

定日镜场的优化设计模型

摘 要

定日镜场能将太阳光反射汇聚到安装在镜场中的吸收塔集热器上,实现能量转换,不同的定日镜场参数会影响定日镜场的效率。本文通过光线反射、入射等模型,计算给定参数下的定日镜场效率,并建立定日镜场的效率优化模型,研究定日镜场参数变化对单位镜面面积平均输出热功率的影响。

针对问题一,本文建立了定日镜场的效率计算模型。首先,建立镜场大地坐标系、镜面坐标系和光锥束坐标系,描述定日镜场的参数。太阳光束的入射光线和反射光线考虑为光锥束,计算太阳光的入射和反射,考虑阴影遮挡损失、余弦损失、集热器截断效率等因素。通过坐标变换和方程联立,判断某入射光线及对应的反射光线是否造成阴影遮挡损失所涉及的塔身阴影损失、入射镜面遮挡损失、反射镜面遮挡损失,再判断成功反射出的光线是否被集热器接收。最后得到光学效率和输出热功率的表达式,进一步计算年平均光学效率、年平均热功率、单位镜面面积年平均输出热功率等。在某个定日镜上取点步长 τ 为1m,在光锥束上取 $\theta_1 = 0.002rad$, $\theta_2 = 0, \pi/2, \pi, 3\pi/2$,进行网格化构建入射光线束和反射光线束,计算得到表1和表2。该定日镜场的年平均输出光学效率0.536230167、年平均输出热功率32.76117051MW、单位镜面面积年平均输出热功率0.521508604kW/m²,最后,进行敏感性分析,分析步长 τ 对结果的影响。

针对问题二,本文提出了单位镜面面积年平均输出热功率的优化模型。问题二是在定日镜场的额定年平均输出热功率为60MW的条件下,设计定日镜场的参数,使得单位镜面面积年平均输出热功率尽量大。决策变量包括吸收塔的位置坐标、定日镜的尺寸(相同)、安装高度、数量和位置。目标函数是单位镜面面积年平均输出热功率的最大值,约束条件包括镜面边长在2m至8m之间、安装高度在2m至6m之间、相邻定日镜底座中心距离比镜面宽度多5m等。先取 w, v 相等,用蜂窝排列法确定圆环内能排列的定日镜数目和位置。然后将蜂窝进行绕原点旋转 μ ,用蒙特卡洛模拟法对部分定日镜进行随机抽样,得到粗决策变量,从而进一步计算光学效率和单位面积输出热功率等。最后遍历 $\mu, w = v, \tilde{h}, X_0, Y_0$ 寻找决策变量的最优解。得到该定日镜场的年平均输出光学效率0.591643667、年平均输出热功率68.24427914MW、单位镜面面积年平均输出热功率0.572538333kW/m²。

针对问题三,本文提出了单位镜面面积年平均输出热功率的优化模型。问题三是在定日镜尺寸和安装高度可以不同的情况下,设计定日镜场的参数,使得单位镜面面积年平均输出热功率尽量大。以单位镜面面积年平均输出热功率为目标函数,以吸收塔位置坐标、定日镜尺寸、安装高度、定日镜数目、定日镜位置为决策变量,以额定功率、相邻定日镜距离、圆形区域半径等为约束条件,建立优化模型。考虑通过第二问求得结果进一步优化,将最外围的定日镜安装高度升高为6m,其他不变,得到该定日镜场的年平均输出光学效率0.496428083、年平均输出热功率60.336111MW、单位镜面面积年平均输出热功率0.506192417kW/m²。

关键词: 坐标旋转 光学效率 输出热功率 锥形光束 蜂窝排列法



一、问题分析

1.1 问题一的分析

问题一是在定日镜场参数确定时，计算定日镜场的效率。同时，题中简化计算“年均”指标只需计算每月 21 日 5 个时间点等。为了得到最大的效率，我们认为经过定日镜中心反射后的反射光线通过集热器中心，那么其他入射光线也能大部分被集热器接受。因此，太阳位置确定后，经过定日镜中心的入射光线和反射光线确定，可根据向量法则计算定日镜的法向量，从而确定每个定日镜场的俯仰角和方位角。阴影遮挡损伤考虑塔身的阴影损伤、后排镜面接受的太阳光线被前镜阻挡的阴影损伤和后排镜面反射的太阳光线被前镜阻挡的遮挡损伤。余弦损伤考虑为入射光线和镜面法线不平行的能量损伤。最后，建立模型表示集热器的接收能量，计算定日镜场的相应量。

1.2 问题二的分析

优化模型下，吸收塔的位置坐标、定日镜尺寸、安装高度、定日镜数量、定日镜位置为决策变量，单位面积年平均输出热功率尽量大为目标函数，每个定日镜尺寸和安装高度相同，在满足约束条件的情况下，使得目标函数尽量大。进行逐步寻优，先确定定日镜尺寸和安装高度，之后逐步寻找其他决策变量的优解，使得目标函数达到局部最优解即可。

1.3 问题三的分析

每个定日镜尺寸和安装高度不同，确定定日镜场的其他参数，在满足约束条件的情况下，使得单位面积年平均输出热功率尽量大。根据问题二的寻优过程，适当调整尺寸和高度。

二、模型假设

为了对模型进行合理简化，我们建立了以下的模型假设：

- 1、假设太阳光直接照射到集热器上，不计入我们模型的计算。
- 2、假设相对于 3000m 的海拔，定日镜作为垂直于太阳光线的平面，计算 DNI 时的海拔可以忽略定日镜的高度。
- 3、假设吸收塔的直径和集热器的直径相同，为 7m。
- 4、假设安装定日镜场时不考虑人工成本、土地利用率等。

三、符号说明

符号表示	符号说明
α_s, γ_s	太阳高度角、太阳方位角
$\tilde{\eta}, \bar{\eta}$	年平均光学效率，平均光学效率
\bar{P}_s, \bar{P}	年输出平均热功率，单位镜面面积年平均输出热功率
T_i, T_z	镜面坐标系、光锥坐标系转到大地坐标系的旋转矩阵
$\alpha, \beta, \alpha_z, \beta_z$	定日镜的方位角、俯仰角，光锥束的方位角和俯仰角



注：其他符号将在下文给出具体说明

四、问题一的模型建立与求解

4.1 问题一的模型准备

4.1.1 镜场大地坐标系的建立

在计算定日镜场的效率时，由于一年中太阳是运动的，因此需要确定太阳的位置才能计算定日镜场的效率。根据题目描述，为了简化计算，“平均”指标只需在每月 21 日的 9:00、10:30、12:00、13:30、15:00 计算，因此当太阳位置位于每月 21 日的 5 个时间点时，太阳位置固定，可计算镜场效率。首先，以圆形区域中心为原点，正东方向为 x 轴方向，正北方向为 y 轴方向，垂直地面方向为 z 轴方向建立定日镜场的大地坐标系。

