5)$$

两边平方项展开后，移项整理可得方程：

$$z = \frac{x^2}{2L_f} - \frac{L_f}{2} - (R - F) \quad (5-6)$$

以上，得到了二维直角坐标系下，开口竖直向上的抛物线方程。

### 5.1.2 极坐标系下二维抛物线方程的构建

由于需要将计算得到的抛物面与球面进行对比，使得工作抛物面尽量贴近理想抛物面。因此为方便求解，将上述抛物线方程转换成极坐标形式。

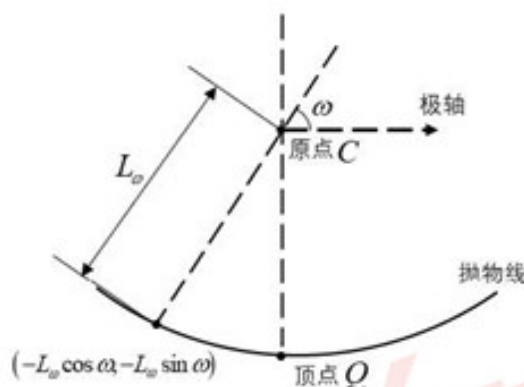


图 2：极坐标系下抛物线示意图

如图 2，以  $C$  点为原点，以原  $x$  轴为极轴，构建极坐标系，有：

$$(x, z) = (-L_\omega \cos \omega, -L_\omega \sin \omega) \quad (5-7)$$

其中， $L_\omega$  是从原点出发，到抛物线的距离。将式(5-7)代入直角坐标下的抛物线方程并整理可得：

$$\frac{L_\omega^2 \cos^2 \omega}{2L_f} + L_\omega \sin \omega - \frac{L_f}{2} - (R - F) = 0 \quad (5-8)$$

为了得到不同偏转角度  $\omega$  与  $L_\omega$  的表达式，需要对式(5-8)的方程进行求解。

(1) 当  $\omega \neq \pi/2$  时，可将式(5-9)看作以  $L_\omega$  为未知量的一元二次方程，故有





$$\Delta = \sin^2 \omega + 4 \frac{\cos^2 \omega}{2L_f} \left( \frac{L_f}{2} + R - F \right) > 0 \quad (5-9)$$

此时方程始终有两个不同的根。根据韦达定理：

$$L_{\omega 1} + L_{\omega 2} = - \frac{-2L_f \sin \omega}{\cos^2 \omega} = \frac{2L_f \sin \omega}{\cos^2 \omega} > 0 \quad (5-10)$$

$$L_{\omega 1} \cdot L_{\omega 2} = \frac{2L_f \left( -\frac{L_f}{2} - (R - F) \right)}{\cos^2 \omega} = \frac{-2L_f^2 - 2L_f(R - F)}{\cos^2 \omega} < 0 \quad (5-11)$$

故可知，此时对于每一个 $\omega$ ，都会产生取值一正一负的两个根。舍去负根，得到正根的解析式：

$$L_{\omega} = \frac{-L_f \sin \omega + L_f \sqrt{\sin^2 \omega + \frac{2\cos^2 \omega}{L_f} \left( \frac{L_f}{2} + R - F \right)}}{\cos^2 \omega} \quad (5-12)$$

(2) 当 $\omega = \pi/2$ 时，可将式(5-9)看作以 $L_{\omega}$ 为未知量的一元一次方程，故有：

$$L_{\omega} = \frac{L_f}{2\sin \omega} + \frac{R - F}{\sin \omega} \quad (5-13)$$

经过化简，可得基于偏转角度 $\omega$ 与 $L_{\omega}$ 的表达式：

$$L_{\omega} = \begin{cases} \frac{-L_f \sin \omega + L_f \sqrt{\sin^2 \omega + \frac{2\cos^2 \omega}{L_f} \left( \frac{L_f}{2} + R - F \right)}}{\cos^2 \omega}, & \omega \neq \frac{\pi}{2} \\ \frac{L_f}{2\sin \omega} + \frac{R - F}{\sin \omega}, & \omega = \frac{\pi}{2} \end{cases} \quad (5-14)$$

由于公式(5-14)中含有未知参数，则将该函数记为：

$$f(\omega, L_f) = \begin{cases} \frac{-L_f \sin \omega + L_f \sqrt{\sin^2 \omega + \frac{2\cos^2 \omega}{L_f} \left( \frac{L_f}{2} + R - F \right)}}{\cos^2 \omega}, & \omega \neq \frac{\pi}{2} \\ \frac{L_f}{2\sin \omega} + \frac{R - F}{\sin \omega}, & \omega = \frac{\pi}{2} \end{cases} \quad (5-15)$$

以上，得到了在不同的偏转角度下，二维抛物线上的点到原点的距离 $L_{\omega}$ 的表达式，接着将此二维抛物线进行旋转即可得到三维的旋转抛物面。

### 5.1.3 理想抛物面与基准球面的相似度计算

将 5.1.2 节得到的二维抛物线以极轴为中轴线旋转，得到反射面板的旋转抛物面，该抛物面可使得沿着中轴线平行射入的电磁波可以被反射到焦点 $P$ 处。为比较这一理想抛物面与基准球面的相似程度，需要在球坐标系下，以口径为 300 米的抛物面在基准球面上的投影所围成的区域为积分域，对理想抛物面到原点的距离与基准球面半径的差值平方进行积分，即：



$$\iint_A [f(\omega, L_f) - R]^2 d\sigma \quad (5-16)$$

积分数值的大小是评价相似程度的准则。其中， $R$  为基准球面半径； $\omega$  为抛物面上的点与原点连线和中垂线所形成的空间角的大小； $d\sigma$  为曲面微元； $A$  为积分域，可表示如下：

$$A: \{(x, y, z) | z = -\sqrt{R^2 - x^2 - y^2}, x^2 + y^2 \leq 150^2\} \quad (5-17)$$

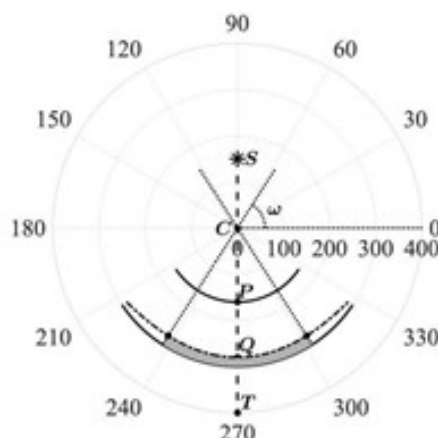


图 3：极坐标系下 FAST 剖面及积分区域示意图

由于抛物面的中轴线是竖直的，所以在球坐标系下可转化为二重积分，故该积分式可化简为：

$$\int_{\beta} \int_{\alpha} r_{\beta} [f(\omega, L_f) - R]^2 d\alpha d\beta \quad (5-18)$$

其中， $\alpha$  为方位角； $\beta$  为仰角； $r_{\beta} = R \cos \beta$  是在  $\beta$  角给定时， $\alpha$  角在基准球面上的轨迹投影出的圆形的半径。根据题目要求的工作口径，可知  $\beta$  的取值范围满足：

$$|\cos \beta| \cdot f(\beta, L_f) = 150 \quad (5-19)$$

根据几何关系，易知式(5-19)有两个解  $\beta_m$  和  $\beta_M$ ，有  $\beta_m < \beta_M$ ，且这两个解的均值为  $\pi/2$ 。为防止对  $\beta$  指向的圆环重复积分，可设  $\beta$  的取值范围为  $\beta \in (\beta_m, \pi/2)$ 。结合上述分析，对式(5-18)进一步化简得：

$$\int_{\beta_m}^{\pi/2} 2\pi r_{\beta} [f(\omega, L_f) - R]^2 d\beta \quad (5-20)$$

以上，得到了球坐标系下，计算理想抛物面和基准球面之间相似程度的准则。

#### 5.1.4 理想抛物面优化模型的建立

由于在现有约束条件下，可求解出多个不同焦距的理想抛物面，故需要对于理想抛物面进行最优化的选取。故构建目标函数如下：



$$L_f^* = \operatorname{argmin}_{L_f} \int_{\beta_m}^{\pi/2} 2\pi r_\beta [f(\omega, L_f) - R]^2 d\beta \quad (5-21)$$

结合式(5-6)，确定形成最优理想抛物面的抛物线方程为：

$$z = \frac{x^2}{2L_f^*} - \frac{L_f^*}{2} - (R - F) \quad (5-22)$$

## 5.2 问题一模型的求解

### 5.2.1 算法设计

首先对目标函数(5-20)的数值特性进行定性分析，可知焦距 $L_f$ 决定了理想抛物面和基准球面之间的相似程度，焦距过大，则理想抛物面会整体低于基准球面；焦距过小，则理想抛物面会整体高于基准球面。易知目标函数的梯度函数在区间内为一个单调连续函数，且区间两端对应的梯度值异号，即目标函数(5-20)在区间(100, 400)上是一个存在极小值点的凸函数。因此，本文基于该性质设计针对梯度函数零点的二分查找法对极小值点进行求解。具体求解算法如下：

#### 算法 1：二分查找算法

- 1: 选定边界值。首先根据题目设定与实际情况，选取 100 与 400 作为二分查找区间的左右端点。
- 2: 取二分查找区间的中点坐标代入公式(5-23)的梯度函数中，计算得到该点所对应的梯度值；
- 3: 若梯度值远大于 0，将二分查找区间的右端点设置为当前中点，并返回第 2 行。
- 4: 若梯度值远小于 0，将二分查找区间的左端点设置为当前中点，并返回第 2 行。
- 5: 若梯度值的绝对值小于机器精度 epsilon，则中点即为待求解的极小值点，程序结束。

### 5.2.2 理想抛物面的求解与分析

基于算法 1，在 MATLAB 中编程进行求解，给出求解出的理想抛物面的剖面视图（图 4 左）及理想抛物面与基准球面的径向高度高低差（图 4 右）：

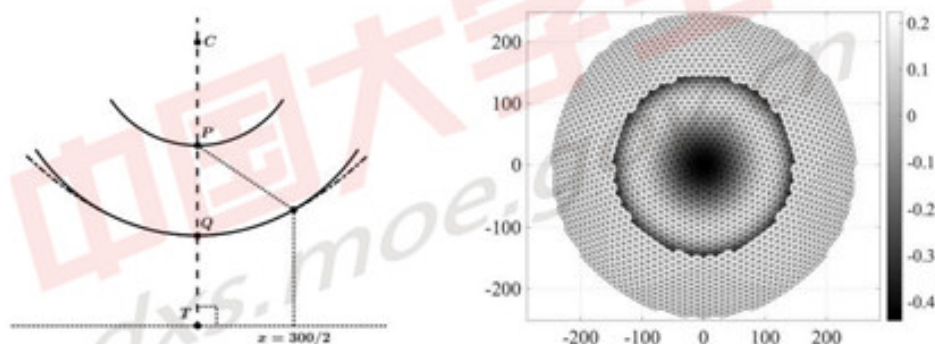


图 4：FAST 2D 剖面最优抛物线（左）、3D 最优抛物面径向高度高低差（右）

最终结果求解出理想抛物面的焦距 $L_f$ 的精确值为 280.854，误差平方在口径上的积





分的最小值为 10.112, 将  $L_f = 280.854$ ,  $R = 300.4$ ,  $F = 0.466R$  代入式(5-6)可得剖切平面上的抛物线方程为:

$$z = \frac{1}{561.708}x^2 - 300.841 \quad (5-23)$$

将抛物线绕  $z$  轴旋转一周后, 可知其旋转抛物面的方程为:

$$z = \frac{1}{561.708}(x+y)^2 - 300.841 \quad (5-24)$$

### 5.2.3 结果检验

基于 5.2.2 节中的精确结果, 对所得理想抛物面的剖面上的径向高度与基准球面半径的高低差进行检验, 可得最优抛物面与基准球面的偏移量:

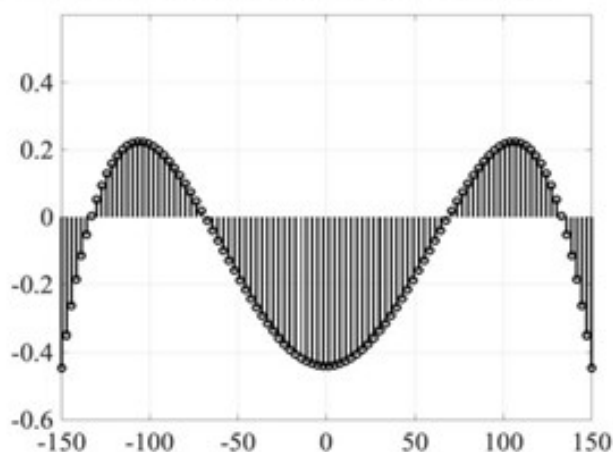


图 5: 最优抛物面与基准球面的偏移量

由图 5 可知, 最优抛物面与基准球面的偏移量在  $(-0.6, +0.4)$  的范围内。因此此最优抛物面在实际应用中, 促动器顶端上下拉动下拉索的长度范围也较小, 结合反射板调节效率因素, 可得本结果合理且较优。

## 六、问题二模型的建立与求解

### 6.1 问题二模型的建立

问题二要求基于问题一所建立的模型, 求解当  $\alpha = 36.795^\circ$ ,  $\beta = 78.169^\circ$  时的理想抛物面, 并以工作抛物面尽量贴近理想抛物面为目标, 在下拉索固定长度、节点之间的伸缩变动率、促动器的伸缩量约束下, 优化相关促动器的伸缩量。

为简化计算, 本文首先建立在标准坐标系上旋转得到的新坐标系, 使得主索节点与理想抛物面在径向方向的距离差值可沿用问题一中的式(5-15), 从而构建本问的目标函数。然后通过分析促动器底端坐标、基准状态与工作状态的主索节点坐标、主索节点的基准坐标与工作坐标以及促动器伸缩量等变量之间的几何关系, 构建本问题的约束条件, 得到反射面板调节优化模型。最后, 设计 BFGS 算法进行模型求解。



### 6.1.1 基于坐标系旋转的理想抛物面与反射面板相似度计算

本题中，由于观测天体 $S$ 的方位对于基准球心 $C$ 存在与竖直方向的倾角，这给构建新的理想抛物面带来了一定的困难。但在求解理想抛物面时，对于不同轴线方向的抛物面来说，其焦距、焦点等性质都是不变的。故将现有空间坐标系，以基准球面的球心 $C$ 为中心、沿基准球面的剖面方向进行旋转，使得问题二中理想抛物面的轴线与旋转后空间直角坐标系 $z$ 轴所在直线重合，此时主索节点与理想抛物面在径向方向的距离差值可沿用问题一中的式(5-15)。

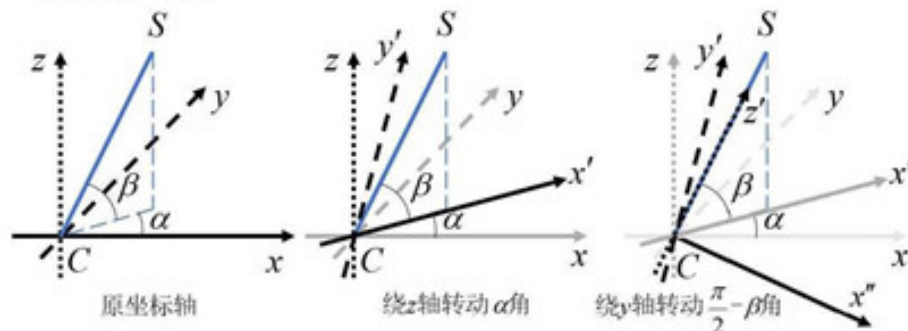


图 6: 坐标系旋转示意图

设第 $i$ 个主索节点的工作坐标为 $(x(i), y(i), z(i))$ 。将原先的坐标轴绕着 $z$ 轴沿正方向旋转角度 $\alpha$ （正向旋转方向与坐标轴指向满足右手定则，下文同义），则坐标相对于坐标系绕 $z$ 轴反向旋转角度 $\alpha$ ；再将此时的坐标轴绕着 $y$ 轴沿正方向旋转角度 $\pi/2 - \beta$ ，则坐标相对于坐标系绕 $y$ 轴反向旋转 $\pi/2 - \beta$ 。根据三维坐标中旋转矩阵的定义，可得旋转坐标系下，第 $i$ 个主索节点的工作坐标 $(x_w(i), y_w(i), z_w(i))$ 为：

$$\begin{pmatrix} x_w(i) \\ y_w(i) \\ z_w(i) \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\pi}{2} - \beta\right) & 0 & -\sin\left(\frac{\pi}{2} - \beta\right) \\ 0 & 1 & 0 \\ \sin\left(\frac{\pi}{2} - \beta\right) & 0 & \cos\left(\frac{\pi}{2} - \beta\right) \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(i) \\ y(i) \\ z(i) \end{pmatrix} \quad (6-1)$$

如图 6，不难看出旋转后坐标系的 $z$ 轴正方向指向被观测体 $S$ 。接着球心 $C$ 为原点，在新的空间直角坐标系上建立空间极坐标系。再将变换后的主索节点的直角坐标转化为新的空间极坐标系下的坐标，得到第 $i$ 个主索节点的坐标为 $(L(i), \beta(i), \alpha(i))$ ，具体形式为：

$$\begin{cases} L(i) = \sqrt{x_w(i)^2 + y_w(i)^2 + z_w(i)^2} \\ \beta(i) = \arcsin\left(-\frac{z_w(i)}{L(i)}\right) \\ \alpha(i) = \begin{cases} 2\pi - \arccos\left(-\frac{x_w(i)}{\sqrt{x_w(i)^2 + y_w(i)^2}}\right), & x_w(i)^2 + y_w(i)^2 \neq 0 \text{ 且 } y_w(i) \geq 0 \\ \arccos\left(-\frac{x_w(i)}{\sqrt{x_w(i)^2 + y_w(i)^2}}\right), & x_w(i)^2 + y_w(i)^2 \neq 0 \text{ 且 } y_w(i) < 0 \\ 0, & x_w(i)^2 + y_w(i)^2 = 0 \end{cases} \end{cases} \quad (6-2)$$

