“FAST”主动反射面的形状调节

摘要

500 米口径球面射电望远镜“FAST”有特殊的使用功能，由主索节点构成的主动反射面主要分为两个状态——基准态（球面）、工作态（近似旋转抛物面）。“FAST”在工作中，支撑其结构的控制众多，而本论文主要研究其主动反射面的形状调节策略。

问题 1 是一个优化问题——在考虑反射面板调节因素（促动器径向伸缩范围为-0.6～+0.6 米）的情况下，满足约束的理想抛物面可能不唯一，需要制定合理的优化指标择优选取。“FAST”主索网受下拉索-促动器结构施加的位移影响实现由基准球面到工作抛物面的变形，该位移沿径向且存在严格的运动范围限制，因此变位时促动器的运动行程是判定理想抛物面优劣的指标之一；此外，对于一个理想抛物面更重要的是，获得天体电磁波经反射面反射后的最佳接收效果。综上两点，我们定义了三项优化指标（1. 促动器的总行程 2. 工作抛物面与基准球面之间的距离差值 3. 工作抛物面的相对反射有效值），结合逐步求精算法，在满足反射面板调节约束的抛物面中择优，确定理想抛物面方程为：$\boldsymbol{(x+y)^2=560.84(z+300.81)}$。

问题 2 是问题 1 的变形——我们经过计算，找到了在问题 2 条件下位于直线 SC 上的主索节点：D27。我们做出假设：基准球面和工作抛物面都是关于轴对称的。通过乘以转移矩阵，将主索网沿中心转动，使得问题 2 可以在问题 1 的环境中求解，问题得到了简化。最后通过乘以逆矩阵，将主索网还原，得到理想抛物面方程：$[(sin36.795°+cos36.795°sin78.169°)x+(sin36.795°sin78.169°-cos36.795°)y-cos78.169°z]^2=560.84(cos78.169°cos36.795°x+sin36.795°cos78.169°y+sin78.169°z+300.81)]$。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保存在 $result.xlsx$ 中，附录 B 展示了该文件的部分截图。

问题 3 重在计算——我们首先计算了照明区域内每块反射面板的法向量，根据光的反射原理，将每块反射面板的三个顶点投影到馈源舱平面得到一块三角区域，再计算三角区域与馈源舱有效区域的重叠部分面积。将工作抛物面照明区域内的所有反射面板投射后计算出的重叠部分面积求和，即可得到馈源舱的接收比：**1.1079%**。

关键字： 加权评分法　优化指标　逐步求精

一、 问题重述

1.1 引言

“FAST”是天文射电望远镜创新领域的重大项目之一。其主索网由 2226 个主索节点构成，要完成天文观测，需要实现主动反射面主索网变形过程的实时精准控制。而多输入、非线性等特征使得这一控制的实现困难重重。因此，建立合理的节点运动轨迹模型具有十分重要的意义。本论文应用基于加权评分法的多目标决策，从逐步求精算法的结果中择优确定理想工作抛物面。通过 MATLAB 仿真实验，验证了所建立的“FAST”索网运动轨迹模型的可行性。最后通过物理分析和数学建模计算求出了反射面调节后馈源舱的接收比，并与基准反射球面的接收比作分析比较。

1.2 问题分析

1.2.1 对于问题 1

该问题是旋转抛物面的确定问题。由旋转抛物面方程，开口方向、对称轴、焦距与顶点确定唯一的抛物面。根据题意，该近似抛物面的开口方向始终面对球心 C，对称轴即被观测体 S 与球心 C 的连线，焦点即直线 SC 与焦面的交点 P。由此问题转化为确定抛物面的顶点。我们采取逐步求精的编程方法，在直线 SC（落在焦面与工作抛物面之间的部分）上查找最优的顶点，记作 V。考虑到题目的限制条件——在反射面板调节约束下确定一个理想抛物面——我们需要严格限制促动器的调节范围，即径向伸缩-0.6～+0.6 米，在此范围内将反射面调节为尽量贴近理想抛物面的工作抛物面，以获得更高的馈源舱接受比。我们制定了三项优化指标并加权来评测逐步求精过程中所取抛物面的优劣，最终得到最优的理想抛物面。

1.2.2 对于问题 2

经过问题 1 的研究，我们知道在本赛题中，被观测体 S 位置确定时，顶点 V 可以唯一确定一个抛物面。经过计算，我们发现在问题 2 的条件下，存在主索节点 D27 位于直线 SC 上。那么若将主索网绕球心转动一定角度，便可以使点 D27 旋转到问题 1 中直线 SC 与基准球面的交点（即点 A0）处，由此问题 2 转化成了问题 1 的变形。我们计算出使点 D27 旋转到点 A0 处的转移矩阵，将主索网上全部节点乘以矩阵，实现了基准球面的整体转动，然后按照问题 1 的求解办法得到照明区域内所有主索节点对应促动器的伸缩量，最后再将主索节点的新坐标乘以上述转移矩阵的逆矩阵，实现还原。不论基准球

面还是工作抛物面都被认为是沿中心对称的，因此上述转动不会改变促动器的伸缩量，只是改变坐标。由此我们解决了问题 2。

1.2.3 对于问题 3

我们假设来自目标天体的平行电磁波是均匀分布的，那么馈源舱接受的电磁波信号强弱可以由反射面板能将电磁波反射到馈源舱有效区域的面积反映出来。由此将问题转化为计算工作抛物面的每块反射面板可以将电磁波反射到馈源舱有效区域的面积。我们首先将每块反射面板的三个顶点反射到馈源舱平面，得到一块三角形区域，再计算其与馈源舱有效区域的重叠部分面积。

二、 符号说明表

打 * 号的量为本论文中给出的定义。

符号	意义	符号	意义
S	被观测体	O	基准球面球心
P	馈源舱接受平面中心	V	工作抛物面顶点
F	焦面与基准球面半径差	V_k	V 点的初始坐标
L	促动器的总行程	W	工作抛物面与基准球面之间的距离幅值 *
Φ_λ	反射面板的相对反射有效值 *	Φ	工作抛物面的相对反射有效值 *
I	工作抛物面理想程度 *		

三、 问题 1——多重指标下的逐步求精算法

3.1 模型假设

1. 根据附录内容，反射面板的厚度（附录 6）、面板上的小孔（附录 3）这些因素对主索网形状的影响均可以忽略。因此在本研究中，不妨认为反射面板组成的主索网是平滑、规则的旋转抛物面。
2. 由于被观测体的距离很远，电磁波信号及反射信号可视为直线传播。
3. 根据附录 5，主索节点调节后相邻节点间距离变化幅度不超过 0.07%，因此可以认为在调节促动器的过程中，主索节点始终沿径向运动。

3.2 建模过程

由旋转抛物面方程，并依照题意，被观测体 S 位置确定时，顶点 V 可以唯一确定一个抛物面：开口方向固定，我们知道对称轴、焦距和顶点唯一确定一个旋转抛物面，而在本赛题中对称轴和焦点是被 S 点和 C 点确定的，因此确定抛物面顶点 V 的坐标是求解本小题的关键之处。由于促动器是径向伸缩的，工作抛物面的顶点 V 必定落在直线 SC 上。可供选择的 V 点数量巨大，因此我们建立指标，采取逐步求精算法，在直线 SC 上搜索 V 的最佳落点。具体过程如下：

3.2.1 旋转抛物面方程

$$z_0' = z_0 + \Delta l \tag{1}$$

式（1）表示，作为工作抛物面顶点的主索节点的 Z 坐标 z_0 以 Δl 为步长变化。此变化过程中，由于焦点不变，焦距 p 不断改变，如式（2）所示。

$$p/2 = f - \Delta l \tag{2}$$

式（1）确定了抛物面的顶点 z_0'，式（2）确定了抛物面的焦距 p，根据旋转抛物面方程（式（3）），可以求出唯一的抛物面。

$$(x+y)^2 = 2p(z - z_0') \tag{3}$$

3.2.2 制定工作抛物面优化指标

根据附录 7，由于促动器的径向的伸缩范围受限，促动器的运动行程是选取工作抛物面的重要指标。

优化指标 1 我们以促动器的总行程 L 最短作为优化指标 1, 下面基于最小二乘法给出 L 的定义：

定义 1 球坐标系中的点坐标可表示为 (r, θ, ϕ)，在本研究中，每个主索节点在球坐标系中的 θ、ϕ 是固定的，因此可以基于最小二乘法定义

$$L = \sum_{\lambda \in A} \|r_\lambda' - r_\lambda\|，A \text{ 为指标集}$$

图 1 是一个剖面。r_λ' 为 OA 的长度，r_λ 为 OB 的长度。

优化指标 2 我们以工作抛物面与基准球面之间的距离幅值 W 最小作为优化指标 2。基于定义 1，我们可以求出主索网由基准球面变形为工作抛物面时每个主索节点沿径向

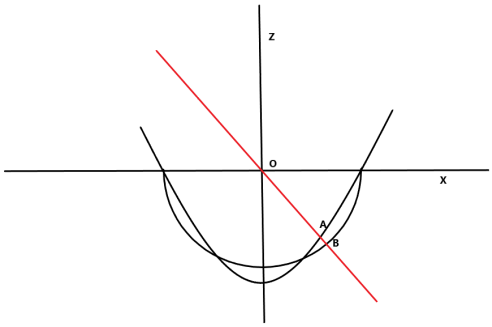


图 **1**　指标 **1** 示意图

移动的距离，在这些距离中同一方向绝对值最大的即为幅值 W。W 是矢量，因此对于每个顶点 V 确定的工作抛物面均有正负两个幅值，记作 W_+，W_-。

根据题意，最终的优化目标为“获得天体电磁波经反射面反射后的最佳接受效果”，下面针对该目标制定优化指标 3。

优化指标 3　我们首先定义反射面板的相对反射有效值 Φ：

定义 2 反射面板的相对反射有效值 Φ_λ，$\lambda \in A$，A 为指标集。

$$\Phi_\lambda = \begin{cases} 1 & \text{存在光线经该面板反射后过 } P, \\ 0 & \text{反之.} \end{cases}$$