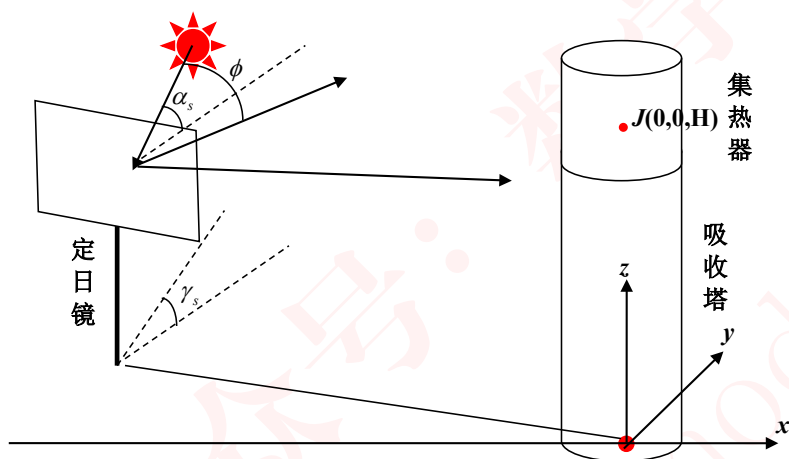


图 1：定日镜场的大地坐标系

在该坐标系中，第 i 个定日镜中心位置坐标 $d_i(x_i, y_i, z_i)$ ，集热器中心坐标 $J(0,0,H)$ 。

4.1.2 太阳主光线入射、反射以及定日镜法向

太阳光并非平行光线，而是具有一定锥形角的一束锥形光线。因此，先对太阳的一条主光线进行描述。进行在计算定日镜场效率时，需要首先需要对定日镜的入射主光线、反射主光线、法向、俯仰角和方位角进行建模。首先，根据太阳高度角和太阳方位角的定义和公式，结合下图，则可得到太阳光的入射主光线方向向量如下：

$$\vec{L}_i = (-\cos \alpha_s \sin \gamma_s, -\cos \alpha_s \cos \gamma_s, -\sin \alpha_s) \quad (1)$$

其中，根据太阳方位角和太阳高度角的公式，ST 为当地时间， δ 为太阳赤纬角，D 为以春分作为第 0 天起算的天数， ω 为太阳时角， φ 为当地纬度：

$$\begin{cases} \sin \alpha_s = \cos \delta \cos \varphi \cos \omega + \sin \delta \sin \varphi \\ \cos \gamma_s = \frac{\sin \delta - \sin \alpha_s \sin \varphi}{\cos \alpha_s \cos \varphi} \\ \omega = \frac{\pi}{12}(ST - 12) \\ \sin \delta = \sin \frac{2\pi}{360} \sin \left(\frac{2\pi}{360} 23.45 \right) \end{cases}$$

而正常来说，对于中国区域，早上太阳光从东边射来，中午太阳光从南边射来，傍晚



太阳光从西边射来。早上的太阳方位角在 90° 左右（但一年当中，有一定的角度范围变化），正中午的太阳方位角在 180° （正南方），傍晚的太阳方位角在 270° 左右（但一年当中，有一定的角度范围变化）。因此，对于太阳方位角的正弦值需要进行分类讨论，如下：

$$\sin \gamma_s = \begin{cases} \sqrt{1 - \cos^2 \gamma_s}, ST < 12 \\ -\sqrt{1 - \cos^2 \gamma_s}, ST \geq 12 \end{cases}$$

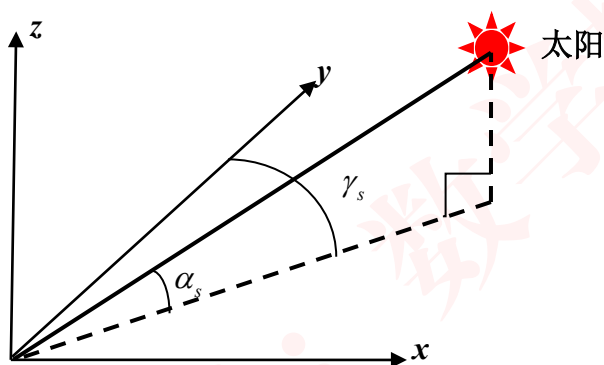


图 2：太阳方位角和高度角图示

我们认为，控制器控制定日镜的法向（即控制定日镜的俯仰角和方位角）使得经过定日镜中心的反射主光线刚好经过集热器中心，即为定日镜法向对不同太阳位置的最佳调节。因此，经过定日镜中心的反射主光线单位向量为：

$$\vec{L}_r = \frac{(-x_i, -y_i, H - z_i)}{\left| (-x_i, -y_i, H - z_i) \right|}$$

根据平行四边形定理，从而得到该时刻太阳位置下，第 i 个定日镜的法向量为（以下简称定日镜法向量）：

$$\vec{n} = (-\vec{L}_i) + \vec{L}_r$$

通过镜场大地坐标系 z 轴的方向向量 $\vec{Z} = (0, 0, 1)$ ，计算定日镜的俯仰角 β ，即定日镜的法向量与镜场大地坐标系投影轴的夹角，如下图：

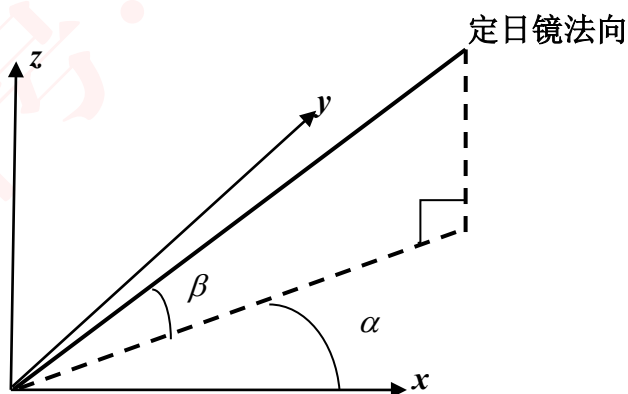


图 3：定日镜的方位角（ α ）和俯仰角（ β ）计算示意图



$$\beta = \frac{\pi}{2} - \arccos \left(\frac{\vec{n} \cdot \vec{Z}}{|\vec{n}| \cdot |\vec{Z}|} \right) \quad (2)$$

将定日镜的法向量投影于大地坐标系的 xoy 平面，得到定日镜法向量的投影向量 \vec{n}_0 ，计算 \vec{n}_0 与大地坐标系 x 轴的方向向量 \vec{S} 的夹角为方位角 α ，如下：

$$\alpha = \begin{cases} \arccos \left(\frac{\vec{n}_0 \cdot \vec{S}}{|\vec{n}_0| \cdot |\vec{S}|} \right), \vec{n}_0(2) \geq 0 \\ -\arccos \left(\frac{\vec{n}_0 \cdot \vec{S}}{|\vec{n}_0| \cdot |\vec{S}|} \right), \vec{n}_0(2) < 0 \end{cases} \quad (3)$$

其中 $\vec{S} = (1, 0, 0)$, $\vec{n}_0 = (\vec{n}_0(1), \vec{n}_0(2), 0)$ 。光学效率包含了阴影遮挡效率、余弦效率、大气透射率、集热器截断效率和镜面反射率。因此确定太阳的反射和入射及定日镜的法向调节后，需表示太阳光线入射反射造成的阴影遮挡效率、集热器接收情况和余弦效率。

4.1.3 定日镜面坐标系的建立

为了方便表示和计算定日镜面上的坐标，我们需要将定日镜面的坐标系转换到镜场大地坐标系或者将镜场大地坐标系转换到定日镜面坐标系。因此，以定日镜中心为原点，以经过原点 d 并垂直镜面的法向量为 z 轴。以经过原点平行镜面的宽为 x 轴，以经过原点平行镜面的长为 y 轴，建立镜面坐标系（见下图）：

