

基于逐步迭代方法的 FAST 节点调节优化

摘要

FAST 望远镜通过调节反射曲面拟合抛物面，以提高观测精度，理想抛物面的选取与节点调节算法关系到拟合精度，本文基于黄金分割法与迭代优化建立了反射平面调节与评估的数学模型。

对于问题一，通过二维投影与极坐标建立简化抛物面方程，并集中优化参数到抛物面顶点偏移量 h ，以节点最小径向移动距离为目标，设置步长试探得到偏移量与移动距离的关系，在可调节区间内呈单下峰函数。再利用黄金分割法逐步缩小区间，逼近极小值点，精度设置为 0.0001，求得最优解 $h = 0.4444m$ ，据此确定抛物线方程 $x^2 + y^2 = 2 \times 280.8616z + 168991.27908$ 。

对于问题二，利用问题一的模型，根据方位角 α 与仰角 β 确定旋转对称轴的方向向量，进一步求得偏移量 $h = 0.4380m$ ，抛物线顶点坐标为 $(-49.3919, -36.9432, -294.4472)$ 。抛物面的拟合为逐步迭代优化过程，优化目标为拟合均方根误差 RMS 最小，借助数值积分的思想，在各反射平面内均匀设下若干个统计点，并计算各节点附近的统计点与理想抛物面的平均距离，以此距离反向移动各节点，迭代计算，直到误差 RMS 值趋于稳定，稳定值 RMS 为 $0.0146m$ 。最终迭代求得各节点的位置坐标与主索节点伸缩量，并验证了主索节点的伸缩量和促动器伸缩量一致。

对于问题三，基于问题二得到的结果点集进行平滑插值，使点更加均匀分布，近似为抛物面。再根据方位角与仰角设置旋转矩阵，对坐标点进行旋转操作，简化竖直抛物面形式。利用其对称性，采用降维的思想进行平面截取，将问题转换为平面求解，分别计算各离散点的电磁波反射情况，确定接收比。通过多平面截取计算，进一步优化结果，提高精度，最终结果为 83.80% 。对于基准球面的反射，可利用球面自身的几何性质，计算得到可被有效接收的区域面积，得到接收比为 51.74%

最后，本文对相关参数作了灵敏度分析，并对模型作出了评价。

关键字： FAST 整网控制 黄金分割法 逐步迭代优化 平滑插值 平面截取降维

一、问题重述

1.1 问题背景

“FAST”天眼——500 米口径球面射电望远镜，是我国重要的一项科研成果，主动性强，精度高，在世界上处于领先地位。FAST 由主动反射面，馈源舱及相关测量系统组成。其中主动反射面由下拉索、促动器及物理支撑结构控制，形成可调节球面。球面由多个三角形反射面板凭借而成，反射面板顶点固定在主索节点上并连接下拉索，并通过促动器主动调节下拉索来控制主索节点的位置，从而调节多个反射面板形成工作抛物面以汇集电磁波至馈源舱，达到观测的目的。馈源舱接受信号区域为直径 1 米的圆盘，且仅可在距离基准球面球心 $F=0.466R$ 的同心球面上运动。主动反射面共有基准态和工作态两个状态。基准态下，反射面形成基准球面，其半径约 300 米，口径为 500 米；工作态下，主动反射面的部分被调节为工作抛物面，为 300 米口径的近似旋转抛物面，要求以馈源舱为焦点，且馈源舱、基准球面球心、被观测物体三点共线。通过调节沿基准球面径向安装的促动器，可以调节主索节点的位置，进而调节反射面板形成工作抛物面。下拉索长度固定，促动器的径向伸缩范围为 $-0.6+0.6$ 米。

1.2 问题要求

- **问题一** 对于基准球面正上方的观测天体，综合考虑反射面板等因素，确定理想抛物面。
- **问题二** 当被测天体方位角 $\alpha=36.795^\circ$ ，仰角 $\beta=78.169^\circ$ 时，建立相关数学模型，确定理想抛物面与各主索节点的移动，要求工作抛物面尽可能贴近理想抛物面。并将主索节点位置、促动器伸缩量等数据记录在“result.xlsx”文件中。
- **问题三** 在第二问确定的主索节点坐标基础上，计算该调节方案计算馈源舱有效区域接收到的反射信号与 300 米口径内反射面的反射信号之比，并计算基准反射球面的接收比，将两者结果进行比较。

二、问题分析

2.1 问题一的分析

题目给出的被测物体在竖直方向上，考虑到理想抛物面和基准球面都具有高度对称性，将问题投影在 xOz 平面上并建立极坐标系进行分析。考虑到应避免主索的过度松弛或绷紧导致应力过度，主索节点的移动位移量应尽可能小。在极坐标下表示出抛物线的

一般方程，初始顶点落在基准球面上，设置顶点在 z 轴上的偏移量 h ，调整并确定 h ，使各主索节点到理想抛物面的径向距离最小。

2.2 问题二的分析

确定理想抛物面的过程与第一问同理。假设反射面板为平面，主索节点只有径向位移。根据天体方位，选取主索节点与坐标原点连线与抛物线主轴夹角在 30° 以内的主索节点，并在照明区域内的每个反射均匀选取若干个样本点。首先将主索节点置于确定的理想抛物面上，选取与其连接的每个反射面板上与其距离最近的 12 个点，计算这些点与抛物面径向距离的平均值，并计算所有样本点和主索节点与抛物面径向距离的均方差值。接下来将得到的平均值作为相应主索节点的径向移动距离进行下一步调节，并计算调节后所有样本点和主索节点与抛物面径向距离的均方差值。如此反复迭代，直至前后两次计算得到的均方差值之差小于 $0.001m$ 停止迭代，此时的方案即为最佳方案。根据主索节点调节后坐标与初始坐标距离之差即可求得促动器伸缩量。

2.3 问题三的分析

将问题二中得到的节点位置坐标进行平滑插值，得到一组分布表现更加光滑的点集，近似为抛物面。将点集进行坐标系旋转操作，使旋转后的坐标系 z 轴与基准球心同被测天体的连线重合，电磁波竖直射下。根据图形的对称性将其降维在平面上以简化运算。选取 xOz 平面截取，得到分布在该平面上的若干散点，分别计算各点处电磁波的反射情况，统计可被馈源舱接收的数目，其与平面上散点点总数比的平方，即为此时的接收比。选取 yOz 平面截取，重复上述操作，得到的结果求平均，可进一步精确，减小误差。对于基准球面的接受比，同样可利用其对称性投影在平面上进行求解，利用几何关系解得可被馈源舱接收信号的部分，其投影面积与总面积比，即为所求。

三、模型假设

- 每个反射面大小规格一致，且大致为一个平面。
- 反射平面足够坚固，节点移动导致的距离变化不会对反射面形状产生影响。
- 基准状态下，反射平面之间的空隙可忽略，球面为完整球面

四、符号说明

符号	说明	单位
θ	向量与极轴夹角	-
R	基准球面半径	m
d	点到抛物面的径向距离	m
h	理想抛物面顶点与球面的偏移量	m
h^*	节点径向位移最小的抛物面偏移量	m
RMS	工作抛物面与理想抛物面的拟合误差	m
\vec{K}	旋转对称轴方向向量	-

五、模型的建立和求解

5.1 问题一模型的建立和求解

5.1.1 对球面半径 R 的修正

根据附件一的各主索节点 XYZ 坐标计算各点与原点的距离，计算结果显示各点并非严格落在半径 300 米的球面上，考虑到工程实现应尽量以实际情况优先，计算各点与原点距离，得到其值基本都为 300.4m，因此球面半径 $R = 300.4m$ 。

5.1.2 一般情况下理想抛物面的极坐标方程

考虑到理想抛物面关于旋转对称轴对称，以球心 C 作为原点，以旋转对称轴为极轴，建立极坐标系，其中极径 r 为抛物面上的点到原点的距离，极角为 θ 。为方便表述，将抛物面和球面投影在 xOz 平面上进行求解，如图1和图2所示。

设 h 表示抛物面顶点与基准球面的垂直距离， P 为抛物面的焦距，由 $F = 0.466R$ ，求得焦距 $P = 2(F + h)$ 。已知了焦点位置和焦距，则抛物线方程可唯一确定。 θ 表示抛物线上各点与原点连线同 z 轴夹角，此时的抛物线方程在直角坐标系下的一般形式为：

$$x^2 = 2P(z + R + h) \quad (1)$$

根据 x, z 和 r, θ 的对应关系：

$$\begin{cases} x = r \sin \theta \\ z = -r \cos \theta \end{cases} \quad (2)$$

将方程转换为极坐标系下，得到抛物面的极坐标方程：

$$r^2 \sin^2 \theta = 2P(-r \cos \theta + R + h) \quad (3)$$

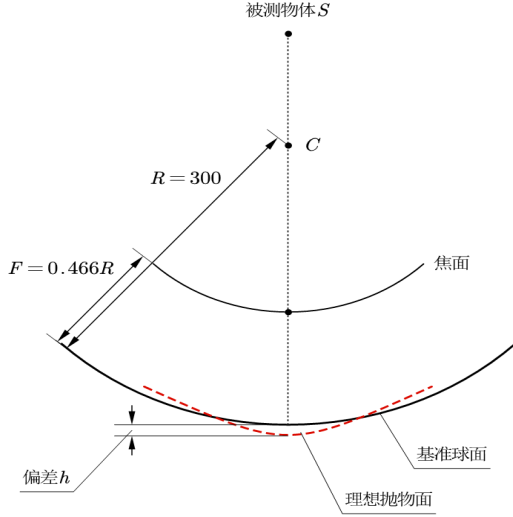


图 1 理想抛物面示意图

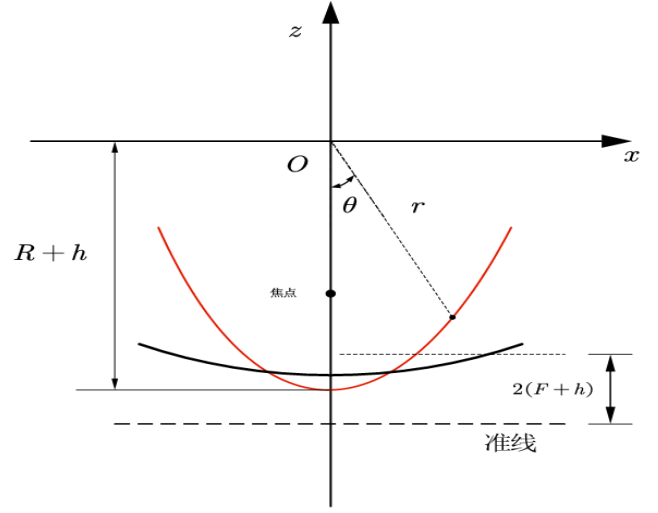


图 2 理想抛物面极坐标平面示意图

将 r 视作因变量, θ 作为自变量, 通过求根公式解得 r 与 θ 的显式关系 (另一解不满足实际情况, 舍去):

$$r = \frac{-2P\cos\theta + \sqrt{4P^2\cos^2\theta + 4\sin^2\theta \cdot 2P(R+h)}}{2\sin^2\theta} \quad (4)$$