进而定义工作抛物面的相对反射有效值 Φ 作为优化指标 3：

定义 3 工作抛物面的相对反射有效值 Φ,

$$\Phi = \sum_{\lambda \in A} \Phi_\lambda$$

综上，为三项指标加权，给出工作抛物面理想程度的定义：

定义 4 工作抛物面的理想程度 I,

$$I_1 = -\frac{100}{m}L + 100 \tag{4}$$

$$I_2 = (-\frac{100}{0.6}W_+ + 100) + (-\frac{100}{0.6}W_- + 100) \tag{5}$$

$$I_3 = \frac{100}{M}\Phi \tag{6}$$

$$I = w_1 I_1 + w_2 I_2 + w_3 I_3 \tag{7}$$

式中，m 为促动器允许移动的最大值（即 *0.6** 照明区域内总节点数），M 为照明区域内反射板的总数。w_1、w_2、w_3 为三项优化指标的权重，我们将在建模过程中衡量三者的比例分配。工作抛物面越理想，I 值越大。

3.2.3 逐步求精算法

由于可供选择的 V 点数量巨大，可以采用逐步求精算法逼近最优解，我们选取基准球面上的 V 点作为第一次循环的基准点，取步长 $\Delta l = 0.1m$。之后的每次外循环取上次循环得到的 V 集合中其确定的工作抛物面 I 值最小的点作为基准点，并将步长缩小至 $\frac{1}{10}$。算法思想如图 2 所示。

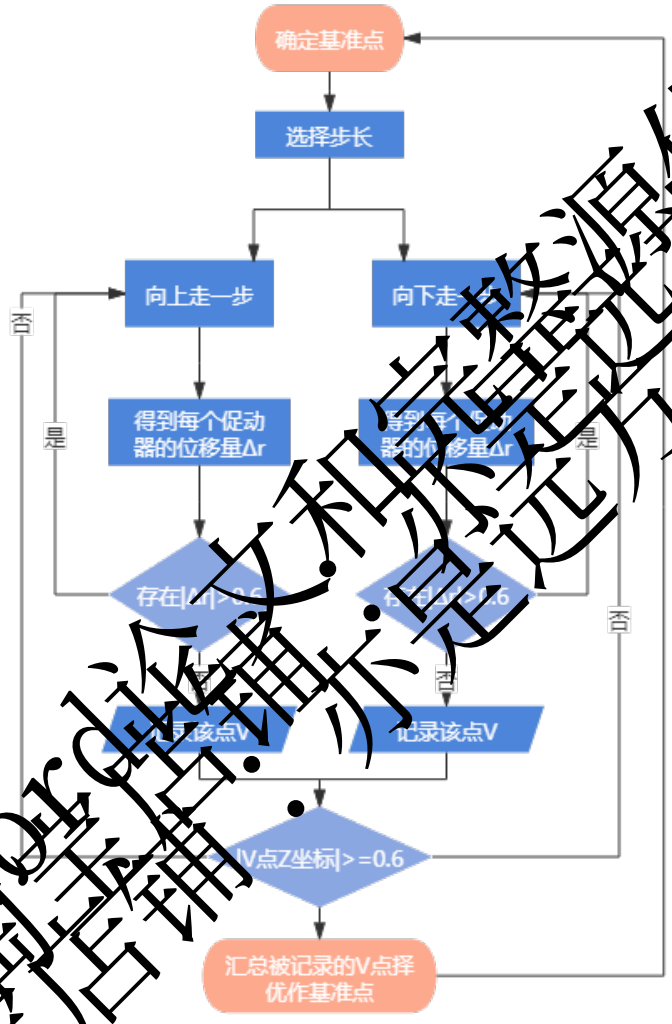


图 2　逐步求精程序框图

3.3 模型求解

通过分析附件 1 的数据，我们发现 Z 坐标相同的主索节点在球坐标系中有相同的 β 角，由旋转抛物面的对称性，两个点对应的 β 角相等则与球心的距离也相等。由此，不妨将照明区域内所有主索节点按 β 角分组，进而将问题降至二维平面（xz 平面），用圆弧和抛物线代替基准球面和工作抛物面。如图 1 所示。

3.3.1 原始数据预处理

1. 根据球坐标公式

$$x = rsin\phi cos\theta \tag{8}$$

$$y = rsin\phi sin\theta \tag{9}$$

$$z = rcos\phi \tag{10}$$

计算出 θ、ϕ 的值

$$\theta = arctan\frac{y}{x} \tag{11}$$

$$\phi = arccos\frac{z}{r} \tag{12}$$

而 β 是的 ϕ 的余角，由此求出了照明区域内所有主索节点在球坐标系中的 β 角。将所有节点按 β 分组，每组取一个代表以降低计算复杂度。

2. 使用 MATLAB 内部函数实现平滑和去噪。

3. 分析附录 1 的数据，我们发现反射面板有一定厚度。结合附录 1 和附录 3 的数据，我们发现主索网单位并不是严格的等边三角形，而是近似的等腰三角形，而且每个三角形近似全等。但我们认为这些因素并不影响本研究的结果。

3.3.2 逐步求精结果

在步长为 $\Delta l = 0.1$ 的第一次循环中，我们发现只有 Δl 为-0.2、-0.3、-0.4、-0.5 以及-0.6 的 V 点满足促动器调节范围的约束。按顺序命名为抛物线 1、2、3、4、5，五条抛物线的相关参数如下：

	抛物线 1	抛物线 2	抛物线 3	抛物线 4	抛物线 5
Δl	-0.2	-0.3	-0.4	-0.5	-0.6
VX	(0.5066,-0.1869)	(0.3996,-0.2869)	(0.2926,-0.387)	(0.185,-0.487)	(0.0785,-0.5871)
L	67.64	62.8131	70.2095	87.3225	112.6378
Φ	1318	1318	1319	1319	1319

如 3.3.3 中分析，在这组抛物线中，抛物线 3 是最优的，因此选定抛物线 3 的顶点作为下次循环的基准点，以 $\Delta l = 0.01m$ 为步长开始下次循环，共得到 21 条抛物线。两次循环中各抛物线的相关参数保存在附录 C 中。最优解附近的 5 条抛物线相关参数如下：

Δl	-0.39	-0.4	-0.41	-0.42	-0.43
W	(0.303,-0.377)	(0.293,-0.387)	(0.282,-0.397)	(0.271,-0.407)	(0.260,-0.417)
L	68.960	70.210	71.459	72.867	74.391
Φ	1319	1319	1319	1319	1319

3.3.3 加权评分方案制定与结果分析

通过分析 3.3.2 中的数据，我们得到以下结论：

1. 根据附件 1 和附件 3，我们计算出照明区域内共有主索节点约 706 个，反射面板约 1323 块。3.3.2 表格显示五个工作抛物面的 Φ 值在 1318-1319 这个非常接近反射面板总数，这说明我们的模型是很有效的——工作抛物面上绝大多数的反射面板可以反射电磁波到馈源仓 P 处。但同时，五个工作抛物面的 Φ 值相差很小，这意味着优化指标 3 对理想程度 I 的影响很小，可以分配较小的权重。

2. 由附录 5 对相邻节点间距离变化幅度的约束，我们认为优化指标 2 更为重要，因为幅值 W 过大可能导致超出 0.07% 的约束。

最终，我们决定了 $w_1 = 0.15$、$w_2 = 0.4$（指标 2 总分是指标 1、3 的 2 倍）、$w_3 = 0.05$ 的权重分配，即 $I = 0.15I_1 + 0.4I_2 + 0.05I_3$。3.3.2 中两次循环的计算结果如下。

第一次循环：

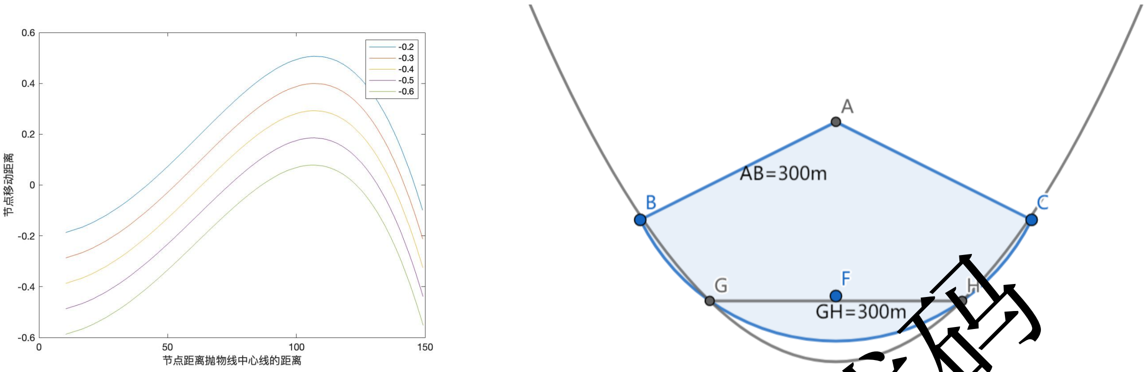
抛物面 1	抛物面 2	抛物面 3	抛物面 4	抛物面 5
51.352	51.795	52.197	52.055	51.622

由此选定抛物线 3 的顶点为下次循环的基准点。

第二次循环各抛物线的评分保存在附录 D 中。最优解附近的 5 条抛物线评分如下：

Δl	-0.39	-0.4	-0.41	-0.42	-0.43
I	52.195	52.197	52.199	52.196	52.188

由此得出结论，$\Delta l = -0.41m$ 时，得到的工作抛物面最理想。



(a) 各工作抛物线与基准圆弧的关系　　**(b)** 抛物线 4(灰) 与基准球面 (蓝) 的关系

图 **3**

综上，当待观测天体 S 位于基准球面正上方，即 α=0°，β=90° 时，结合考虑反射面板调节因素，理想抛物面方程为：$(x+y)^2 = 560.84(z+300.81)$。

四、 问题 2——通过坐标旋转简化问题

4.1 模型假设

理想球面和旋转抛物面是关于轴对称的。在本研究中所有主索节点和反射面以贴近理想基准球面、工作抛物面为目标，因此我们做出假设：基准球面和工作抛物面是关于轴对称的。由此保证了转动不会影响结果正确性。

4.2 建模过程

4.2.1 转移矩阵

我们以使点 D27 与点 A0 重合为目标，在三维空间旋转主索网，得到一个 3×3 的转移矩阵 A。

4.2.2 多重指标逐步求精

由 4.2.1 计算出的转移矩阵得到点 D27 与点 A0 重合时所有主索节点的坐标值，将数据代入问题 1 的模型可以求出变形后各促动器顶端的伸缩量，以及此时的理想抛物面方程、照明区域内所有主索节点的坐标。