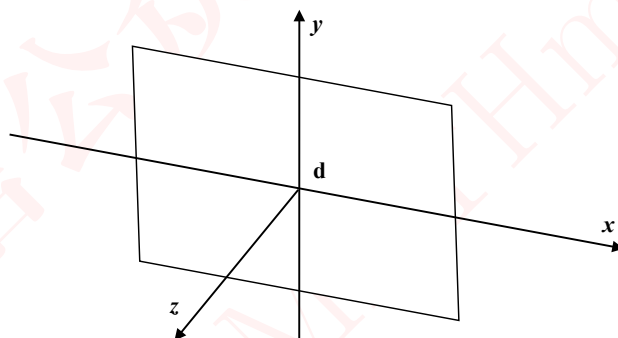


图 4：定日镜面的坐标系

4.1.4 光锥坐标系的建立

太阳光并非平行光线，而是具有一定锥形角的一束锥形光线，因此太阳入射光线经定日镜任意一点的反射光线也是一束锥形光线。在考虑阴影遮挡和集热器吸收能量时，需要考虑锥形光束的吸收和遮挡。因此，我们需要建立起以主入射光线和主反射光线为锥体中心线的光锥坐标系。大地坐标系下的反射光锥束的中心主光线 $\vec{L}_z = (m, n, p)$ ，以主光线为 z 轴，则 z 轴的方向向量为 $\vec{s} = (0, 0, 1)$ ，则锥体光线束的俯仰角 β_z 为：

$$\beta_z = \frac{\pi}{2} - \arccos \left(\frac{\vec{L}_z \cdot \vec{s}}{|\vec{L}_z| \cdot |\vec{s}|} \right)$$

锥体光线束的主光线与 x 轴的夹角即方位角 α_z 为：



$$\alpha_z = \begin{cases} \arccos \frac{\vec{L}'_z \cdot \vec{s}_1}{|\vec{L}'_z| |\vec{s}_1|}, y \leq 0 \\ -\arccos \frac{\vec{L}'_z \cdot \vec{s}_1}{|\vec{L}'_z| |\vec{s}_1|}, y > 0 \end{cases}$$

其中, $\vec{s} = (0, 0, 1), \vec{s}_1 = (1, 0, 0)$ 。描述光锥直线, 侧边任一光线:

$$\vec{f} = (\sin \theta_1 \cos \theta_2, \sin \theta_1 \sin \theta_2, \cos \theta_1) \quad (4)$$

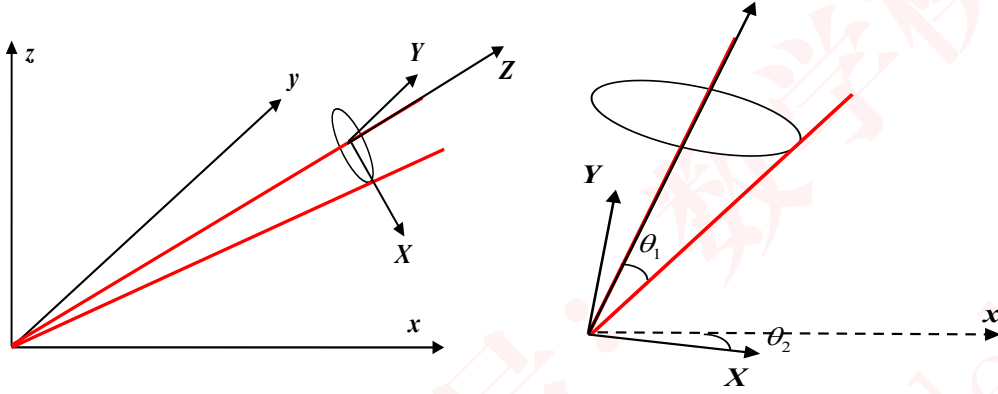


图 5: 光锥束的坐标系 (X-Y-Z)

其中, $\theta_1 \in (0, 4.65 \text{ mrad}), \theta_2 \in (0, 2\pi)$ 。

4.2 问题一的模型建立

为计算一束锥形光线的入射和反射受到的阴影遮挡损失和集热器吸收能量等, 需要对镜面坐标系、镜场大地坐标系和光锥坐标系进行坐标变换。为此, 建立定日镜场的效率计算模型, 计算光学效率、输出热功率等:

4.2.1 镜面、大地、光锥坐标系的变换

首先, 需要将镜面坐标系转成镜场大地坐标系, 对于旋转后的矩阵, 根据欧拉角旋转矩阵来求解。将镜面坐标系转化为镜场大地坐标系, 第 i 个定日镜的俯仰角为 β_i 和方位角为 α_i 。旋转矩阵与俯仰角 β_i 和方位角 α_i 有关, 因此首先绕 x 轴旋转 α_i , 再绕 z 轴旋转 β_i 。则第 i 个定日镜的镜面坐标系到大地坐标系的转换关系矩阵为:

$$T_i = \begin{bmatrix} \cos \alpha_i \cos(\pi/2 - \beta_i) & -\sin \alpha_i & \cos \alpha_i \sin(\pi/2 - \beta_i) \\ \sin \alpha_i \cos(\pi/2 - \beta_i) & \cos \alpha_i & \sin \alpha_i \sin(\pi/2 - \beta_i) \\ -\sin(\pi/2 - \beta_i) & 0 & \cos(\pi/2 - \beta_i) \end{bmatrix} \quad (5)$$

而将光锥坐标系转到镜场大地坐标系上时, 其所对应光锥的俯仰角 β_z 和方位角 α_z 发生变换, 且与每束光锥在大地坐标系里的不同俯仰角 β_z 和方位角 α_z 有关, 其旋转矩阵也会发生变换:



$$T_z = \begin{bmatrix} \cos \alpha_z \cos(\pi/2 - \beta_z) & -\sin \alpha_z & \cos \alpha \sin(\pi/2 - \beta_z) \\ \sin \alpha_z \cos(\pi/2 - \beta_z) & \cos \alpha_z & \sin \alpha \sin(\pi/2 - \beta_z) \\ -\sin(\pi/2 - \beta_z) & 0 & \cos(\pi/2 - \beta_z) \end{bmatrix} \quad (6)$$

4.2.2 阴影遮挡效率

阴影遮挡效率 η_{sb} 由阴影遮挡损失计算，阴影遮挡损失分为三部分。若塔身对某入射光线造成阴影遮挡损失，那么不再考虑该入射光线被前定日镜所阻挡和其所对应的反射光线被前定日镜所遮挡。若塔未对某入射光线造成遮挡，则需要考虑二、三部分。对于，某一束入射光锥束，首先判断是否被塔身遮挡，然后再判断是否被前定日镜遮挡。之后，入射光锥束变为反射光锥束，判断反射光锥束是否会被前定日镜遮挡。若反射光锥束不被前定日镜遮挡，则成功反射出，再考虑是否被集热器接收。