接着，根据问题一中理想抛物面上偏转角度 $\omega$ 与 $L_\omega$ 的表达式(5-15)，得到在新的坐





标系中, 理想抛物面上仰角为 $\beta(i)$ 的点与原点 $C$ 的距离:

$$L_w(i) = \begin{cases} \frac{-L_f \sin \beta(i) + \sqrt{L_f^2 + 2(R-F)L_f \cos^2 \beta(i)}}{\cos^2 \beta(i)}, & \beta(i) \neq \frac{\pi}{2} \\ \frac{L_f}{2 \sin \beta(i)} + \frac{R-F}{\sin \beta(i)}, & \beta(i) = \frac{\pi}{2} \end{cases} \quad (6-3)$$

为保证反射面在工作状态下与理想抛物面尽量贴合, 需要使得在同一径向方向上的主索节点与理想抛物面上的点距离原点 $C$ 的距离差尽量小。故可构建如下优化目标:

$$(x^*(i), y^*(i), z^*(i)) = \underset{x(0), y(0), z(0), i \in \mathcal{I}}{\operatorname{argmin}} \sum_{\beta_m < \beta(i) < \frac{\pi}{2}} (L_w(i) - L(i))^2 \quad (6-4)$$

其中,  $\mathcal{I}$ 为所有主索节点所用到的下标集合;  $\beta_m$ 为新的极坐标下仰角取值的最小值。由于理想抛物面具有照明区域的限制, 5.1.3 节中已经据此给出了仰角的取值范围式, 这在本问题中也同样具有约束效果。注意到, 在基于促动器伸缩完成的主索节点位置变化过程中, 第 $i$ 个主索节点的坐标 $(x_w(i), y_w(i), z_w(i))$ 可以由促动器的伸缩量唯一确定。根据附件 2 中数据, 可知促动器底端的坐标 $(x^-(i), y^-(i), z^-(i))$ 与促动器顶端基准状态下的坐标 $(x^o(i), y^o(i), z^o(i))$ 。设促动器在工作状态时的顶端坐标为 $(x^+(i), y^+(i), z^+(i))$ 。由于促动器底端与促动器顶端始终在一条直线上, 故存在比例关系:

$$\frac{x^+(i) - x^o(i)}{x^o(i) - x^-(i)} = \frac{L_s(i)}{\sqrt{[x^o(i) - x^-(i)]^2 + [y^o(i) - y^-(i)]^2 + [z^o(i) - z^-(i)]^2}} \quad (6-5)$$

其中,  $L_s(i)$ 为每个促动器的伸缩量。同样地,  $y$ 轴和 $z$ 轴方向均同在相同的比例关系, 因此有:

$$\begin{cases} x^+(i) = \frac{L_s(i)(x^o(i) - x^-(i))}{\sqrt{[x^o(i) - x^-(i)]^2 + [y^o(i) - y^-(i)]^2 + [z^o(i) - z^-(i)]^2}} + x^o(i) \\ y^+(i) = \frac{L_s(i)(y^o(i) - y^-(i))}{\sqrt{[x^o(i) - x^-(i)]^2 + [y^o(i) - y^-(i)]^2 + [z^o(i) - z^-(i)]^2}} + y^o(i) \\ z^+(i) = \frac{L_s(i)(z^o(i) - z^-(i))}{\sqrt{[x^o(i) - x^-(i)]^2 + [y^o(i) - y^-(i)]^2 + [z^o(i) - z^-(i)]^2}} + z^o(i) \end{cases} \quad (6-6)$$

以上, 经过推导最终得到了第 $i$ 个促动器的伸缩量 $L_s(i)$ 与对应主索节点的坐标 $(x_w(i), y_w(i), z_w(i))$ 之间的关系。因此加入 $L_s(i)$ 作为式(6-4)中反射面与理想抛物面相似度计算目标函数的决策变量, 整理可得:

$$(x^*(i), y^*(i), z^*(i), L_s^*(i)) = \underset{x(0), y(0), z(0), L_s(i), i \in \mathcal{I}}{\operatorname{argmin}} \sum_{\beta_m < \beta(i) < \frac{\pi}{2}} [L_w(i) - L(i)]^2 \quad (6-7)$$

### 6.1.2 约束条件的确定

#### (1) 节点之间的伸缩变动率约束

根据假设 6, 相邻节点之间的距离可能会发生微小变化, 变化幅度的极限为 0.07%, 故可构建如下约束条件:



$$\begin{cases} \frac{[x(i_1) - x(i_2)]^2 + [y(i_1) - y(i_2)]^2 + [z(i_1) - z(i_2)]^2}{[x_0(i_1) - x_0(i_2)]^2 + [y_0(i_1) - y_0(i_2)]^2 + [z_0(i_1) - z_0(i_2)]^2} \geq (1 - 0.07\%)^2, (i_1, i_2) \in E \\ \frac{[x(i_1) - x(i_2)]^2 + [y(i_1) - y(i_2)]^2 + [z(i_1) - z(i_2)]^2}{[x_0(i_1) - x_0(i_2)]^2 + [y_0(i_1) - y_0(i_2)]^2 + [z_0(i_1) - z_0(i_2)]^2} \leq (1 + 0.07\%)^2, (i_1, i_2) \in E \end{cases} \quad (6-8)$$

其中,  $E = \{(i_1, i_2) \in \mathcal{I} \times \mathcal{I}\}$ 。

## (2) 促动器伸缩范围约束

根据题设, 促动器伸缩量只能在  $-0.6m$  到  $+0.6m$  之间, 故具有如下约束条件:

$$-0.6 \leq L_s(i) \leq 0.6 \quad (6-9)$$

## (3) 下拉索长度约束

主索节点与促动器顶端是由固定长度的下拉索连接的, 可通过基准状态下的主索节点的坐标与促动器顶端基准坐标, 计算出每一段下拉索的长度。设每一段下拉索的长度为  $L_s(i)$ , 基准状态下主索节点坐标为  $(x_0(i), y_0(i), z_0(i))$ , 促动器顶端基准坐标为  $(x^o(i), y^o(i), z^o(i))$ , 则有:

$$L_s(i) = \sqrt{[x_0(i) - x^o(i)]^2 + [y_0(i) - y^o(i)]^2 + [z_0(i) - z^o(i)]^2}, i \in \mathcal{I} \quad (6-10)$$

设促动器在工作状态时的顶端坐标为  $(x^+(i), y^+(i), z^+(i))$ , 则存在如下约束条件:

$$[x(i) - x^+(i)]^2 + [y(i) - y^+(i)]^2 + [z(i) - z^+(i)]^2 = L_s(i)^2 \quad (6-11)$$

## 6.1.3 反射面板调节优化模型的构建

综上所述, 本文建立的反射面板调节优化模型为:

$$(x^*(i), y^*(i), z^*(i), L_s^*(i)) = \underset{x(i), y(i), z(i), L_s(i), i \in \mathcal{I}}{\operatorname{argmin}} \sum_{\beta_m < \beta(i) < \frac{\pi}{2}} [L_w(i) - L(i)]^2 \quad (6-12)$$

$$s.t. \begin{cases} \frac{[x(i_1) - x(i_2)]^2 + [y(i_1) - y(i_2)]^2 + [z(i_1) - z(i_2)]^2}{[x_0(i_1) - x_0(i_2)]^2 + [y_0(i_1) - y_0(i_2)]^2 + [z_0(i_1) - z_0(i_2)]^2} \geq (1 - 0.07\%)^2, (i_1, i_2) \in E \\ \frac{[x(i_1) - x(i_2)]^2 + [y(i_1) - y(i_2)]^2 + [z(i_1) - z(i_2)]^2}{[x_0(i_1) - x_0(i_2)]^2 + [y_0(i_1) - y_0(i_2)]^2 + [z_0(i_1) - z_0(i_2)]^2} \leq (1 + 0.07\%)^2, (i_1, i_2) \in E \\ -0.6 \leq L_s(i) \leq 0.6, \forall i \in \mathcal{I} \\ [x(i) - x^+(i)]^2 + [y(i) - y^+(i)]^2 + [z(i) - z^+(i)]^2 = L_s(i)^2, \forall i \in \mathcal{I} \end{cases}$$

## 6.2 问题二模型的求解

### 6.2.1 算法设计

由于问题二共计需要求解 2226 个主索节点的  $xyz$  坐标, 以及其对应的促动器伸缩量  $L_s$ , 需要求解的决策变量数量庞大且其中带有非线性的等式和不等式约束, 因此必须寻找适应于高维大规模问题求解的算法。

对于模型中非线性约束, 首先基于拉格朗日乘子法的思想, 引入一个乘数  $\lambda$ , 将等





式非线性约束转化为目标函数的一部分。接着,在不考虑不等式约束的情况下进行求解,当结果违反不等式约束时,将其转化为等式约束并再此求解,直到得到不违反约束的最优解。注意到待求解的模型中的非线性约束都能表示为决策变量的二次型,具有光滑和凸的性质,且容许在数值定义域上轻微违反约束时稳定求值,因此此处使用拟牛顿方法中的 BFGS 方法,以在不直接计算 Hessian 矩阵的情况下实现这一高维问题的超线性收敛。具体求解算法如下:

---

#### 算法 2: BFGS 算法

---

- 1: 初始化参数。为循环变量  $k$  赋初始值  $k=0$ , 为切线刚度矩阵  $B_0$  赋初始值为单位矩阵, 设置初始解  $x_0$  为主索网节点在附件中的基准态坐标。
  - 2: 循环: 根据条件  $B_k p_k = -\nabla f(x_k)$ , 计算  $p_k$ 。
  - 3: 由式  $x_{k+1} = x_k + \alpha_k p_k$  得到  $x_{k+1}$ , 其中  $\alpha_k$  的值从 1.0 开始折半查找, 直到步长足够小时, 函数值低于上一步。
  - 4: 计算  $s_k = \alpha_k p_k$ ,  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
  - 5: 计算新的切线刚度矩阵  $B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$ 。
  - 6: 判断: 将当前解代入原式验证, 若拉格朗日方程组等式两边的误差小于 epsilon, 则得到待求解, 程序结束; 否则返回第 2 行。
- 

#### 6.2.2 反射面调节情况的求解与分析

由于问题二中使用到了旋转的坐标系, 故本问理想抛物面的结果可以在旋转坐标变换的条件下, 得到了顶点坐标为  $(-49.392, -36.943, -294.450)$  的抛物面。接着, 基于算法 2, 在 MATLAB 中编程进行求解, 给出求解出的 3D 的促动器提升量图 (图 7 左) 及 3D 的理想抛物面的拟合误差图 (图 7 右):

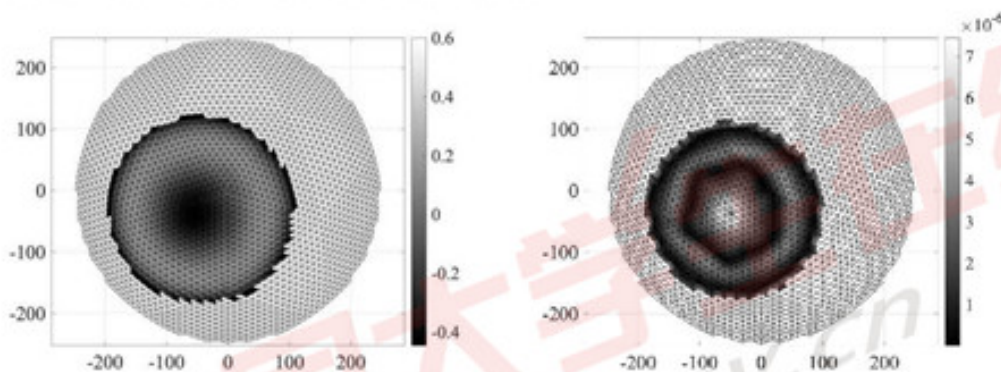


图 7: 3D 促动器提升量 (左)、3D 理想曲面拟合误差 (右)

由图 7 左可知, 此状态下促动器的提升量处于  $-0.4m$  到  $0.6m$  的范围内, 此时反射面较为贴近所求理想抛物面, 且促动器顶端上下拉动下拉索的长度范围也较小。由图 7 (右) 可知, 此状态下理想抛物面的拟合误差处于  $0$  到  $7 \times 10^{-6}$  的范围内, 拟合误差也较小, 说明此时的结果为合理最优解。



### 6.2.3 结果检验

基于 6.2.2 节中的结果, 求出主网节点与理想抛物面间的球面径向误差以及主网节点间距离的变化率:

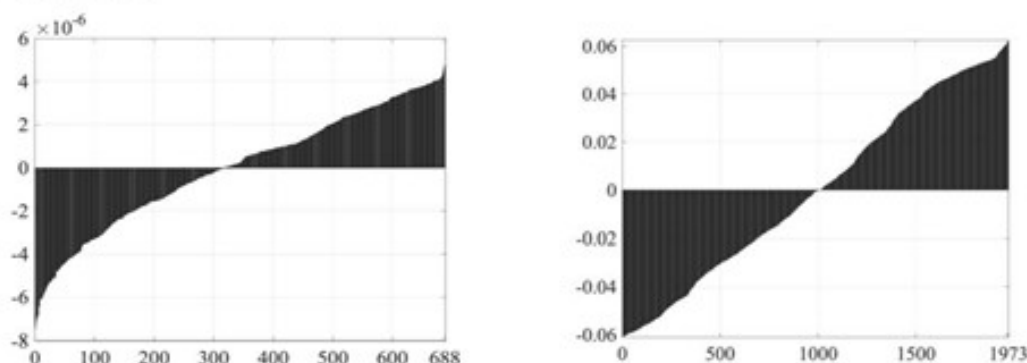


图 8: 主网节点与理想抛物面球面径向误差 (左)、主网节点间距离变化率 (单位: 百分数) (右)

由图 8 (左) 可知, 此状态下主网节点与理想抛物面间的球面径向误差处于  $-8 \times 10^{-6}$  到  $6 \times 10^{-6}$  的范围内, 可见此径向误差非常小, 此时反射面较为贴近所求理想抛物面。由图 8 (右) 可知, 此状态下主网节点间距离的变化率处于  $-0.06\%$  到  $0.06\%$  的范围内, 充分利用而并没有违反  $0.07\%$  的裕度, 说明此时的结果是合理的。

## 七、问题三模型的建立与求解

### 7.1 问题三模型的建立

问题三要求基于问题二的反射面调节方案, 计算调节后馈源舱的接收比, 并与基准球面接收比作比较。对于该问题, 本文首先通过使用旋转变换, 将倾斜入射光线转化为垂直入射光线。其次, 通过求解线性方程组, 依次确定出入射光线与三角形面板的相交判定式、交点坐标、三角形面板的法线向量, 并利用光线垂直入射的性质, 使用法线向量简化计算得到出射光线的方向角。再次, 通过联立射线方程与馈源舱所在的目标高度, 得出射线方程的步长和出射光到达目标高度时的坐标, 并与馈源舱的有效区域进行比对, 作为入射光线是否被有效接收的判定式。最后, 将 300 米口径内实际接收区域作为积分域, 将入射光线有效接收判别式作为被积函数, 使用蒙特卡洛算法进行积分, 计算有效接收的光源面积和调节前后的信号接收比。

#### 7.1.1 反射光线几何模型的建立

首先, 要在新坐标系中计算反射点在反馈面板三角形区域内的坐标。设新坐标系的空间变量为  $(x_w, y_w, z_w)$ , 在  $x_w^2 + y_w^2 \leq 150^2$  的范围内, 沿着  $z$  轴负方向均有来自于被观测物体的电磁波照射在反射面上。记反射面板上一三角形三个顶点  $B_1 B_2 B_3$  的坐标分别为  $B_1(x_w(j_1), y_w(j_1), z_w(j_1))$ ,  $B_2(x_w(j_2), y_w(j_2), z_w(j_2))$  和  $B_3(x_w(j_3), y_w(j_3), z_w(j_3))$ , 表示三角形  $j$  访问的主索节点的坐标。其中编号  $j \in \mathcal{J} = \{j | j = (i_1, i_2, i_3) \in \mathcal{I} \times \mathcal{I} \times \mathcal{I}\}$ 。