4.2.3 还原主索网

求出 4.2.1 中转移矩阵的逆矩阵，实现所有主索节点的还原，进而得到真正的理想抛物面方程和照明区域内所有主索节点的坐标。

4.3 模型求解

4.3.1 计算转移矩阵

记 α_0=36.795°，β_0=78.169°，转移矩阵 A 为

$$\mathbf{A} = \begin{pmatrix} sin\alpha_0 & -cos\alpha_0 & 0 \\ sin\beta_0 cos\alpha 0 & sin\beta_0 sin\alpha_0 & -cos\beta_0 \\ cos\beta_0 cos\alpha_0 & cos\beta_0 sin\alpha_0 & sin\beta_0 \end{pmatrix}$$

A 是正交阵，因此 A^{-1}=A^T。

通过转移矩阵 A 及其逆可以实现主索网的转动。

4.3.2 确定理想抛物面

根据问题 1 的逐步求精算法，列出第一次循环各抛物线的相关参数和评分表、第二次循环中最优解附近的 5 条抛物线的相关参数和评分表。

	抛物线 1	抛物线 2	抛物线 3	抛物线 4	抛物线 5
Δl	-0.2	-0.3	-0.4	-0.5	-0.6
W	(0.5067,-0.1868)	(0.3996,-0.2868)	(0.2925,-0.3869)	(0.1855,-0.487)	(0.0786,-0.5871)
L	52.4735	54.8659	61.2932	75.4383	97.5311
Φ	1318	1318	1319	1319	1319
I	51.642	52.278	52.521	52.480	52.151

Δl	-0.39	-0.4	-0.41	-0.42	-0.43
W	(0.303,-0.377)	(0.293,-0.387)	(0.282,-0.397)	(0.271,-0.407)	(0.260,-0.417)
L	68.960	70.210	71.459	72.867	74.391
Φ	1319	1319	1319	1319	1319

显然当 $\Delta l = -0.41$ 时，对应的工作抛物面理想程度最高，此时的理想抛物面方程为：$(x+y)^2 = 560.84(z+300.81)$。

4.3.3 计算促动器顶端伸缩量

根据理想抛物面方程，可以计算出促动器顶端的伸缩量，由于绕中心轴转动不会改变促动器的伸缩量，因此这就是最终结果。

4.3.4 还原主索网得到结果

将主索节点乘以 A^{-1}，可以将主索网还原到转动前的位置，进而得到理想抛物面方程：$[(sin36.795° + cos36.795°sin78.169°)x + (sin36.795°sin78.169° - cos36.795°)y - cos78.169°z]^2 = 560.84(cos78.169°cos36.795°x + sin36.795°cos78.169°y + sin78.169°z + 300.81)$

4.4 结果分析与检验

综上，当待观测天体方位于 α=36.795°，β=78.169° 时，理想抛物面方程为：$[(sin36.795° + cos36.795°sin78.169°)x + (sin36.795°sin78.169° - cos36.795°)y - cos78.169°z]^2 = 560.84(cos78.169°cos36.795°x + sin36.795°cos78.169°y + sin78.169°z + 300.81)$。理想抛物面的顶点坐标、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详细结果见附录 B。

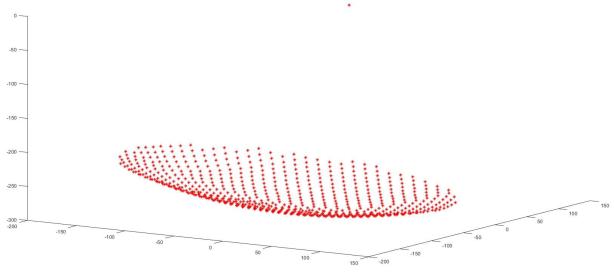


图 4　问题 2 理想工作抛物面示意图

五、 问题 3——用通量来反映接收信号量

5.1 模型假设

由于目标天体距离很远，我们不妨假设其发出的平行电磁波是均匀分布的。馈源舱可接收的有效区域是一个平面，因此我们可以采用通量来反映有效区域内的接受量，我们知道在场强均匀的情况下，通量与面积成正比，因此将问题转化为计算重叠部分面积是合理的。

5.2 建模过程

根据附件 3 的数据和对问题 1 的研究，我们得到了照明区域内所有反射面板的法向量。根据光的反射原理，可以计算每块反射面板出射光的角度，我们首先将每块反射面板的三个顶点反射到馈源舱平面，得到一个三角形区域，然后计算该三角区域与馈源舱有效区域的重叠部分面积（如 5.4 分析，该面积存在一个放大比，因此需要乘以一个比例系数。这里认为 S_g 是经过放大比处理的面积和）。所有反射面板计算完毕后，对所有面积求和，用面积和 S_g 表示馈源舱有效区域接收到的反射信号。再计算工作抛物面照明区域的面积 S_r，$\frac{S_g}{S_r}$ 即为所求的馈源舱接收比。

5.3 模型求解

5.3.1 求法向量

1. 若三个顶点不共线共面

由

$$\begin{vmatrix} (x-x_1) & (y-y_1) & (z-z_1) \\ (x_2-x_1) & (y_2-y_1) & (z_2-z_1) \\ (x_3-x_1) & (y_3-y_1) & (z_3-z_1) \end{vmatrix} = 0 \tag{13}$$

可以解出平面方程，进而得出法向量 n_1。

5.3.2 求反射电磁波方向

根据几何光学，可以得到反射电磁波的方向 $n_2 = n_1 + a$ 。如图 5 所示。

作平行于反射光方向、且经过反射面板三个顶点的直线，与馈源舱接收平面的交

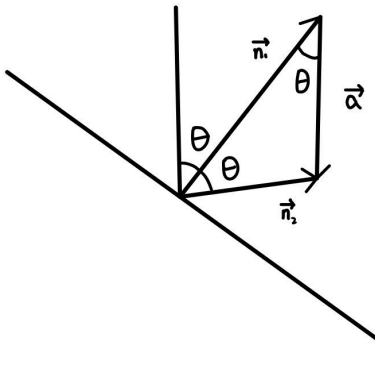


图 5　反射示意图

点，得到一个三角形区域。

5.3.3 求三角与圆的重叠面积

把圆形看作一千边形，使用 MATLAB 内部函数 $intersect()$ 求解。误差可以忽略不计。

5.4 结果分析与检验

由于是对抛物面，与馈源舱接收平面成一定角度，因此三角形面积经过了一次放大，而电磁波经过反射后，又会使三角形面积放大一次。因此最终得到的三角形面积是大于实际面积的，我们需要将它乘以一个比例系数 k_l。

如图 6 所示，放大比 $=|\frac{|AB|}{|AD|}| = |\frac{|AB|}{|AC|cos\theta_j}| = |\frac{sin(\theta_j+\theta_l)}{cos\theta_j sin\theta_l}|$。其中，$\theta_j$ 是平面法向量与竖直轴夹角，θ_l 是反射光线与 xy 平面夹角。

综上，基于第 2 问的反射面调节方案，调节后馈源舱的接收比为：1.1079%。

用同样的方法计算出基准反射球面的接收比为：0.8138518%。对比发现，工作抛物面的接收比相对基准反射球面提升了 36.1391%。这证明我们的模型是有效的，经过从基准球面到理想工作抛物面的变形，对目标天体的观测取得了更好的效果。

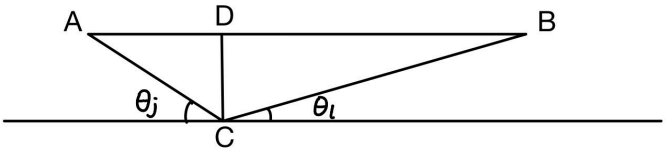


图 **6**　放大比示意图

六、 模型评价与拓展

最后我们对本论文建立的数学模型的优缺点进行评估。

优点：

1. 模型可以在比较小的时间复杂度下迅速找到理想抛物面。
2. 模型可迁移性较强，在观测天体旋转时，仍可通过坐标的旋转变换找到理想抛物面。
3. 本模型综合考虑了多种指标，并根据实际情况分析给出合理权重，找到更加符合实际的理想抛物面。

不足：

1. 模型只考虑了三个顶点都在光照区域内的反射面板，忽略了仅有一个或两个顶点在光照区域的反射面板，从而导致了一定的误差。
2. 没有考虑力学上的因素，边界处的下拉索可能会由于突变而造成受力过大，从而对设备造成影响。

参考文献

[1] 商文忠. FAST 反射面支承结构优化研究. 哈尔滨工业大学工学硕士学位论文, 2007.

[2] 罗斌，郭正兴，姜鹏. FAST 主动反射面索网结构设计与施工技术研究. 东南大学出版社, 2016.