1) 塔对镜场造成的阴影损失

首先考虑经过 a 镜上的入射光线是否经过塔身，即被塔身遮挡，当遮挡后即不需考虑是否该光线的入射和反射，示意图如下图 7。

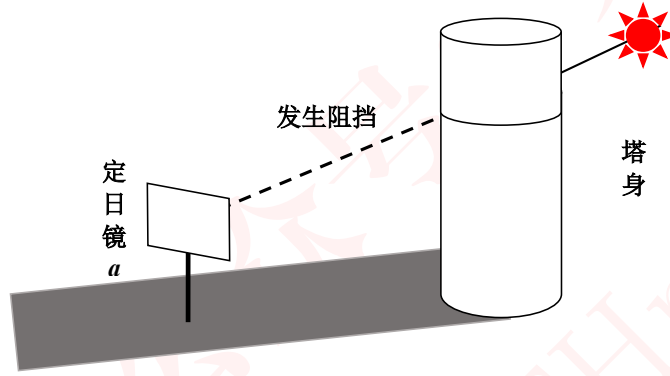


图 6: 塔损失模型

入射光线可表示为：

$$\vec{L}_i = (-\cos \alpha_s \sin \gamma_s, -\cos \alpha_s \cos \gamma_s, -\sin \alpha_s) = (m, n, p)$$

描述光锥束侧边任一光线：

$$\vec{f} = (\sin \theta_1 \cos \theta_2, \sin \theta_1 \sin \theta_2, \cos \theta_1)$$

再将光锥坐标系进行坐标变换，换到地面坐标。结果如下：

$$\vec{f}' = T_z \cdot \vec{f}$$

其中 $\vec{f}' = (m', n', p')$ ，取镜面上任意一点 $D(x_0, y_0, z_0)$ ，联立方程：

$$\begin{cases} \frac{x-x_0}{m'} = \frac{y-y_0}{n'} = \frac{z-z_0}{p'} = t \\ x^2 + y^2 = R^2 \end{cases} \Rightarrow \begin{cases} x = x_0 + m't \\ y = y_0 + n't \\ z = z_0 + p't \end{cases} \quad (7)$$

得到以下式子：

$$(m'^2 + n'^2)t^2 + 2(m'x_0 + n'y_0)t + x_0^2 + y_0^2 - R^2 = 0 \quad (8)$$



其所对应的判别式如下：

$$\begin{cases} \Delta = 4(m'x_0 + n'y_0)^2 - 4(m'^2 + n'^2)(x_0^2 + y_0^2 - R^2) < 0, \text{ 无遮挡} \\ \Delta = 4(m'x_0 + n'y_0)^2 - 4(m'^2 + n'^2)(x_0^2 + y_0^2 - R^2) \geq 0, \text{ 有交点} \end{cases} \quad (9)$$

若 $\Delta \geq 0$ ，则：

$$\begin{cases} \min(z_0 + t_1 p', z_0 + t_2 p') \leq 84, \text{ 遮挡} \\ \min(z_0 + t_1 p', z_0 + t_2 p') > 84, \text{ 无遮挡} \end{cases}$$

若为判断入射主光线 (m, n, p) 是否被塔身遮挡，同理如上联立方程求判别式。且每块定日镜的入射光线均需要判断是否被塔身遮挡，则此时将第 i 根入射光锥束是否被塔身遮挡记为 b_1^i ，其为 0, 1 变量：

$$b_1^i = \begin{cases} 0, & \text{if } \min(z_0 + t_1 p', z_0 + t_2 p') > 84, \text{ 未挡住} \\ 1, & \text{if } \min(z_0 + t_1 p', z_0 + t_2 p') \leq 84, \text{ 挡住} \end{cases} \quad (10)$$

2) 后定日镜接收的入射太阳光被前定日镜所阻挡的阴影损失

当 a 镜的入射光线未被塔身遮挡时，需要考虑后定日镜接收的太阳光被前定日镜所阻挡的阴影损失和后定日镜在反射太阳光时被前定日镜阻挡而未到达集热器上的遮挡损失。首先考虑后定日镜接收的太阳光被前定日镜所阻挡的阴影损失，求阴影遮挡损失，即计算经过 a 镜坐标系中的某一点 $D_1(x_1, y_1, 0)$ 的入射光线是否被 b 镜遮挡，判断其与 b 镜平面的交点是否在 b 镜面内。

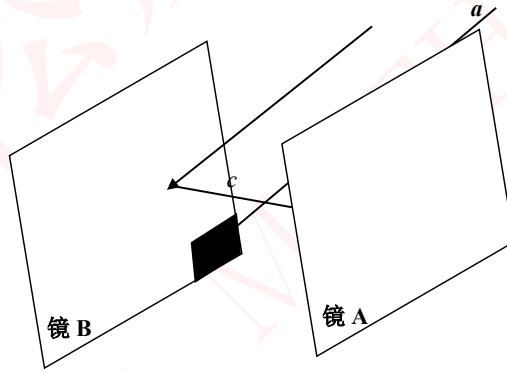


图 7：塔身不遮挡下，定日镜的阴影遮挡损伤模型图示

首先进行光线的坐标系变换，将大地坐标系中的入射主光线转换到 b 镜的镜面坐标系中，则转换后的 b 镜坐标系下的入射光线（束）为：

$$\vec{L}' = (T_b)^T \cdot \vec{L}_i = (\tilde{m}, \tilde{n}, \tilde{p})$$

以该入射主光线为 z 轴建立光锥束坐标系，其坐标如下：

$$\vec{f}_2 = (\sin \theta_1 \cos \theta_2, \sin \theta_1 \sin \theta_2, \cos \theta_1)$$

再将光锥坐标系进行坐标变换，换到镜场大地坐标系。结果如下：

$$\vec{f}_2' = T_z \cdot \vec{f}_2$$



再将镜场大地坐标系的 \vec{f}_2' 转换到 b 镜坐标系下，结果如下：

$$\vec{f}_2'' = (T_b)^T \cdot \vec{f}_2'$$

然后进行点的变换，将 a 镜坐标系上的一点转换到大地坐标系上，再转到 b 镜的坐标系下，其点为 $D_1''(x_1'', y_1'', z_1'')$ 。经过 a 镜坐标系中的某一点 $D_1(x_1, y_1, z_1)$ ，则其转换到大地坐标系上的坐标为：

$$D_1' = T_a \cdot D_1 + O_a \quad (11)$$

其中， $D_1'(x_1', y_1', z_1')$ O_a 为 a 镜原点在大地坐标系的坐标，即 (x_a, y_a, z_a) 。再转换到 b 镜坐标系下得到：

$$D_1'' = (T_b)^T \cdot (D_1' - O_b) \quad (12)$$

其中 O_b 为 b 镜原点在大地坐标系的坐标，即 (x_b, y_b, z_b) 。最后判断入射光线与 b 镜平面的交点是否在 b 镜的镜面上。设 b 镜坐标系下的入射光线向量 $\vec{L}_i' = (a_i, b_i, c_i)$ ，计算经过 $D_1''(x_1'', y_1'', z_1'')$ 的入射主光线与 b 镜的交点 $(x_i, y_i, 0)$ 是否在 b 镜内：

$$\frac{x_2 - x_1''}{a_i} = \frac{y_2 - y_1''}{b_i} = \frac{-z_1''}{c_i} \quad (13)$$

若 $-3 \leq x_i \leq 3, -3 \leq y_i \leq 3$ ，且交点经过转换到镜场大地坐标系后的 Z 坐标值要大于 z_1' ，则表明经过 a 镜某点的入射主光线被 b 镜阻挡。同理，光锥束的光线经过转换后，结合 $D_1''(x_1'', y_1'', z_1'')$ ，求 b 镜的交点 $(x_i, y_i, 0)$ 进行判断。最后，引入 b_2^i 表示入射主光线或者光锥束是否被前定日镜阻挡，且每个定日镜均需要进行判断：

$$b_2^i = \begin{cases} 0, & \text{else, 未挡住} \\ 1, & \text{if } -3 \leq x_i, y_i \leq 3 \text{ and } z_1' < Z, \text{ 挡住} \end{cases} \quad (14)$$