设 M 为主索节点的集合, 对于某个主索节点 M_i , (i 为节点编号, $i = 1, 2, \dots, n$, n 为节点个数), 其 XYZ 坐标为 (x_i, y_i, z_i) , 其到原点的距离 D_i 为

$$D_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (5)$$

由题可知, 通过天体所在方位可以确定抛物面的旋转对称轴, 设旋转对称轴的方向向量为 \vec{K} , 则由方位角 α 和仰角 β 确定其表达式为

$$\vec{K} = (-\cos\beta\cos\alpha, -\cos\beta\sin\alpha, -\sin\beta) \quad (6)$$

以基准球面的球心 C 作为原点, 则球心 C 的 XYZ 坐标为 $(0, 0, 0)$, 设某个主索节点 M_i 的 XYZ 坐标为 (x_i, y_i, z_i) , 向量 \vec{CM}_i 与旋转对称轴向量 \vec{K} 的夹角为 θ_i , 由向量夹角公式可得:

$$\theta_i = \arccos \frac{\vec{K} \cdot \vec{CM}_i}{|\vec{K}| |\vec{CM}_i|} \quad (7)$$

由于主索节点径向移动, 因此移动前后的 θ_i 不变, 由抛物面的极坐标方程式 (4) 求得同一夹角下抛物面上的点到原点的距离 r_i , 即:

$$r_i = \frac{-P\cos\theta_i + \sqrt{P^2\cos^2\theta_i + \sin^2\theta_i \cdot 2P(R+h)}}{\sin^2\theta_i} \quad (8)$$

则主索节点 M_i 沿径向移动的距离 Δd_i 为:

$$\Delta d_i = r_i - D_i \quad (9)$$

5.1.3 问题一单目标优化模型的建立

1. 优化目标 给定抛物面顶点与基准球面的垂直距离 h 的值, 根据式 (9) 可以求得每个主索节点移动到理想抛物面的位移。

根据工程实际要求, 结合反射面板调节的因素, 主索的过度紧绷或松弛导致应力过大, 对工程的寿命与精度都有一定的影响, 为避免这种情况, 主索节点的位移应在合理范围内且尽可能的小。各节点仅能沿径向移动, 所以应使各节点到达抛物面所需的平均径向距离最小, 基于此标准对决策变量 h 提出优化目标:

$$\min D = \sqrt{\frac{\sum_{i=1}^n \Delta d_i^2}{n}} \quad (10)$$

其中 D 为各节点径向移动距离均方。

2. 约束条件 因为抛物面的口径为 300m, 又由于球面半径为 300.4m, 抛物面的左右边缘点与球心构成等边三角形, 因此边缘点与原点的连线同旋转对称轴的夹角为 30° 。对于需要移动的主索节点, 其与原点的连线同旋转对称轴的夹角必然小于等于 30° , 如图3所示。因此需要移动的主索节点满足的约束条件为:

$$\theta_i \leq 30^\circ \quad (11)$$

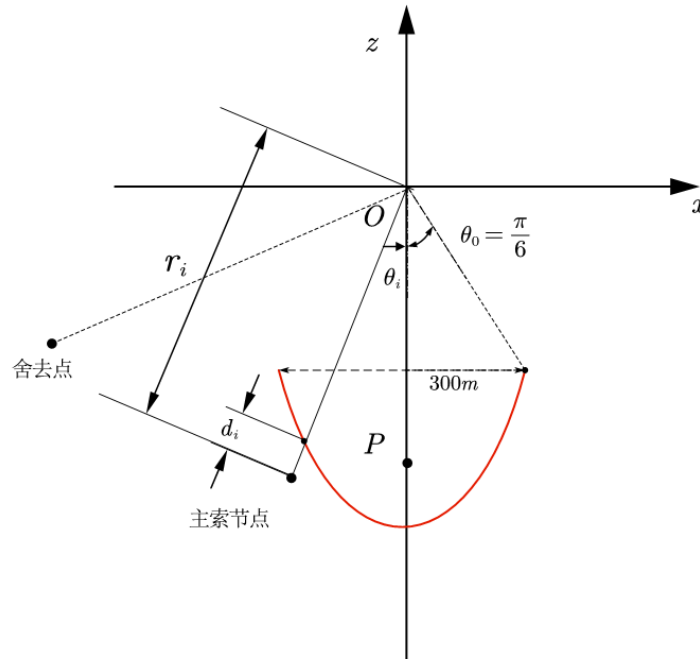


图 3 主索节点径向位移

综上, 可以建立以 h 为决策变量, 径向移动距离均方最小为目标函数的单目标规划

模型:

$$\hat{h} = \arg \min_h D = \sqrt{\frac{\sum_{i=1}^n \Delta d_i^2}{n}} \quad (12)$$

$$\begin{cases} \theta_i = \arccos \frac{\vec{K} \cdot \vec{CM}_i}{|\vec{K}| |\vec{CM}_i|} \leq 30^\circ \\ r_i = \frac{-P \cos \theta_i + \sqrt{P^2 \cos^2 \theta_i + \sin^2 \theta_i \cdot 2P(R+h)}}{\sin^2 \theta_i} \\ D_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \\ \Delta d_i = r_i - D_i \end{cases}$$

5.1.4 基于黄金分割法优化求解 h

根据附件一中主索节点的坐标, 计算各节点在极坐标系下的 r_i 与 θ_i , 判断是否满足 $\theta_i \leq \frac{\pi}{6}$, 满足条件的节点即为需要移动以拟合理想抛物面的节点, 反之舍去。

在问题 1 中, $\alpha = 0^\circ, \beta = 90^\circ$, 以 $0m$ 为初值, $0.01m$ 为步长, $0.6m$ 为终值遍历 h , 求出相应的径向移动距离均方 D , 得到的关系图如图4。

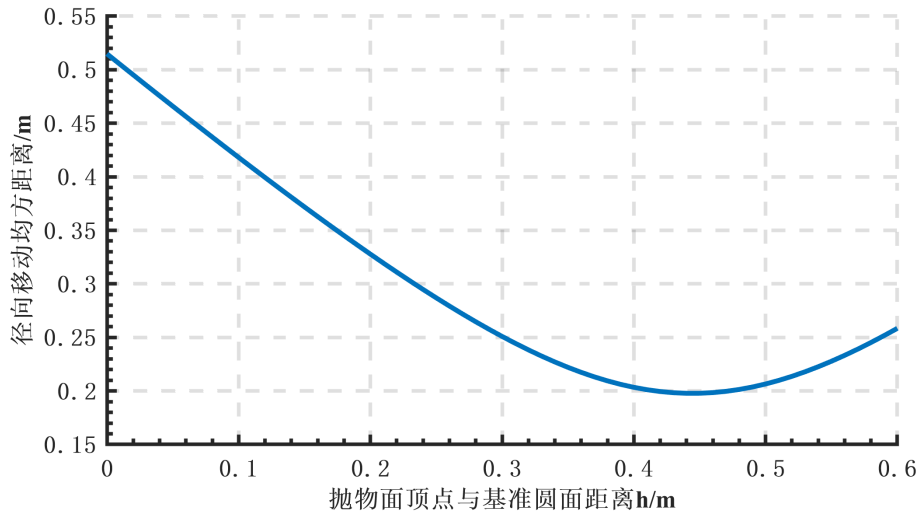


图 4 问题 1 径向移动均方距离随 h 的变化

根据曲线图, 初步判断 D 随 h 增长的关系 D 是先减后增, 则问题转换为求 D 在区间 $[0m, 0.6m]$ 的极小值点 h^* 。 D 关于 h 的函数为区间 $[0m, 0.6m]$ 上的下单峰函数, 可采用采用一维搜索方法的黄金分割法进行求解。求解步骤如下:

STEP1: 在初始区间 $[a, b]$ 长度的 0.618 处和 0.382 处分别插入两点 h_1, h_2 , 称作搜索点, 计算并比较 $D(h_1), D(h_2)$ 。

STEP2: 对于单下峰函数, 当 $D(h_1) < D(h_2)$ 时, 必有 $h^* \in [h_1, b]$; 当 $D(h_1) > D(h_2)$ 时, 则有 $h^* \in [a, h_2]$ 。据此更新区间。

STEP3: 更新区间, 当区间长度小于精度 δ 时 (取 0.0001), 选取此区间内的任一搜索点作为 h^* 的近似解。否则, 再次插入新的搜索点, 进行迭代运算。

根据算法, 编程最终求得 $h = 0.4444m$ 。据此, 即可确定理想抛物面。

5.2 问题二模型的建立和求解

5.2.1 问题二理想抛物面的确定

问题二中的被测物体的位置有了方向的变换，利用问题一建立的模型，调整方位角 $\alpha = 36.795^\circ$ ，仰角 $\beta = 78.169^\circ$ ，求得旋转对称轴的方向向量为

$$\vec{K} = (-\cos \beta \cos \alpha, -\cos \beta \sin \alpha, -\sin \beta) = (-0.1642, -0.12282, -0.9788) \quad (13)$$

利用问题 1 的算法，计算得到此时的偏移量 $h = 0.4380m$ ，据此确定此时的理想抛物面。则抛物面的顶点坐标为

$$(R + h)\vec{K} = (-49.3919, -36.9432, -294.4472) \quad (14)$$

5.2.2 问题二各主索点径向位移的求解

问题需要使工作平面尽可能地贴近理想抛物面，该拟合过程可看作参数优化过程，其优化参数为主索节点与抛物面之间的径向距离。将三个相互连接的主索节点构成的三角形反射单元看作平面，从抛物面的侧面看去，由图5所示。显然并非当主索节点落在