附录 A　计算程序

问题 1 计算程序

```
% 计算符合光照条件的点在angle_1中的索引，存储在fit_triangle 中

axis_1 = zeros(4300,1);
axis_2 = zeros(4300,1);
axis_3 = zeros(4300,1);
fit_triangle = zeros(4300,3);
belong_index = 0;

for i = 1:4300
    a = triangle(i,1);
    a = string(a);
    axis_1(i,1) = find(strcmp(a, point_name));
    b = triangle(i,2);
    b = string(b);
    axis_2(i,1) = find(strcmp(b, point_name));
    c = triangle(i,3);
    c = string(c);
    axis_3(i,1) = find(strcmp(c, point_name));
end
% 找到符合半径条件的反射面板，有若三角形有两个点都在口径范围内，则视为满足条件
for i = 1:4300
    mat_angle_1 = cell2mat(angle_1(i,5));
%     belong_index = belong_index + (abs(mat_angle_1(axis_1(i,1),1)) > 60) +
     (abs(mat_angle_1(axis_2(i,1),1)) > 60) + (abs(mat_angle_1(axis_3(i,1),1)) > 60);
    if axis_1(i,1) <= length(update_length(:,1)) && axis_2(i,1) <= length(update_length(:,1))
        && axis_3(i,1) <= length(update_length(:,1))
       fit_triangle(i,1) = axis_1(i,1);
       fit_triangle(i,2) = axis_2(i,1);
       fit_triangle(i,3) = axis_3(i,1);
    end
end
% 去除不满足条件的三角面片（去除0值）
fit_triangle = fit_triangle(any(fit_triangle,2),:);

% 根据update_length伸长量来计算移动后的坐标索引,计算得到的索引存在new_axis中

new_axis = zeros(2226,3);
r_new_axis = zeros(2226,3);
dl_index = 1:5;
index_3 = zeros(length(dl_index),1);


global o;
```

```
for i = 1:length(o)
   if inv_r_base_axis(i,1) < 0
       o(i,2) = o(i,2) + 180;
   end
end

for i = 1:length(dl_index)
    for j = 1:length(update_length)
       new_axis(j,1) =
           update_length(j,dl_index(i))*cosd(angle_1_mat(j,1))*cosd(angle_1_mat(j,2));
       new_axis(j,2) =
           update_length(j,dl_index(i))*cosd(angle_1_mat(j,1))*sind(angle_1_mat(j,2));
       new_axis(j,3) = update_length(j,dl_index(i))*sind(angle_1_mat(j,1));
    end
    [new_angle, new_line, temp_a, temp_b, temp_c, index_3(i,1)] =
         triangle_calculator(fit_triangle);
end
% global r_mat;
% for i = 1:length(update_length)
%     r_new_axis(i,:) = new_axis(i,:)*r_mat;
% end
%
% plot3(r_new_axis(:,1),r_new_axis(:,2),r_new_axis(:,3),'r*');

% 生成一个有空位的元胞数组存储符合条件的结点，用于索引
% global angle_1
% fit_point_name_2 = cell(length(r_new_axis()),1);
% for i = 1:length(r_new_axis(:,1))
%     if r_new_axis(i,1) >= 0
%        fit_point_name_2(i,1) = angle_1(i,1);
%     end
% end
% fit_point_name_2_nan = fit_point_name_2;
% 生成一个没有空位的元胞数组存储符合条件的节点，用于计数
% fit_point_name_2(cellfun(@isempty,fit_point_name_2)) = [];

% 计算了每次抛物线上下移动相等的距离所产生的新的伸长量，第一次迭代
surface_1 = surface.';
o = zeros(2226,2);
rad = zeros(2226,2);
min_index = -0.6;
max_index = 0.6;
for i = 1:2226
%    第一问
    x = surface(i,1);
    y = surface(i,2);
    z = surface(i,3);
```

```
    beta_d = asind(z/sqrt(x^2+y^2+z^2));
    theta_d = atand(y/x);
    beta = asin(z/sqrt(x^2+y^2+z^2));
    theta = atan(y/x);
    o(i,1) = real(beta_d);
    o(i,2) = real(theta_d);
    rad(i,1) = real(beta);
    rad(i,2) = real(theta);
end

z0 = -300.4;
F = 139.8;
unique_rad = unique(rad(:,1));
unique_angle = unique(o(:,1));
for i = 2:length(unique(rad(:,1)))
    if abs(unique_rad(i)) < pi/3
%           由于光照区域角度是60度，所以大于一定角度的点直接舍去，根据角度索引筛出符合条件的点
%           将不在光照区域内的角度除去
        unique_rad(i) = 0;
        unique_angle(i) = 0;
        continue;
    end
end
% 去除不满足的点
unique_rad(unique_rad == 0) = [];
unique_angle(unique_angle == 0) = [];
% 能够满足正负0.6的顶点移动距离各自的顶点移动距离
success_dl = zeros(num,1);
success_index = 1;
fail_up = 0;
fail_down = 0;
max_move_length = zeros(num,0);
min_move_length = zeros(num,0);
begin = 0.1;
pace = 0.1;
final = 0.6;
num = ((final - begin)/pace + 1)*2;
update_length = zeros(2226,num);
sum = zeros(1,num);
show_length = zeros(1,length(unique_rad));

figure;
for dl = begin:pace:final
    fail_down = 0;
    x = (-500:500);
%     顶点向上移动
```

```
    z1 = z0 + dl;

%   up_result代表向上移动后，各主索节点的横坐标
    up_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%           垂直的线不画出
            continue;
        else
            z_line = tan(unique_rad(i))*x;
%           plot(x,z_line);
            syms x_1;
            eqn = tan(unique_rad(i))*x_1 == x_1^2/(4*(F-dl))+z1;
%           联立方程
            S = solve(eqn, x_1);
            S_double = double(S);
            if abs(S_double(1,1)) > abs(S_double(2,1))
                up_result(1,i) = S_double(2,1);
            else
                up_result(1,i) = S_double(1,1);
            end
        end
    end
    up_result(up_result==0) = [];
%   根据满足条件的各点横坐标计算各点到c的距离，将多次的结果都存储在update_length中，
    print_up是各下拉锁移动距离
    print_up = zeros(length(up_result),1);
    for i = 1:length(up_result)
        if i == 1
            print_up(i,1) = 0.2;
        else
            print_up(i,1) = up_result(i)/cos(unique_rad(i)) + z0;
%           找每一个相应的主索节点
            angle_x = find(abs(angle_1_mat(:,1) - unique_angle(i)) < 0.000001);
            for j = 1:length(angle_x)
                update_length(angle_x(j,1),int32(num/2+dl/pace)) =
                    up_result(i)/cos(unique_rad(i));
            end
%           第一问得到706行的update_length，第二问得到692行update_length
%           对每一次各个长度的求和
            sum(int32(num/2+dl/pace)) = sum(int32(num/2+dl/pace)) + abs(print_up(i,1));
            if abs(print_up(i,1)) > 0.6
%               有超过界限的直接排除
                fail_up = 1;
                break;
            end
        end
```

```
    end
%     假如促动器全部满足条件，视为成功,将dl存储,并且画出横坐标与伸长量print_up的图,
%    并且记录最大移动量和最小移动量
    if fail_up == 0
        success_dl(success_index,1) = dl;
        success_index = success_index + 1;
        plot(up_result(1,2:end), -print_up(2:end,1));
        hold on
        max_move_length(int32(num/2+dl/pace),1) = max(print_up(2:end,1));
        min_move_length(int32(num/2+dl/pace),1) = min(print_up(2:end,1));
    end
% 将成功指标置为成功
    fail_up = 0;
%     顶点向下移动
    z2 = z0 - dl;
    down_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%             垂直的线不画出
            continue;
        else
            z_line = tan(unique_rad(i))*x;
            syms x_2;
            eqn = tan(unique_rad(i))*x_2 == x_2^2/(4*(f+dl))+z2;
%             联立方程
            S = solve(eqn, x_2);
            S_double = double(S);
            if abs(S_double(1,1)) < abs(S_double(2,1))
                down_result(1,i) = S_double(2,1);
            else
                down_result(1,i) = S_double(1,1);
            end
        end
    end
    down_result(down_result==0) = [];
%     促动器运动距离有效出光范围的，失败，继续下一次循环
    print_down = zeros(length(down_result),1);
    for i = 1:length(down_result)
        if i == 1
            print_down(i,1) = -dl;
        else
            print_down(i,1) = down_result(i)/cos(unique_rad(i)) + z0;
    %         检索当前角度的主索节点
            angle_x = find(abs(angle_1_mat(:,1) - unique_angle(i)) < 0.0001);
            for j = 1:length(angle_x)
                update_length(angle_x(j,1),int32(num/2+1-dl/pace)) =
                    abs(down_result(i)/cos(unique_rad(i)));
```

```
            end
            sum(int32(num/2+1-dl/pace)) = sum(int32(num/2+1-dl/pace)) + abs(print_down(i,1));
            if abs(print_down(i,1)) > 0.6
                fail_down = 1;
                break;
            end
        end
    end
%       假如促动器全部满足条件，视为成功,将dl存储,并且画出横坐标与伸长量print_down的图，
%    并且记录最大移动量和最小移动量
    if fail_down == 0
        success_dl(success_index,1) = -dl;
        success_index = success_index + 1;
        plot(down_result(1,2:end), -print_down(2:end,1));
        hold on
        max_move_length(int32(num/2+1-dl/pace),1) = max(print_down(2:end,1));
        min_move_length(int32(num/2+1-dl/pace),1) = min(print_down(2:end,1));
    end
%     将成功指标置为成功
    fail_down = 0;
end
hold off
%%
update_length(all(update_length==0,2),:) = [];
%%

% 加精计算-0.3 - -0.5的区间，第二次迭代
% 计算了每次抛物线上下移动相等的距离所产生的新的伸长量
surface_1 = surface.';
o = zeros(2226,2);
rad = zeros(2226,2);
min_index = -0.6;
max_index = 0.6;
for i = 1:2226
%     第一列
    x = surface(i,1);
    y = surface(i,2);
    z = surface(i,3);

    beta_d = asind(z/sqrt(x^2+y^2+z^2));
    theta_d = atand(y/x);
    beta = asin(z/sqrt(x^2+y^2+z^2));
    theta = atan(y/x);
    o(i,1) = real(beta_d);
    o(i,2) = real(theta_d);
    rad(i,1) = real(beta);
    rad(i,2) = real(theta);
```

```
end

z0 = -300.4;
F = 139.8;
unique_rad = unique(rad(:,1));
unique_angle = unique(o(:,1));
for i = 2:length(unique(rad(:,1)))
    if abs(unique_rad(i)) < pi/3
%           由于光照区域角度是60度，所以大于一定角度的点直接舍去,根据角度索引筛出符合条件的点
%           将不在光照区域内的角度除去
        unique_rad(i) = 0;
        unique_angle(i) = 0;
        continue;
    end
end
% 去除不满足的点
unique_rad(unique_rad == 0) = [];
unique_angle(unique_angle == 0) = [];
% 能够满足正负0.6的顶点移动距离条件的顶点移动距离
begin = 0.3;
pace = 0.01;
final = 0.5;
num = int32((final - begin)/pace + 1);
success_dl = zeros(num,1);
success_index = 1;
fail_up = 0;
fail_down = 0;
max_move_length = zeros(num,0);
min_move_length = zeros(num,0);
update_length = zeros(2000,num);
sum = zeros(1,num);
show_length = zeros(1,length(unique_rad));

figure;
for dl = begin:pace:final
    fail_down = 0;
    x = (-500:500);

%    顶点向下移动
    z2 = z0 - dl;
    down_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%           垂直的线不画出
            continue;
        else
            z_line = tan(unique_rad(i))*x;
```

```
        syms x_2;
        eqn = tan(unique_rad(i))*x_2 == x_2^2/(4*(F+dl))+z2;
%          联立方程
        S = solve(eqn, x_2);
        S_double = double(S);
        if abs(S_double(1,1)) > abs(S_double(2,1))
            down_result(1,i) = S_double(2,1);
        else
            down_result(1,i) = S_double(1,1);
        end
    end
end
down_result(down_result==0) = [];
%   促动器运动距离有超出范围的，失败，继续下一次循环
print_down = zeros(length(down_result),1);
for i = 1:length(down_result)
    if i == 1
        print_down(i,1) = -dl;
    else
        print_down(i,1) = down_result(i)/cos(unique_rad(i)) + x0;
%   检索当前角度的主索节点
        angle_x = find(abs(angle_1_mat(:,1) - unique_angle(i)) < 0.0001);
        for j = 1:length(angle_x)
            update_length(angle_x(j,1),int32((dl - begin)/pace + 1)) =
                abs(down_result(i)/cos(unique_rad(i)));
        end
        sum(int32((dl - begin)/pace + 1)) = sum(int32((dl - begin)/pace + 1)) +
             abs(print_down(i,1));
        if abs(print_down(i,1)) > 0.6
            fail_down = 1;
            break;
        end
    end
end
%    假如促动器全部满足条件，视为成功,将dl存储,并且画出横坐标与伸长量print_down的图,
%  并且记录最大移动量和最小移动量
if fail_down == 0
    success_dl(success_index,1) = -dl;
    success_index = success_index + 1;
    plot(down_result(1,2:end), -print_down(2:end,1));
    hold on
    max_move_length(int32((dl - begin)/pace + 1),1) = max(print_down(2:end,1));
    min_move_length(int32((dl - begin)/pace + 1),1) = min(print_down(2:end,1));
end
%   将成功指标置为成功
fail_down = 0;
end
```

```
hold off
%%
update_length(all(update_length==0,2),:) = [];
%%

% 一些变量的初始化
surface_1 = surface.';
o = zeros(2226,2);
rad = zeros(2226,2);
for i = 1:2226
    x = surface(i,1);
    y = surface(i,2);
    z = surface(i,3);
    beta_d = asind(z/sqrt(x^2+y^2+z^2));
    theta_d = atand(y/x);
    beta = asin(z/sqrt(x^2+y^2+z^2));
    theta = atan(y/x);
    o(i,1) = real(beta_d);
    o(i,2) = real(theta_d);
    rad(i,1) = real(beta);
    rad(i,2) = real(theta);
end

z0 = -300.4;
F = 139.8;
unique_rad = unique(rad(:,1));
unique_angle = unique(o(:,1));
update_length = zeros(2226,1);
for i = 2:length(unique(rad(:,1)))
    if abs(unique_rad(i)) < pi/3
%           由于光照区域角度是60度，所以大于一定角度的点直接舍去,根据角度索引筛出符合条件的点
%           将不在光照区域内的角度舍去
        unique_rad(i) = 0;
        unique_angle(i) = 0;
        continue;
    end
end
%  去除不满足的点
unique_rad(unique_rad == 0) = [];
unique_angle(unique_angle == 0) = [];
% 首次成功的顶点移动距离
first_success_dl = 0;
begin = 0.1;
pace = 0.1;
final = 0.6;
num = ((final - begin)/pace + 1)*2;
sum = zeros(1,num);
```

```
show_length = zeros(1,length(unique_rad));
```