3) 后定日镜在反射太阳光时被前定日镜阻挡而未到达集热器上的遮挡损失

反射光线（束）和入射光线（束）一一对应，即锥形光束入射也是锥体光束反射。入射光线（束）转换到大地坐标系上为 \vec{f}_2' 。根据几何关系，求对应的反射光线满足：

$$\vec{g} - \vec{f}_2' = -|\vec{f}_2'| \cdot \cos \theta \cdot \vec{n} \quad (15)$$

其中， \vec{n} 为镜面法向量。 θ 为入射光线与法线夹角。得到：

$$\vec{g} = \vec{f}_2' - 2 \cdot \vec{f}_2' \cdot \vec{n} \cdot \vec{n}$$

\vec{L}_r 由以上公式计算。接着，同 2) 所述，计算定日镜在反射太阳光时被前定日镜阻挡而未到达集热器上的遮挡损失。即判断经过 a 镜坐标系中的某一点 $D_1(x_1, y_1, z_1)$ 的反射光线落入 b 镜坐标系中，首先进行光线的坐标系变换，将大地坐标系中的反射光线转换到 b 镜的镜面坐标系中，则转换后的 b 镜坐标系下的反射光线为：

$$\vec{L}_r' = T_b^T \cdot \vec{L}_r$$

其中 \vec{L}_r 为反射光线向量。设 b 镜坐标系下的反射光线 $\vec{L}_r' = (a_r, b_r, c_r)$ ，反射光线与 b 镜



平面的交点为 $(x_r, y_r, 0)$ ，计算反射光线与 b 镜的交点是否在 b 镜内：

$$\frac{x_r - x_1''}{a_r} = \frac{y_r - y_1''}{b_r} = \frac{-z_1''}{c_r} \quad (16)$$

若 $-3 \leq x_r \leq 3, -3 \leq y_r \leq 3$ ，且反射光线与 b 镜的交点相对于大地坐标系的 Z 坐标值大于 z_1' ，则表明该经过 a 镜某点的反射光线被 b 镜阻挡。

$$b_3^i = \begin{cases} 0, & \text{else, 未挡住} \\ 1, & \text{if } -3 \leq x_r, y_r \leq 3 \text{ and } z_1' < Z, \text{ 挡住} \end{cases} \quad (17)$$

最后，将综合以上三部分的阴影遮挡损失，可得到阴影遮挡效率， N 为总光线数量。

$$\eta_{sb} = 1 - \sum_{i=1}^N (b_1^i + b_2^i + b_3^i) / N$$

$$\begin{cases} \text{if } b_1^i = 1 \Rightarrow b_2^i = 0, b_3^i = 0 \\ \text{if } b_1^i = 0 \Rightarrow b_2^i = 0, 1 \\ \text{if } b_2^i = 0 \Rightarrow b_3^i = 0, 1 \\ \text{if } b_2^i = 1 \Rightarrow b_3^i = 1 \end{cases} \quad (18)$$

4.2.3 集热器截断效率

考虑经过 a 镜上一点 (x_0, y_0, z_0) 的反射光线向量 \vec{L}_r 。在未被前定日镜遮挡后，成功反射后是否与集热器存在交点。将反射光线与集热器方程进行联立：

$$\begin{cases} x^2 + y^2 = R^2, z \in [76, 84] \\ \frac{x - x_0}{-x_i} = \frac{y - y_0}{-y_i} = \frac{z - z_0}{H - z_i} = t \end{cases} \Rightarrow \begin{cases} x = -x_i t + x_0 \\ y = -y_i t + y_0 \\ z = (H - z_i) t + z_0 \end{cases} \quad (19)$$

得到方程如下：

$$(-x_i t + x_0)^2 + (-y_i t + y_0)^2 = R^2 \quad (20)$$

整理得到方程如下：

$$\begin{cases} \Delta \geq 0, & \text{有交点} \\ \Delta < 0, & \text{无法接收} \end{cases}$$

若 $\Delta \geq 0$ 有交点，两个解为 t_1, t_2 ，则考虑 z 坐标最小时在集热器的高度范围内：

$$\begin{cases} \min\{(H - z_0)t_1 + z_i, (H - z_0)t_2 + z_i\} \in [76, 84], & \text{成功接收} \\ \min\{(H - z_0)t_1 + z_i, (H - z_0)t_2 + z_i\} \notin [76, 84], & \text{无法接收} \end{cases}$$

引入 $0, 1$ 变量 r_j ，表示第 j 根成功反射的光线是否接收。且每块定日镜均需要进行反射光线或光锥束接收计算：

$$r_j = \begin{cases} 0, & \text{阴影遮挡或未被接收} \\ 1, & \min\{(H - z_0)t_1 + z_i, (H - z_0)t_2 + z_i\} \in [76, 84] \end{cases} \quad (21)$$



使用 r_j 表示集热器截断效率，其中， N' 表示总光线数：

$$\eta_{trunc} = \frac{\sum_{j=1}^{N'} r_j}{N' - \sum_{i=1}^{N'} (b_1^i + b_2^i + b_3^i)} \quad (22)$$

4.2.4 余弦效率

余弦损失，太阳光入射方向与镜面采光口法线方向不平行引起的接收能量损伤。

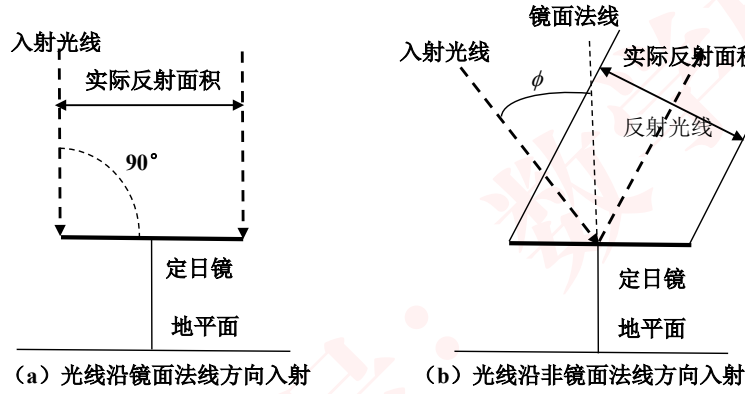


图 8：余弦损失的示意图

因此，余弦效率的表达式如下：

$$\eta_{\cos} = \cos \phi = \frac{\vec{L}_i \cdot \vec{L}_r}{|\vec{L}_i| \cdot |\vec{L}_r|} = 1 - \frac{S - \cos \phi \cdot S}{S} \quad (23)$$