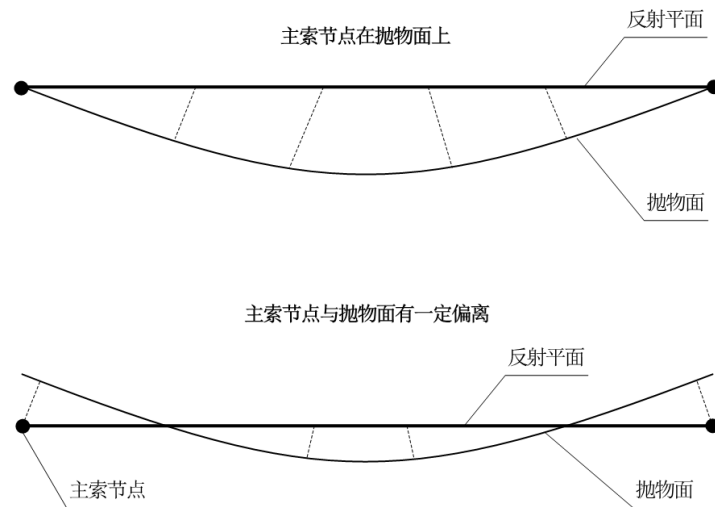


图 5 问题 2 最优主索节点位置的比较

抛物面上时，拟合精度最好，而是当又有一定偏离，使反射平面与抛物面的整体距离最小，达到最优拟合精度。

各主索点径向位移的求解步骤如下：

步骤一：确定初始位移

计算各主索节点到理想抛物面的径向距离 ΔD_0 ，将主索节点径向移动到抛物面上，即第一步的位移为 ΔD_0 。此时得到一个较好的初始解，但此时的解并不是最优；

步骤二：计算每个主索节点周围统计点到抛物面平均距离和误差

为得更优的位移量，首先确定各主索节点连接的反射平面，借助数值积分的思想，在每个主索节点周围的若干三角形平面单元上均匀地选取若干个统计点，如图6所示。计算周围各统计点与理想抛物面的平均径向距离 $\Delta D1$ (有正有负)，并计算各统计点与理想抛物面的径向距离的均方根值 RMS (即公式 (10))

$$RMS = \sqrt{\frac{\sum_{i=1}^n \Delta d_i^2}{n}} \quad (15)$$

对每个主索节点的 RMS 求和取平均再开平方，得到总的误差 RMS 。为了得到 $\Delta D1$ 和

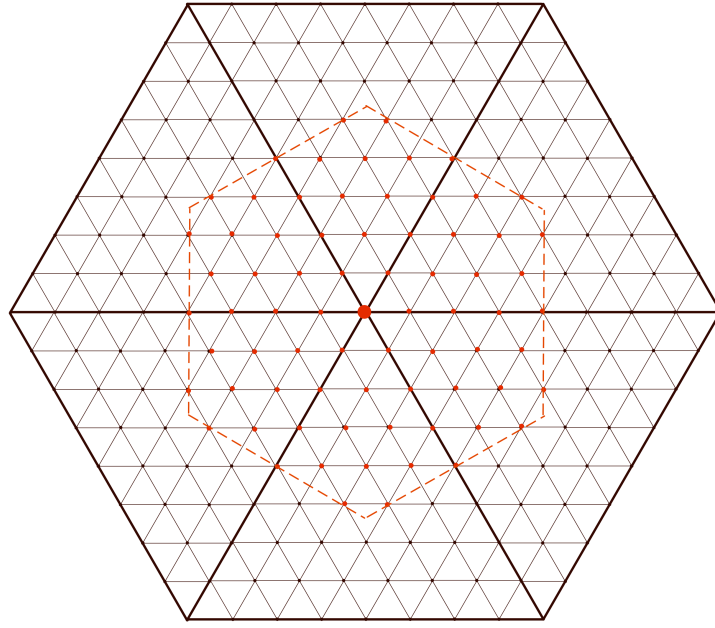


图 6 问题 2 周围统计点

RMS ，需要遍历该主索节点周围的三角形平面，得到所有统计点的 XYZ 坐标。以单个三角形平面为例，如图7所示。

在图7中，A 点为我们讨论的主索节点，A 与 B、C 两个节点构成三角形平面。将三角形的 3 条边 8 等分，对于图7中红色标注的统计点，其 XYZ 坐标的递推公式为

$$\begin{aligned} P(x, y, z) &= A(x, y, z) + i \cdot \overrightarrow{AD} + j \cdot \overrightarrow{DE} \\ i &= 1, 2, 3, 4; j = 0, \dots, i-1 \\ \overrightarrow{AD} &= \overrightarrow{AB}/8; \overrightarrow{AE} = \overrightarrow{AC}/8 \\ D(x, y, z) &= A(x, y, z) + \overrightarrow{AD} \\ E(x, y, z) &= A(x, y, z) + \overrightarrow{AE} \\ \overrightarrow{DE} &= E(x, y, z) - D(x, y, z) \end{aligned} \quad (16)$$

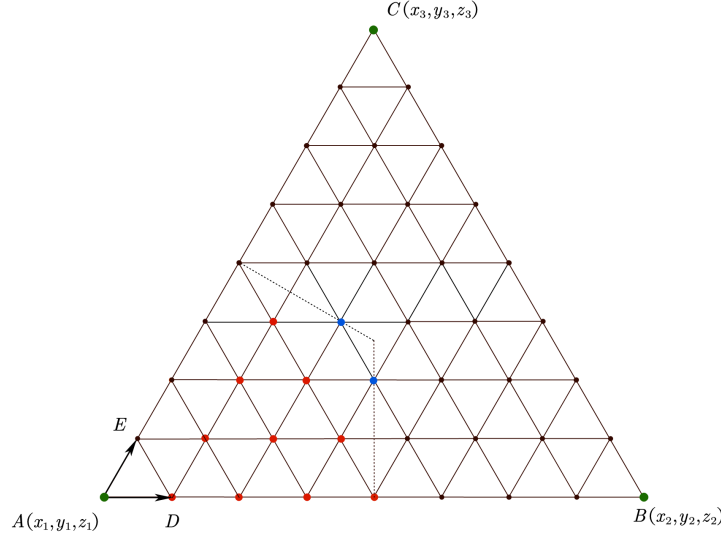


图 7 问题 2 单个三角形平面统计点

对于外围 2 个蓝色的点，其递推公式为：

$$P(x, y, z) = A(x, y, z) + 5 \cdot \overrightarrow{AD} + \begin{cases} 2\overrightarrow{DE} \\ 3\overrightarrow{DE} \end{cases} \quad (17)$$

步骤三：以步骤二中得到的平均距离反向移动各主索节点

在步骤二中得到了各主索节点周围统计点平均距离 $\Delta D1$ ，反向移动各主索节点，重新计算总和 RMS。重复步骤二，进行迭代计算，直到 RMS 值趋于稳定，本模型中的判断标准选取的是前后的 RMS 差值小于 0.001。

最终总和的径向位移为 $\Delta D = \Delta D0 + \Delta D1 + \dots$ ，移动后主索节点的 XYZ 坐标由原来的位置坐标径向移动 ΔD 后得到。

5.2.3 问题二径向位移和坐标求解结果

1. 主索节点位移与促进器伸缩量的讨论

现讨论主索节点伸缩量与对应促进器伸缩量的关系。节点、促进器顶端与基准球面球心三点并不在同一直线上，节点与促进器均沿径向移动，且连接两者的下拉索长度固定，如图8显示

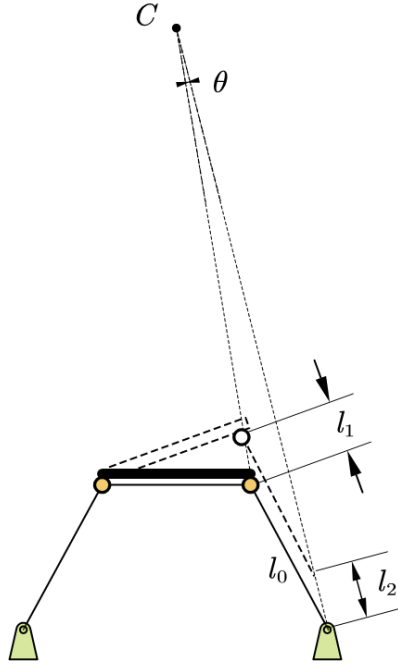


图 8 主索节点偏移

设节点坐标 (r_1, θ_1) ，促进器上端坐标 (r_2, θ_2) ，节点和促进器上端的径向移动距离分别是 l_1, l_2 ，则由余弦定理可以得到关系式：

$$\cos\theta = \frac{(r_1 - l_1)^2 + (r_2 - l_2)^2 - l_0^2}{2(r_1 - l_1)(r_2 - l_2)} \quad (18)$$

l_0 是拉索长度，为已知量，则得到了 l_1 与 l_2 的关系式，可由主索节点的偏移计算出对应促进器的伸缩量。

将附件 1 附件 2 中各点坐标值导入计算夹角余弦值 $\cos\theta$ ，得到的结果均接近于 1，精度达到了 0.0000001，则可以认为基准球面球心、主索节点和促进器顶端在同一直线上，在模型建立与计算过程中可视作 $l_1 = l_2$ ，即主索节点的偏移等于促进器的伸缩量。

2. 促进器伸缩量和调整后主索节点坐标求解结果

由上述迭代方法，计算位移量和位移后坐标，前 10 个主索节点的编号、移动后 XYZ 坐标和促进器伸缩量如表 1。误差 RSM 收敛过程如图 9，由图可知，迭代两次即收敛。

由于径向伸缩范围为 -0.6 到 0.6 米，因此检查求解结果的最大位移量为 0.4977m，满足要求，因此此时求得的结果为最优结果。移动前后主索节点的位置分布如图 10。

5.3 问题三模型的建立和求解

5.3.1 坐标系旋转

问题二中求得的抛物面开口指向被测天体，在原始坐标系下呈斜向状态，中轴与 z 轴之间较大的夹角，使得对应的抛物面方程较为复杂，计算困难，不便于光线反射的

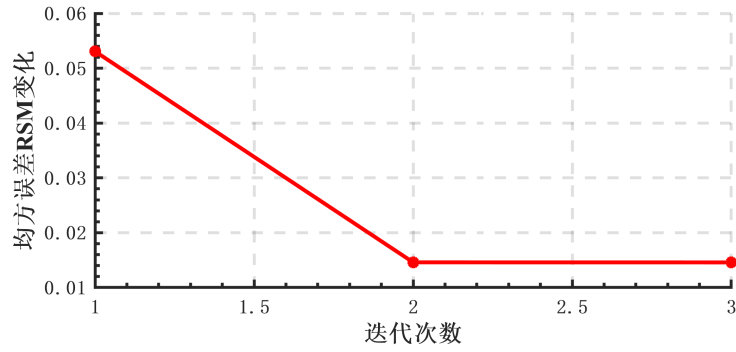


图 9 均方误差 **RSM** 随迭代次数的变化

表 1 前 10 个主索节点的编号、移动后 **XYZ** 坐标和促动器伸缩量

节点编号	X 坐标 (米)	Y 坐标 (米)	Z 坐标 (米)	伸缩量 (米)
A0	0	0	-300.5107	-0.1107
B1	6.108298	8.407685	-300.2447	-0.024492
C1	9.884553	-3.211602	-300.2765	-0.056327
D1	0	-10.39692	-300.3913	-0.17119
E1	-9.889597	-3.213241	-300.4297	-0.20966
A1	-6.11	8.410028	-300.3283	-0.10821
A3	0	16.81867	-299.9409	-0.012042
B2	12.20589	16.80055	-299.6195	0.061749
B3	15.99264	5.196494	-299.8997	0.029247
C2	19.75289	-6.41777	-299.6704	0.010753