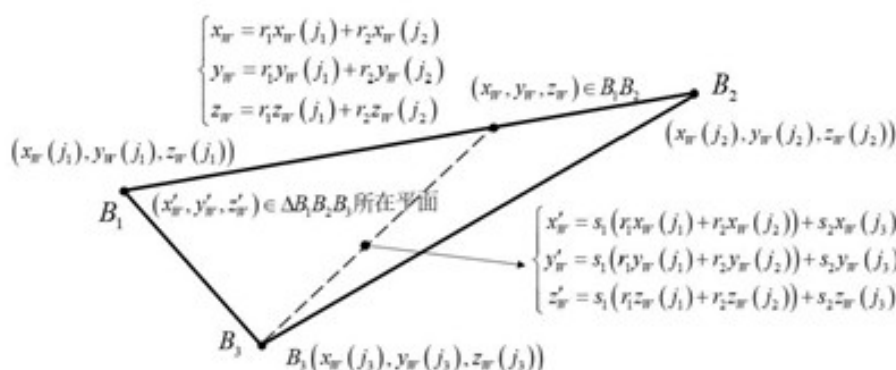


图 9: 反馈面板三角形区域示意图

设一点 $(x_w, y_w, z_w)$ 在线段 $B_1B_2$ 上, 则其在 $x$ 轴的坐标可以表示为:

$$x_w = r_1 x_w(j_1) + r_2 x_w(j_2) \quad (7-1)$$

其中,  $r_1, r_2 > 0$  且  $r_1 + r_2 = 1$ 。接着, 设另一点 $(x'_w, y'_w, z'_w)$ 在点 $(x_w, y_w, z_w)$ 与 $B_3$ 为端点的线段上, 该点在 $x$ 轴上的坐标可以表示为:

$$x'_w = s_1 r_1 x_w(j_1) + s_1 r_2 x_w(j_2) + s_2 x_w(j_3) \quad (7-2)$$

其中,  $s_1, s_2 > 0$  且  $s_1 + s_2 = 1$ 。

记 $s_1 r_1 = t_1$ ,  $s_1 r_2 = t_2$ ,  $s_2 = t_3$ 。易证 $t_1 + t_2 + t_3 = 1$ 且 $t_1 > 0$ ,  $t_2 > 0$ ,  $t_3 > 0$ 。同理可得该点在 $y$ 轴与 $z$ 轴的坐标。以上, 得到了一点在以 $B_1B_2B_3$ 为顶点形成的三角形内部平面上的坐标为:

$$\begin{pmatrix} x'_w \\ y'_w \\ z'_w \end{pmatrix} = \begin{pmatrix} x_w(j_1) & x_w(j_2) & x_w(j_3) \\ y_w(j_1) & y_w(j_2) & y_w(j_3) \\ z_w(j_1) & z_w(j_2) & z_w(j_3) \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (7-3)$$

将 $t_3 = 1 - t_1 - t_2$ 代入式(7-3)约去 $z$ 轴的坐标, 并根据矩阵运算法则可得:

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} x_w(j_1) - x_w(j_3) & x_w(j_2) - x_w(j_3) \\ y_w(j_1) - y_w(j_3) & y_w(j_2) - y_w(j_3) \end{pmatrix}^{-1} \begin{pmatrix} x'_w - x_w(j_3) \\ y'_w - y_w(j_3) \end{pmatrix} \quad (7-4)$$

注意到三个点 $B_1B_2B_3$ 为三角形的三个顶点, 式(7-4)中的逆矩阵总是存在。因此, 当坐标 $(x'_w, y'_w)$ 给定时, 可以根据式(7-4)计算 $t_1$ 和 $t_2$ 的值, 进而判断该点的水平面投影是否在三角形的水平面投影的内部。根据实际条件, 对于任意在 $x_w^2 + y_w^2 \leq 150^2$ 范围内的 $(x'_w, y'_w)$ , 都有一个 $j \in \mathcal{J}$ 使得其在 $j$ 所指代的三角形内部。故可进一步根据式(7-3)求得 $(x'_w, y'_w)$ 对应的 $z$ 轴坐标为:

$$z'_w = t_1 z_w(j_1) + t_2 z_w(j_2) + t_3 z_w(j_3) \quad (7-5)$$

其次, 要确定反射直线的解析表达式。计算反馈面上三个点 $B_1B_2B_3$ 构成平面的法向量 $\vec{n} = (e_1; e_2; e_3)$ , 有:

$$\begin{pmatrix} x_w(j_1) - x_w(j_2) & y_w(j_1) - y_w(j_2) & z_w(j_1) - z_w(j_2) \\ x_w(j_2) - x_w(j_3) & y_w(j_2) - y_w(j_3) & z_w(j_2) - z_w(j_3) \\ x_w(j_3) - x_w(j_1) & y_w(j_3) - y_w(j_1) & z_w(j_3) - z_w(j_1) \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (7-6)$$



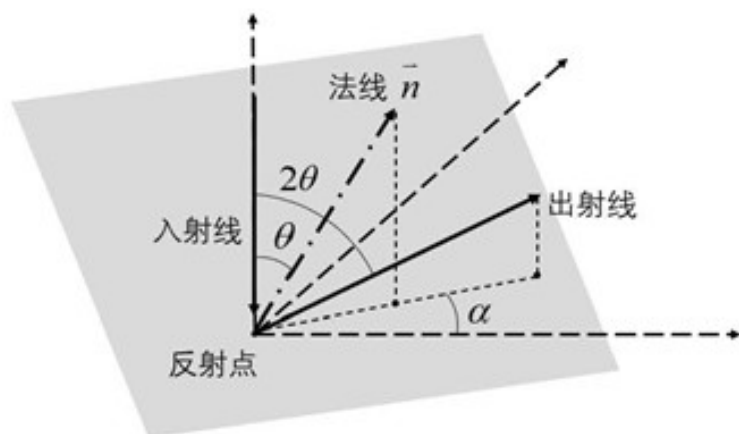


图 10: 各角度三维示意图

注意到, 法向量没有归一化且式(7-6)左侧矩阵的秩为 2。且考虑到问题所求的抛物面开口向上, 即  $e_3 > 0$ , 故令  $e_3 = 1$ , 将式(7-6)进行化简并根据矩阵运算法则可得:

$$\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} x_w(j_1) - x_w(j_2) & y_w(j_1) - y_w(j_2) \\ x_w(j_2) - x_w(j_3) & y_w(j_2) - y_w(j_3) \end{pmatrix}^{-1} \begin{pmatrix} z_w(j_2) - z_w(j_1) \\ z_w(j_3) - z_w(j_2) \end{pmatrix} \quad (7-7)$$

由于入射光垂直于水平面向下, 由方向向量  $\vec{n}_1 = (0, 0, 1)$ 。因此  $\vec{n}$  与  $\vec{n}_1$  的夹角:

$$\theta = \arccos \frac{\vec{n} \cdot \vec{n}_1}{|\vec{n}| \cdot |\vec{n}_1|} \quad (7-8)$$

接着, 将法向量  $\vec{n}$  投影到水平面后与  $x$  轴的方向向量  $\vec{n}_x = (1, 0, 0)$  的夹角  $\alpha$  表示为:

$$\alpha = \begin{cases} \arccos \frac{\vec{n} \cdot \vec{n}_x}{|\vec{n}| \cdot |\vec{n}_x|}, & e_2 \geq 0 \\ -\arccos \frac{\vec{n} \cdot \vec{n}_x}{|\vec{n}| \cdot |\vec{n}_x|}, & e_2 < 0 \end{cases} \quad (7-9)$$

另外, 由于出射光与入射光所在平面垂直于反射面, 可知出射光与  $z$  轴的夹角为  $2\theta$ 。因此出射光的方向向量:

$$\vec{n}_2 = (\sin(2\theta)\cos\alpha, \sin(2\theta)\sin\alpha, \cos(2\theta)) \quad (7-10)$$

以上, 可得出射光所在的直线的标准方程为:

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{pmatrix} x'_w \\ y'_w \\ z'_w \end{pmatrix} + r \begin{pmatrix} \sin(2\theta)\cos\alpha \\ \sin(2\theta)\sin\alpha \\ \cos(2\theta) \end{pmatrix} \quad (7-11)$$

最后, 要确定馈源舱接收平面的解析表达式。根据题目, 馈源舱接收平面的法线方向指向理想抛物面的顶点, 距离原点的距离为。考虑到由于馈源舱接收信号有效区域为直径 1 米的中心圆盘, 因此其平面方程为





$$\begin{cases} z_w = -(R-F) \\ x_w^2 + y_w^2 \leq 0.5^2 \end{cases} \quad (7-12)$$

综上, 在新坐标系中构建了反射光线所在直线与馈源舱接收平面的解析表达式。

### 7.1.2 反射光线吸收与信号比计算模型的建立

在确定了反射光线与馈源舱接收平面的解析表达式后, 即可判断反射光线是否被接收并计算反射信号比。

首先, 将接收面平面方程与出射光所在直线方程进行联立, 可以得到出射光线命中馈源舱所在 $z$ 平面的行进步长:

$$r = \frac{F - R - z'_w}{\cos(2\theta)} \quad (7-13)$$

结合式(7-11)和(7-13), 整理可得:

$$\begin{cases} x_w = x'_w + \frac{F - R - z'_w}{\cos(2\theta)} \sin(2\theta) \cos \alpha \\ y_w = y'_w + \frac{F - R - z'_w}{\cos(2\theta)} \sin(2\theta) \sin \alpha \end{cases} \quad (7-14)$$

接着, 可以基于入射线指向的水平面上点的坐标 $(x'_w, y'_w)$ 并根据式(7-14)求得 $x_w$ 和 $y_w$ 。因此可以通过:

$$I(x'_w, y'_w) = \begin{cases} 1, & x_w^2 + y_w^2 \leq 0.5^2 \text{ 且 } x_w'^2 + y_w'^2 \geq 0.5^2 \\ 0, & \text{其他} \end{cases} \quad (7-15)$$

判断反射光线是否能够被接受面吸收。进一步, 对判断函数 $I(x_w, y_w)$ 在 $x_w^2 + y_w^2 \leq 150^2$ 范围内积分, 可以得到被馈源舱吸收的光线所占的面积:

$$\iint_{x_w^2 + y_w^2 \leq 150^2} I(x_w, y_w) dx_w dy_w \quad (7-16)$$

接着, 注意到有效口径内的每一个点都必然有且仅有一个相应的三角形, 则平面内接收到的光线面积为 $150^2 \pi$ 。最后, 可得接收与反射信号之比为:

$$\eta = \frac{1}{150^2 \pi} \iint_{x_w^2 + y_w^2 \leq 150^2} f(x_w, y_w) dx_w dy_w \quad (7-17)$$

## 7.2 问题三模型的求解

### 7.2.1 算法设计

对于问题三所求的积分数值, 如果直接对整个口径圆面进行积分, 则对于每个输入的光线坐标, 算法都将消耗大量时间遍历 $xOy$ 平面上的各个投影三角形, 以确定这一坐标所属的三角形法线, 用于计算出射光线的射线方程。为了避免这一逐点依次进行三角形遍历的时间成本, 我们将圆形口径积分域分解为各个互不重叠的 $xOy$ 投影三角形积分域, 通过对各个三角形积分域内均匀地采样各个光线的接收判定信息, 使得一次性遍历到各个投影三角形时, 可以大批量地使用同一个法线向量处理多个投影到该三角形上的



入射光线，甚至使用足够高密度的采样从入射光线的接收信息还原出馈源器有效区域在各个投影三角形上的像。为了实现三角形区域的均匀采样式积分，我们设计了包含对边界条件精细修正的蒙特卡洛积分算法，算法细节如下：

---

### 算法 3：蒙特卡洛算法

---

- 1: 设置接收面积近似值  $a: = 0$ ;
  - 2: 遍历三角形  $j \in \mathcal{J}$ ;
  - 3:     设置接收次数初始值  $d: = 0$ ;
  - 4:     抽样次数 10000, 循环;
  - 5:         每次在 0 到 1 之间平均随机抽样生成一个  $t_1$  和一个  $t_2$ ;
  - 6:         如果  $t_1 + t_2 > 1$ ;
  - 7:              $t_1 := 1 - t_1$ ;
  - 8:              $t_2 := 1 - t_2$ ;
  - 9:          $t_3 = 1 - t_1 - t_2$ ;
  - 10:        使用式(7-3)从  $(t_1, t_2, t_3)$  计算出  $(x, y, z)$ ;
  - 11:        使用  $x$  和  $y$  计算出  $xOy$  平面上的投影向量长度;
  - 12:        如果向量长度超过口径的一半 150 米;
  - 13:            跳过这次接收计算，跳转到第 16 行;
  - 14:        使用  $(x, y, z)$  计算出是否接收，如果接收:
  - 15:             $d := d + 1$ ;
  - 16:        如果抽样次数没满，跳转到第 4 行循环;
  - 17:        计算出三角形  $j$  在  $xOy$  平面上的投影面积;
  - 18:         $a_1 := a_1 + d/10000 \cdot (\text{阴影面积})$
  - 19:        如果遍历三角形没结束，跳转到第 2 行继续遍历
  - 20: 输出  $a_1/(150^2\pi)$  即为所求接收率;
- 

#### 7.2.2 问题三的结果与分析

基于算法 3，在 MATLAB 中编程进行求解，给出求解出的在相同区间下，工作态（左）与基准态（右）中，反射面板与有效入射点区域的纹理对比图。其中，黑点表示光波经反射后，可以被馈源舱接受平面的中心有效区域吸收的反射面板的入射点区域。图 11 中所选取的区间横坐标范围为  $(-90^\circ, -30^\circ)$ ，对照图 5，可得此区间基准态和工作态之间偏移量的相对斜率最大，夹角差异最大，导致基准态时部分反射面板中的“有效接受反射命中点”逐渐偏移至反射面板外部，进而反射率也相应降低。因此工作态和基准态下，最终产生的反射效果相差较大。最终计算得到的调节前接收比 0.811%，调节后接收比 1.103%，提升了 36%。





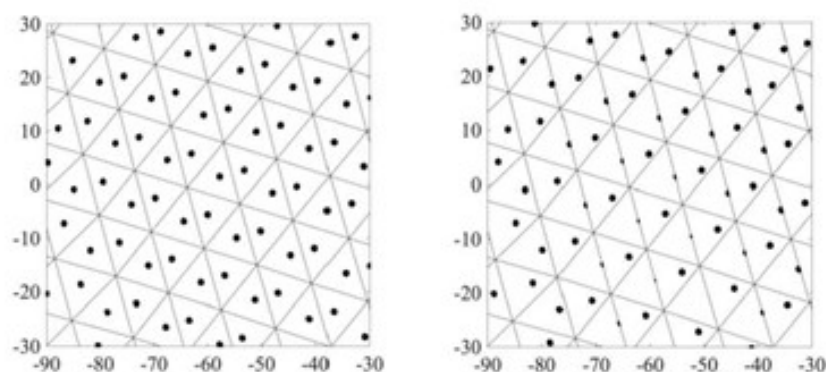


图 11：在相同区间下，工作态（左）与基准态（右）纹理对比  
将计算得到的此情况下反射光线聚焦于焦点的状态进行可视化处理，得到图 12。

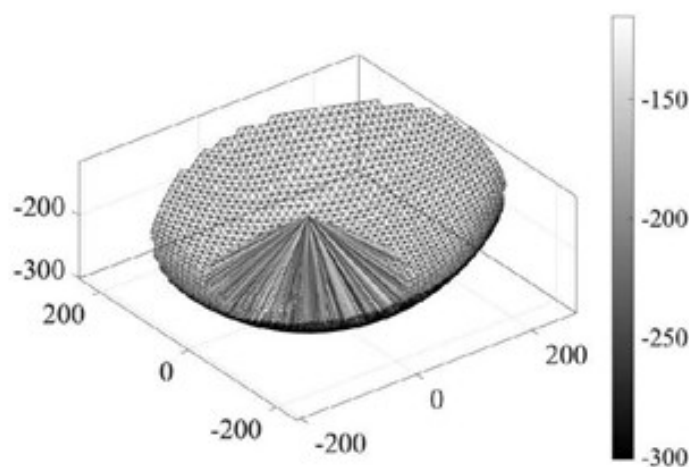


图 12：反射光线聚焦于焦点的示意图

由图 12 可观察到，在此情况下，由观测点 $S$ 方向射出的平行均匀光波被精准地反射到了馈源舱有效接收区域，故所得结果符合题设。

### 7.2.3 问题三结果检验

基于 7.2.2 节中的结果，对蒙特卡洛积分的稳定性进行检验。在工作态与基准态下各积分 100 次，得到数值波动量如图 12：

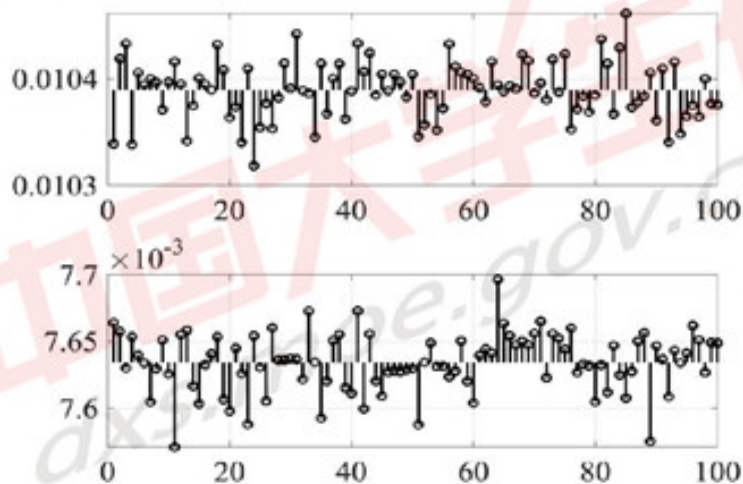


图 13：工作态（上）与基准态（下）下多次蒙特卡洛积分的数值波动量



观察光斑分布偏移,由图 13 可得工作态下与基准态下,多次蒙特卡洛积分的数值波动量范围都在 $10^{-3}$ 的维度上,波动量较小,可以认为结果合理。

## 八、模型的评价与推广

### 8.1 模型的评价

本文构建的 FAST 射电望远镜的工作抛物面调整模型,对于利用促动器调整工作抛物面的过程进行了合理假设,在保证模型求解结果精确性的条件下,有效降低了模型的复杂程度。在求解过程中,利用了二分法, BFGS 法,蒙特卡洛积分等方法对于模型进行求解,使得算法的收敛速度快且结果精确。

在第一问的理想抛物面求解过程中,考虑到抛物面是由抛物线沿轴线旋转得来,故将空间的问题转化为平面坐标系下的问题,使得模型描述更为简便。对于限定了焦距的抛物线方程,本文以使其与基准球面的近似程度最大为最优化条件,对于其焦距进行了优化,从而确定了最优的抛物面方程。

在第二问的工作抛物面调整过程中,考虑到抛物面的开口方向与原先的坐标轴方向并不相同,本文将原先的直角坐标系进行旋转,使得原先的坐标轴方向依然指向被测物体,使得抛物面方程的描述更为简化,构建出的模型更加简明。并且在此过程中,通过极坐标变换,利用了第一问得出的结论,节省了公式推导的工作量。在求解时利用 BFGS 法,将约束条件转化为二次型,提高了算法的收敛速度与精确度。

在第三问的馈源舱接收比计算模型中,本文将直角坐标,方位角等概念结合使用,在旋转后的直角坐标中通过构建参数方程判断三角形平面与直线的相交关系,使得模型的推导过程更加简单。在求解有效接受的光源面积时,使用蒙特卡洛积分的方法进行求解,在一定程度上提高了解算的速度。

### 8.2 模型的改进

与实际情况相比,题目中缺乏边界结构如何固定的相关信息,如支撑结构等,这就使得在确定整个工作抛物面时约束条件是不够充分的。若能够提供最外圈反射面板的固定方式,并且合理假设在边界的节点之间的距离约束等条件,可以对于工作抛物面的边界条件进行建模,并且通过各个节点之间的连接关系间接地传递约束,从而给出 FAST 射电望远镜所有促动器伸缩量的调整策略,构建出可以投入实际调度的工作抛物面调整模型。

## 九、参考文献

- [1] 杨凡,李广云,王力. 三维坐标转换方法研究[J]. 测绘通报, 2010 (6): 5-7
- [2] 龚纯,王正林. 精通 MATLAB 最优化计算[M]. 电子工业出版社, 2009.
- [3] Beavis B, Dobbs I. Optimisation and stability theory for economic analysis[M]. Cambridge university press, 1990.