4.2.5 定日镜场的效率计算模型

1) 光学效率

$$\eta_i^j = \eta_{sb} \eta_{\cos} \eta_{at} \eta_{trunc} \eta_{ref}$$

其中， $\eta_{ref} = 0.92$, $\eta_{at} = 0.99321 - 0.0001176d_{HR} + 1.97 \times 10^{-8} \times d_{HR}^2$ ($d_{HR} \leq 1000$)， η_i^j 表示第 i 面定日镜在第 j 个时刻的光学效率。那么，某月 21 日的平均光学效率为：

$$\bar{\eta} = \sum_{j=1}^5 \sum_{i=1}^N \eta_i^j / 5 \quad (24)$$

年平均光学效率为每个月的平均光学效率和的平均值：

$$\tilde{\eta} = (\bar{\eta}_1 + \bar{\eta}_2 + \dots + \bar{\eta}_{12}) / 12$$

其中， $\bar{\eta}_1$ 表示 1 月 21 日的平均光学效率。

2) 输出热功率

定日镜场的输出热功率如下，其为某时刻的输出热功率：

$$E_{field}^j = DNI \cdot \sum_{i=1}^n A_i \eta_i^j$$



其中, A_i 为第 i 面定日镜的采光面积, η_i^j 表示第 j 个时刻光学效率, 法向直接辐射辐照度 DNI 的计算公式如下, 其中 $G_0 = 1.366 \text{ kW/m}^2$, $H = 3000 \text{ m}$:

$$DNI = G_0 \left[a + b \exp \left(-\frac{c}{\sin \alpha_s} \right) \right]$$

$$a = 0.4237 - 0.00821(6 - H)^2$$

$$b = 0.5055 + 0.00595(6.5 - H)^2$$

$$c = 0.2711 + 0.01858(2.5 - H)^2$$

因此, 平均输出热功率 \bar{E}_{field} 为:

$$\bar{E}_{field} = \frac{1}{5} \sum_{j=1}^5 E_{field}^j = \frac{1}{5} DNI \cdot \sum_{j=1}^5 \sum_{i=1}^n A_i \eta_i^j \quad (25)$$

\bar{E}_{field}^1 表示 1 月 21 日的平均输出热功率, 年输出平均热功率 \bar{P} 为:

$$\bar{P} = \sum_{i=1}^{12} \bar{E}_{field}^i / 12 \quad (26)$$

3) 单位面积镜面平均输出热功率

单位面积镜面的平均输出热功率为平时输出热功率与镜面面积的比值, 如下式:

$$AVE_{field} = \bar{E}_{field} / \sum_{i=1}^n A_i \quad (27)$$

单位镜面面积年平均输出热功率如下:

$$\bar{P}_s = \bar{P} / \sum_{i=1}^n A_i \quad (28)$$

4.3 问题一模型的求解

4.3.1 利用邻接矩阵寻找问题定日镜

按照附件顺序进行定日镜编号。首先, 通过计算每个定日镜中心在镜场大地坐标系 xoy 的 x, y 坐标之间的距离, 判断 a 定日镜与其他定日镜之间是否属于相邻或者较为相近的定日镜。由于定日镜具有一定尺寸, 且太阳定日镜的太阳影子所造成阴影遮挡损失一般与其相邻的定日镜有关, 由于在 a 定日镜上取点描述入射、反射光线和光锥束, 需要网格化取点和构造外侧锥束线, 且 a 定日镜需要与其他 1744 块定日镜进行阴影遮挡损失计算, 计算量至少 $1744 \times 1744 \times \tau^2 \times \lambda^2 \times \rho^2$, τ, λ, ρ 为镜面网格化步长、锥体束 θ_1 角度步长、锥体束 θ_2 角度步长。因此在计算 a 定日镜与其他定日镜的阴影遮挡损失时, 考虑利用邻接矩阵减少计算量。每个定日镜中心在镜场大地坐标系 xoy 的 x, y 坐标之间的距离计算公式如下:

$$d = \sqrt{(x_k - x_m)^2 + (y_k - y_m)^2}$$

通过距离 d 判断是否相邻, 考虑题中要求相邻定日镜底座中心之间的距离比镜面宽度多 5 m。先计算距离, 得到 d' 矩阵, 观察距离大小, 给定 $d = 20$, 邻接矩阵如下:



$$I(i, j) = \begin{cases} 0, d > 20 \\ 1, d \leq 20 \end{cases}$$

其中， $I(i, j)=1$ 表示第*i*块定日镜和第*j*块定日镜相邻，否则不相邻。根据此减少*a*定日镜的阴影遮挡损失的定日面遍历数，更加精确地定位问题定日镜。

4.3.2 网格化镜面和取点求解

在某个定日镜上取点，步长 1.5m，即镜上取 25 个点，在光锥束上取 $\theta_1 = 0.002rad$ ， $\theta_2 = 0, \pi/2, \pi, 3\pi/2$ ，即光锥束取 4 个点。首先，计算以 3 月 21 日为起止的天数 D：

$$D = [306, 337, 0, 31, 61, 92, 122, 153, 184, 214, 245, 275]$$

根据不同月份、时间内，计算所对应太阳位置。而后需要计算定日镜的俯仰角和方位角。而后，需要遍历每一个入射光线圆锥的光线，根据入射主光线，计算主光线锥体系的旋转矩阵，再通过旋转矩阵将不同坐标系的坐标进行转换。

接着判断入射光线（锥束）是否被塔身遮挡，若被塔身遮挡则造成阴影遮挡损失。然后，判断入射光线（锥束）是否被其他光镜遮挡。根据几何关系，计算所对应的反射光线（锥束），判断是否被其他镜遮挡。若未造成遮挡，则计算集热器是否接受。之后计算定日镜场的计算。

4.3.3 结果计算

根据定日镜场效率计算的公式，求解 5 个时刻的平均值，可得到表 1：

表 1：问题 1 的每月 21 日平均光学效率及输出功率

日期	平均光学效率	平均余弦效率	平均阴影遮挡效率	平均截断效率	单位面积镜面平均输出热功率(kW/m ²)
1 月 21 日	0.504022	0.719936	0.896844	0.905984	0.439239
2 月 21 日	0.52647	0.74044	0.917321	0.890886	0.496346
3 月 21 日	0.541603	0.76114	0.922758	0.881162	0.538716
4 月 21 日	0.554958	0.77934	0.922698	0.878468	0.571094
5 月 21 日	0.564145	0.789317	0.922121	0.880157	0.589456
6 月 21 日	0.567412	0.792359	0.921968	0.881265	0.595407
7 月 21 日	0.564043	0.78921185	0.92213983	0.88012325	0.58926025
8 月 21 日	0.55441	0.778637	0.922747	0.878471	0.569872
9 月 21 日	0.541021	0.760092	0.922691	0.881786	0.536912
10 月 21 日	0.524031	0.737835	0.915258	0.892657	0.489964
11 月 21 日	0.501759	0.718196	0.894627	0.907023	0.433597
12 月 21 日	0.490888	0.711082	0.8825	0.912004	0.40824



表 2：问题 1 的年平均光学效率及输出功率表

年平均光学效率	年平均余弦效率	年平均阴影遮挡效率	年平均截断效率	年平均输出热功率 (MW)	单位面积镜面年平均输出热功率(kW/m ²)
0.536230167	0.756465488	0.913639403	0.889165521	32.76117051	0.521508604

4.4 问题一的结果分析

取不同的镜面网格化步长下的 1 月 21 日上午 9:00 的单位镜面面积平均输出热功率, 当步长从 1.5 开始后, 单位镜面面积平均输出热功率增长的速度变得缓慢。因此, 我们认为当步长为 1.5 时, 单位镜面面积平均输出热功率接近稳定。

表 3：不同的镜面网格化步长下的 1 月 21 日上午 9:00 的单位面积平均输出热功率

τ	6	3	1.5	1	0.5	0.25
AVE_{field}	0.29982404	0.35749342	0.37905415	0.389954	0.39836452	0.40294417