求解，通过对坐标系的旋转操作，将旋转后的 z 轴与基准球面球心同被测天体的连线重合，此时的抛物面转换为了问题一中的状态，电磁波沿 z 轴负方向竖直射下，大大简化了计算。

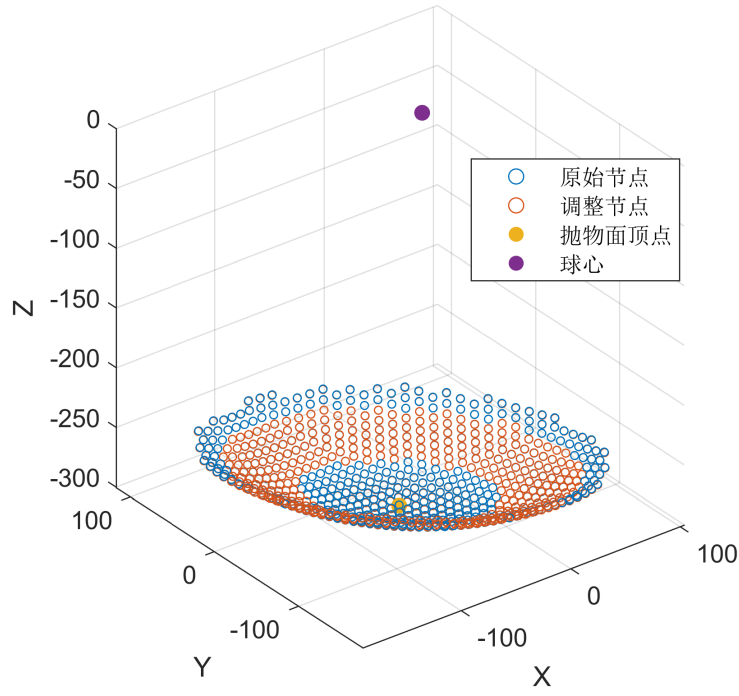


图 10 问题 2 原始节点和调整节点位置分布

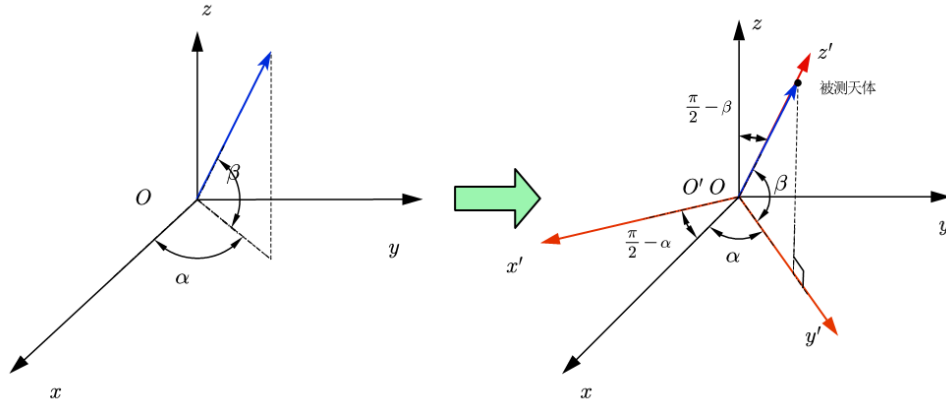


图 11 坐标系旋转示意

由图11显示，将坐标系先绕 z 轴顺时针旋转 $\frac{\pi}{2} - \alpha$ ，再绕 x 轴顺时针旋转 $\frac{\pi}{2} - \beta$ ，使 z 轴与目标直线重合，达到目标坐标系。

以 z 轴旋转为例，考虑到旋转过程中，点在 z 轴上的坐标不会发生变化，为简化问题，将坐标系投影在 xOy 平面上进行求解。如图12所示，其中 xOy 为原始坐标系， $x'O'y'$ 为旋转后的坐标系。

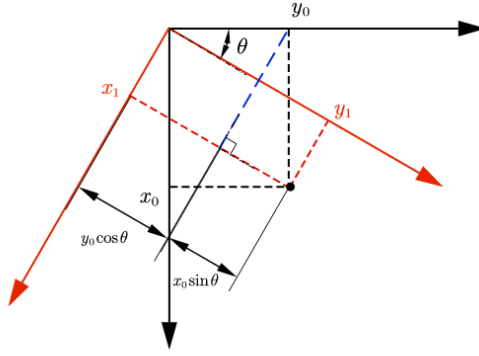


图 12 坐标系旋转平面示意

利用几何关系，可以求得旋转后的坐标变换公式 (即求坐标轴上的投影长度)：

$$\begin{cases} x_1 = x_0 \cos \theta - y_0 \sin \theta \\ y_1 = y_0 \cos \theta + x_0 \sin \theta \\ z_1 = z_0 \end{cases} \quad (19)$$

同理，针对绕 x 轴的旋转坐标变换公式如下：

$$\begin{cases} x_1 = x_0 \cos \theta + z_0 \sin \theta \\ y_1 = y_0 \\ z_1 = z_0 \cos \theta - x_0 \sin \theta \end{cases} \quad (20)$$

借助矩阵计算，可以更快的求解旋转坐标的变换，其中 z 轴旋转矩阵：

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

x 轴旋转矩阵：

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix} \quad (22)$$

结合被测天体的方位角和仰角，根据图11，可以确定矩阵形式的坐标变换公式：

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\frac{\pi}{2} - \beta) & -\sin(\frac{\pi}{2} - \beta) \\ 0 & \sin(\frac{\pi}{2} - \beta) & \cos(\frac{\pi}{2} - \beta) \end{bmatrix} \begin{bmatrix} \cos(\frac{\pi}{2} - \theta) & -\sin(\frac{\pi}{2} - \theta) & 0 \\ \sin(\frac{\pi}{2} - \theta) & \cos(\frac{\pi}{2} - \theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (23)$$

其中 $\begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ 表示原始坐标系下点的坐标， $\begin{bmatrix} x' & y' & z' \end{bmatrix}^T$ 表示旋转后坐标系中的坐标。

5.3.2 截面降维计算

问题二中求解的结果得到调整后各节点的位置坐标，由于拟合优化操作，这些节点近似分布在一个抛物面上，但仍不够精确，存在着偏差，对结果点集进行三维散点插值，得到新的更加密集的三维点，更新后的点集分布更有规律，形成的抛物面更加光滑。

将插值后的各点坐标左乘旋转矩阵，得到旋转后的坐标点，此时各点拟合形成的抛物面中轴接近 z 轴，开口向上，电磁波竖直向下射入。整体图形具有对称性，类比问题一，可将其投影在平面上简化求解，同时根据光的反射定理，入射光、反射光与法线在同一平面上，平面的截取不会对问题求解造成影响。此处选取 xOz 平面 yOz 两个平面进行截取，得到分布在 xOz 平面 yOz 平面上的若干散点，且散点分布近似抛物线。 xOz 截面上的抛物线散点及局部放大如图13. 图13中各散点的横轴间距是相等 (插值的时候

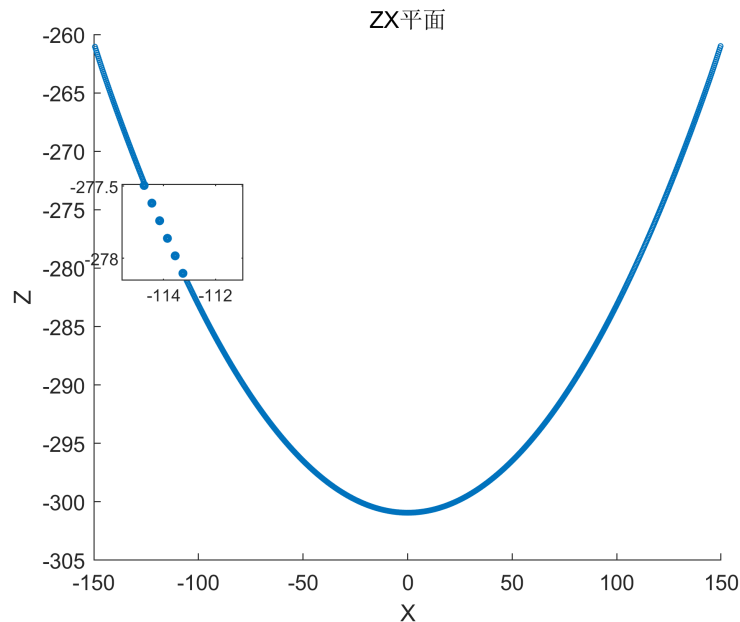


图 13 ZX 平面上的抛物线及局部放大

使用等间隔插值)，电磁波竖直入射到这些散点上，计算反射电磁波是否反射到了馈源舱，统计反射到馈源舱的电磁波比例。

因为各散点的横轴间距是相等，因此此时计算的比例即可代表 xOz 截面的接收比例。假设得到的比例为 λ_1 ，即为 xOz 截面的接收比例。按同样的步骤计算 yOz 截面的接收比例 λ_2 。由于电磁波在一个面上入射，因此总的接收比例 λ 为：