## 十、附录

### 10.1 主要计算程序：基于 MATLAB R2019a 开发

#### 10.1.1 主要计算程序

##### prob1\_calc.m: 问题 1 的求解核心算法

```
function [Lf, angle300] = prob1_calc()
    %
    % 问题 1 的核心计算部分
    % 积分输出最优焦距
    % 并且与 Constant 中固定的焦距做比较
    %
    warning('off','MATLAB:integral:MaxIntervalCountReached');
    warning('off','MATLAB:integral:MinStepSize');
    angle_min = acos(150/Constant.sphere_radius);
    Lf = fzero(@(Lf)int_dL(Lf, angle_min), [100, 400]);
    disp("问题 1 最优焦距 Lf: "+mat2str(Lf,17));
    disp("问题 1 最小误差平方积分: "+mat2str(int_L(Lf, angle_min),17));
    assert(abs(int_dL(Lf, angle_min)) < sqrt(eps));
    assert(Lf == Constant.Lf);
    [~, ~, angle] = Constant.get_rad_angle();
    f = @(angle)abs(Polar.para(Lf, angle)*cos(angle))-150;
    angle300 = fzero(f, [pi/2, angle(end)]);
    assert(abs(f(angle300)) <= sqrt(eps));
end
```

```
function v = int_L(Lf, angle_min)
```

```
    v = integral(@f, angle_min, pi/2);
```

```
    function err = f(angle)
```

```
        [L, ~] = Polar.para(Lf, angle);
```

```
        err = (L-Constant.sphere_radius).*(L-Constant.sphere_radius);
```

```
        radius = Constant.sphere_radius.*cos(angle);
```

```
        err = err.*(2.*pi.*radius);
```

```
    end
```

```
end
```

```
function v = int_dL(Lf, angle_min)
```

```
    v = integral(@f, angle_min, pi/2);
```



```

function derr = f(angle)
    [L, dL] = Polar.para(Lf, angle);
    derr = 2.*dL.*(L-Constant.sphere_radius);
    radius = Constant.sphere_radius.*cos(angle);
    derr = derr.*(2.*pi.*radius);
end
end

```

### prob2\_calc.m: 问题 2 的求解核心算法

```

function [fval, opt_X, opt_XYZ, opt_Lz, opt_ABR, index, W, Winv, ...
    edge, edge_distSQ, motor_hi, motor_unit, motor_distSQ] = prob2_calc()

clf('reset');
rng('default');
clear global

% 初始化数据集
data = Data();
% 初始化旋转变换矩阵
[W, Winv] = Polar.compose_rotate(Constant.alpha, Constant.beta);
% 选择 300 口径内的节点
[~, angle300] = prob1_calc();
main_XYZ = data.main_node_coordinate;
main_ABR = Polar.xyz2abr(main_XYZ*W);
index = main_ABR(:,2) >= pi-angle300;
N_nodes = sum(index);
main_XYZ = main_XYZ(index, :); % 过滤主网节点
edge = data.edge(all(index(data.edge),2),:); % 过滤节点之间的连边
index2(index) = 1:sum(index); % 构建过滤之后的新的节点下标顺序
edge = index2(edge); % 使用新的节点下标对连边节点序号重定向
assert(all(all(edge))); % 连边节点序号不能有空泡
% 初始化邻边长度
edge_distSQ = main_XYZ(edge(:,1),:)-main_XYZ(edge(:,2),:);
edge_distSQ = sum(edge_distSQ.*edge_distSQ,2);
% 促动器预处理
motor_hi = data.motor_higher_base(index, :);
motor_lo = data.motor_lower(index, :);

```





```

motor_unit = motor_hi-motor_lo;
motor_unit = motor_unit./sqrt(sum(motor_unit.*motor_unit,2));
motor_distSQ = main_XYZ-motor_hi;
motor_distSQ = sum(motor_distSQ.*motor_distSQ, 2);

objective = @(X)obj_SSE(X, W, N_nodes);
nonlcon = @(X)constraints(X, edge, edge_distSQ, motor_hi, motor_unit,
motor_distSQ, N_nodes);

f_testkit(...
    pack(main_XYZ, rand(N_nodes, 1)), ...
    objective,...
    nonlcon...
);

opts = optimoptions('fmincon', 'Display', 'iter-detailed');
opts.FunValCheck = 'on';
opts.SpecifyObjectiveGradient = true;
opts.SpecifyConstraintGradient = true;
opts.MaxFunctionEvaluations = inf;
opts.MaxIterations = inf;
opts.HonorBounds = false;
% opts.SubproblemAlgorithm = 'cg';
opts.HessianApproximation = 'lbfgs';
saved = load('prob2');
if true
    opt_X = saved.opt_X;
else

    opt_X = fmincon(objective, pack(main_XYZ, zeros(N_nodes, 1)), [], [], [], [],...
        pack(-inf(size(main_XYZ)), repmat(-0.6, N_nodes, 1)),... % lb
        pack( inf(size(main_XYZ)), repmat( 0.6, N_nodes, 1)),... % ub
        nonlcon, opts); %#ok<UNRCH>
end
fval = obj_SSE(opt_X, W, N_nodes);
[opt_XYZ, opt_Lz] = unpack(opt_X, N_nodes);

```



```

    opt_ABR = Polar.xyz2abr(opt_XYZ*W');
end

```

方和

```

function [SSE, dSSE_dXYZ] = obj_SSE(X, W, N_nodes)
%
% 目标函数
% 球面径向投影 Lw（理想抛物面）和主网节点实际投影位置 L 之间的误差平
%
[XYZ, Lz] = unpack(X, N_nodes);
[abr, ~, db, dr] = Polar.xyz2abr(XYZ*W');
[Lw, ~, dangle] = Polar.para(Constant.Lf, abr(:,2,:));
err = Lw-abr(:,3);
SSE = err'*err;
dSSE_dXYZ = 2.*err.*(dangle.*db-dr);
dSSE_dXYZ = dSSE_dXYZ*W;
dSSE_dXYZ = [dSSE_dXYZ(:); zeros(size(Lz))];
end

```

```

function [c,ceq,gc,gceq] = constraints(...
    X, ...
    edge, edge_distSQ, ...
    motor_hi, motor_unit, motor_distSQ, N_nodes)
%
% 约束条件（二次型）
% 条件 1: 保持主网节点之间的连接索的距离保持固定
% 条件 2: 保持主网节点与相应的促动器之间的距离（下拉索）保持固定
%
[XYZ, Lz] = unpack(X, N_nodes);
errdist = XYZ(edge(:,1),:)-XYZ(edge(:,2),:);
edist = sum(errdist.*errdist,2);
c = [edist-edge_distSQ*(1.0007*1.0007); edge_distSQ*(0.9993*0.9993)-edist];
I1 = sub2ind(size(XYZ), repmat(edge(:,1),1,3), repmat(1:3,numel(edist),1));
I2 = sub2ind(size(XYZ), repmat(edge(:,2),1,3), repmat(1:3,numel(edist),1));
J = repmat((1:numel(edist)).', 1, 3);
gc = [

```



```

    sparse(...
    reshape([I1 I2],[],1),...
    reshape([ J   J],[],1),...
    reshape(2.*[errdist -errdist],[],1),...
    numel(X),numel(edist)), ...
    -sparse(...
    reshape([I1 I2],[],1),...
    reshape([ J   J],[],1),...
    reshape(2.*[errdist -errdist],[],1),...
    numel(X),numel(edist))
    ];

    motor_hi_new = motor_hi+Lz.*motor_unit;
    errdist = XYZ-motor_hi_new;
    edist = sum(errdist.*errdist,2);
    ceq = edist-motor_distSQ;

    J = repmat((1:numel(edist)).', 1, 4);
    gceq = sparse(...
        reshape(1:numel(X),[],1),...
        reshape(J,[],1),...
        reshape(2.*[errdist -sum(errdist.*motor_unit,2)],[],1),...
        numel(X),numel(edist));
end

function [SSE, dSSE] = constr_test_ceq(X, constraints)
    [~,ceq,~,gceq] = constraints(X);
    SSE = ceq'*ceq;
    dSSE = full(2.*gceq*ceq);
end
function [SSE, dSSE] = constr_test_c(X, constraints)
    [c,~,gc,~] = constraints(X);
    SSE = c'*c;
    dSSE = full(2.*gc*c);
end

function f_testkit(data, objective, constraints)

```





```

%
% 函数微分正确性检查
%
f = objective;
for i = 1:10
    index = randi(numel(data)/4*3);
    ratio = linspace(0.1, 10.0, 20);
    diff_ = arrayfun(@(r)derr_diff(r, index), ratio);
    ana_ = arrayfun(@(r)derr_ana(r, index), ratio);
    disp("信噪比: "+norm(diff_-ana_)/norm(ana_));
    plot(ratio, diff_-ana_);
    hold on
end
f = @(X)constr_test_ceq(X, constraints);
for i = 1:10
    index = randi(numel(data));
    ratio = linspace(0.1, 10.0, 20);
    ana_ = arrayfun(@(r)derr_ana(r, index), ratio);
    diff_ = arrayfun(@(r)derr_diff(r, index), ratio);
    disp("信噪比: "+norm(diff_-ana_)/norm(diff_));
    plot(ratio, diff_-ana_);
    hold on
end
f = @(X)constr_test_c(X, constraints);
for i = 1:10
    index = randi(numel(data)/4*3);
    ratio = linspace(0.1, 10.0, 20);
    ana_ = arrayfun(@(r)derr_ana(r, index), ratio);
    diff_ = arrayfun(@(r)derr_diff(r, index), ratio);
    disp("信噪比: "+norm(diff_-ana_)/norm(diff_));
    plot(ratio, diff_-ana_);
    hold on
end
function dSSE = derr_diff(ratio, index)
    assert(isscalar(ratio));
    dSSE = (f(set(index, ratio+1e-4))-f(set(index, ratio-1e-4)))/2e-4;
end

```



```

function dSSE = derr_ana(ratio, index)
    assert(isscalar(ratio));
    [~, dSSE] = f(set(index, ratio));
    dSSE = dSSE(index)*data(index);
end
function newdata = set(index, ratio)
    newdata = data;
    newdata(index) = newdata(index)*ratio;
end
end
function X = pack(XYZ, Lz)
    X = [XYZ(:); Lz];
end
function [XYZ, Lz] = unpack(X, N_nodes)
    assert(numel(X) == 4*N_nodes);
    XYZ = reshape(X(1:3*N_nodes), [], 3);
    Lz = X(3*N_nodes+1:end);
End

```

**prob3\_calc.m:** 问题 3 的求解核心算法（含画图输出，检验等）

```

clf()
clear()
rng('default');
close('all');
clear global
global opt_XYZ index W
[~,~, opt_XYZ, ~, ~, index, W, ~] = prob2_calc();clf('reset');
% 画图
clf('reset');
% plot_working_illu_3d(true);
clf('reset');
% plot_2d_texture(true);
clf('reset');
% plot_2d_texture(false);

rec_ratio_work = [0.011020300922443 0.0110003218576217 0.0110518782939506

```





```

0.0110643817438084 0.0109730660550652 0.0110257601080111 0.0110223227972293
0.0110485078071177 0.0110240151884694 0.0110130230869417 0.0110164104558229
0.0110550668326781 0.0110459270545307 0.0109882612985835 0.0110436769357324
0.0110367034749252 0.0110481766120668 0.0110321572590037 0.0110749365711392
0.0110386802631623 0.0110153292520637 0.0110192330970475 0.0109874238643901
0.0110296311887876 0.0109757385346491 0.0109755111105056 0.0110288022004406
0.0109799794453533 0.0110238540462687 0.0110453042446409 0.0110284372401636
0.0110669362584378 0.0110526144686468 0.0110334430375848 0.0109741922281317
0.0110478434113818 0.0110341899191246 0.0110517591813282 0.0110496603830398
0.0110046254841212 0.0110235600140406 0.0110550698235744 0.0110649714478939
0.0110474009766273 0.0110420096870245 0.0110365756602061 0.0110091725112358
0.0110382613793932 0.0110600521586413 0.0110028232503258 0.0110465102463241
0.0110136604630215 0.0110101135220489 0.0110183405377215 0.0110043363971838
0.0109958279906651 0.0110837873612865 0.011065917248056 0.0110277008429329
0.0110127664607068 0.0110545986531244 0.0110345008277633 0.0110429260222507
0.0110505079293461 0.011039406367263 0.0110103799922359 0.0110289593537192
0.0110541998462448 0.0110513385417147 0.0110529854247001 0.0110445898913447
0.0110180088017255 0.011022189720714 0.0110627343553963 0.0110285841846507
0.0110817414943682 0.0109907491678801 0.0110200180167431 0.0110311268543086
0.0110146584529961 0.0110255093771213 0.0110528637613645 0.0110777319262726
0.0110169034101532 0.0110789763845019 0.011086907752962 0.0110068509917417
0.0110202757278154 0.0110403474743263 0.0110425429291141 0.0110162286817121
0.0110382443277745 0.0110068533880955 0.0110452395673006 0.010996333441479
0.011000327429337 0.0110230589753898 0.0110471891212048 0.0110296744899169
0.0110090500347684];

```

```

rec_ratio_base = [0.00809957344004469 0.00816030276752031 0.00815114719028782
0.00809170932206714 0.00813359107499093 0.00815073014613017 0.00808926197249174
0.00808170453150241 0.00806468281488732 0.0081182224458027 0.00809664564432054
0.00803929331956928 0.00813862975913561 0.00812651614405913 0.00807548662594117
0.00809838752451361 0.00809400286528839 0.00811009080826381 0.00810822712967782
0.00809315525807824 0.00806486974020875 0.008134630287794 0.00807892790322006
0.00805598714908852 0.00813180369790822 0.00810668350665907 0.00807828088140091
0.00811052825834018 0.00810927517071732 0.00811143502132163 0.00810309576115239
0.00810982854434356 0.00811344740473194 0.00814411524881943 0.00810455307681079
0.00806302524392838 0.00808600818409412 0.00812354564034997 0.00811186777478928
0.00809026737854994 0.00808663956574123 0.00814128338195877 0.00807940794169834
0.00812115941770167 0.0081024286210036 0.0080924217013025 0.00809501225610337
0.00809902994305578 0.00812940497617186 0.00809554341149597 0.00809631643018426
0.00804895426011689 0.00809575027496546 0.00812928273720676 0.00811334164853845
0.00812004359958459 0.00810587906967159 0.00807971230125123 0.00810458060650855
0.00809081157217132 0.00807630129046536 0.00812220517987459 0.00811206634654426
0.00810709806891063 0.00815164906999222 0.00813766254713846 0.00814321524369861
0.00810908041014188 0.00812934844902824 0.00810621605696549 0.00811063002236979