问题 2 计算程序

```
% 根据update_length伸长量来计算移动后的坐标索引,计算得到的索引存在new_axis中
% 每次进行基准球面的计算时，需要先运行此脚本
new_axis = zeros(2226,3);
r_new_axis = zeros(2226,3);
dl_index = 1:5;
index_3 = zeros(length(dl_index),1);

global o;
for i = 1:length(o)
   if inv_r_base_axis(i,1) < 0
       o(i,2) = o(i,2) + 180;
   end
end



for j = 1:length(update_length)
   new_axis(j,1) = update_length(j,3)*cosd(o(j,1))*cosd(o(j,2));
   new_axis(j,2) = update_length(j,3)*cosd(o(j,1))*sind(o(j,2));
   new_axis(j,3) = update_length(j,3)*sin(o(j,1));
end
%    new_axis(1,1) = 0;
%    new_axis(1,2) = 0;
%    [new_angle, new_line, temp_a, temp_b, temp_, index_3] =
    triangle_calculator(fit_triangle);
global r_mat;
for i = 1:length(update_length)
   r_new_axis(i,:) = new_axis(i,:) * r_mat;
end

plot3(r_new_axis(:,1),r_new_axis(:,2),r_new_axis(:,3),'r*');

% 生成一个有空位的元胞数组存储符合条件的结点，用于索引
global angle_1
fit_point_name_2 = cell(length(r_new_axis()),1);
for i = 1:length(r_new_axis(:,1))
   if r_new_axis(i,1) ~= 0
      fit_point_name_2(i,1) = angle_1(i,1);
   end
end
fit_point_name_2_nan = fit_point_name_2;
% 生成一个没有空位的元胞数组存储符合条件的节点，用于计数
fit_point_name_2(cellfun(@isempty,fit_point_name_2)) = [];
```

```
% 根据update_length伸长量来计算移动后的坐标索引,计算得到的索引存在new_axis中

new_axis = zeros(2226,3);
r_new_axis = zeros(2226,3);
dl_index = 1:5;
index_3 = zeros(length(dl_index),1);

global o;
for i = 1:length(o)
   if inv_r_base_axis(i,1) < 0
      o(i,2) = o(i,2) + 180;
   end
end

for i = 1:length(dl_index)
    for j = 1:length(update_length)
       new_axis(j,1) = update_length(j,dl_index(i))*cosd(o(j,1))*cosd(o(j,2));
       new_axis(j,2) = update_length(j,dl_index(i))*cosd(o(j,1))*sind(o(j,2));
       new_axis(j,3) = update_length(j,dl_index(i))*sind(o(j,1));
    end
%    new_axis(1,1) = 0;
%    new_axis(1,2) = 0;
%    [new_angle, new_line, temp_a, temp_b, temp_c, index_3] =
    triangle_calculator(fit_triangle);
end
global r_mat;
for i = 1:length(update_length)
    r_new_axis(i,:) = new_axis(i,:)*r_mat;
end

plot3(r_new_axis(:,1),r_new_axis(:,2),r_new_axis(:,3),'r*');

% 生成一个没有空位的元胞数组存储符合条件的结点，用于索引
global angle_1
fit_point_name_2 = cell(length(r_new_axis()),1);
for i = 1:length(r_new_axis(:,1))
    if r_new_axis(i,1) ~= 0
       fit_point_name_2(i,1) = angle_1(i,1);
    end
end
fit_point_name_2_nan = fit_point_name_2;
% 生成一个没有空位的元胞数组存储符合条件的节点，用于计数
fit_point_name_2(cellfun(@isempty,fit_point_name_2)) = [];

% 计算了每次抛物线上下移动相等的距离所产生的新的伸长量,第一次迭代
surface_1 = surface.';
o = zeros(2226,2);
```

```
rad = zeros(2226,2);
for i = 1:2226
% 处理第二问，需要将旋转坐标轴后的点坐标带入
    x = inv_r_base_axis(i,1);
    y = inv_r_base_axis(i,2);
    z = inv_r_base_axis(i,3);

    beta_d = asind(z/sqrt(x^2+y^2+z^2));
    theta_d = atand(y/x);
    beta = asin(z/sqrt(x^2+y^2+z^2));
    theta = atan(y/x);
    o(i,1) = real(beta_d);
    o(i,2) = real(theta_d);
    rad(i,1) = real(beta);
    rad(i,2) = real(theta);
end

z0 = -300.4;
F = 139.8;
unique_rad = unique(rad(:,1));
unique_angle = unique(o(:,1));
for i = 2:length(unique(rad(:,1)))
    if abs(unique_rad(i)) < pi/3
%           由于光照区域角度是60度，所以大于一定角度的点直接舍去，根据角度索引筛出符合条件的点
%           将不在光照区域内的角度除去
        unique_rad(i) = 0;
        unique_angle(i) = 0;
        continue;
    end
end
% 去除不满足的点
unique_rad(unique_rad == 0) = [];
unique_angle(unique_angle == 0) = [];
% 能够满足正负0.6的顶点移动距离条件的顶点移动距离
success_dl = zeros(num,1);
success_index = 1;
fail_up = 0;
fail_down = 0;
max_move_length = zeros(num,0);
min_move_length = zeros(num,0);
begin = 0.1;
pace = 0.1;
final = 0.6;
num = ((final - begin)/pace + 1)*2;
update_length = zeros(2226,num);
sum = zeros(1,num);
show_length = zeros(1,length(unique_rad));
```

```
figure;
for dl = begin:pace:final
    fail_down = 0;
    x = (-500:500);
%     顶点向上移动
    z1 = z0 + dl;
%     z = x.^2/(4*(F-dl))+z1;
%     plot(x,z)
%     hold on

%   up_result代表向上移动后，各主索节点的横坐标
    up_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%           垂直的线不画出
            continue;
        else
            z_line = tan(unique_rad(i))*x;
%           plot(x,z_line);
            syms x_1;
            eqn = tan(unique_rad(i))*x_1 == x_1^2/(4*(F-dl))+z1;
%           联立方程
            S = solve(eqn, x_1);
            S_double = double(S);
            if abs(S_double(1,1)) < abs(S_double(2,1))
                up_result(1,i) = S_double(2,1);
            else
                up_result(1,i) = S_double(1,1);
            end
        end
    end
    up_result(up_result==0) = [];
%   根据满足条件的各点横坐标计算各点到c的距离，将多次的结果都存储在update_length中，
%   print_up是各下拉索移动距离的绝对值
    print_up = zeros(length(up_result),1);
    for i = 1:length(up_result)
        print_up(i,1) = up_result(i)/cos(unique_rad(i)) + z0;
%       检索当前角度的主索节点
        angle_x = find(abs(o(:,1) - unique_angle(i)) < 0.000001);
        for j = 1:length(angle_x)
            update_length(angle_x(j,1),int32(num/2+dl/pace)) =
                abs(up_result(i)/cos(unique_rad(i)));
        end
%       第一问得到706行的update_length，第二问得到692行update_length
%       对每一次各个长度的求和
        sum(int32(num/2+dl/pace)) = sum(int32(num/2+dl/pace)) + abs(print_up(i,1));
```

```
        if abs(print_up(i,1)) > 0.6
            fail_up = 1;
            break;
        end
    end
%    假如促动器全部满足条件，视为成功
        if fail_up == 0
            success_dl(success_index,1) = dl;
            success_index = success_index + 1;
            plot(up_result(1,2:end), -print_up(2:end,1));
            hold on
            max_move_length(int32(num/2+dl/pace),1) = max(print_up(2:end,1));
            min_move_length(int32(num/2+dl/pace),1) = min(print_up(2:end,1));
        end


%    顶点向下移动
    z2 = z0 - dl;
    down_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%            垂直的线不画出
             continue;
        else
            z_line = tan(unique_rad(i))*x;
%            plot(x,z_line);
            syms x_2;
            eqn = tan(unique_rad(i))*x_2 == x_2^2/(4*(F+dl))+z2;
%            联立方程
            S = solve(eqn, x_2);
            S_double = double(S);
            if abs(S_double(1,1)) > abs(S_double(2,1))
                down_result(1,i) = S_double(2,1);
            else
                down_result(1,i) = S_double(1,1);
            end
        end
    end
    down_result(down_result==0) = [];
%    促动器运动距离有超出范围的，失败，继续下一次循环
    print_down = zeros(length(down_result),1);
    for i = 1:length(down_result)
        print_down(i,1) = down_result(i)/cos(unique_rad(i)) + z0;
%        检索当前角度的主索节点
        angle_x = find(abs(o(:,1) - unique_angle(i)) < 0.0001);
        for j = 1:length(angle_x)
            update_length(angle_x(j,1),int32(num/2+1-dl/pace)) =
```

```
                abs(down_result(i)/cos(unique_rad(i)));
        end
        sum(int32(num/2+1-dl/pace)) = sum(int32(num/2+1-dl/pace)) + abs(print_down(i,1));
        if abs(print_down(i,1)) > 0.6
            fail_down = 1;
            break;
        end
    end
    %    假如促动器全部满足条件，视为成功
   if fail_down == 0
        success_dl(success_index,1) = -dl;
        success_index = success_index + 1;
        plot(down_result(1,2:end), -print_down(2:end,1));
        hold on
        max_move_length(int32(num/2+1-dl/pace),1) = max(print_down(2:end,1));
        min_move_length(int32(num/2+1-dl/pace),1) = min(print_down(2:end,1));
   end
end
%%
% update_length(all(update_length==0,2),:) = [];
%%

% 加精计算-0.3 - -0.5的区间，第二次迭代
% 计算了每次抛物线上下移动相等的距离所产生的新的伸长量
surface_1 = surface.';
o = zeros(2226,2);
rad = zeros(2226,2);
for i = 1:2226
% 处理第二问，需要将旋转后坐标轴后的点坐标带入
    x = inv_r_base_axis(i,1);
    y = inv_r_base_axis(i,2);
    z = inv_r_base_axis(i,3);

    beta_d = asind(z/sqrt(x^2+y^2+z^2));
    theta_d = atand(y/x);
    beta = asin(z/sqrt(x^2+y^2+z^2));
    theta = atan(y/x);
    o(i,1) = real(beta_d);
    o(i,2) = real(theta_d);
    rad(i,1) = real(beta);
    rad(i,2) = real(theta);
end

z0 = -300.4;
F = 139.8;
unique_rad = unique(rad(:,1));
unique_angle = unique(o(:,1));
```

```
for i = 2:length(unique(rad(:,1)))
    if abs(unique_rad(i)) < pi/3
%           由于光照区域角度是60度，所以大于一定角度的点直接舍去,根据角度索引筛出符合条件的点
%           将不在光照区域内的角度除去
        unique_rad(i) = 0;
        unique_angle(i) = 0;
        continue;
    end
end
% 去除不满足的点
unique_rad(unique_rad == 0) = [];
unique_angle(unique_angle == 0) = [];
% 能够满足正负0.6的顶点移动距离条件的顶点移动距离
success_dl = zeros(num,1);
success_index = 1;
fail_up = 0;
fail_down = 0;
max_move_length = zeros(num,0);
min_move_length = zeros(num,0);
begin = 0.3;
pace = 0.01;
final = 0.5;
num = ((final - begin)/pace + 1)*2;
update_length = zeros(2226,num);
sum = zeros(1,num);
show_length = zeros(1,length(unique_rad));

figure;
for dl = begin:pace:final
    fail_down = 0;
    x = (-500:500);

%     顶点开始移动
    z2 = z0 - dl;
    down_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%             垂直的线不画出
            continue;
        else
            z_line = tan(unique_rad(i))*x;
%             plot(x,z_line);
            syms x_2;
            eqn = tan(unique_rad(i))*x_2 == x_2^2/(4*(F+dl))+z2;
%             联立方程
            S = solve(eqn, x_2);
            S_double = double(S);
```

```
            if abs(S_double(1,1)) > abs(S_double(2,1))
                down_result(1,i) = S_double(2,1);
            else
                down_result(1,i) = S_double(1,1);
            end
        end
    end
    down_result(down_result==0) = [];
%     促动器运动距离有超出范围的，失败，继续下一次循环
    print_down = zeros(length(down_result),1);
    for i = 1:length(down_result)
        print_down(i,1) = down_result(i)/cos(unique_rad(i)) + z0;
%         检索当前角度的主索节点
        angle_x = find(abs(o(:,1) - unique_angle(i)) < 0.0001);
        for j = 1:length(angle_x)
            update_length(angle_x(j,1),int32((dl - begin)/pace + 1)) =
                abs(down_result(i)/cos(unique_rad(i)));
        end
        sum(int32((dl - begin)/pace + 1)) = sum(int32((dl - begin)/pace + 1)) +
            abs(print_down(i,1));
        if abs(print_down(i,1)) > 0.6
            fail_down = 1;
            break;
        end
    end
    %     假如促动器全部满足条件，视为成功
    if fail_down == 0
        success_dl(success_index,1) = dl;
        success_index = success_index + 1;
        plot(down_result(2:end),print_down(2:end,1));
        hold on
        max_move_length(int32((dl - begin)/pace + 1),1) = max(print_down(2:end,1));
        min_move_length(int32((dl - begin)/pace + 1),1) = min(print_down(2:end,1));
    end
    %     将成功指标置为成功
    fail_down = 0;
end
hold off
%%
update_length(all(update_length==0,2),:) = [];
%%

% 加精计算-0.3 - -0.5的区间，第二次迭代
% 计算了每次抛物线上下移动相等的距离所产生的新的伸长量
surface_1 = surface.';
o = zeros(2226,2);
rad = zeros(2226,2);
```

```
for i = 1:2226
% 处理第二问，需要将旋转坐标轴后的点坐标带入
    x = inv_r_base_axis(i,1);
    y = inv_r_base_axis(i,2);
    z = inv_r_base_axis(i,3);

    beta_d = asind(z/sqrt(x^2+y^2+z^2));
    theta_d = atand(y/x);
    beta = asin(z/sqrt(x^2+y^2+z^2));
    theta = atan(y/x);
    o(i,1) = real(beta_d);
    o(i,2) = real(theta_d);
    rad(i,1) = real(beta);
    rad(i,2) = real(theta);
end

z0 = -300.4;
F = 139.8;
unique_rad = unique(rad(:,1));
unique_angle = unique(o(:,1));
for i = 2:length(unique(rad(:,1)))
    if abs(unique_rad(i)) < pi/3
%           由于光照区域角度是60度，所以大于一定角度的点直接舍去，根据角度索引筛出符合条件的点
%           将不在光照区域内的角度除去
        unique_rad(i) = 0;
        unique_angle(i) = 0;
        continue;
    end
end
% 去除不满足的点
unique_rad(unique_rad == 0) = [];
unique_angle(unique_angle == 0) = [];
% 能够满足最大60度倾角移动距离条件的顶点移动距离
success_di = zeros(num,1);
success_index = 1;
fail_up = 0;
fail_down = 0;
max_move_length = zeros(num,0);
min_move_length = zeros(num,0);
begin = 0.3;
pace = 0.01;
final = 0.5;
num = ((final - begin)/pace + 1)*2;
update_length = zeros(2226,num);
sum = zeros(1,num);
show_length = zeros(1,length(unique_rad));
```

```
figure;
for dl = begin:pace:final
    fail_down = 0;
    x = (-500:500);

%     顶点向下移动
    z2 = z0 - dl;
    down_result = zeros(1,length(unique_rad));
    for i = 1:length(unique_rad)
        if unique_rad(i) == -pi/2 && unique_rad(i) == pi/2
%             垂直的线不画出
            continue;
        else
            z_line = tan(unique_rad(i))*x;
%             plot(x,z_line);
            syms x_2;
            eqn = tan(unique_rad(i))*x_2 == x_2^2/(4*(F+dl));
%             联立方程
            S = solve(eqn, x_2);
            S_double = double(S);
            if abs(S_double(1,1)) > abs(S_double(2,1))
                down_result(1,i) = S_double(2,1);
            else
                down_result(1,i) = S_double(1,1);
            end
        end
    end
    down_result(down_result==0) = [];
%     促动器运动距离有超出范围的，失败，继续下一次循环
    print_down = zeros(length(down_result),1);
    for i = 1:length(down_result)
        print_down(i,1) = down_result(i)/cos(unique_rad(i)) + z0;
%         找到当前对称的点并下压
        angle_x = find(abs(o(:,1) - unique_angle(i)) < 0.0001);
        for j = 1:length(angle_x)
            update_length(angle_x(j,1),int32((dl - begin)/pace + 1)) =
                abs(down_result(i)/cos(unique_rad(i)));
        end
        sum(int32((dl - begin)/pace + 1)) = sum(int32((dl - begin)/pace + 1)) +
            abs(print_down(i,1));
        if abs(print_down(i,1)) > 0.6
            fail_down = 1;
            break;
        end
    end
    %     假如促动器全部满足条件，视为成功
  if fail_down == 0
```

```
        success_dl(success_index,1) = -dl;
        success_index = success_index + 1;
        plot(down_result(1,2:end), -print_down(2:end,1));
        hold on
        max_move_length(int32((dl - begin)/pace + 1),1) = max(print_down(2:end,1));
        min_move_length(int32((dl - begin)/pace + 1),1) = min(print_down(2:end,1));
   end
   %     将成功指标置为成功
    fail_down = 0;
end
hold off
%%
update_length(all(update_length==0,2),:) = [];
%%

% 尝试将问题二转化为问题一，生成旋转矩阵，运行第二问的程序首先需要运行该脚本

beta_2 = 78.169;
alpha_2 =36.795;
r_mat = [sind(alpha_2), -cosd(alpha_2), 0;
        sind(beta_2)*cosd(alpha_2), sind(beta_2)*sind(alpha_2), -cosd(beta_2);
        cosd(beta_2)*cosd(alpha_2), cosd(beta_2)*sind(alpha_2), sind(beta_2)];
r_base_axis = zeros(length(base_axis), 3);
inv_r_base_axis = zeros(length(base_axis), 3);
couple = zeros(length(base_axis), 2);
index = 1;
base_axis = cell2mat(angle_1(:,2:4));
% 将原来的基准面向目标光线旋转
for i = 1:length(base_axis)
    r_base_axis(i,:) = base_axis(i,:)*r_mat;
end
% 将轴旋转至和目标光线平行
for i = 1:length(base_axis)
    inv_r_base_axis(i,:) = base_axis(i,:)/r_mat;
end
% 验证是否配对
for i = 1:length(base_axis)
   for j = 1:length(base_axis)
    if abs(r_base_axis(i,1) - base_axis(j,1)) < 0.001 && abs(r_base_axis(i,2) - base_axis(j,2))
         < 0.001 ...
        && abs(r_base_axis(i,3) - base_axis(j,3)) < 0.001
        couple(index,1) = j;
        couple(index,2) = i;
        index = index + 1;
    end
   end
end
```