$$\lambda = \left(\frac{\lambda_1 + \lambda_2}{2} \right)^2 \quad (24)$$

5.3.3 截面接收比例的计算

以 xOz 截面为例，如图14所示，电磁波竖直入射在 A 点，A 点右邻点记为 B。入射

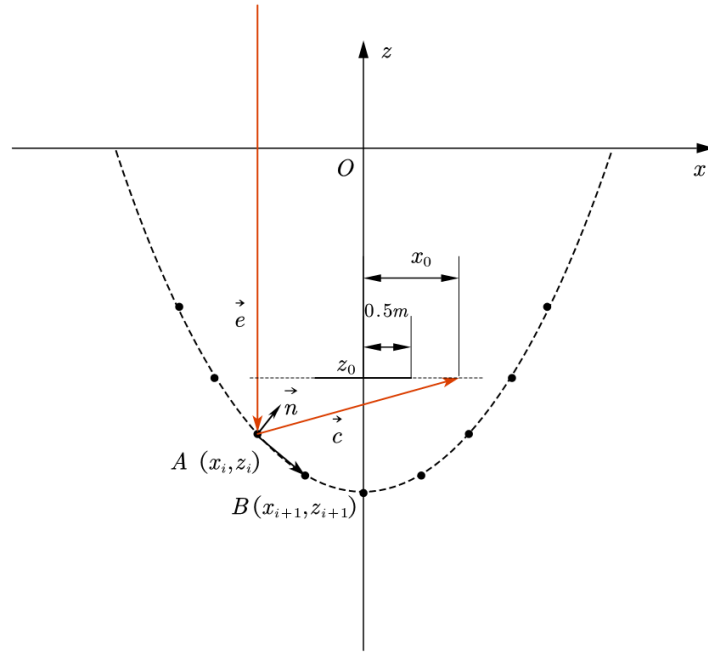


图 14 问题 3 离散抛物面反射原理

向量 \vec{e} 为 $(0, -1)$ ，法向量 \vec{n} 为：

$$\vec{n} = (-\Delta z, \Delta x) = (z_i - z_{i+1}, x_{i+1} - x_i) \quad (25)$$

则反射向量 \vec{c} 为：

$$\vec{c} = (c_x, c_z) \vec{e} - 2 \left| \vec{e} \cdot \frac{\vec{n}}{|\vec{n}|} \right| \frac{\vec{n}}{|\vec{n}|} = \quad (26)$$

反射电磁波的直线方程为：

$$z - z_i = \frac{c_z}{c_x} (x - x_i) \quad (27)$$

将馈源舱的 z 坐标 $z_0 = -(1 - 0.466)R$ 代入反射电磁波的直线方程，求得交点的 x 坐标 x_0 为：

$$x_0 = \frac{c_x}{c_z} (z_0 - z_i) + x_i \quad (28)$$

如果 $x_0 \leq 0.5m$ ，说明反射电磁波被馈源舱接收。遍历每一个散点，统计被接收的比例即可得到 xOz 截面上的接收比。同理可以求得 yOz 截面的接收比。按公式 (24) 计算得到整体的接收比。

5.3.4 抛物面接收比计算结果

按上述方法计算 xOz 截面和 yOz 截面的接收比，得到 xOz 截面的接收比 $\lambda_1 = 0.9209$ ， yOz 截面的接收比 $\lambda_2 = 0.9099$ ，按公式 (24) 计算得到整体的接收比 $\lambda = 0.8380$ 。所以馈源舱有效区域接收到的反射信号与 300 米口径内反射面的反射信号之比为 83.80%。

5.3.5 基准球面接收比的计算

当反射面是基准球面时，从 300m 口径边缘入射的电磁波一定无法被馈源舱接收，当入射电磁波沿旋转对称轴入射时，则一定能被馈源舱接收。

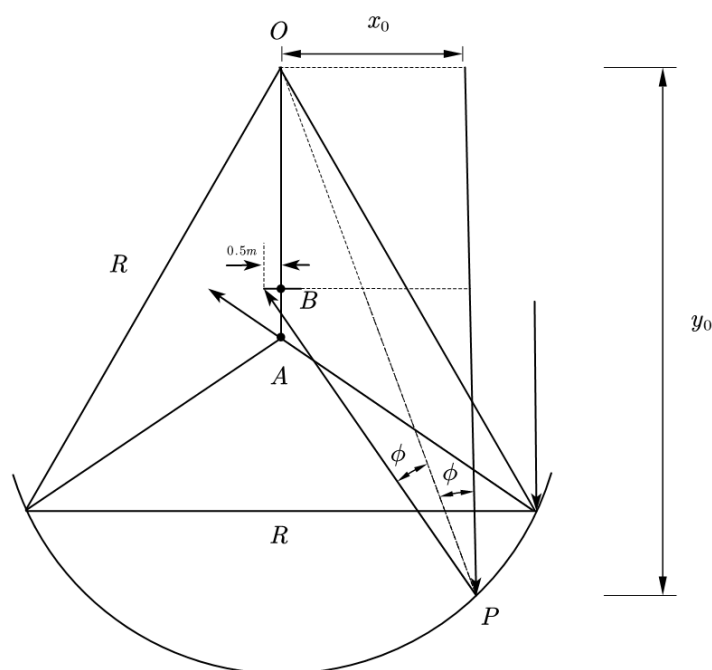


图 15 问题 3 球面反射示意图

因此存在一个临界距离，其入射的电磁波经过反射后正好通过馈源舱边缘（半径

0.5m), 设这个临界距离为 x_0 , 入射点纵坐标为 y_0 , 则 x_0 和 y_0 满足如下关系:

$$\begin{aligned} x_0^2 + y_0^2 &= R^2 \\ \tan \phi &= \frac{x_0}{y_0} \\ \tan(2\phi) &= \frac{2 \tan \phi}{1 - \tan^2 \phi} = \frac{0.5 + x_0}{y_0 - (1 - 0.466)R} \end{aligned} \quad (29)$$

解得 $x_0 = 107.8965m$, 则基准球面接收比为:

$$\left(\frac{x_0}{150} \right)^2 = 0.5174 \quad (30)$$

5.3.6 抛物面和基准球面接收比的比较

计算得到的抛物面的接收比为 83.80%, 基准球面接收比为 51.74%, 相比之下, 接收比提高了 61.96%。因此把球面调整为抛物面后接收比改善较为显著。

5.4 灵敏性分析

考虑到 h 值的改变会影响到问题 2 的收敛时误差和最大位移值, 以及问题 3 的的接收比, 因此对 h 作灵敏性分析, 其中最大位移值随 h 的改变比例的变化如图16, 收敛时误差随 h 的改变比例的变化如图

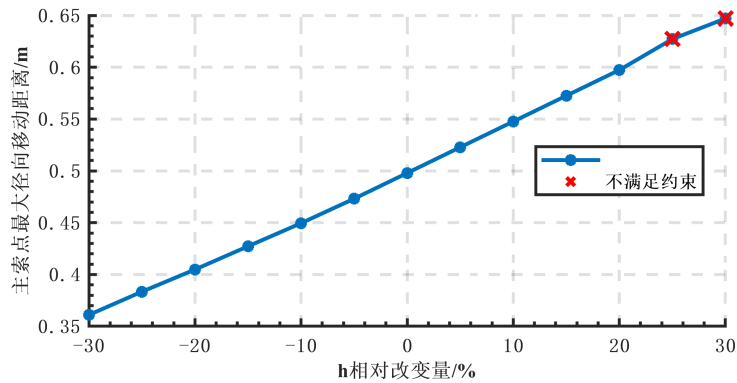


图 16 主索点最大径向移动距离随 h 改变量的变化

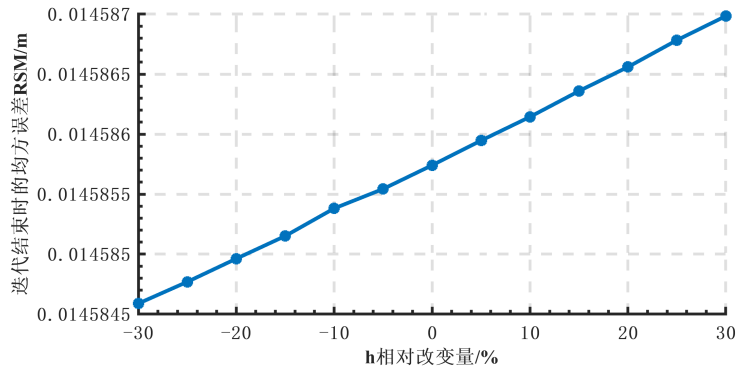


图 17 迭代结束时的均方误差 RSM 随 h 改变量的变化

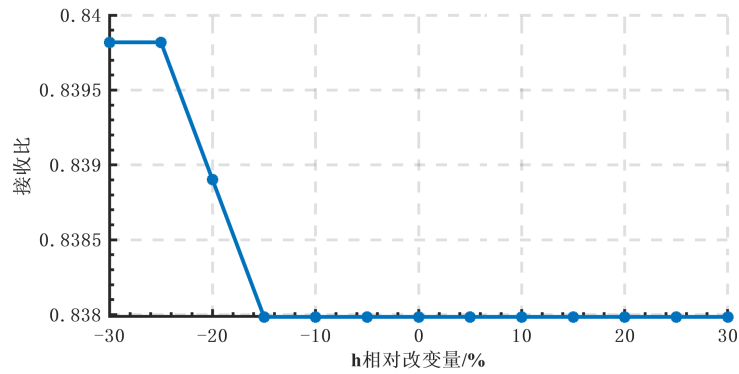


图 18 接收比随 h 改变量的变化

可以看出，迭代结束时的均方误差 RSM 和接收比基本不变， h 对主索点最大径向移动改变影响较大，这是显然的。

六、模型的评价

6.1 模型的优点

- 模型建立过程中，运用建立极坐标与旋转坐标系等操作，简化了计算形式，计算过程更加清晰。
- 问题二求解节点移动的模型中，对各节点单独进行分析求解，以达到全局最优，迭代次数少计算速度快，且不会由于随即性进入局部最优解。
- 问题三中对节点进行平滑插值，简化计算量的同时，兼顾了平面内部点的反射情况。

6.2 模型的缺点

- 模型将反射面近似为平面，与实际情况并不完全一致，存在误差。
- 问题二求解过程中没有考虑部分处于理想抛物面的反射面的情况，遗漏了少量拟合点的计算。
- 问题三中平面截取次数不够充分，接受比计算精度仍可提升。

参考文献

- [1] 狄杰建. 索网式可展开天线结构的反射面精度优化调整技术研究[D]. [出版地不详]: 西安电子科技大学, 2005.
- [2] 姜鹏, 王启明, 赵清. 巨型射电望远镜索网结构的优化分析与设计[J]. 工程力学, 2013, 30(02):400-405.
- [3] 商文念. FAST 反射面支承结构优化研究[D]. [出版地不详]: 哈尔滨工业大学, 2007.

- [4] 沈世钊, 范峰, 钱宏亮. FAST 主动反射面支承结构总体方案研究[J]. 建筑结构学报, 2010, 31(12):1-8.
- [5] 王志远. 基于迭代学习理论的 FAST 整网控制策略的研究[D]. [出版地不详]: 东北大学, 2015.
- [6] 徐双. 基于改进 PSO_BP 的 FAST 节点位移控制研究[D]. [出版地不详]: 东北大学, 2014.