```





```
0.00812537388975942 0.00809493844001929 0.00811284919748417 0.00812982625291347
0.00811444175548698 0.00812948462742588 0.00812471872392731 0.00809515370106578
0.00810070201885477 0.00809154975241786 0.00809924616852124 0.00808220053817524
0.00812402217146259 0.00811192024738402 0.00806217864697205 0.00809729197336589
0.00814243000329029 0.00814908482651054 0.00805549860700152 0.00812911963465325
0.00809820772628644 0.0080799192180059 0.00812099010446983 0.00811185636211377
0.00810677088931511 0.00812457458272667 0.00811715912091946 0.00810254234380183
0.00812512633851956];
```

```
subplot(2,1,1);
hold on
plot([1:100;1:100],[repmat(mean(rec_ratio_work),size(rec_ratio_work));rec_ratio_work]
,'k-','LineWidth',2);
plot(1:100,rec_ratio_work,'ko','LineWidth',2);
box on
grid on
style('fontname','fontsize');
% ylim([0.01033, 0.01045]);
subplot(2,1,2);
hold on
plot([1:100;1:100],[repmat(mean(rec_ratio_base),size(rec_ratio_base));rec_ratio_base],'k-','LineWidth',2);
plot(1:100,rec_ratio_base,'ko','LineWidth',2);
box on
grid on
style('fontname','fontsize');
fastprint('图片/prob3.检验.多次蒙特卡洛积分的数值波动量');
function [data, XYZ, tri, tri_area] = get_data(WORKING)
%
% 提取工作态或者非工作态下需要使用的所有数据
%
data = Data();
global opt_XYZ index W
XYZ = data.main_node_coordinate; % 产生一个对齐到光线入射角度的坐标格式
if WORKING
    XYZ(index,:) = opt_XYZ; % 主索网调整到工作抛物面
end
XYZ = XYZ*W;
```



```

% 计算三角形的面积
triangle = data.triangle;
X = reshape(XYZ(triangle,1),size(triangle));
Y = reshape(XYZ(triangle,2),size(triangle));
% Z = reshape(XYZ(triangle,3),size(triangle));
triangle_area = 0.5.*abs(X(:,1).*(Y(:,2)-Y(:,3)) + X(:,2).*(Y(:,3)-Y(:,1)) +
X(:,3).*(Y(:,1)-Y(:,2)));

% 筛选出所有可能跟选中顶点有关系的三角形
tri = triangle(any(index(triangle),2),:);
tri_area = triangle_area(any(index(triangle),2));
end
function [rec_ratio, xyz, uvw, data, XYZ] = mc_integral(WORKING)
%
% 提取出当前有效工作界面（由有效三角形定义）中的
% 所有 xyz 映射到 uvw 的射线关系
% (蒙特卡洛法，顺便就完成了积分)
%
[data, XYZ, tri, tri_area] = get_data(WORKING);
area = zeros(size(tri,1),1);
xyz = cell(size(tri));
uvw = cell(size(tri));
N = 10000;
for i = 1:size(tri,1)
    X = XYZ(tri(i,:),1);
    Y = XYZ(tri(i,:),2);
    Z = XYZ(tri(i,:),3);
    t123 = rand(N,2);
    t123(t123(:,1)+t123(:,2)>1,:) = [1, 1] - t123(t123(:,1)+t123(:,2)>1,:);
    t123(:,end+1) = 1-t123(:,1)-t123(:,2); %#ok<AGROW>
    [xyz_,uvw_,ratio] = calc_t_fast(t123, X,Y,Z);
    xyz{i} = xyz_(~isnan(ratio),:);
    uvw{i} = uvw_(~isnan(ratio),:);
    area(i) = mean(~isnan(ratio));
end
assert(~any(isnan(area)));

```



```

% area = area(~isnan(area));
% tri_area = tri_area(~isnan(area));
rec_ratio = sum(tri_area.*area)./(150*150*pi);
if WORKING
    disp("工作接收比: "+mat2str(rec_ratio,17));
else
    disp("基准接收比: "+mat2str(rec_ratio,17));
end
end

function plot_2d_texture(WORKING)
%
% 画图
% 2d, 放大界面纹理, 工作界面和基准态都要绘制
%
[~, xyz, ~, data, XYZ] = mc_integral(WORKING);
tri = data.triangle(:,[1:3 1]).';
plot(reshape(XYZ(tri,1),size(tri)),reshape(XYZ(tri,2),size(tri)), 'k');
hold on;
xyz = vertcat(xyz{:});
plot(xyz(:,1),xyz(:,2),'k');
axis equal
xlim([-90,-30]);
ylim([-30,30]);
style('fontname', 'fontsize');
if WORKING
    fastprint('图片/prob3.结果.纹理对比(工作态)');
else
    fastprint('图片/prob3.结果.纹理对比(基准态)');
end
end

function plot_working_illu_3d(WORKING)
%
% 画图
% 3d, 体现出工作界面的反射光聚焦于焦点
% 这张图只在工作界面时绘制, 不在基准态绘制

```





```

%
assert(WORKING);
[~, xyz, uvw, data, XYZ] = mc_integral(WORKING);
xyz = vertcat(xyz{:});
uvw = vertcat(uvw{:});
xyz = xyz(1:100:end,:);
uvw = uvw(1:100:end,:);
p = plot3(...
    [xyz(:,1),xyz(:,1)+uvw(:,1)].',...
    [xyz(:,2),xyz(:,2)+uvw(:,2)].',...
    [xyz(:,3),xyz(:,3)+uvw(:,3)].','-');
color = mat2cell(repmat(rand(numel(p),1),1,3),ones(numel(p),1),3);
[p.Color] = color{:};
hold('on');

X = XYZ(:,1); Y = XYZ(:,2); Z = XYZ(:,3);
trisurf(data.triangle, X,Y,Z, 'FaceColor', 'None');
axis('equal');
colormap('gray');
colorbar();
grid('on');
box('on');
style('fontsize','fontname');

fastprint('图片/prob3.结果.反射光线聚焦于焦点的示意图');
end

function [xyz, uvw, ratio] = calc_t_fast(t123,X,Y,Z)
%
% 寻找距离入射(x,y)最近的 XYZ 坐标点，并提取出以该点为顶点的所有三角
形
% 依次求解它们的 t1, t2, t3
%
xyz = t123 * [X Y Z];

```



```

% 分界线：上面是命中点，下面是法线
a = X(1)-X(2); b = Y(1)-Y(2);
c = X(2)-X(3); d = Y(2)-Y(3);
uvw = ([a b; c d]\[Z(2)-Z(1); Z(3)-Z(2)]);
uvw(:,end+1) = 1;

uvw_abr = Polar.xyz2abr(-uvw);
uvw_abr(:,2) = pi/2-(pi/2-uvw_abr(:,2)).*2;
uvw_abr(:,3) = 1;
uvw = -Polar.abr2xyz(uvw_abr);

ratio = -(1-Constant.FR_ratio).*Constant.sphere_radius-xyz(:,3))./uvw(:,3);

ratio(ratio < 0) = NaN; % 同向达不到馈源舱的情况下 pass
dst = xyz+ratio.*uvw;
assert(all(abs(dst(:,3))+(1-Constant.FR_ratio).*Constant.sphere_radius) < sqrt(eps) |
isnan(ratio)));
ratio(sum(dst(:,1:2)).*dst(:,1:2),2) > 0.5.*0.5) = NaN; % 反射馈源舱平面时超过馈
源舱圆圈的情况下 pass
ratio(sum(xyz(:,1:2)).*xyz(:,1:2),2) < 0.5.*0.5) = NaN; % 入射被馈源舱遮挡的情况
下 pass
ratio(sum(xyz(:,1:2)).*xyz(:,1:2),2) > 150.*150) = NaN; % 超出 150 口径圆面 pass

xyz(isnan(ratio),:) = NaN;
uvw = ratio.*uvw;
assert(size(xyz,1) == size(uvw,1));
end

prob1.m: 问题 1 的结果可视化、检验
% 本文件原名 prob1.m
%#ok<*DEFNU>
clear
clear global
close all
[Lf, angle300] = prob1_calc();

clf reset

```



```
illu2d(Lf, angle300);
```

```
clf reset
```

```
plot3d(Lf, angle300)
```

```
clf reset
```

```
error_plot(Lf, angle300)
```

```
function error_plot(Lf, angle300) % 横向剖面高低差
    angle = linspace(angle300, -angle300+pi, 100);
    Lw = Polar.para(Lf, angle);
    err = Constant.sphere_radius-Lw;
    plot([-Lw.*cos(angle); -Lw.*cos(angle)], [0.*err; err], '-k', 'LineWidth',2);
    hold('on');
    grid('on');
    plot(-Lw.*cos(angle), err, 'ok', 'LineWidth',2);
    xlim([-150, 150]);
    ylim([-0.6, 0.6]);
    style('fontsize', 'fontname');
    % fastprint('图片/prob1.检验.剖面上的高低差(径向高度 vs 基准球面半径)');
end

function plot3d(Lf, angle300) % 3d 高低差
    d = Data();
    XYZ = d.main_node_coordinate;
    ABR = Polar.xyz2abr(XYZ);
    index = ABR(:,2) >= pi-angle300;
    X = XYZ(:,1); Y = XYZ(:,2); Z = XYZ(:,3);
    err = -Polar.para(Lf, ABR(:,2));
    err(~index) = NaN;
    trisurf(d.triangle,X,Y,Z,err+Constant.sphere_radius,'FaceColor','interp');
    view(2);
    colorbar();
    axis('equal');
    style('fontsize', 'fontname');
    colormap('gray');
```





```

box('on');
% fastprint('图片/prob1.结果.3D 最优抛物面高低差(径向高度)');
end
function illu2d(Lf, angle300) % 横向剖面最优示意图
    font_opts = {'VerticalAlignment', 'middle', 'HorizontalAlignment', 'center', 'FontSize',
13, 'Interpreter', 'latex'};
    clf('reset');
    hold on
    [spher_rad, focal_rad, angle] = Constant.get_rad_angle();
    p_S = illu.p_S;
    p_P = [0 -focal_rad];
    p_Q = [0 -focal_rad-0.5*Lf];
    p_T = [0 -focal_rad-Lf];
    hold('on');
    % 基准圆弧
    plot(-spher_rad*cos(angle), -spher_rad*sin(angle), 'k', 'LineWidth', 2);
    % 焦面圆弧
    plot(-focal_rad*cos(angle), -focal_rad*sin(angle), 'k', 'LineWidth', 2);
    % 标记定点
    plot(p_S(1), p_S(2), '*k', 'MarkerSize', 14, 'LineWidth', 2);
    plot(p_P(1), p_P(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
    plot(p_Q(1), p_Q(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
    plot(p_T(1), p_T(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
    plot(0, 0, 'k', 'MarkerSize', 22, 'LineWidth', 2);
    text(p_S(1)+20, p_S(2), '\boldmathSS$', font_opts{:});
    text(p_P(1)+15, p_P(2)+18, '\boldmathSP$', font_opts{:});
    text(p_Q(1)+15, p_Q(2)+18, '\boldmathSQ$', font_opts{:});
    text(p_T(1)-18, p_T(2)+18, '\boldmathST$', font_opts{:});
    text(0+20, 0, '\boldmathSC$', font_opts{:});
    % SCPQT 中轴线
    plot([p_S(1), p_T(1)], [p_S(2), p_T(2)], '--k', 'LineWidth', 2);
    % 准线
    plot(p_T(1)+[-300 300], p_T(2)+[0 0], 'k', 'LineWidth', 2);
    plot(p_T(1)+[0 30 30], p_T(2)+[30 30 0], 'k', 'LineWidth', 2); % 垂足
    % 计算抛物线
    Lw = Polar.para(Lf, angle);
    L300 = Polar.para(Lf, angle300);

```



```

p_x = [-L300.*cos(angle300) -L300.*sin(angle300)];
% 抛物线
plot(-Lw.*cos(angle), -Lw.*sin(angle), '-.k', 'LineWidth', 2);
plot(p_x(1), p_x(2), '.k', 'MarkerSize', 22); % 300 口径点
plot([p_P(1) p_x(1)], [p_P(2) p_x(2)], ':k', 'LineWidth', 2); % 焦点连线
plot([p_x(1) p_x(1)], [p_T(2) p_x(2)], ':k', 'LineWidth', 2); % 准线连线
text(p_x(1), p_T(2)-20, '\boldmath$x=300/2$', font_opts{:});
% 格式
axis('equal');
set(gca,'Visible','off');
% ylim([-300, 200]);
style('fontsize','fontname');
% fastprint('图片/prob1.结果.2D 剖面最优抛物线');
end

```

## prob2.m: 问题 2 的结果可视化、检验

```

% 本文件原名 prob2.m
clc
clf
clear
rng('default');
close all
clear('global');
data = Data();
[fval, opt_X, opt_XYZ, opt_Lz, opt_ABR, index, W, Winv, ...
    edge, edge_distSQ, motor_hi, motor_unit, motor_distSQ] = prob2_calc();
close all
xlsx_output(opt_XYZ, opt_Lz, index, Winv)
disp("问题 2 最小误差平方和: "+fval);
return
clf reset
errplot_3d(data, opt_XYZ, opt_ABR, index);

clf reset
plot_3d_lz(data, opt_XYZ, opt_Lz, index);

```



```

clf reset
plot_2d_main_node_dist(opt_XYZ, edge, edge_distSQ);

clf reset
plot_2d_fiterr(opt_ABR);

function xlsx_output(opt_XYZ, opt_Lz, index, Winv)
    if exist('result.xlsx', 'file')
        delete('result.xlsx');
    end
    copyfile('数据/4. 工作抛物面的主索节点.xlsx', 'result.xlsx');
    % 理想抛物面顶点坐标
    XYZ = Polar.abr2xyz([0 pi/2 Polar.para(Constant.Lf, pi/2)]);
    XYZ = XYZ*Winv';
    t1 = table(XYZ(1), XYZ(2), XYZ(3));
    disp("理想抛物面顶点坐标");
    disp(head(t1));
    writetable(t1, 'result.xlsx', 'Sheet', '理想抛物面顶点坐标', 'Range', 'A2:C2',
'WriteVariableNames', false);
    % 调整后主索节点编号及其坐标
    data = readtable('数据/1. 主索节点坐标和编号.csv');
    t2 = table(data.x____(index), opt_XYZ(:,1), opt_XYZ(:,2), opt_XYZ(:,3));
    disp("调整后主索节点编号及坐标");
    disp(head(t2));
    writetable(t2, 'result.xlsx', 'Sheet', '调整后主索节点编号及坐标', 'Range', ['A2:D'
num2str(height(t2)+1)], 'WriteVariableNames', false);
    % 促动器顶端伸缩量
    t3 = table(data.x____(index), opt_Lz);
    disp("促动器顶端伸缩量");
    disp(head(t3));
    writetable(t3, 'result.xlsx', 'Sheet', '促动器顶端伸缩量', 'Range', ['A2:B'
num2str(height(t3)+1)], 'WriteVariableNames', false);
    xlsx_check();
end
function xlsx_check()

```





```

t1 = readtable('result.xlsx','Sheet','理想抛物面顶点坐标');
t2 = readtable('result.xlsx','Sheet','调整后主索节点编号及坐标');
t3 = readtable('result.xlsx','Sheet','促动器顶端伸缩量');
[W, ~] = Polar.compose_rotate(Constant.alpha, Constant.beta);
% 抛物面顶端 W 旋转后的取值 Lw 要与 focal_rad+Lf/2 一样
ABR = Polar.xyz2abr(t1{1,1:3}*W');
err = ABR(3)-((1-Constant.FR_ratio)*Constant.sphere_radius+Constant.Lf/2);
disp('检查: 抛物面顶端旋转后是否是 focal_rad+Lf/2 的取值(相减误差): '+err);
assert(abs(err) < eps);
% 重新构建 index
raw_code = readtable('数据/1. 主索节点坐标和编号.csv');
raw_code = raw_code.x____;
index_code = string(t2.x____);
index = cellfun(@(s)any(s == index_code), raw_code);
assert(all(string(raw_code(index)) == index_code));
disp('构建 index 成功');
% 0.6
assert(all(t3{:,2}<0.6));
assert(all(t3{:,2}>-0.6));
disp('Lz 没有超过±0.6: 成功');
% 007
data = Data();
edge = data.edge(all(index(data.edge),2),:); % 过滤节点之间的连边
index2(index) = 1:sum(index); % 构建过滤之后的新的节点下标顺序
edge = index2(edge); % 使用新的节点下标对连边节点序号重定向
assert(all(all(edge))); % 连边节点序号不能有空泡
raw_XYZ = data.main_node_coordinate(index,:);
opt_XYZ = t2{:,2:4};
raw_edist = raw_XYZ(edge(:,1),:)-raw_XYZ(edge(:,2),:);
raw_edist = sqrt(sum(raw_edist.*raw_edist,2));
opt_edist = opt_XYZ(edge(:,1),:)-opt_XYZ(edge(:,2),:);
opt_edist = sqrt(sum(opt_edist.*opt_edist,2));
ratio = (opt_edist./raw_edist)*100-100;
assert(all(ratio<0.07));
assert(all(ratio>-0.07));
disp('节点之间的距离变化率没有超过±0.07%: 成功');
% motor 距离(下拉索)

```



```

motor_unit = data.motor_higher_base(index,:)-data.motor_lower_base(index,:);
motor_unit = motor_unit./sqrt(sum(motor_unit.*motor_unit,2));
assert(all(abs(sum(motor_unit.*motor_unit,2)-1) < sqrt(eps)));
motor_top = data.motor_higher_base(index,:)+t3{:,2}.*motor_unit;
opt_pdist = opt_XYZ-motor_top;
opt_pdist = sqrt(sum(opt_pdist.*opt_pdist,2));
raw_pdist = data.main_node_coordinate(index,:)-data.motor_higher_base(index,:);
raw_pdist = sqrt(sum(raw_pdist.*raw_pdist,2));
assert(max(abs(raw_pdist-opt_pdist)) < sqrt(eps));
disp('下拉索距离不变: 成功');
% 最优贴合程度
opt_ABR = Polar.xyz2abr(opt_XYZ*W);
err = opt_ABR(:,3)-Polar.para(Constant.Lf, opt_ABR(:,2));
assert(max(abs(err)) < 1e-5);
disp('贴合程度至多 1e-5: 成功');
end

function plot_2d_fiterr(opt_ABR) %ok<*DEFNU>
    Lw = Polar.para(Constant.Lf, opt_ABR(:,2));
    err = opt_ABR(:,3)-Lw;
    bar(sort(err), 'k');
    grid('on');
    style('fontsize', 'fontname');
    xticks([xticks numel(err)]);
    fastprint('图片/prob2.检验.主网节点与理想抛物面之间的球面径向误差');
end

function plot_2d_main_node_dist(opt_XYZ, edge, edge_distSQ)
    errdist = opt_XYZ(edge(:,1),:)-opt_XYZ(edge(:,2),:);
    edist = 100*(sqrt(sum(errdist.*errdist,2))./sqrt(edge_distSQ)-1);
    bar(sort(edist), 'k');
    grid('on');
    style('fontsize', 'fontname');
    xticks([xticks numel(edist)]);
    fastprint('图片/prob2.检验.主网节点之间距离的变化率(百分比)');
end