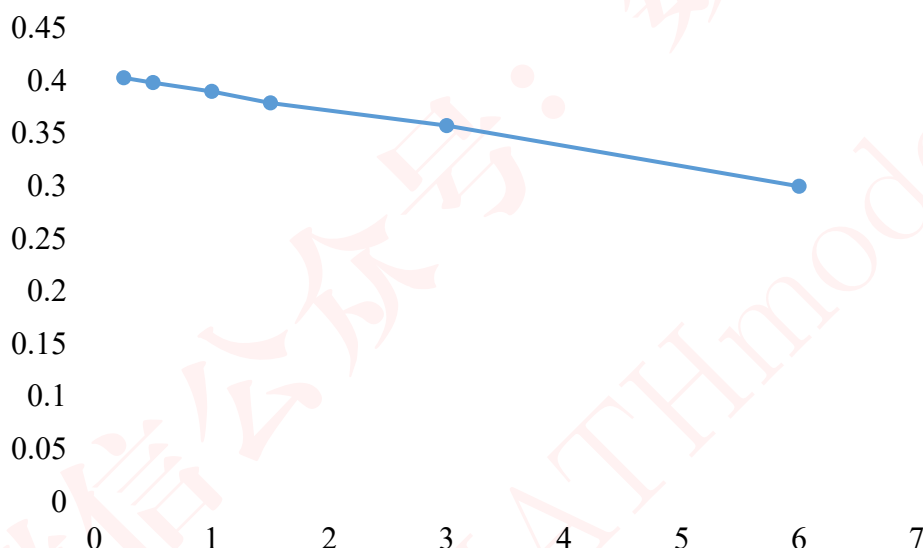


图 9：镜面取点步长变小，单位面积平均输出热功率变化

在 1 月 21 日上午 9:00, 定日镜场的单位面积平均输出热功率与安装高度的变化:

表 4：高度变化下, 1 月 21 日上午 9:00 的单位面积平均输出热功率

高度	3	3.5	4	4.5	5	5.5	6
AVE_{field}	0.35798	0.35772	0.35749	0.35730	0.35705	0.35663	0.35641

说明高度对单位面积平均输出热功率的变化影响小。

五、问题二的模型建立与求解

5.1 问题二的模型建立

根据 4.3.1 计算邻接矩阵, 判断是否相邻。按设计要求, 定日镜场的额定年平均输出热功率 (以下简称额定功率) 为 60MW, 若所有定日镜尺寸及安装高度相同, 设计定日镜场的参数, 使得在约束条件下, 使得单位镜面面积年平均输出热功率尽量大。因此, 建立以下优化模型:



1) 决策变量: 吸收塔位置坐标、定日镜尺寸（宽度和高度）、安装高度、定日镜数目、定日镜位置。

吸收塔位置坐标: (X_o, Y_o) ;

定日镜尺寸: w, v , 其中 w 为定日镜宽度, v 为定日镜高度。每个定日镜的尺寸相同均为 w, v ;

安装高度: \tilde{h} ;

定日镜数目: N ;

定日镜位置: $P_i(x_i, y_i, z_i)$, 其中 $z_i = h$, 所有定日镜的安装高度相同。

2) 目标函数: $\max\{\bar{P}_s\} = \max\left\{\left[\sum_{i=1}^{12} \bar{E}_{field}^i / (12 * \tilde{S})\right]\right\}$, 其中 $\tilde{S} = A_i \cdot N$, $A_i = w \cdot v$ 为第 i 面定日镜的面积（均相同）。

由于吸收塔位置坐标 (X_o, Y_o) 发生变化, 计算定日镜场的输出热功率时, 塔身对定日镜场的阴影遮挡损失和反射光线在集热器上的接收判别方程发生变化。根据问题一的 4.2.2 和 4.2.3,

● 阴影遮挡损失变化如下, 首先是联立方程发生变化:

$$\begin{cases} \frac{x-x_0}{m'} = \frac{y-y_0}{n'} = \frac{z-z_0}{p'} = t \\ (x-X_0)^2 + (y-Y_0)^2 = R^2 \end{cases} \Rightarrow \begin{cases} x = x_0 + m't \\ y = y_0 + n't \\ z = z_0 + p't \end{cases} \quad (29)$$

得到以下关于 t 的方程式子:

$$At^2 + Bt + C = 0 \quad (30)$$

其所对应的判别式如下:

$$\begin{cases} \Delta = B^2 - 4AC < 0, \text{ 无遮挡} \\ \Delta = B^2 - 4AC \geq 0, \text{ 有交点} \end{cases}$$

接着判断 $\Delta \geq 0$ 时的交点 t_1, t_2 是否在大于 84m。

● 反射光线在集热器上的接收判别方程变化如下:

考虑经过 a 镜上一点 (x_0, y_0, z_0) 的反射光线向量 \vec{L}_r 。在未被前定日镜遮挡后, 成功反射后是否与集热器存在交点。将反射光线与集热器方程进行联立:

$$\begin{cases} (x-X_0)^2 + (y-Y_0)^2 = R^2, z \in [76, 84] \\ \frac{x-x_0}{-x_i} = \frac{y-y_0}{-y_i} = \frac{z-z_0}{H-z_i} = t \end{cases} \Rightarrow \begin{cases} x = -x_i t + x_0 \\ y = -y_i t + y_0 \\ z = (H-z_i)t + z_0 \end{cases} \quad (31)$$

得到方程如下:

$$(-x_i t + x_0 - X_0)^2 + (-y_i t + y_0 - Y_0)^2 = R^2$$



整理得到方程如下：

$$\begin{cases} \Delta \geq 0, & \text{有交点} \\ \Delta < 0, & \text{无法接收} \end{cases}$$

若 $\Delta \geq 0$ 有交点，两个解为 t_1, t_2 ，则考虑 z 坐标最小时在集热器的高度 76-84m 范围内。

3) 约束条件：

- 吸收塔周围 100m 不安装定日镜，定日镜场的圆形区域半径为 350m：

$$x_i^2 + y_i^2 \leq 350^2, (x_i - X_0)^2 + (y_i - Y_0)^2 \geq 100^2$$

- 镜面宽度不小于镜面高度，且镜面边长在 2m 至 8m 之间：

$$w \geq v, 2 \leq w \leq 8, 2 \leq v \leq 8$$

- 安装高度在 2m 至 6m 之间，安装高度保证水平转轴转动时不触及地面：

$$2 \leq \tilde{h} \leq 6, v/2 < \tilde{h}$$

- 相邻定目镜底座中心距离比镜面宽度多 5m：

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} > w + 5$$

其中， i, j 表示两个相邻的定日镜，由邻接矩阵计算。

- 达到额定功率：

$$\bar{P}_s \cdot \tilde{S} \geq 60$$

优化模型如下，其中， i, j 表示两个相邻的定日镜：

$$\begin{aligned} \max \{ \bar{P}_s \} = \max \left\{ \left[\sum_{i=1}^{12} \bar{E}_{field}^i / (12 * \tilde{S}) \right] \right\} \\ \begin{cases} (x_i - X_0)^2 + (y_i - Y_0)^2 \geq 100^2 \\ w \geq v, 2 \leq w \leq 8, 2 \leq v \leq 8 \\ 2 \leq \tilde{h} \leq 6, v/2 < \tilde{h} \\ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} > w + 5 \\ x_i^2 + y_i^2 \leq 350^2 \\ \bar{P}_s \cdot \tilde{S} \geq 60 \end{cases} \end{aligned} \quad (32)$$