问题 3 计算程序

```
% 计算旋转坐标轴后基准球面的反射吸收率指标，方便与抛物面进行比较
axis_1 = zeros(4300,1);
axis_2 = zeros(4300,1);
axis_3 = zeros(4300,1);
fit_triangle = zeros(4300,3);
belong_index = 0;

for i = 1:4300
    a = triangle(i,1);
    a = string(a);
    axis_1(i,1) = find(strcmp(a, point_name));
    b = triangle(i,2);
    b = string(b);
    axis_2(i,1) = find(strcmp(b, point_name));
    c = triangle(i,3);
    c = string(c);
    axis_3(i,1) = find(strcmp(c, point_name));
end
% 找到符合半径条件的反射面板，若三角形三个点全在光照区域内，视作有效顶点
for i = 1:4300
% 当旋转后的基准面的每个三角形都在光照区域内，这个三角形有效
    if abs(o(axis_1(i,1),1)) > 60 && abs(o(axis_2(i,1),1)) > 60 && abs(o(axis_3(i,1),1)) > 60
        fit_triangle(i,1) = axis_1(i,1);
        fit_triangle(i,2) = axis_2(i,1);
        fit_triangle(i,3) = axis_3(i,1);
    end
%     belong_index = 0;
end
% 去除不满足条件的反射面板（去除0值）
fit_triangle = fit_triangle(any(fit_triangle,2),:);
new_axis = inv_r_base*axis;

[new_angle, new_line, temp_a, temp_b, temp_c, valid_num, area_c] =
    triangle_calculator(fit_triangle);

% 计算每个符合条件反射面板的法向量以及法向量与z轴的夹角

axis_1 = zeros(4300,1);
axis_2 = zeros(4300,1);
axis_3 = zeros(4300,1);
fit_triangle = zeros(4300,3);
belong_index = 0;
```

```
for i = 1:4300
    a = triangle(i,1);
    a = string(a);
    axis_1(i,1) = find(strcmp(a, point_name));
    b = triangle(i,2);
    b = string(b);
    axis_2(i,1) = find(strcmp(b, point_name));
    c = triangle(i,3);
    c = string(c);
    axis_3(i,1) = find(strcmp(c, point_name));
end
% 找到符合半径条件的反射面板，若三角形三个点全在光照区域内，视作有效顶点
for i = 1:4300
%     等于0代表该点无效
    pre_1 = (new_axis(axis_1(i,1),2) == 0);
    pre_2 = (new_axis(axis_2(i,1),2) == 0);
    pre_3 = (new_axis(axis_3(i,1),2) == 0);
%     mat_angle_1 = cell2mat(angle_1(:,5));
%     belong_index = belong_index + (abs(mat_angle_1(axis_1(i,1),1)) > 60) +
     (abs(mat_angle_1(axis_2(i,1),1)) > 60) + (abs(mat_angle_1(axis_3(i,1),1)) > 60);
    if pre_1 + pre_2 + pre_3 == 0
        fit_triangle(i,1) = axis_1(i,1);
        fit_triangle(i,2) = axis_2(i,1);
        fit_triangle(i,3) = axis_3(i,1);
    end
%     belong_index = 0;
end
% 去除不满足条件的三角面板（去除0值）
fit_triangle = fit_triangle(any(fit_triangle,2),:);

[new_angle, new_line, temp_a, temp_b, temp_c, valid_num, area_r] =
    triangle_calculator(fit_triangle);
```