```



```

function plot_3d_lz(data, opt_XYZ, opt_Lz, index) % 三维促动器提升量(Flat)
    raw_XYZ = data.main_node_coordinate;
    raw_XYZ(index,:) = opt_XYZ;
    X = raw_XYZ(:,1); Y = raw_XYZ(:,2); Z = raw_XYZ(:,3);
    Lz = NaN(size(index));
    Lz(index) = opt_Lz;
    trisurf(data.triangle, X,Y,Z, Lz);
    colorbar();
    view(2);
    axis('equal');
    style('fontsize', 'fontname');
    colormap('gray');
    cc = caxis();
    cc(2) = 0.6;
    caxis(cc);
    box('on');
    fastprint('图片/prob2.结果.3D 促动器提升量');
end

```

```

function errplot_3d(data, opt_XYZ, opt_ABR, index) % 三维拟合误差(interp)
    raw_XYZ = data.main_node_coordinate;
    raw_XYZ(index,:) = opt_XYZ;
    X = raw_XYZ(:,1); Y = raw_XYZ(:,2); Z = raw_XYZ(:,3);
    err = NaN(size(index));
    err(index) = abs(Polar.para(Constant.Lf, opt_ABR(:,2)) - opt_ABR(:,3));
    trisurf(data.triangle, X,Y,Z, err, 'FaceColor', 'interp');
    colorbar();
    view(2);
    axis('equal');
    style('fontsize', 'fontname');
    colormap('gray');
    box('on');
    fastprint('图片/prob2.结果.3D 理想曲面拟合误差');
end

```

### 10.1.2 常数定义与数据处理





### Constant.m: 程序用到的所有常量数值定义

% 本文件原名 Constant.m

```
classdef Constant
```

```
    properties(Constant)
```

```
        FR_ratio = 0.466; % 焦距比
```

```
        sphere_radius = 300.4; % 基准态球面半径，单位是米
```

```
        sphere_half_caliber = 500/2; % 基准态球面的口径的一半
```

```
        paraboloid_half_caliber = 300/2; % 工作状态球面的口径的一半
```

```
        Lf = 280.85417567168855; % 第一问算出的最优焦距，用作求解完的检查
```

```
        alpha = deg2rad(36.795); % 第二问平面角
```

```
        beta = deg2rad(78.169); % 第二问仰角
```

```
    end
```

```
    methods(Static)
```

```
        function [spher_rad, focal_rad, angle] = get_rad_angle()
```

```
            spher_rad = Constant.sphere_radius;
```

```
            spher_cli = Constant.sphere_half_caliber;
```

```
            focal_rad = (1-Constant.FR_ratio) * spher_rad;
```

```
            angle = pi/2-asin(spher_cli/spher_rad); % 使用半径和半口径计算出俯角，
```

俯角变成仰角

```
            angle = linspace(angle, pi-angle, 100);
```

```
        end
```

```
    end
```

```
end
```

### Data.m: 将题目附件 123 映射到程序内数组

% 本文件原名 Data.m

```
classdef Data
```

```
    properties
```

```
        main_node_code % 主索节点的编号，strings 转序数
```



```

main_node_coordinate % 主索节点的坐标, XYZ
motor_lower % 促动器下端点的坐标, XYZ
motor_higher_base % 促动器上端点的坐标, XYZ (基准态)
triangle % 三角形反射面板的顶点序号, string 转序数
edge % 不重复的连边顶点序号, 按照左小右大表示, 格式是序数
end
methods
function d = Data
    main_node = readtable('数据/1. 主索节点坐标和编号.csv');
    d.main_node_code = string(main_node{:,1});
    d.main_node_coordinate = main_node{:,2:end};

    motor = readtable('数据/2. 促动器上下端点坐标和编号.csv');
    assert(all(d.main_node_code == string(motor{:,1})));
    d.motor_lower = motor{:,2:4};
    d.motor_higher_base = motor{:,5:end};

    triangle = readtable('数据/3. 反射面板的顶点编号.csv');
    d.triangle = string(triangle{:, :});

    d.check()

    d = d.convert_node_id();
    d.edge = d.get_edges();

end
function e = get_edges(d)
    tri = d.triangle;
    e = [tri(:,1:2); tri(:,2:3); tri(:,1:3)];
    e(e(:,1) > e(:,2), :) = e(e(:,1) > e(:,2), [2,1]);
    e = unique(e, 'rows');
    assert(size(e,1) == 6525);
    assert(~any(e(:,1) == e(:,2)));
end
function check(d)
    assert(isstring(d.main_node_code));

```



```

assert(numel(d.main_node_code) == 2226);
assert(size(d.main_node_coordinate,1) == 2226);
assert(size(d.main_node_coordinate,2) == 3);

assert(ismatrix(d.motor_lower));
assert(ismatrix(d.motor_higher_base));
assert(size(d.motor_lower,1) == 2226);
assert(size(d.motor_lower,2) == 3);
assert(size(d.motor_higher_base,1) == 2226);
assert(size(d.motor_higher_base,2) == 3);

assert(isstring(d.triangle))
assert(size(d.triangle,1) == 4300);
assert(size(d.triangle,2) == 3);
end

function d = convert_node_id(d)
    % 先给主节点排序，确定一个顺序查找表 abc(i)，并且有 index(i) == [原
    % 再给 triangle 排序，确定一个顺序表 xyz(j)，并且有 index2(j) == [tri

    [abc, index] = sort(d.main_node_code);
    [xyz, index2] = sort(d.triangle(:));
    j = 1;
    indexed = zeros(size(xyz));
    for i = 1:numel(abc)
        while abc(i) == xyz(j)
            indexed(j) = index(i);
            j = j+1;
            if j > numel(xyz)
                break
            end
        end
    end
    indexed(index2) = indexed;
    indexed = reshape(indexed, [], 3);
    assert(all(all(d.main_node_code(indexed) == d.triangle)));

```





```

        d.main_node_code = 1:numel(d.main_node_code);
        d.main_node_code = d.main_node_code(:);
        d.triangle = indexed;
    end

    function length = get_pull_length(d)
        main = d.main_node_coordinate;
        higher = d.motor_higher_base;
        length = sqrt(((main-higher).*(main-higher)) * [1 1 1]');
    end
end
end
end

```

### **Polar.m: 极坐标与直角坐标相互转换，抛物面径向距离**

% 本文件原名 Polar.m

```

classdef Polar
    methods(Static)
        function [L, dL_dLf, dL_dangle] = para(Lf, angle)
            %
            % 极坐标下，给定抛物线焦距，在特定的 omega(angle)下
            % 输出相应的距离坐标 L
            %
            cos_angle = cos(angle); % diff(cos) == -sin
            sin_angle = sin(angle); % diff(sin) == cos
            cos2_angle = cos_angle.*cos_angle;

            RF = (1-Constant.FR_ratio)*Constant.sphere_radius;
            index = cos_angle == 0 | abs(angle - pi/2) < eps;

            delta = Lf.*Lf + 2.*RF.*Lf.*cos2_angle;
            L = (-Lf.*sin_angle+sqrt(delta))./cos2_angle;
            L(index) = (Lf./2+RF)./sin_angle(index);

            ddelta_dLf = 2.*Lf + 2.*RF.*cos2_angle;
            dL_dLf = -sin_angle./cos2_angle;
            dL_dLf = dL_dLf+1./cos2_angle.*0.5./sqrt(delta).* ddelta_dLf;
        end
    end
end

```



```

dL_dLf(index) = 1./(2.*sin_angle(index));

ddelta_dangle = 2.*RF.*Lf.*2.*cos_angle.*-sin_angle;
dL_dangle = -Lf./cos2_angle.*cos_angle - L./cos2_angle.*2.*cos_angle.*-
sin_angle;

dL_dangle = dL_dangle+1./cos2_angle.*0.5./sqrt(delta).*ddelta_dangle;
dL_dangle(index) = -L(index)./sin_angle(index).*cos_angle(index);
end
function [abr, dalpha, dbeta, dr] = xyz2abr(XYZ)
%
% 笛卡尔坐标转极坐标
% 输出:
%   abr   : [alpha beta r]
%   dalpha: d{alpha}/d{[X Y Z]}
%   dbeta : d{beta}/d{[X Y Z]}
%   dr    : d{r}/d{[X Y Z]}
%
X = XYZ(:,1);
Y = XYZ(:,2);
Z = XYZ(:,3);
norm2_sq = X.*X+Y.*Y;
norm3_sq = norm2_sq+Z.*Z;
norm2 = sqrt(norm2_sq);
norm3 = sqrt(norm3_sq);
r = norm3;
beta = asin(-Z./r);
alpha = acos(-X./norm2);
alpha(Y>=0) = 2*pi-alpha(Y>=0);
alpha(~norm2) = 0;
abr = [alpha, beta, r];
if nargin == 1
    return
end
ZEROS = zeros(size(Z));

dnorm2_sq = 2.*[X Y ZEROS];

```



```

dnorm3_sq = 2.*[X Y Z];
dnorm2 = 0.5.*dnorm2_sq./norm2;
dnorm3 = 0.5.*dnorm3_sq./norm3;
dr = dnorm3;
dbeta = r./norm2.*([ZEROS ZEROS -1./r] + (Z./r).*(dr./r));
dalpha = -abs(norm2./Y).*([-1./norm2 ZEROS ZEROS] +
(X./norm2).*(dnorm2./norm2));
dalpha(Y>=0, :) = -dalpha(Y>=0, :);
dalpha(~norm2, :) = 0;

```

end

function XYZ = abr2xyz(abr)

```

%
% 极坐标格式转笛卡尔坐标
%
alpha = abr(:,1);
beta = abr(:,2);
r = abr(:,3);
Z = -r.*sin(beta);
r = r.*cos(beta);
X = -r.*cos(alpha);
Y = -r.*sin(alpha);
XYZ = [X, Y, Z];

```

end

function [W, Winv] = compose\_rotate(alpha, beta)

```

%
% 组合旋转（3d 旋转矩阵）
% 先逆向旋转以清除 alpha 角(xOy 平面上的)
% 再正向旋转以将 beta 补偿到 pi/2
%
Wxy = [
cos(alpha) sin(alpha) 0
-sin(alpha) cos(alpha) 0
0 0 1];
Wxy_inv = [
cos(alpha) -sin(alpha) 0
sin(alpha) cos(alpha) 0
0 0 1];

```





```

        sin(alpha) cos(alpha) 0
        0                0 1
    ];
    Wxz = [
        cos(pi/2-beta) 0 -sin(pi/2-beta)
        0                1                0
        sin(pi/2-beta) 0 cos(pi/2-beta)
    ];
    Wxz_inv = [
        cos(pi/2-beta) 0 sin(pi/2-beta)
        0                1                0
        -sin(pi/2-beta) 0 cos(pi/2-beta)
    ];
    W = Wxz*Wxy;
    Winv = Wxy_inv*Wxz_inv;
    assert(all(all(abs(W*Winv-eye(3)) < eps())));
end
end
end

```

### 10.1.3 画图程序

#### illu.m: 与示意图有关的常量定义

% 本文件原名 illu.m

```

classdef illu
    properties(Constant)
        p_S = [0 150]; % 被观测点的方位
    end
end

```

#### illu1.m: 剖面直角坐标示意图

% 本文件原名 illu1.m

```

font_opts = {'VerticalAlignment', 'middle', 'HorizontalAlignment', 'center', 'FontSize', 10,
'Interpreter', 'latex'};
close all
hold on

```



```

[spher_rad, focal_rad, angle] = Constant.get_rad_angle();
p_spher      = [-spher_rad*cos(angle( 1)), -spher_rad*sin(angle( 1))];
p_spher_end = [-spher_rad*cos(angle(end)), -spher_rad*sin(angle(end))];
p_focal      = [-focal_rad*cos(angle( 1)), -focal_rad*sin(angle( 1))];
p_focal_end = [-focal_rad*cos(angle(end)), -focal_rad*sin(angle(end))];
v_sph = [cos(pi/2+angle(1)), sin(pi/2+angle(1))];
p_S = illu.p_S;
p_P = [0 -focal_rad];
p_Q = [0 -spher_rad+20];
p_T = [0 p_P(2)+2*(p_Q(2)-p_P(2))];

hold('on');
% 基准圆弧
draw_2d_arc_focal_and_base(spher_rad, focal_rad, angle);
% 焦距比和 R300
standard_distance_tag(angle, p_spher, p_spher_end, p_focal, p_focal_end, v_sph,
font_opts);

% 标记定点
plot(p_S(1), p_S(2), '*k', 'MarkerSize', 14, 'LineWidth', 2);
plot(p_P(1), p_P(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
plot(p_Q(1), p_Q(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
plot(p_T(1), p_T(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
plot(0,0,'k', 'MarkerSize', 22, 'LineWidth', 2);

text(p_S(1)+10, p_S(2)+30, '被观测体', font_opts{1:end-2});
text(p_S(1)+20, p_S(2), '\boldmathSS$', font_opts{:});
text(p_P(1)+15, p_P(2)+15, '\boldmathSP$', font_opts{:});
text(p_Q(1)+15, p_Q(2)+15, '\boldmathSQ$', font_opts{:});
text(p_T(1)+15, p_T(2)+10, '\boldmathST$', font_opts{:});
text(0+15, 0, '\boldmathSC$', font_opts{:});

% SCPQT 中轴线
plot([p_S(1), p_T(1)], [p_S(2), p_T(2)], '--k', 'LineWidth', 2);

% 文字 tag
text(p_spher_end(1)-40, p_spher_end(2)-70, '基准球面',...

```



```

    'Rotation', rad2deg(angle(end-11)-pi/2), font_opts{1:end-2});
text(p_focal_end(1), p_focal_end(2)-25, '焦面',...
    'Rotation', rad2deg(angle(end-4)-pi/2), font_opts{1:end-2});
text(-160, p_T(2)+15, '抛物面的准线', font_opts{1:end-2});

% PQ 长度
plot(p_P(1)+[-350 0], p_P(2)+[0 0], 'k', 'LineWidth', 2);
plot(p_Q(1)+[-350 0], p_Q(2)+[0 0], 'k', 'LineWidth', 2);
myquiver(8,[p_P(1) p_Q(1)]-325, [p_P(2) p_Q(2)], 'k', 'LineWidth', 2);
text((p_P(1)+p_Q(1))/2-300, (p_P(2)+p_Q(2))/2, '\boldmath$|PQ|=L_f/2$',...
    'Rotation', 90, font_opts{:});

% QT 长度以及垂直的准线
plot(p_T(1)+[-350 300], p_T(2)+[0 0], 'k', 'LineWidth', 2);
myquiver(8,[p_Q(1) p_T(1)]-325, [p_Q(2) p_T(2)], 'k', 'LineWidth', 2);
text((p_Q(1)+p_T(1))/2-300, (p_Q(2)+p_T(2))/2, '\boldmath$|QR|=L_f/2$',...
    'Rotation', 90, font_opts{:});
plot(p_T(1)+[0 30 30], p_T(2)+[30 30 0], 'k', 'LineWidth', 2); % 垂足

% 计算抛物线
Lf = 2*(p_P(2)-p_Q(2));
L = Polar.para(Lf, angle);

xangle = fzero(@(angle)abs(Polar.para(Lf, angle)*cos(angle))-150, [pi/2, angle(end)]);
xL = Polar.para(Lf, xangle);
p_x = [-xL.*cos(xangle) -xL.*sin(xangle)];

plot(-L.*cos(angle), -L.*sin(angle), '-k', 'LineWidth', 2);
plot(p_x(1), p_x(2), 'k', 'MarkerSize', 22);
plot([p_P(1) p_x(1)], [p_P(2) p_x(2)], 'k', 'LineWidth', 2);
plot([p_x(1) p_x(1)], [p_T(2) p_x(2)], 'k', 'LineWidth', 2);

text(p_spher_end(1)-50, p_spher_end(2)-10, '抛物面',...
    'Rotation', rad2deg(angle(end-14)-pi/2), font_opts{1:end-2});
title('\boldmath$y=\frac{x^2}{2L_f}-\frac{L_f}{2}-(1-0.446)R$', 'Interpreter', 'latex');
text(p_x(1), p_T(2)-15, '\boldmath$х=300/2$', font_opts{:});

```





```

%
axis('equal');
% ylim([-525, 200]);
set(gca, 'Visible', 'off');
style('fontname');
fastprint('图片/prob1.示意图 1.准线等辅助线');
function draw_2d_arc_focal_and_base(spher_rad, focal_rad, angle)
    % 基准圆弧
    plot(-spher_rad*cos(angle),-spher_rad*sin(angle), 'k', 'LineWidth', 2);
    % 焦面圆弧
    plot(-focal_rad*cos(angle),-focal_rad*sin(angle), 'k', 'LineWidth', 2);
end
function standard_distance_tag(angle, p_spher, p_spher_end, p_focal, ~, v_sph, font_opts)

    % 焦径比
    plot(p_spher(1)+[0 50*v_sph(1)], p_spher(2)+[0 50*v_sph(2)], 'k', 'LineWidth', 2);
    plot(p_focal(1)+[0 50*v_sph(1)], p_focal(2)+[0 50*v_sph(2)], 'k', 'LineWidth', 2);
    myquiver(8,[p_spher(1) p_focal(1)]+25*v_sph(1), [p_spher(2)
p_focal(2)]+25*v_sph(2), 'k', 'LineWidth', 2);
    text(...
        (p_spher(1)+p_focal(1)+80*v_sph(1))/2, ...
        (p_spher(2)+p_focal(2)+80*v_sph(2))/2,...
        '\boldmathSF=0.466RS', font_opts{:}, 'Rotation', rad2deg(angle(1)));

    % 基准半径
    plot(p_spher(1)+100*[0 v_sph(1)], p_spher(2)+100*[0 v_sph(2)], 'k', 'LineWidth', 2);
    plot(
        0+100*[0 v_sph(1)], 0+100*[0 v_sph(2)], 'k', 'LineWidth',
2);
    myquiver(15,[p_spher(1) 0]+75*v_sph(1), [p_spher(2) 0]+75*v_sph(2), 'k',
'LineWidth', 2);
    text(...
        (p_spher(1)+0+180*v_sph(1))/2, ...
        (p_spher(2)+0+180*v_sph(2))/2, ...
        '\boldmathSR=300.4\mathrm{m}S', font_opts{:}, 'Rotation', rad2deg(angle(1)));

    % D=500m