附录 A 文件列表

表 2 Add caption

文件名	文件描述
Data1.mat	附件 1 数据
Data2.mat	附件 2 数据
Data3.mat	附件 3 数据
problem1.m	问题 1 求解 h
problem2_1.m	问题 1 求解 h
problem2_2.m	问题 2 求解其他要求的数据
problem3.m	问题 3 求解抛物面接收比
solvex0.m	问题 3 球面接收比求解
linminxin.m	灵敏性分析
huangjin.m	黄金分割法
result.xlsx	问题二结果表格

附录 B 代码

problem1.m

```
1 %求解问题1理想抛物面h
2 clear,clc
3 load Data1.mat
4 R=300.4;
5 F=0.466*R;
6 alpha=0;beta=pi/2;
7 M=cell2mat(Data1(:,2:4));
8 K=[-cos(beta)*cos(alpha),-cos(beta)*sin(alpha),-sin(beta)];
9 M1=sqrt(sum(M.*M,2));
10 K1=sqrt(sum(K.*K));
11 costheta=(M*K')./(K1*M1);
12 select_index=find(costheta>cos(pi/6));
13 select_point=M(select_index,:);
14 theta=acos(costheta(select_index));
15 select_point_d=M1(select_index);
16 fitness=[];
17 for h=0:0.01:0.6
18     P=2*(F+h);
```

```

19     r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(
20     sin(theta)).^2);
21     r(1)=R+h;
22     D=r-select_point_d;
23     fitness=[fitness,sqrt(mean(D.*D))];
24 end
25 h=0:0.01:0.6;
26 figure('Position',[466.6,395.4,559.9,294.4]);
27 plot(h,fitness);xlabel('抛物面顶点与基准圆面距离h/m');ylabel('径向移动均方距离/m')
28 beautiplot('small')
29 exportgraphics(gcf,'img\问题1径向移动均方距离随h的变化.png','Resolution',400)
30 best_h=h(find(fitness==min(fitness)))%best_h =0.4400
31 %%
32 getfitness_p1(0.39,theta,select_point_d)%测试
33 getfitness_p1(0.44,theta,select_point_d)%测试
34 %%
35 f=@(h) getfitness_p1(h,theta,select_point_d);
36 [result,x]=huangjin(f,[0,0.6],100)%result = 0.1978;x =0.4444
37 %%
38 h=0.4444;
39 P=2*(F+h);
40 r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(sin(
41 theta)).^2);
42 r(1)=R+h;
43 max(r.*sin(theta))
44 %%
45 function fit=getfitness_p1(h,theta,select_point_d)
46 R=300.4;
47 F=0.466*R;
48 P=2*(F+h);
49 r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(sin(
50 theta)).^2);
51 r(1)=R+h;
52 D=r-select_point_d;
53 fit=sqrt(mean(D.*D));
54 end

```

problem2_1.m

```

1 %求解问题2理想抛物面h
2 clear,clc
3 load Data1.mat
4 R=300.4;
5 F=0.466*R;
6 alpha=36.795/180*pi;beta=78.169/180*pi;
7 M=cell2mat(Data1(:,2:4));
8 K=[-cos(beta)*cos(alpha),-cos(beta)*sin(alpha),-sin(beta)];
9 M1=sqrt(sum(M.*M,2));

```

```

10 K1=sqrt(sum(K.*K));
11 costheta=(M*K')./(K1*M1);
12 select_index=find(costheta>cos(pi/6));
13 select_point=M(select_index,:);
14 source_point=select_point;
15 save source_point.mat source_point
16 theta=acos(costheta(select_index));
17 select_point_d=M1(select_index);
18 fitness=[];
19 for h=0:0.01:0.6
20     P=2*(F+h);
21     r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(sin(theta)).^2);
22     r(1)=R+h;
23     D=r-select_point_d;
24     fitness=[fitness,sqrt(mean(D.*D))];
25 end
26 h=0:0.01:0.6;
27 plot(h,fitness)
28 best_h=h(find(fitness==min(fitness)))%best_h =0.4400
29 %%
30
31 f=@(h) getfitness_p1(h,theta,select_point_d);
32 [result,best_h2]=huangjin(f,[0,0.6],100)%result = 0.2032;best_h2 =0.4380
33 %%
34 % getfitness_p1(h,theta,select_point_d)
35 h=0.4380;
36 R=300.4;
37 F=0.466*R;
38 P=2*(F+h);
39 r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(sin(theta)).^2);
40 r(1)=R+h;
41 D=r-select_point_d;
42 %fit=sqrt(mean(D.*D));
43 mean(D)
44
45 function fit=getfitness_p1(h,theta,select_point_d)
46 R=300.4;
47 F=0.466*R;
48 P=2*(F+h);
49 r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(sin(theta)).^2);
50 r(1)=R+h;
51 D=r-select_point_d;
52 fit=sqrt(mean(D.*D));
53 %fit=mean(D);

```

54 end

problem2_2.m

```
1 clear,clc
2 load Data1.mat
3 load Data3.mat
4 global K K1 R F h P
5 Data3=string(Data3);
6 R=300.4;
7 F=0.466*R;
8 alpha=36.795/180*pi;beta=78.169/180*pi;
9 M=cell2mat(Data1(:,2:4));
10 K=[-cos(beta)*cos(alpha),-cos(beta)*sin(alpha),-sin(beta)];
11 M1=sqrt(sum(M.*M,2));
12 K1=sqrt(sum(K.*K));
13 costheta=(M*K')./(K1*M1);
14 select_index=find(costheta>cos(pi/6));
15 select_point=M(select_index,:);
16 source_point=select_point;
17 all_point_bianhao=string(Data1(:,1));
18 select_point_bianhao=all_point_bianhao(select_index);
19 theta=acos(costheta(select_index));
20 select_point_d=M1(select_index);
21 h=0.4380;
22 P=2*(F+h);
23 r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(sin(theta)).^2);
24 peak_point_xyz=(R+h)*K;%抛物面顶点坐标
25 Delta_d0=r-select_point_d;%初始需要移动的距离Delta_d0
26 m_arrow=select_point./((sqrt(sum(select_point.*select_point,2))));%主索点单位方向向量
27 select_point=select_point+m_arrow.*Delta_d0;
28 lianjietu=zeros(length(all_point_bianhao));%所有主索点连接图矩阵
29 Data3_weizhi=zeros(size(Data3));
30 for ii=1:4300
31     for jj=1:3
32         Data3_weizhi(ii,jj)=find(all_point_bianhao==Data3(ii,jj));
33     end
34 end
35
36 select_weizhi=cell(1,length(select_index));
37 for ii=1:length(select_index)
38     [row_index,col_index]=find(Data3_weizhi==select_index(ii));
39     weizhi1=Data3_weizhi(row_index(col_index==1),:);
40     weizhi2=Data3_weizhi(row_index(col_index==2),:);
41     weizhi3=Data3_weizhi(row_index(col_index==3),:);
42     weizhi2=[weizhi2(:,2),weizhi2(:,1),weizhi2(:,3)];
43     weizhi3=[weizhi3(:,3),weizhi3(:,1),weizhi3(:,2)];
```



```

44 weizhi=[weizhi1;weizhi2;weizhi3];
45 willdelete=ones(1,size(weizhi,1));
46 for iii=1:size(weizhi,1)
47     for jjj=1:size(weizhi,2)
48         if ~isempty(find(select_index==weizhi(iii,jjj)))
49             weizhi(iii,jjj)=find(select_index==weizhi(iii,jjj));
50         else
51             willdelete(iii)=0;
52         end
53     end
54 end
55 weizhi=weizhi(find(willdelete==1),:);
56 select_weizhi(ii)={weizhi};
57 end
58 %%
59
60 RMS_array=[10];
61 Sum_Delta_d=-Delta_d0;
62 for k=1:100
63     All_RMS=[];
64     Delta_d=[];
65     for i=1:length(select_weizhi)
66         now_weizhi=cell2mat(select_weizhi(i));
67         all6_tonji_point_xyz=[];%每个主索点周围的统计点
68         for ii=1:size(now_weizhi,1)
69             each_weizhi=now_weizhi(ii,:);
70             xyz1=select_point(each_weizhi(1),:);
71             xyz2=select_point(each_weizhi(2),:);
72             xyz3=select_point(each_weizhi(3),:);
73             single_tonji_point_xyz=get_tonji_point(xyz1,xyz2,xyz3);%单个三角形内统
计点坐标
74             all6_tonji_point_xyz=[all6_tonji_point_xyz;single_tonji_point_xyz];
75         end
76         all6_2_tonji_point_xyz=[select_point(i,:);all6_tonji_point_xyz];%每个主索点
周围的统计点+主索点本身
77         [meanDr,singleRMS]=get_meanDr_singleRMS(all6_2_tonji_point_xyz);
78         Delta_d=[Delta_d,meanDr];
79         All_RMS=[All_RMS,singleRMS];
80     end
81     RMS_now=sqrt(mean(All_RMS.*All_RMS));
82     RMS_array=[RMS_array,RMS_now];
83     m_arrow=select_point./(sqrt(sum(select_point.*select_point,2)));%主索点单位方向
向量
84     select_point=select_point+m_arrow.*Delta_d';
85     Sum_Delta_d=Sum_Delta_d-Delta_d';
86     if RMS_array(end-1)-RMS_array(end)<0.001%收敛时为0.0146
87         RMS_array(end)