关键程序

```
% 根据输入的（n,3）的矩阵序列，在new_axis中找到组成三角形的三个点，并且计算：各个三角形的法向量；
  各个三角形反射光线方向向量；
% 各个三角形在馈源口平面的投影，以及所有三角形与馈源仓交集区域的总面积（作为光线吸收率的重要指标）
% A = "A0";
% temp = string(new_axis(:,1));
% [i,j] = find(strcmp(A, temp));

function [j_angle, j_line, temp_a, temp_b, temp_c, valid_num, sum_area] =
    triangle_calculator(new_fit_index)
    global new_axis;
    j_line = zeros(length(new_fit_index),3);
    light_normal_vector = zeros(length(new_fit_index(:,1)),3);
```

```
    global temp_a;
    global temp_b;
    global temp_c;
    global new_axis;
    x_in = zeros(length(new_fit_index(:,1)),1);
    y_in = zeros(length(new_fit_index(:,1)),1);
    z_in = zeros(length(new_fit_index(:,1)),1);
    x_zp = zeros(length(new_fit_index(:,1)),3);
    y_zp = zeros(length(new_fit_index(:,1)),3);
    z_p = -160.2;
    sum_area = 0;
    valid_num = 0;
    D = zeros(length(new_fit_index),1);
    for i = 1:length(new_fit_index(:,1))
%        start_points =
     [new_axis(new_fit_index(i,1),1:3);new_axis(new_fit_index(i,2),1:3);new_axis(new_fit_index(i,3),1:3)];
%        end_points =
     [new_axis(new_fit_index(i,1),1:3);new_axis(new_fit_index(i,2),1:3);new_axis(new_fit_index(i,3),1:3)];
%        X=[start_points(:,1) end_points(:,1)]';
%        Y=[start_points(:,2) end_points(:,2)]';
%        Z=[start_points(:,3) end_points(:,3)]';
%        line(X, Y ,Z)
%        获取同一个面的三个点坐标
        temp_a(i,:) = new_axis(new_fit_index(i,1),1:3);
        temp_b(i,:) = new_axis(new_fit_index(i,2),1:3);
        temp_c(i,:) = new_axis(new_fit_index(i,3),1:3);
%        求解平面的法向量
        j_line(i, 1) = (temp_b(i,2) - temp_a(i,2))*(temp_c(i,3) - temp_a(i,3)) - (temp_b(i,3) -
            temp_a(i,3))*(temp_c(i,2) - temp_a(i,2));
        j_line(i, 2) = (temp_c(i,1) - temp_a(i,1))*(temp_b(i,3) - temp_a(i,3)) - (temp_b(i,1) -
            temp_a(i,1))*(temp_c(i,3) - temp_a(i,3));
        j_line(i, 3) = (temp_b(i,1) - temp_a(i,1))*(temp_c(i,2) - temp_a(i,2)) - (temp_c(i,1) -
            temp_a(i,1))*(temp_b(i,2) - temp_a(i,2));
        if j_line(i,3) < 0
            j_line(i,:) = -j_line(i,:);
        end
        D(i,1) = - (j_line(i, 1)*temp_a(i,1) + j_line(i, 2)*temp_a(i,2) + j_line(i,
            3)*temp_a(i,3));
    end
%   求解法向量和z轴的夹角
    j_angle = zeros(length(new_fit_index(:,1)),1);
    for i = 1:length(new_fit_index(:,1))
        j_angle(i,1) = acosd(abs(j_line(i,3)/sqrt(j_line(i,1)^2 + j_line(i,2)^2 +
            j_line(i,3)^2)));
    end
%   求解反射光线的方向
    for i = 1:length(new_fit_index(:,1))
```

```matlab
        light_normal_vector(i,1) = j_line(i, 1);
        light_normal_vector(i,2) = j_line(i, 2);
        light_normal_vector(i,3) = j_line(i, 3) - sqrt(j_line(i,1)^2 + j_line(i,2)^2 +
            j_line(i,3)^2)/(2*cosd(j_angle(i)));
%         求解交点
        x_in(i,1) = -(j_line(i, 1)*(j_line(i, 3)*z_p + D(i,1))/(j_line(i, 1)^2 + j_line(i, 2)^2
            + j_line(i, 3)*light_normal_vector(i,3)));
        y_in(i,1) = -(j_line(i, 2)*(j_line(i, 3)*z_p + D(i,1))/(j_line(i, 1)^2 + j_line(i, 2)^2
            + j_line(i, 3)*light_normal_vector(i,3)));
        z_in(i,1) = ((j_line(i, 1)^2 + j_line(i, 2)^2)*z_p -
            light_normal_vector(i,3)*D(i,1))/(j_line(i, 1)^2 + j_line(i, 2)^2 + j_line(i,
            3)*light_normal_vector(i,3));
%         通过公式查看交点是否在三角形内部，假如在内部则计数加1
% u = ((v1•v1)(v2•v0)-(v1•v0)(v2•v1)) / ((v0•v0)(v1•v1) - (v0•v1)(v1•v0))
%
% v = ((v0•v0)(v2•v1)-(v0•v1)(v2•v0)) / ((v0•v0)(v1•v1) - (v0•v1)(v1•v0))
        v0 = temp_c(i,:) - temp_a(i,:);
        v1 = temp_b(i,:) - temp_a(i,:);
        v2 = [x_in(i,1),y_in(i,1),z_in(i,1)] - temp_a(i,:);
        U = (dot(v1, v1)*dot(v2, v0) - dot(v1, v0)*dot(v2, v1)) / (dot(v0, v0)*dot(v1, v1) -
            dot(v0, v1)*dot(v1, v0));
        V = (dot(v1, v1)*dot(v2, v0) - dot(v1, v0)*dot(v2, v1)) / (dot(v0, v0)*dot(v1, v1) -
            dot(v0, v1)*dot(v1, v0));
        if U >= 0 && V >= 0 && U+V <= 1
            valid_num = valid_num + 1;
        end
    end
%     计算经过三角形的反射光线到馈源舱平面的投影
    for i = 1:length(new_fit_index(:,1))
        x_zp(i,1) = light_normal_vector(i,1)/light_normal_vector(i,3)*(z_p - temp_a(i,3)) +
            temp_a(i,1);
        x_zp(i,2) = light_normal_vector(i,1)/light_normal_vector(i,3)*(z_p - temp_b(i,3)) +
            temp_b(i,1);
        x_zp(i,3) = light_normal_vector(i,1)/light_normal_vector(i,3)*(z_p - temp_c(i,3)) +
            temp_c(i,1);
        y_zp(i,1) = light_normal_vector(i,2)/light_normal_vector(i,3)*(z_p - temp_a(i,3)) +
            temp_a(i,2);
        y_zp(i,2) = light_normal_vector(i,2)/light_normal_vector(i,3)*(z_p - temp_b(i,3)) +
            temp_b(i,2);
        y_zp(i,3) = light_normal_vector(i,2)/light_normal_vector(i,3)*(z_p - temp_c(i,3)) +
            temp_c(i,2);
    end

%    计算所有三角形与馈源仓的交集面积之和
    circle_x = zeros(1000,1);
    cirlce_y = zeros(1000,1);
    theta = 2*pi/1000;
```

```
    for i = 1:1000
        circle_x(i,1) = 0.5*cos((i-1)*theta);
        cirlce_y(i,1) = 0.5*sin((i-1)*theta);
    end
    circle = polyshape(circle_x, cirlce_y);
    for i = 1:length(new_fit_index(:,1))
        poly1 = polyshape(x_zp(i,:), y_zp(i,:));
        polyout = intersect(circle, poly1);
        single_area = area(polyout);
        single_area = single_area * cosd(j_angle(i))*light_normal_vector(i,3)/...
            (sind(j_angle(i))*sqrt(norm(light_normal_vector(i,:))^2 -
                light_normal_vector(i,3)^2) + cosd(j_angle(i))*light_normal_vector(i,3));
        sum_area = sum_area + single_area;
    end
end
```