```



```

myquiver(20,[-250 250], [-450 -450],':k', 'LineWidth', 2);
plot([-250 p_spher(1)], [-460 p_spher_end(2)],':k', 'LineWidth', 2);
plot([ 250 p_spher_end(1)], [-460 p_spher_end(2)],':k', 'LineWidth', 2);
text(0, -440, '\boldmathSD=500\mathrm{m}S', font_opts{:});
end
function myquiver(r,x,y,varargin)
    rad = atan2(y(1)-y(2), x(1)-x(2));
    rad = [rad+deg2rad(30), rad-deg2rad(30)];
    plot(x,y,varargin{:});
    L = sqrt((x(1)-x(2))*(x(1)-x(2))+(y(1)-y(2))*(y(1)-y(2)))/r;
    plot(x(1)-[0,L]*cos(rad(1)),y(1)-[0,L]*sin(rad(1)),varargin{:});
    plot(x(1)-[0,L]*cos(rad(2)),y(1)-[0,L]*sin(rad(2)),varargin{:});
    plot(x(2)+[0,L]*cos(rad(1)),y(2)+[0,L]*sin(rad(1)),varargin{:});
    plot(x(2)+[0,L]*cos(rad(2)),y(2)+[0,L]*sin(rad(2)),varargin{:});
end

```

### illu2.m: 剖面极坐标示意图

```

% 本文件原名 illu2.m
% 剖面示意图
font_opts = {'VerticalAlignment', 'middle', 'HorizontalAlignment', 'center', 'FontSize', 18,
'Interpreter', 'latex'};
close all
[spher_rad, focal_rad, angle] = Constant.get_rad_angle();
p_S = [pi/2 illu.p_S(2)];
p_P = [-pi/2 focal_rad];
p_Q = [-pi/2 spher_rad-20];
p_T = [-pi/2 p_P(2)+2*(p_Q(2)-p_P(2))];
% 基准圆弧
polarplot(angle+pi,angle.*0+spher_rad,'k','LineWidth', 2);
hold('on');
% 焦面圆弧
polarplot(angle+pi,angle.*0+focal_rad,'k','LineWidth', 2);
% 标记定点
polarplot(p_S(1), p_S(2),'k','MarkerSize', 14, 'LineWidth', 2);
polarplot(p_P(1), p_P(2),'k','MarkerSize', 22, 'LineWidth', 2);
polarplot(p_Q(1), p_Q(2),'k','MarkerSize', 22, 'LineWidth', 2);
polarplot(p_T(1), p_T(2),'k','MarkerSize', 22, 'LineWidth', 2);

```



```

polarplot(0,0,'k', 'MarkerSize', 22, 'LineWidth', 2);
text(p_S(1)-0.20, p_S(2), '\boldmathSS$', font_opts{:});
text(p_P(1)+0.12, p_P(2)-18, '\boldmathSP$', font_opts{:});
text(p_Q(1)+0.08, p_Q(2)-23, '\boldmathSQ$', font_opts{:});
text(p_T(1)+0.06, p_T(2)-18, '\boldmathST$', font_opts{:});
text(0.15, -30, '\boldmathSC$', font_opts{:});
% SCPQT 中轴线
polarplot([p_S(1), p_T(1)], [p_S(2), p_T(2)], '--k', 'LineWidth', 2);
% 计算抛物线
Lf = 2*(p_Q(2)-p_P(2));
L = Polar.para(Lf, angle);
xangle = fzero(@(angle)abs(Polar.para(Lf, angle)*cos(angle))-150, [pi/2*1.01,
angle(end)]);
xL = Polar.para(Lf, xangle);
theta = linspace(-xangle, xangle-pi, 150);
r = Polar.para(Lf, theta+pi);
r(2:2:end) = spher_rad;
polarplot(reshape(theta,2,[]), reshape(r,2,[]), 'k', 'linewidth', 2, 'color', [0.7, 0.7, 0.7]);
polarplot([0 0], [0 p_T(2)], 'k', 'linewidth', 2);
polarplot(linspace(0,-xangle+pi,100), repmat(60,1,100), 'k', 'linewidth', 2);
text((-xangle+pi)/2-0.05, 90, '\boldmath$\omega$', font_opts{:});
polarplot(angle+pi, L, '-.k', 'LineWidth', 2);
polarplot(xangle+pi, xL, 'k', 'MarkerSize', 22);
polarplot(-xangle, xL, 'k', 'MarkerSize', 22);
polarplot([-xangle xangle], [-p_S(2) spher_rad], 'k', 'linewidth', 2);
polarplot([xangle xangle]+pi, [-p_S(2) spher_rad], 'k', 'linewidth', 2);
% polarplot(-p_x(1), p_x(2), 'k', 'MarkerSize', 22);

set(gca,'RAxisLocation',0);
rlim([0, p_T(2)]);
% 极坐标
style('fontsize', 'fontname');
fastprint('图片/prob1.示意图 2.极坐标下的  $\omega$  角与球面径向误差平方积分');

style.m: 为输出图片统一文字大小和字体
% 本文件原名 style.m
function style(varargin)

```





```

for v = varargin
    switch lower(char(v))
        case 'fontsize'
            set(gca(), 'fontsize', 22);
        case 'fontname'
            set(gca(), 'fontname', 'times new roman');
    end
end
end
end

```

### fastprint.m: 简易绘图输出

```

% 本文件原名 fastprint.m
function r = fastprint(filename)
    assert(isequal(class(filename),'char') || (isequal(class(filename),'string') &&
isscalar(filename)))
    if isstring(filename)
        filename = char(filename);
    end
    if numel(filename) >= 4 && isequal(filename(end-3:end),'.png')
        filename = filename(1:end-4);
    end
    filename = [filename, '_', datestr(now(), 'yyyy.mm.dd.HH_MM'), '.png'];
    print('-dpng', '-r300', filename);
    disp(filename);
    r = @()system(['open ', filename]);
end

```

### 10.2 附件 result.xlsx 数据

cc	X_1	Y_1	Z_1	x_2	X_2	Y_2	Z_2	x_3	X_3	Y_3	Z_3
A0	-0.037	-0.019	-300.468	E9	-32.031	10.370	-298.683	D19	-29.795	-40.965	-296.498
B1	6.075	8.387	-300.197	E10	-28.225	21.953	-298.325	D20	-39.656	-33.755	-296.286
C1	9.845	-3.228	-300.228	A7	-24.374	33.496	-297.478	D21	-49.353	-26.488	-295.567
D1	-0.042	-10.411	-300.340	A12	-18.305	42.019	-296.750	E16	-58.963	-19.174	-294.334
E1	-9.924	-3.233	-300.378	A13	-6.122	42.075	-297.208	E17	-55.469	-7.605	-295.481
A1	-6.142	8.387	-300.279	A14	6.089	42.077	-297.175	E18	-51.882	3.974	-296.120
A3	-0.029	16.796	-299.894	A15	18.270	42.025	-296.661	E19	-48.132	15.602	-296.266
B2	12.177	16.782	-299.574	B11	30.365	41.811	-295.699	E20	-44.303	27.198	-295.913
B3	15.958	5.180	-299.853	B12	34.307	30.377	-296.658	E21	-40.377	38.655	-295.090
C2	19.714	-6.431	-299.625	B13	38.120	18.810	-297.154	A16	-36.388	50.048	-293.802



C3	9.844	-13.621	-299.979	B14	41.890	7.197	-297.160	A23	-30.307	58.569	-292.865
D2	-0.046	-20.793	-299.839	B15	45.599	-4.403	-296.671	A24	-18.214	58.790	-293.809
D3	-9.932	-13.632	-300.140	C11	49.126	-15.970	-295.708	A25	-6.112	58.904	-294.278
E2	-19.801	-6.445	-299.920	C12	39.469	-23.262	-296.719	A26	6.098	58.910	-294.277
E3	-16.033	5.174	-300.070	C13	29.645	-30.474	-297.286	A27	18.203	58.808	-293.816
A2	-12.238	16.781	-299.715	C14	19.764	-37.661	-297.373	A28	30.302	58.601	-292.906
A5	-6.132	25.211	-299.217	C15	9.872	-44.788	-296.967	B22	42.341	58.289	-291.560
A6	6.081	25.211	-299.157	D11	-0.046	-51.731	-296.080	B23	46.360	46.940	-292.933
B4	18.263	25.156	-298.610	D12	-9.971	-44.812	-297.138	B24	50.298	35.499	-293.854
B5	22.077	13.590	-299.128	D13	-19.871	-37.702	-297.706	B25	54.136	24.018	-294.312
B6	25.848	1.978	-299.147	D14	-29.762	-30.524	-297.754	B26	57.904	12.404	-294.302
C4	29.558	-9.623	-298.664	D15	-39.594	-23.314	-297.277	B27	61.536	0.857	-293.820
C5	19.732	-16.835	-299.251	E11	-49.251	-16.019	-296.297	B28	65.064	-10.712	-292.866
C6	9.848	-24.022	-299.355	E12	-45.710	-4.437	-297.192	C22	68.470	-22.250	-291.444
D4	-0.047	-31.149	-298.962	E13	-41.983	7.176	-297.593	C23	58.921	-29.581	-292.846
D5	-9.943	-24.037	-299.523	E14	-38.192	18.796	-297.485	C24	49.261	-36.869	-293.819
D6	-19.830	-16.859	-299.567	E15	-34.360	30.366	-296.881	C25	39.533	-44.077	-294.343
E4	-29.657	-9.650	-299.090	A11	-30.400	41.797	-295.816	C26	29.655	-51.262	-294.405
E5	-25.933	1.964	-299.491	A17	-24.321	50.310	-294.984	C27	19.798	-58.297	-293.993
E6	-22.147	13.583	-299.384	A18	-12.215	50.490	-295.679	C28	9.883	-65.238	-293.097
A4	-18.318	25.154	-298.780	A19	-0.012	50.535	-295.903	D22	-0.041	-72.050	-291.714
A8	-12.207	33.580	-298.174	A20	12.191	50.497	-295.647	D23	-9.973	-65.265	-293.252
A9	-0.021	33.609	-298.372	A21	24.298	50.325	-294.938	D24	-19.900	-58.348	-294.306
A10	12.163	33.582	-298.082	B16	36.371	50.074	-293.777	D25	-29.773	-51.330	-294.861
B7	24.327	33.502	-297.314	B17	40.341	38.674	-294.949	D26	-39.666	-44.153	-294.910
B8	28.162	21.961	-298.065	B18	44.247	27.215	-295.655	D27	-49.405	-36.946	-294.447
B9	31.950	10.383	-298.326	B19	48.052	15.622	-295.895	D28	-59.066	-29.649	-293.474
B10	35.696	-1.212	-298.090	B20	51.780	4.002	-295.650	E22	-68.598	-22.304	-291.995
C7	39.366	-12.805	-297.355	B21	55.349	-7.565	-294.932	E23	-65.184	-10.755	-293.388
C8	29.571	-20.028	-298.167	C16	58.830	-19.120	-293.734	E24	-61.640	0.825	-294.281
C9	19.727	-27.220	-298.506	C17	49.215	-26.425	-294.948	E25	-57.986	12.379	-294.673
C10	9.849	-34.370	-298.357	C18	39.524	-33.690	-295.715	E26	-54.193	23.996	-294.572
D7	-0.048	-41.466	-297.709	C19	29.675	-40.905	-296.030	E27	-50.330	35.474	-293.991
D8	-9.947	-34.390	-298.529	C20	19.777	-48.054	-295.863	E28	-46.370	46.906	-292.942
D9	-19.831	-27.253	-298.835	C21	9.876	-55.037	-295.219	A22	-42.331	58.240	-291.444
D10	-29.681	-20.066	-298.620	D16	-0.044	-61.929	-294.080	A30	-36.238	66.746	-290.411
E7	-39.478	-12.844	-297.883	D17	-9.971	-55.063	-295.384	A31	-24.193	67.022	-291.597
E8	-35.785	-1.236	-298.539	D18	-19.883	-48.101	-296.190	A32	-12.108	67.191	-292.311
A33	-0.003	67.253	-292.558	B40	66.541	40.788	-290.135	B54	91.293	-8.632	-286.202
A34	12.104	67.208	-292.347	B41	70.375	29.301	-290.567	B55	94.188	-20.081	-284.713
A35	24.196	67.057	-291.686	B42	74.142	17.667	-290.540	C46	96.952	-31.500	-282.796
A36	36.258	66.805	-290.586	B43	77.774	6.114	-290.051	C47	87.969	-39.103	-284.572
B29	48.271	66.448	-289.062	B44	81.303	-5.462	-289.098	C48	78.873	-46.648	-285.964
B30	52.326	55.137	-290.630	B45	84.713	-17.008	-287.686	C49	69.258	-53.967	-287.085





B31	56.298	43.746	-291.755	C37	87.535	-28.443	-285.963	C50	59.542	-61.201	-287.779
B32	60.181	32.293	-292.426	C38	78.505	-36.026	-287.587	C51	49.699	-68.434	-288.018
B33	63.967	20.792	-292.639	C39	68.946	-43.352	-288.945	C52	39.777	-75.556	-287.798
B34	67.650	9.257	-292.389	C40	59.279	-50.636	-289.876	C53	29.893	-82.551	-287.100
B35	71.227	-2.291	-291.673	C41	49.549	-57.839	-290.363	C54	19.960	-89.414	-285.932
B36	74.692	-13.836	-290.492	C42	39.662	-65.025	-290.393	C55	9.974	-95.695	-284.442
C29	78.040	-25.359	-288.849	C43	29.806	-72.056	-289.953	D46	-0.028	-101.825	-282.496
C30	68.541	-32.707	-290.435	C44	19.891	-78.993	-289.036	D47	-10.037	-95.718	-284.544
C31	58.943	-40.007	-291.601	C45	9.968	-85.802	-287.639	D48	-20.034	-89.462	-286.152
C32	49.257	-47.244	-292.329	D37	-0.034	-92.009	-285.912	D49	-29.981	-82.622	-287.441
C33	39.496	-54.409	-292.606	D38	-10.042	-85.828	-287.762	D50	-39.882	-75.642	-288.246
C34	29.671	-61.487	-292.420	D39	-19.976	-79.046	-289.296	D51	-49.820	-68.528	-288.543
C35	19.796	-68.467	-291.759	D40	-29.907	-72.130	-290.346	D52	-59.673	-61.292	-288.335
C36	9.888	-75.336	-290.617	D41	-39.780	-65.113	-290.898	D53	-69.386	-54.046	-287.609
D29	-0.038	-82.080	-288.989	D42	-49.684	-57.932	-290.945	D54	-78.978	-46.708	-286.384
D30	-9.971	-75.363	-290.758	D43	-59.423	-50.725	-290.482	D55	-88.026	-39.137	-284.816
D31	-19.891	-68.520	-292.050	D44	-69.084	-43.427	-289.508	E46	-96.928	-31.504	-282.780
D32	-29.782	-61.560	-292.850	D45	-78.615	-36.081	-288.031	E47	-94.192	-20.094	-284.743
D33	-39.625	-54.494	-293.150	E37	-87.593	-28.472	-286.211	E48	-91.301	-8.651	-286.235
D34	-49.400	-47.331	-292.945	E38	-84.781	-17.038	-287.952	E49	-87.835	2.922	-287.388
D35	-59.091	-40.087	-292.230	E39	-81.367	-5.492	-289.350	E50	-84.222	14.487	-288.054
D36	-68.679	-32.773	-291.009	E40	-77.823	6.087	-290.250	E51	-80.496	26.126	-288.219
E29	-78.146	-25.405	-289.285	E41	-74.169	17.637	-290.654	E52	-76.637	37.718	-287.904
E30	-74.797	-13.876	-290.926	E42	-70.373	29.261	-290.567	E53	-72.737	49.183	-287.111
E31	-71.321	-2.324	-292.066	E43	-66.510	40.737	-290.003	E54	-68.723	60.570	-285.870
E32	-67.724	9.230	-292.708	E44	-62.549	52.171	-288.973	E55	-64.332	71.524	-284.345
E33	-64.015	20.765	-292.857	E45	-58.511	63.506	-287.497	A46	-59.850	82.377	-282.418
E34	-60.203	32.263	-292.523	A37	-54.066	74.410	-285.741	A57	-53.934	91.265	-280.954
E35	-56.295	43.706	-291.718	A47	-48.157	83.300	-284.378	A58	-42.193	92.123	-282.687
E36	-52.300	55.076	-290.456	A48	-36.390	84.095	-285.899	A59	-30.407	92.825	-283.991
A29	-48.225	66.363	-288.753	A49	-24.323	84.392	-287.107	A60	-18.310	93.059	-284.987
A38	-42.332	75.272	-287.491	A50	-12.215	84.555	-287.859	A61	-6.115	93.246	-285.507
A39	-30.308	75.604	-288.923	A51	0.000	84.640	-288.144	A62	6.114	93.270	-285.586
A40	-18.214	75.831	-289.889	A52	12.220	84.592	-287.976	A63	18.320	93.136	-285.229
A41	-6.111	75.950	-290.387	A53	24.339	84.465	-287.357	A64	30.440	92.945	-284.424
A42	6.112	75.963	-290.424	A54	36.433	84.209	-286.311	B61	86.882	34.648	-285.900
A43	18.221	75.868	-290.008	B48	68.842	60.690	-286.450	B62	90.644	23.006	-285.855
A44	30.331	75.671	-289.148	B49	72.832	49.273	-287.537	B63	94.247	11.347	-285.366
A45	42.385	75.377	-287.858	B50	76.715	37.785	-288.183	B64	97.777	-0.238	-284.408
B37	54.161	74.545	-286.295	B51	80.535	26.169	-288.377	B65	100.767	-11.676	-283.143
B38	58.589	63.610	-287.921	B52	84.236	14.518	-288.112	B66	103.605	-23.119	-281.459
B39	62.600	52.244	-289.251	B53	87.834	2.946	-287.382	C57	97.365	-42.169	-281.274
C58	88.308	-49.739	-282.821	C78	9.989	-115.152	-277.045	D85	-59.765	-92.620	-279.553
C59	79.143	-57.214	-283.985	D67	-0.017	-121.086	-274.685	D86	-69.661	-85.444	-279.588