```

```

88         break
89     end
90 end
91
92 %% 数据矫正
93 load('Data2.mat')
94 A_point=[0,0,0];
95 B_point=select_point;
96 D_point=source_point;
97 E_point=cell2mat(Data2(:,5:7));
98 E_point=E_point(select_index,:);
99 cos_theta=(sum(E_point.*D_point,2))./(sqrt(sum(E_point.*E_point,2)).*sqrt(sum(
    D_point.*D_point,2)));
100 b_length=sqrt(sum(B_point.*B_point,2));
101 DE=E_point-D_point;
102 a_length=sqrt(sum(DE.*DE,2));
103 c_length=b_length.*cos_theta+sqrt((b_length.*cos_theta).^2-b_length.^2+a_length.^2)
    ;
104 AE_length=sqrt(sum(E_point.*E_point,2));
105 Delta_D=AE_length-c_length;
106 chazhi=Delta_D-Sum_Delta_d;
107 max_D=max(abs(Sum_Delta_d))%0.4977
108 %% 作图
109 figure('Position',[307.4,401,560,245.6])
110 plot(1:3,RMS_array(2:4),'r*-')
111 xlabel('迭代次数'),ylabel('均方误差RSM变化');beautiplot
112 exportgraphics(gcf,'img\均方误差RSM随迭代次数的变化.png','Resolution',400)
113 %%
114 figure;scatter3(source_point(:,1),source_point(:,2),source_point(:,3),10);hold on
115 scatter3(select_point(:,1),select_point(:,2),select_point(:,3),10)
116 xlabel('X'),ylabel('Y');zlabel('Z')
117 scatter3(peak_point_xyz(1),peak_point_xyz(2),peak_point_xyz(3),40,'filled')%抛物面
    顶点
118 scatter3([0],[0],[0],40,'filled')
119 legend('原始节点','调整节点','抛物面顶点','球心')
120 axis equal
121 %% 保存数据
122 %save select_point.mat select_point
123 % resulttable=[{'X坐标（米）','Y坐标（米）','Z坐标（米）'};num2cell(peak_point_xyz)];
124 % writecell(resulttable,'result.xlsx','Sheet','理想抛物面顶点坐标')
125 % result2=[select_point_bianhao,select_point];
126 % resulttable=[{'节点编号','X坐标（米）','Y坐标（米）','Z坐标（米）'};result2];
127 % writematrix(resulttable,'result.xlsx','Sheet','调整后主索节点编号及坐标')
128 % result3=[select_point_bianhao,Delta_D];
129 % resulttable=[{'对应主索节点编号','伸缩量（米）'};result3];
130 % writematrix(resulttable,'result.xlsx','Sheet','促动器顶端伸缩量')
131

```

```

132
133 %%
134 % for ii=1:4300
135 %     lianjietu(Data3_weizhi(ii,1),Data3_weizhi(ii,2))=1;
136 %     lianjietu(Data3_weizhi(ii,1),Data3_weizhi(ii,3))=1;
137 %     lianjietu(Data3_weizhi(ii,2),Data3_weizhi(ii,1))=1;
138 %     lianjietu(Data3_weizhi(ii,3),Data3_weizhi(ii,1))=1;
139 % end
140 % select_lianjietu=lianjietu(select_index,select_index);
141 % for i=1:length(all_point_bianhao)
142 %
143 % end

```

problem3.m

```

1 % load select_point.mat
2 % old_point=select_point;
3 load source_point.mat
4 old_point=source_point;
5 alpha=36.795/180*pi;beta=78.169/180*pi;
6 A=[cos(pi/2-alpha),-sin(pi/2-alpha),0;
7     sin(pi/2-alpha),cos(pi/2-alpha),0;
8     0,0,1];
9 B=[1,0,0;
10    0,cos(pi/2-beta),-sin(pi/2-beta);
11    0,sin(pi/2-beta),cos(pi/2-beta)];
12 C=B*A;%旋转矩阵
13 new_point=C*old_point';
14 new_point=new_point';
15 x=new_point(:,1);y=new_point(:,2);z=new_point(:,3);
16 figure;scatter3(old_point(:,1),old_point(:,2),old_point(:,3))
17 figure;scatter3(x,y,z)
18 %%
19 [X,Y,Z]=griddata(x,y,z,linspace(min(x),max(x),1000)',linspace(min(y),max(y),1000)',
20     'v4');%插值
21 %figure;meshc(X,Y,Z)
22 size2cutslice = size(Z,1);
23 cutZ = zeros(1,size2cutslice);
24 cutX = zeros(1,size2cutslice);
25 cutY = zeros(1,size2cutslice);
26 row = floor(size2cutslice/2);
27 for i=1:size2cutslice
28     cutZ(i) = Z(row,i);
29     cutX(i) = X(row,i);
30     cutY(i) = Y(row,i);
31 end
32 figure;scatter(cutX,cutZ,5),xlabel('X'),ylabel('Z');title('ZX平面');magnify
33 % xlim(-150,150),ylim(-250,-310);axis equal

```

```

33
34 R=300.4;R1=-(1-0.466)*R;
35 DX=cutX(2:end)-cutX(1:end-1);
36 DZ=cutZ(2:end)-cutZ(1:end-1);
37 n_arrow=[-DZ;DX];%法向量
38 N_arrow=n_arrow./ (sqrt(sum(n_arrow.*n_arrow,1)));%单位法向量
39 Rushe_arrow=[zeros(1,size(N_arrow,2));-ones(1,size(N_arrow,2))];%入射向量
40 Chushe_arrow=Rushe_arrow-2*sum(Rushe_arrow.*N_arrow).*N_arrow;%出射向量
41 K_=Chushe_arrow(1,:)./Chushe_arrow(2,:);%斜率倒数
42 jiao_X=K_.*(R1-cutZ(1:end-1))+cutX(1:end-1);%交点X坐标
43 bili1=sum(abs(jiao_X)<=0.5)/length(jiao_X)%在区域内比例 :0.9209
44
45 size2cutslice = size(Z,1);
46 cutZ = zeros(1,size2cutslice);
47 cutX = zeros(1,size2cutslice);
48 cutY = zeros(1,size2cutslice);
49 row = floor(size2cutslice/2);
50 for i=1:size2cutslice
51     cutZ(i) = Z(i,row);
52     cutX(i) = X(i,row);
53     cutY(i) = Y(i,row);
54 end
55 figure;scatter(cutY,cutZ,5),xlabel('Y'),ylabel('Z');title('ZY平面');
56 % xlim(-150,150),ylim(-250,-310);axis equal
57 %%
58 R=300.4;R1=-(1-0.466)*R;
59 DX=cutY(2:end)-cutY(1:end-1);
60 DZ=cutZ(2:end)-cutZ(1:end-1);
61 n_arrow=[-DZ;DX];%法向量
62 N_arrow=n_arrow./ (sqrt(sum(n_arrow.*n_arrow,1)));%单位法向量
63 Rushe_arrow=[zeros(1,size(N_arrow,2));-ones(1,size(N_arrow,2))];%入射向量
64 Chushe_arrow=Rushe_arrow-2*sum(Rushe_arrow.*N_arrow).*N_arrow;%出射向量
65 K_=Chushe_arrow(1,:)./Chushe_arrow(2,:);%斜率倒数
66 jiao_X=K_.*(R1-cutZ(1:end-1))+cutY(1:end-1);%交点X坐标
67 bili2=sum(abs(jiao_X)<=0.5)/length(jiao_X)%在区域内比例 0.9099
68 bili=((bili1+bili2)/2)^2%0.8380

```

solvex0.m

```

1 syms x0
2 R=300.4;
3 f=@(x0) (0.5+x0)/((sqrt(R^2-x0^2))-(1-0.466)*R)-(2*x0/(sqrt(R^2-x0^2)))/(1-x0^2/(
    sqrt(R^2-x0^2))^2)
4 fsolve(f,150)

```

problem3.m

```

1 % load select_point.mat
2 % old_point=select_point;

```

```

3 load source_point.mat
4 old_point=source_point;
5 alpha=36.795/180*pi;beta=78.169/180*pi;
6 A=[cos(pi/2-alpha),-sin(pi/2-alpha),0;
7     sin(pi/2-alpha),cos(pi/2-alpha),0;
8     0,0,1];
9 B=[1,0,0;
10    0,cos(pi/2-beta),-sin(pi/2-beta);
11    0,sin(pi/2-beta),cos(pi/2-beta)];
12 C=B*A;%旋转矩阵
13 new_point=C*old_point';
14 new_point=new_point';
15 x=new_point(:,1);y=new_point(:,2);z=new_point(:,3);
16 figure;scatter3(old_point(:,1),old_point(:,2),old_point(:,3))
17 figure;scatter3(x,y,z)
18 %%
19 [X,Y,Z]=griddata(x,y,z,linspace(min(x),max(x),1000)',linspace(min(y),max(y),1000)',
20     'v4');%插值
21 %figure;meshc(X,Y,Z)
22 size2cutslice = size(Z,1);
23 cutZ = zeros(1,size2cutslice);
24 cutX = zeros(1,size2cutslice);
25 cutY = zeros(1,size2cutslice);
26 row = floor(size2cutslice/2);
27 for i=1:size2cutslice
28     cutZ(i) = Z(row,i);
29     cutX(i) = X(row,i);
30     cutY(i) = Y(row,i);
31 end
32 figure;scatter(cutX,cutZ,5),xlabel('X'),ylabel('Z');title('ZX平面');magnify
33 % xlim(-150,150),ylim(-250,-310);axis equal
34 R=300.4;R1=-(1-0.466)*R;
35 DX=cutX(2:end)-cutX(1:end-1);
36 DZ=cutZ(2:end)-cutZ(1:end-1);
37 n_arrow=[-DZ;DX];%法向量
38 N_arrow=n_arrow./sqrt(sum(n_arrow.*n_arrow,1));%单位法向量
39 Rushe_arrow=[zeros(1,size(N_arrow,2));-ones(1,size(N_arrow,2))];%入射向量
40 Chushe_arrow=Rushe_arrow-2*sum(Rushe_arrow.*N_arrow).*N_arrow;%出射向量
41 K_=Chushe_arrow(1,:)./Chushe_arrow(2,:);%斜率倒数
42 jiao_X=K_.*(R1-cutZ(1:end-1))+cutX(1:end-1);%交点X坐标
43 bili1=sum(abs(jiao_X)<=0.5)/length(jiao_X)%在区域内比例 :0.9209
44
45 size2cutslice = size(Z,1);
46 cutZ = zeros(1,size2cutslice);
47 cutX = zeros(1,size2cutslice);
48 cutY = zeros(1,size2cutslice);

```

```

49 row = floor(size2cutslice/2);
50 for i=1:size2cutslice
51     cutZ(i) = Z(i,row);
52     cutX(i) = X(i,row);
53     cutY(i) = Y(i,row);
54 end
55 figure;scatter(cutY,cutZ,5),xlabel('Y'),ylabel('Z');title('ZY平面');
56 % xlim(-150,150),ylim(-250,-310);axis equal
57 %%
58 R=300.4;R1=-(1-0.466)*R;
59 DX=cutY(2:end)-cutY(1:end-1);
60 DZ=cutZ(2:end)-cutZ(1:end-1);
61 n_arrow=[-DZ;DX];%法向量
62 N_arrow=n_arrow./sqrt(sum(n_arrow.*n_arrow,1));%单位法向量
63 Rushe_arrow=[zeros(1,size(N_arrow,2))-ones(1,size(N_arrow,2))];%入射向量
64 Chushe_arrow=Rushe_arrow-2*sum(Rushe_arrow.*N_arrow).*N_arrow;%出射向量
65 K_=Chushe_arrow(1,:)/Chushe_arrow(2,:);%斜率倒数
66 jiao_X=K_.*(R1-cutZ(1:end-1))+cutY(1:end-1);%交点X坐标
67 bili2=sum(abs(jiao_X)<=0.5)/length(jiao_X)%在区域内比例 0.9099
68 bili=((bili1+bili2)/2)^2%0.8380