附录 B ”result.xlsx”文件部分截图

A	B	C
X坐标（米）	Y坐标（米）	Z坐标（米）
0	0	[illegible]

图 7 result.xlsx

A	B	C	D
节点编号	X坐标（米）	Y坐标（米）	Z坐标（米）
D27	-49.386703	[illegible]	[illegible]
D35	-59.080166	[illegible]	[illegible]
D20	[illegible]	[illegible]	-296.2591
D34	-49.381044	-47.833443	[illegible]
D21	[illegible]	[illegible]	[illegible]
D26	[illegible]	[illegible]	[illegible]
D28	[illegible]	[illegible]	[illegible]
D15	-39.568378	[illegible]	[illegible]
D43	[illegible]	[illegible]	[illegible]
E184	[illegible]	[illegible]	[illegible]
C170	[illegible]	[illegible]	-248.53984
D154	[illegible]	[illegible]	-244.30387
A55	[illegible]	[illegible]	-284.90521
A93	[illegible]	[illegible]	-268.87184
D36	[illegible]	[illegible]	-290.97799
A95	[illegible]	116.731456	-273.06576
D19	[illegible]	[illegible]	-296.46957
D33	[illegible]	-54.500566	-293.12023

(a) result.xlsx

对应主索节点编号	伸缩量（米）
D27	-0.41
D35	-0.396922096
D20	-0.396917935
D34	-0.396738752
D21	-0.396678982
D26	-0.392323478
D28	-0.392266133
D15	-0.375360367
D43	-0.373500929
E184	-0.371305501
C170	-0.370702887
D154	-0.368119222
A55	-0.36622547
A93	-0.36405282
D36	-0.362404103
A95	-0.362250482
D19	-0.36190664
D33	-0.361867468

(b) result.xlsx

图 8

附录 C 问题 1——各抛物面相关参数与评分表

顶点	移动总距离	正最大值	负最大值	有效面板	I1	I2	I2	I3	总分
0.2	67.647	0.5066	0.1869	1318	84.03045326	15.56666667	68.85	99.62207105	51.35233821
0.3	62.8131	0.3996	0.2869	1318	85.17160057	33.4	52.18333333	99.62207105	51.99017697
0.4	70.2095	0.2926	0.387	1319	83.42551936	51.23333333	35.5	99.69765684	52.19204408
0.5	87.3225	0.1855	0.487	1319	79.38562323	69.08333333	18.83333333	99.69765684	52.05939299
0.6	112.6378	0.0785	0.5871	1319	73.40939566	86.91666667	2.15	99.69765684	51.62295886
0.3	62.81310236	0.399588	0.286901363	1319	85.17160001	33.40200103	52.18310613	99.69765684	51.99466571
0.31	63.06817323	0.3888842	0.296908281	1319	85.11138498	35.18596746	50.51528645	99.69765684	52.03209215
0.32	63.36933776	0.3781804	0.306915199	1319	85.04028854	36.96992579	48.84746686	99.69765684	52.06788318
0.33	63.77674594	0.3674767	0.316922116	1319	84.94411097	38.75387602	47.17964737	99.69765684	52.09990884
0.34	64.22566111	0.3567731	0.326929032	1319	84.83813477	40.53781816	45.51182797	99.69765684	52.13046151
0.35	64.72390727	0.3460695	0.336935948	1319	84.72051292	42.3217522	43.84400866	99.69765684	52.15926412
0.36	65.54774008	0.3353659	0.346942863	1319	84.52602925	44.10567814	42.17618944	99.69765684	52.17653426
0.37	66.5503631	0.3246624	0.356949778	1319	84.28933827	45.88959599	40.50837032	99.69765684	52.18747011
0.38	67.71124505	0.313959	0.366956692	1319	84.01528681	47.67350576	38.84055129	99.69765684	52.19279868
0.39	68.9603884	0.3032556	0.376963606	1319	83.72039934	49.45740656	37.17273235	99.69765684	[illegible]
0.4	70.20953818	0.2925522	0.386970519	1319	83.42551035	51.2412984	35.50491351	99.69765684	[illegible]
0.41	71.4586944	0.2818489	0.396977431	1319	83.13061983	53.02518215	33.83709476	99.69765684	[illegible]
0.42	72.86681467	0.2711457	0.406984343	1319	82.79820239	54.80905782	32.16927611	99.69765684	[illegible]
0.43	74.39130215	0.2604424	0.416991255	1319	82.43831394	56.59292541	30.50145754	[illegible]	[illegible]
0.44	75.95817246	0.2497393	0.426998166	1319	82.0684201	58.37678491	28.83363907	[illegible]	[illegible]
0.45	77.69656307	0.2390362	0.437005076	1319	81.65803516	60.16063634	27.16582069	[illegible]	52.16417093
0.46	79.4906245	0.2283331	0.447011986	1319	81.23450791	61.94447969	25.49800241	[illegible]	[illegible]
0.47	81.29610458	0.2176301	0.457018895	1319	80.80828504	63.72831496	[illegible]	[illegible]	52.12952527
0.48	83.28368819	0.2069271	0.467025803	1319	80.33907267	65.51214216	[illegible]	[illegible]	52.10554705
0.49	85.30309547	0.1962242	0.477032711	1319	79.86234762	67.29596129	[illegible]	[illegible]	[illegible]
0.5	87.32251244	0.1855214	0.487039619	1319	79.38562029	69.07977234	[illegible]	[illegible]	[illegible]

图 **9**　问题 **1** 各抛物面相关参数与评分表

附录 D　问题 2——各抛物面相关参数与评分表

顶点	移动总距离	正最大值	负最大值	有效面板	I1	I2	I2	I3	总分
0.2	59.4735	0.5067	0.1868	1318	[illegible]	[illegible]	[illegible]	99.62207105	51.64176809
0.3	54.8659	0.3996	0.2868	[illegible]	[illegible]	[illegible]	[illegible]	99.62207105	52.27826007
0.4	61.2932	0.2925	[illegible]	1319	[illegible]	51.25	[illegible]	99.69765684	52.52111042
0.5	75.4383	0.1855	0.487	1319	[illegible]	[illegible]	[illegible]	99.69765684	52.48022161
0.6	97.5311	0.0786	[illegible]	[illegible]	[illegible]	[illegible]	2.15	99.69765684	52.15123199
0.3	62.81310236	[illegible]	[illegible]	[illegible]	85.17160001	[illegible]	52.19238283	99.69765684	51.99758458
0.31	63.06817323	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	50.52455817	99.69765684	52.03517806
0.32	63.36933776	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	48.85673367	99.69765684	52.07112266
0.33	63.77674594	[illegible]	[illegible]	[illegible]	[illegible]	38.75303983	47.18890917	99.69765684	52.10327909
0.34	64.22566111	[illegible]	[illegible]	[illegible]	[illegible]	40.53726267	45.52108483	99.69765684	52.13394206
0.35	64.72390727	[illegible]	[illegible]	1319	[illegible]	42.3214175	43.85326067	99.69765684	52.16283105
0.36	65.54774008	[illegible]	[illegible]	1319	[illegible]	44.1055185	42.1854365	99.69765684	52.18016923
0.37	66.5503631	[illegible]	[illegible]	1319	84.28933827	45.8896115	40.5176125	99.69765684	52.19117318
0.38	[illegible]	[illegible]	[illegible]	1319	84.01528681	47.67369633	38.8497885	99.69765684	52.1965698
0.39	[illegible]	[illegible]	[illegible]	[illegible]	83.72039934	49.457773	37.18196467	99.69765684	52.19883781
0.4	[illegible]	[illegible]	[illegible]	1319	83.42551035	51.2416945	35.514141	99.69765684	52.20104359
0.41	[illegible]	[illegible]	[illegible]	1319	83.13061983	53.02527083	33.84631733	99.69765684	52.20311108
0.42	[illegible]	[illegible]	[illegible]	1319	82.79820239	54.80883883	32.17849367	99.69765684	52.1995462
[illegible]	[illegible]	[illegible]	[illegible]	1319	82.43831394	56.592399	30.51067017	99.69765684	52.1918576
[illegible]	[illegible]	[illegible]	[illegible]	1319	82.0684201	58.375951	28.84284683	99.69765684	52.18266499
[illegible]	[illegible]	[illegible]	[illegible]	1319	81.65803516	60.159495	27.1750235	99.69765684	52.16739552
[illegible]	[illegible]	[illegible]	0.446957	1319	81.23450791	61.94303083	25.50720033	99.69765684	52.15015149
[illegible]	[illegible]	[illegible]	0.456964	1319	80.80828504	63.72639567	23.83937717	99.69765684	52.13243473
0.48	[illegible]	[illegible]	0.466971	1319	80.33907267	65.50965933	22.17155417	99.69765684	52.10822914
0.49	[illegible]	[illegible]	0.476978	1319	79.86234762	67.29291483	20.50373133	99.69765684	52.08289345
0.5	[illegible]	[illegible]	0.486985	1319	79.38562029	69.07616233	18.8359085	99.69765684	52.05755422

图 **10**　问题 **2** 各抛物面相关参数与评分表