C60	69.467	-64.491	-284.871	D68	-10.030	-115.161	-277.093	D87	-79.430	-78.142	-279.133
C61	59.694	-71.783	-285.300	D69	-20.028	-109.064	-279.088	D88	-88.707	-70.870	-278.266
C62	49.803	-78.963	-285.290	D70	-29.998	-102.798	-280.657	D89	-97.836	-63.377	-276.976
C63	39.841	-85.988	-284.830	D71	-39.923	-96.370	-281.784	D90	-106.828	-55.780	-275.225
C64	29.925	-92.917	-283.881	D72	-49.931	-89.435	-282.551	D91	-115.639	-48.117	-273.029
C65	19.982	-99.284	-282.614	D73	-59.857	-82.346	-282.834	E79	-124.168	-40.349	-270.452
C66	9.982	-105.496	-280.902	D74	-69.681	-75.115	-282.627	E80	-121.819	-29.049	-272.983
D56	-0.022	-111.520	-278.749	D75	-79.385	-67.752	-281.928	E81	-119.176	-17.670	-275.125
D57	-10.035	-105.514	-280.978	D76	-88.577	-60.308	-280.852	E82	-116.348	-6.238	-276.819
D58	-20.043	-99.323	-282.788	D77	-97.623	-52.766	-279.316	E83	-113.352	5.188	-278.058
D59	-29.999	-92.978	-284.159	D78	-106.510	-45.140	-277.325	E84	-110.109	16.519	-278.882
D60	-39.929	-86.066	-285.205	E67	-115.224	-37.445	-274.889	E85	-106.493	28.163	-279.311
D61	-49.905	-79.049	-285.737	E68	-112.679	-26.095	-277.268	E86	-102.708	39.783	-279.274
D62	-59.806	-71.869	-285.780	E69	-109.952	-14.701	-279.197	E87	-98.777	51.321	-278.781
D63	-69.578	-64.566	-285.327	E70	-107.047	-3.283	-280.670	E88	-94.733	62.393	-277.920
D64	-79.235	-57.270	-284.348	E71	-103.970	8.139	-281.689	E89	-90.445	73.404	-276.686
D65	-88.357	-49.768	-283.025	E72	-100.436	19.793	-282.343	E90	-86.032	84.337	-275.052
D66	-97.341	-42.164	-281.238	E73	-96.735	31.411	-282.525	E91	-81.516	95.144	-273.039
E56	-106.142	-34.496	-279.003	E74	-92.876	42.978	-282.243	A79	-76.826	105.745	-270.714
E57	-103.509	-23.117	-281.173	E75	-88.869	54.474	-281.509	A93	-70.895	114.664	-268.950
E58	-100.689	-11.684	-282.890	E76	-84.636	65.519	-280.446	A94	-59.353	115.816	-271.227
E59	-97.699	-0.255	-284.144	E77	-80.280	76.470	-278.979	A95	-47.642	116.756	-273.140
E60	-94.160	11.320	-285.063	E78	-75.809	87.316	-277.123	C96	98.021	-73.983	-274.572
E61	-90.537	22.965	-285.471	A67	-71.233	98.047	-274.893	C97	88.762	-81.284	-275.447
E62	-86.744	34.587	-285.406	A80	-65.330	106.986	-273.209	C98	79.372	-88.448	-275.953
E63	-82.804	46.125	-284.880	A81	-53.676	107.987	-275.342	C99	69.518	-95.685	-276.086
E64	-78.847	57.558	-283.873	A82	-41.922	108.847	-277.059	C100	59.565	-102.768	-275.793
E65	-74.542	68.557	-282.576	A83	-30.124	109.542	-278.361	C101	49.824	-109.416	-275.121
E66	-70.109	79.472	-280.877	A84	-18.339	109.986	-279.279	C102	40.008	-115.898	-274.039
A56	-65.573	90.257	-278.798	A85	-6.141	110.185	-279.834	C103	30.037	-122.207	-272.553
A68	-59.638	99.124	-277.244	C82	97.899	-63.424	-277.161	C104	20.022	-128.329	-270.645
A69	-47.954	100.055	-279.176	C83	88.707	-70.885	-278.242	C105	10.013	-134.169	-268.357
A70	-36.200	100.834	-280.688	C84	79.396	-78.127	-278.971	D92	-0.010	-139.803	-265.658
A71	-24.387	101.456	-281.780	C85	69.606	-85.408	-279.350	D93	-10.036	-134.154	-268.342
A72	-12.208	101.726	-282.538	C86	59.705	-92.577	-279.295	D94	-20.047	-128.313	-270.646
A73	-0.006	101.845	-282.853	C87	49.737	-99.594	-278.798	D95	-30.065	-122.197	-272.588
A74	12.201	101.792	-282.740	C88	39.955	-106.165	-277.928	D96	-40.037	-115.895	-274.115
C69	97.661	-52.787	-279.407	C89	30.012	-112.533	-276.664	D97	-49.854	-109.419	-275.229
C70	88.546	-60.295	-280.722	C90	20.013	-118.750	-274.961	D98	-59.595	-102.770	-275.911
C71	79.317	-67.710	-281.652	C91	10.008	-124.782	-272.828	D99	-69.541	-95.680	-276.178
C72	69.596	-75.053	-282.268	D79	-0.013	-130.516	-270.317	D100	-79.370	-88.427	-275.970
C73	59.770	-82.276	-282.455	D80	-10.039	-124.781	-272.846	D101	-88.715	-81.229	-275.338
C74	49.852	-89.366	-282.201	D81	-20.049	-118.758	-275.025	D102	-97.917	-73.900	-274.261
C75	39.855	-96.311	-281.499	D82	-30.054	-112.554	-276.788	D103	-106.995	-66.365	-272.751



C76	29.940	-102.754	-280.451	D83	-40.004	-106.198	-278.114	D104	-115.900	-58.724	-270.801
C77	19.980	-109.038	-278.967	D84	-49.793	-99.636	-279.033	D105	-124.530	-51.004	-268.460
E92	-132.966	-43.206	-265.705	E116	-110.344	67.059	-271.045	C152	20.073	-157.672	-255.090
E93	-130.712	-31.938	-268.426	E117	-106.153	78.026	-269.866	C153	10.031	-163.190	-252.225
E94	-128.254	-20.617	-270.723	E118	-101.765	88.980	-268.305	D137	-0.006	-166.787	-250.064
E95	-125.524	-9.201	-272.625	E119	-97.258	99.723	-266.403	D138	-10.040	-163.115	-252.097
E96	-122.594	2.231	-274.088	C128	88.555	-101.807	-268.866	D139	-20.071	-157.543	-254.872
E97	-119.446	13.565	-275.127	C129	79.035	-108.783	-268.930	D140	-30.101	-151.756	-257.292
E98	-116.112	24.878	-275.736	C130	69.410	-115.608	-268.612	D141	-40.066	-145.778	-259.337
E99	-112.426	36.522	-275.939	C131	59.674	-122.245	-267.919	D142	-50.103	-139.525	-261.014
E100	-108.558	48.106	-275.694	C132	49.859	-128.707	-266.833	D143	-59.946	-133.147	-262.275
E101	-104.604	59.218	-275.063	C133	39.975	-134.978	-265.353	D144	-69.731	-126.563	-263.133
E102	-100.495	70.245	-274.034	C134	30.032	-141.044	-263.481	D145	-79.393	-119.810	-263.575
E103	-96.165	81.237	-272.628	C135	20.044	-146.899	-261.216	D146	-89.434	-112.519	-263.588
E104	-91.698	92.120	-270.842	C136	10.027	-152.533	-258.560	D147	-98.844	-105.419	-263.167
E105	-87.084	102.795	-268.731	D121	-0.006	-157.941	-255.518	D148	-108.131	-98.147	-262.322
C112	88.705	-91.609	-272.316	D122	-10.038	-152.483	-258.476	D149	-117.235	-90.753	-261.057
C113	79.248	-98.671	-272.603	D123	-20.052	-146.819	-261.082	D150	-126.270	-83.133	-259.357
C114	69.332	-105.833	-272.498	D124	-30.033	-140.949	-263.321	D151	-135.012	-75.493	-257.262
C115	59.657	-112.567	-272.007	D125	-39.967	-134.874	-265.180	D152	-143.576	-67.732	-254.763
C116	49.866	-119.136	-271.123	D126	-49.840	-128.600	-266.647	D153	-151.913	-59.899	-251.881
C117	40.014	-125.504	-269.844	D127	-59.638	-122.130	-267.710	E137	-158.439	-51.477	-249.739
C118	30.013	-131.728	-268.149	D128	-69.346	-115.472	-268.361	E138	-158.115	-40.829	-251.892
C119	20.044	-137.685	-266.070	D129	-78.946	-108.633	-268.592	E139	-155.971	-29.591	-254.771
C120	10.021	-143.434	-263.590	D130	-88.424	-101.624	-268.398	E140	-153.601	-18.276	-257.256
D106	-0.007	-148.946	-260.720	D131	-97.762	-94.456	-267.778	E141	-151.016	-6.955	-259.333
D107	-10.038	-143.403	-263.541	D132	-106.947	-87.139	-266.733	E142	-148.183	4.525	-261.016
D108	-20.060	-137.639	-266.005	D133	-115.963	-79.685	-265.267	E143	-145.166	15.861	-262.271
D109	-30.027	-131.679	-268.089	D134	-124.797	-72.110	-263.383	E144	-141.933	27.205	-263.119
D110	-40.025	-125.456	-269.799	D135	-133.439	-64.424	-261.093	E145	-138.506	38.485	-263.561
D111	-49.871	-119.090	-271.089	D136	-141.877	-56.641	-258.406	E146	-134.690	50.297	-263.595
D112	-59.651	-112.517	-271.969	E121	-150.105	-48.773	-255.334	E147	-130.871	61.457	-263.225
D113	-69.316	-105.770	-272.427	E122	-148.072	-37.571	-258.399	E148	-126.861	72.564	-262.472
D114	-79.208	-98.593	-272.452	E123	-145.812	-26.312	-261.066	E149	-122.692	83.555	-261.345
D115	-88.621	-91.497	-272.039	E124	-143.329	-15.012	-263.329	E150	-118.294	94.571	-259.844
D116	-97.903	-84.234	-271.194	E125	-140.625	-3.688	-265.182	C168	40.151	-156.748	-253.619
D117	-107.005	-76.839	-269.923	E126	-137.705	7.642	-266.623	C169	30.141	-162.605	-251.300
D118	-116.008	-69.251	-268.211	E127	-134.573	18.961	-267.651	C171	10.033	-171.963	-246.644
D119	-124.741	-61.609	-266.097	E128	-131.236	30.251	-268.268	D155	-10.047	-171.861	-246.481
D120	-133.290	-53.851	-263.574	E129	-127.699	41.496	-268.480	D156	-20.064	-168.051	-248.304
E106	-141.613	-46.015	-260.661	E130	-123.970	52.679	-268.293	D157	-30.110	-162.373	-250.884
E107	-139.478	-34.781	-263.553	E131	-120.056	63.788	-267.715	D158	-40.107	-156.468	-253.117
E108	-137.111	-23.479	-266.044	E132	-115.966	74.810	-266.757	D159	-50.175	-150.328	-254.972
E109	-134.524	-12.160	-268.119	E133	-111.708	85.729	-265.430	D160	-60.164	-143.963	-256.449





E110	-131.687	-0.729	-269.792	E134	-107.292	96.543	-263.744	D161	-69.994	-137.485	-257.497
E111	-128.661	10.602	-271.031	C147	69.831	-126.810	-263.627	D162	-79.717	-130.800	-258.154
E112	-125.417	21.933	-271.854	C148	60.017	-133.365	-262.704	D163	-89.824	-123.616	-258.380
E113	-121.976	33.207	-272.263	C149	50.153	-139.725	-261.393	D164	-99.778	-116.222	-258.164
E114	-118.203	44.829	-272.258	C150	40.098	-145.962	-259.671	D165	-109.138	-109.037	-257.511
E115	-114.368	55.976	-271.847	C151	30.114	-151.914	-257.579	D166	-118.329	-101.683	-256.454
D167	-127.455	-94.140	-254.951	D196	-50.160	-171.307	-241.798	D240	-79.883	-172.745	-232.873
D168	-136.381	-86.451	-253.046	D197	-60.213	-165.169	-243.617	D241	-90.243	-166.185	-233.728
D169	-145.047	-78.753	-250.737	D198	-70.169	-158.794	-245.082	D242	-100.462	-159.371	-234.213
D170	-153.496	-70.937	-248.052	D199	-80.018	-152.184	-246.187	D243	-110.578	-152.172	-234.382
D171	-160.152	-62.573	-246.125	D200	-90.254	-145.257	-246.788	D244	-120.516	-144.734	-234.164
E154	-166.611	-54.127	-243.891	D201	-100.340	-138.090	-246.984	D245	-130.127	-137.091	-233.612
E155	-166.362	-43.505	-246.149	D202	-110.259	-130.694	-246.768	D246	-139.530	-129.230	-232.684
E156	-165.900	-32.827	-248.095	D203	-119.994	-123.083	-246.138	D247	-148.675	-121.699	-231.136
E157	-163.661	-21.537	-250.782	D204	-129.301	-115.743	-244.994	D248	-157.590	-113.969	-229.255
E158	-161.176	-10.212	-253.087	D205	-138.409	-108.224	-243.473	D249	-164.791	-106.029	-228.070
E159	-158.478	1.260	-254.986	D206	-147.306	-100.539	-241.579	D250	-171.794	-97.961	-226.607
E160	-155.536	12.730	-256.488	D207	-155.982	-92.698	-239.322	D265	-110.495	-162.601	-227.711
E161	-152.435	24.087	-257.554	D208	-162.912	-84.496	-237.795	D266	-120.526	-155.271	-227.677
E162	-149.102	35.409	-258.233	D209	-169.636	-76.185	-235.979	D267	-130.234	-147.779	-227.292
E163	-145.419	47.257	-258.498	D210	-176.155	-67.773	-233.876	D268	-139.753	-140.024	-226.557
E164	-141.498	59.029	-258.348	E191	-182.469	-59.277	-231.490	D269	-149.020	-132.057	-225.476
E165	-137.602	70.187	-257.793	E192	-182.398	-48.703	-233.923				
E166	-133.499	81.247	-256.882	E193	-182.083	-38.073	-236.080				
D174	-20.084	-176.709	-242.524	E194	-181.548	-27.396	-237.938				
D175	-30.061	-172.749	-244.147	E195	-180.789	-16.688	-239.498				
D176	-40.094	-166.941	-246.545	E196	-178.395	-5.240	-241.779				
D177	-50.193	-160.938	-248.550	E197	-175.733	6.215	-243.696				
D178	-60.230	-154.674	-250.208	E198	-172.808	17.663	-245.244				
D179	-70.139	-148.183	-251.501	E199	-169.630	29.090	-246.419				
D180	-79.924	-141.593	-252.350	E200	-166.254	40.987	-247.114				
D181	-90.096	-134.550	-252.763	E201	-162.618	52.822	-247.412				
D182	-100.133	-127.250	-252.762	D217	-60.050	-175.359	-236.775				
D183	-109.967	-119.743	-252.340	D218	-70.090	-169.078	-238.407				
D184	-119.247	-112.462	-251.475	D219	-79.999	-162.569	-239.688				
D185	-128.464	-105.032	-250.150	D220	-90.303	-155.893	-240.381				
D186	-137.495	-97.407	-248.440	D221	-100.470	-148.831	-240.769				
D187	-146.286	-89.639	-246.360	D222	-110.506	-141.516	-240.752				
D188	-154.851	-81.875	-243.861	D223	-120.343	-134.002	-240.324				
D189	-161.644	-73.596	-242.134	D224	-129.850	-126.245	-239.579				
D190	-168.245	-65.199	-240.110	D225	-139.066	-118.804	-238.237				
E172	-174.623	-56.726	-237.803	D226	-148.097	-111.175	-236.522				
E173	-174.468	-46.136	-240.147	D227	-156.894	-103.398	-234.448				
E174	-174.075	-35.466	-242.206	D228	-163.967	-95.368	-233.076				





E175	-173.458	-24.777	-243.958	D229	-170.845	-87.128	-231.450
E176	-171.102	-13.457	-246.468	D230	-177.527	-78.757	-229.541
E177	-168.566	-2.002	-248.557	D231	-183.991	-70.313	-227.352
E178	-165.754	9.482	-250.276	E213	-190.006	-40.617	-229.671
E179	-162.684	20.922	-251.616	E214	-189.543	-29.928	-231.641
E180	-159.479	32.274	-252.512	E215	-188.862	-19.222	-233.309
E181	-155.959	44.145	-252.990	E216	-187.892	-8.569	-234.729
E182	-152.170	55.976	-253.068	E217	-185.373	2.895	-236.841
E183	-148.122	67.685	-252.763	E218	-182.582	14.387	-238.596
D195	-40.023	-177.194	-239.644	E219	-179.542	25.836	-239.978