```

linminxin.m

```

1 %灵敏性分析
2 clear
3 h0=0.4380;
4 Last_RMS=[];
5 maxD_array=[];
6 bili_array=[];
7 kkk=-30:5:30;
8 for h=h0+kkk/100*h0
9     load Data1.mat
10    load Data3.mat
11    Data3=string(Data3);
12    R=300.4;
13    F=0.466*R;
14    alpha=36.795/180*pi;beta=78.169/180*pi;
15    M=cell2mat(Data1(:,2:4));
16    K=[-cos(beta)*cos(alpha),-cos(beta)*sin(alpha),-sin(beta)];
17    M1=sqrt(sum(M.*M,2));
18    K1=sqrt(sum(K.*K));
19    costheta=(M*K')./(K1*M1);
20    select_index=find(costheta>cos(pi/6));
21    select_point=M(select_index,:);
22    source_point=select_point;
23    all_point_bianhao=string(Data1(:,1));
24    select_point_bianhao=all_point_bianhao(select_index);
25    theta=acos(costheta(select_index));

```

```

26 select_point_d=M1(select_index);
27
28 % h=h0+kkk/100*h0;%h改变量
29 P=2*(F+h);
30 r=(-2*P*cos(theta)+sqrt(4*(P*cos(theta)).^2+8*(sin(theta)).^2*P*(R+h)))/(2*(
sin(theta)).^2);
31 peak_point_xyz=(R+h)*K;%抛物面顶点坐标
32 Delta_d0=r-select_point_d;%初始需要移动的距离Delta_d0
33 m_arrow=select_point./((sqrt(sum(select_point.*select_point,2))));%主索点单位方向
向量
34 select_point=select_point+m_arrow.*Delta_d0;
35 lianjietu=zeros(length(all_point_bianhao));%所有主索点连接图矩阵
36 Data3_weizhi=zeros(size(Data3));
37 for ii=1:4300
38     for jj=1:3
39         Data3_weizhi(ii,jj)=find(all_point_bianhao==Data3(ii,jj));
40     end
41 end
42
43 select_weizhi=cell(1,length(select_index));
44 for ii=1:length(select_index)
45     [row_index,col_index]=find(Data3_weizhi==select_index(ii));
46     weizhi1=Data3_weizhi(row_index(col_index==1),:);
47     weizhi2=Data3_weizhi(row_index(col_index==2),:);
48     weizhi3=Data3_weizhi(row_index(col_index==3),:);
49     weizhi2=[weizhi2(:,2),weizhi2(:,1),weizhi2(:,3)];
50     weizhi3=[weizhi3(:,3),weizhi3(:,1),weizhi3(:,2)];
51     weizhi=[weizhi1;weizhi2;weizhi3];
52     willdelete=ones(1,size(weizhi,1));
53     for iii=1:size(weizhi,1)
54         for jjj=1:size(weizhi,2)
55             if ~isempty(find(select_index==weizhi(iii,jjj)))
56                 weizhi(iii,jjj)=find(select_index==weizhi(iii,jjj));
57             else
58                 willdelete(iii)=0;
59             end
60         end
61     end
62     weizhi=weizhi(find(willdelete==1),:);
63     select_weizhi(ii)={weizhi};
64 end
65
66 clc
67 RMS_array=[10];
68 Sum_Delta_d=-Delta_d0;
69 for k=1:100
70     All_RMS=[];

```

```

71     Delta_d=[];
72     for i=1:length(select_weizhi)
73         now_weizhi=cell2mat(select_weizhi(i));
74         all6_tonji_point_xyz=[];%每个主索点周围的统计点
75         for ii=1:size(now_weizhi,1)
76             each_weizhi=now_weizhi(ii,:);
77             xyz1=select_point(each_weizhi(1),:);
78             xyz2=select_point(each_weizhi(2),:);
79             xyz3=select_point(each_weizhi(3),:);
80             single_tonji_point_xyz=get_tonji_point(xyz1,xyz2,xyz3);%单个三角形
内统计点坐标
81             all6_tonji_point_xyz=[all6_tonji_point_xyz;single_tonji_point_xyz];
82         end
83         all6_2_tonji_point_xyz=[select_point(i,:);all6_tonji_point_xyz];%每个主
索点周围的统计点+主索点本身
84         [meanDr,singleRMS]=get_meanDr_singleRMS(all6_2_tonji_point_xyz);
85         Delta_d=[Delta_d,meanDr];
86         All_RMS=[All_RMS,singleRMS];
87     end
88     RMS_now=sqrt(mean(All_RMS.*All_RMS));
89     RMS_array=[RMS_array,RMS_now];
90     m_arrow=select_point./(sqrt(sum(select_point.*select_point,2)));%主索点单位
方向向量
91     select_point=select_point+m_arrow.*Delta_d';
92     Sum_Delta_d=Sum_Delta_d-Delta_d';
93     max_D=max(abs(Sum_Delta_d));
94     if RMS_array(end-1)-RMS_array(end)<0.001
95
96         break
97     end
98 end
99 Last_RMS=[Last_RMS,RMS_now];
100 maxD_array=[maxD_array,max_D];
101 old_point=select_point;
102 A=[cos(pi/2-alpha),-sin(pi/2-alpha),0;
103     sin(pi/2-alpha),cos(pi/2-alpha),0;
104     0,0,1];
105 B=[1,0,0;
106     0,cos(pi/2-beta),-sin(pi/2-beta);
107     0,sin(pi/2-beta),cos(pi/2-beta)];
108 C=B*A;%旋转矩阵
109 new_point=C*old_point';
110 new_point=new_point';
111 x=new_point(:,1);y=new_point(:,2);z=new_point(:,3);
112 [X,Y,Z]=griddata(x,y,z,linspace(min(x),max(x),1000)',linspace(min(y),max(y)
,1000),'v4');%插值
113

```



```

114 size2cutslice = size(Z,1);
115 cutZ = zeros(1,size2cutslice);
116 cutX = zeros(1,size2cutslice);
117 cutY = zeros(1,size2cutslice);
118 row = floor(size2cutslice/2);
119 for i=1:size2cutslice
120     cutZ(i) = Z(row,i);
121     cutX(i) = X(row,i);
122     cutY(i) = Y(row,i);
123 end
124
125 R=300.4;R1=-(1-0.466)*R;
126 DX=cutX(2:end)-cutX(1:end-1);
127 DZ=cutZ(2:end)-cutZ(1:end-1);
128 n_array=[-DZ;DX];%法向量
129 N_array=n_array./sqrt(sum(n_array.*n_array,1));%单位法向量
130 Rushe_array=[zeros(1,size(N_array,2));-ones(1,size(N_array,2))];%入射向量
131 Chushe_array=Rushe_array-2*sum(Rushe_array.*N_array).*N_array;%出射向量
132 K_=Chushe_array(1,:)./Chushe_array(2,:);%斜率倒数
133 jiao_X=K_.*(R1-cutZ(1:end-1))+cutX(1:end-1);%交点X坐标
134 bili1=sum(abs(jiao_X)<=0.5)/length(jiao_X);%在区域内比例
135
136 size2cutslice = size(Z,1);
137 cutZ = zeros(1,size2cutslice);
138 cutX = zeros(1,size2cutslice);
139 cutY = zeros(1,size2cutslice);
140 row = floor(size2cutslice/2);
141 for i=1:size2cutslice
142     cutZ(i) = Z(i,row);
143     cutX(i) = X(i,row);
144     cutY(i) = Y(i,row);
145 end
146
147 R=300.4;R1=-(1-0.466)*R;
148 DX=cutY(2:end)-cutY(1:end-1);
149 DZ=cutZ(2:end)-cutZ(1:end-1);
150 n_array=[-DZ;DX];%法向量
151 N_array=n_array./sqrt(sum(n_array.*n_array,1));%单位法向量
152 Rushe_array=[zeros(1,size(N_array,2));-ones(1,size(N_array,2))];%入射向量
153 Chushe_array=Rushe_array-2*sum(Rushe_array.*N_array).*N_array;%出射向量
154 K_=Chushe_array(1,:)./Chushe_array(2,:);%斜率倒数
155 jiao_X=K_.*(R1-cutZ(1:end-1))+cutY(1:end-1);%交点X坐标
156 bili2=sum(abs(jiao_X)<=0.5)/length(jiao_X);%在区域内比例
157 bili=((bili1+bili2)/2)^2;
158 bili_array=[bili_array,bili]
159 end
160 %%

```

```

161 close all
162 kkk=-30:5:30;
163 dh=kkk;
164 figure('Position',[526.6,365,559.9,268.8]),plot(dh,maxD_array,'*-');hold on
165 bad_index=find(maxD_array>0.6);
166 scatter(dh(bad_index),maxD_array(bad_index),100,'rx');
167 legend(' ','不满足约束')
168 xlabel('h相对改变量/%');ylabel('主索点最大径向移动距离/m');beautiplot('small')
169 exportgraphics(gcf,'img\主索点最大径向移动距离随h改变量的变化.png','Resolution'
    ,400)
170 figure('Position',[526.6,365,559.9,268.8]),plot(dh,Last_RMS,'*-');
171 xlabel('h相对改变量/%');ylabel('迭代结束时的均方误差RSM/m');beautiplot('small')
172 exportgraphics(gcf,'img\迭代结束时的均方误差RSM随h改变量的变化.png','Resolution'
    ,400)
173 figure('Position',[526.6,365,559.9,268.8]),plot(dh,bili_array,'*-');
174 xlabel('h相对改变量/%');ylabel('接收比');beautiplot('small')
175 exportgraphics(gcf,'img\接收比随h改变量的变化.png','Resolution',400)

```

huangjin.m

```

1 function [result,x]=huangjin(f,x3,n)
2 tol=1e-6;%精度
3 x1=x3(1);
4 x2=x3(2);
5 i=1;
6 while i < n
7     %取中间值
8     a=x1+0.382*(x2-x1);
9     b=x1+0.618*(x2-x1);
10    fa=f(a);
11    fb=f(b);
12    % 判断fa fb大小，缩小区间
13    if fa < fb
14        x2=b;
15    else
16        x1=a;
17    end
18    if abs(x1-x2) < tol
19        result=f((x1+x2)/2);
20        x=(x1+x2)/2;
21        break;
22    end
23    i=i+1;
24 end
25 end

```