5.2 问题二的模型求解

5.2.1 圆环最大镜面面积的蜂窝排列

考虑 E_{field} 与第 i 块面板的光学效率和面积有关。而我们认为在时刻变化时，光学效率基本变化不大。因此，为了首先满足题中的额定功率 60MW，在满足约束条件的情况下，首先需要给定宽度使得定日镜场内排列的定日镜面积尽量大，从而确定定日镜的尺寸、数目和位置。即在内径 $r = 100m$ ，外径 $R = 350m$ 的圆环中尽可能多的排列定日镜，以下我们采用蜂窝排列法，使得每个定日镜与其相邻的定日镜成为正三角形，



如下布局。根据蜂窝排列法，按照约束条件，得到正六边形边长的限制范围为大于 $w+5 \leq \kappa$ 。一行行进行考虑，对于第一、二行：

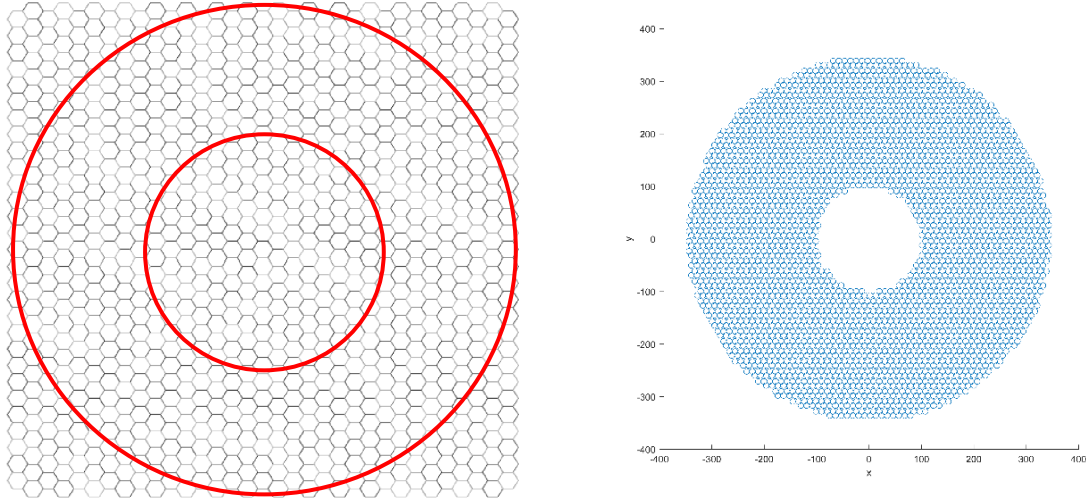


图 10：蜂窝排列法确定定日镜尺寸、数目和位置（左）和计算后的图（右）

$$\begin{aligned}
 \text{line1: } & \begin{cases} x_{11} = -R \\ y_{11} = R \end{cases}, \begin{cases} x_{12} = x_{11} + w + 5 \\ y_{12} = y_{11} \end{cases}, \dots, \begin{cases} x_{1n} = x_{11} + (n-1)(w+5) \\ y_{1n} = y_{11} \end{cases} \\
 \text{line2: } & \begin{cases} x_{21} = -R + \frac{w+5}{2} \\ y_{21} = R - \frac{w+5}{2}\sqrt{3} \end{cases}, \begin{cases} x_{22} = -R + \frac{w+5}{2} + w + 5 \\ y_{22} = R - y_{21} \end{cases}, \dots, \begin{cases} x_{2n} = -R + \frac{w+5}{2} + (n-1)(w+5) \\ y_{2n} = R - y_{21} \end{cases}
 \end{aligned}$$

根据上式给出蜂窝排列法的正六边形顶点及中心点的坐标，将坐标与圆环范围判断：

$$x_{mn}^2 + y_{mn}^2 \geq 100^2, x_{mn}^2 + y_{mn}^2 \leq 350^2$$

最后得到定日镜的位置坐标，类似于附件。

5.2.2 定日镜场参数的寻优

通过蜂窝排列法确定了圆环内能排列下符合要求的镜面全面积，蜂窝中正六边形的顶点和中心点安排定日镜。因此，确定了定日镜的数目、位置和数量后，接着寻优高度和吸收塔的位置坐标，此时计算量为 $N \times N \times \tilde{h} \times w \times 350 \times 350$ ，计算量大，难以全局遍历寻优。因此，我们采用蒙特卡洛模拟的方法，对 N 块定日镜进行随机抽样，随机选取 10% 的定日镜进行光学效率、单位面积输出热功率等计算，计算 10 次，来预测决策变量的粗略值使得 \bar{P}_s 最大，以此来减少计算量。之后，再进一步减少步长，计算额定功率是否达到 60MW 即可，作为局部最优解。

不同的定日镜位置会影响太阳光的光学效率，因此通过蜂窝排列法得到的定日镜排列是一种初始排列方案。通过二维旋转，将定日镜的位置坐标统一进行旋转 μ ，坐标变换后的坐标与原坐标的关系如下，其中 $\mu \in [0, \pi/6]$ ：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \mu & -\sin \mu \\ \sin \mu & \cos \mu \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

通过蜂窝排列法确定粗排列，且定日镜为正方形，之后通过旋转确定定日镜的位置。



因此，决策变量变为 $\mu, w=v, \tilde{h}, X_0, Y_0$ ，减少了决策变量。而对 μ 进行遍历后，对于确定 $w=v=6, \tilde{h}=4, X_0=0, Y_0=0$ 的情况下，结果影响不大。而对 \tilde{h} 进行遍历后，在确定 $w=v=6, \mu=0, X_0=0, Y_0=0$ 的情况下，其也变化不大。因此，在进行指标影响的研究后，我们需要进一步遍历 $w=v, X_0, Y_0$ ，而 μ, \tilde{h} 取一个在其他决策变量改变的情况下的最佳值，最后得到相应的参数。最后，确定参数后，蒙特卡洛模拟 30%，计算 5 次，取步长为 1.5，光锥束同问题一，得到的结果如下表：

表 5：问题 2 的每月 21 日平均光学效率及输出功率

日期	平均光学效率	平均余弦效率	平均阴影遮挡效率	平均截断效率	单位面积镜面平均输出热功率(kW/m ²)
1 月 21 日	0.596392	0.900449	0.86744	0.878184	0.519412
2 月 21 日	0.604102	0.890944	0.890552	0.873959	0.569552
3 月 21 日	0.597382	0.874067	0.900939	0.86939	0.594206
4 月 21 日	0.583787	0.848828	0.903563	0.87136	0.60082
5 月 21 日	0.573441	0.826211	0.906433	0.875483	0.599229
6 月 21 日	0.569729	0.816672	0.90771	0.878365	0.597893
7 月 21 日	0.575872	0.827351	0.908685	0.875577	0.601676
8 月 21 日	0.587265	0.851509	0.9057	0.871392	0.603707
9 月 21 日	0.601599	0.877273	0.903057	0.869969	0.597065
10 月 21 日	0.609563	0.895238	0.892403	0.875476	0.569916
11 月 21 日	0.603862	0.904399	0.873217	0.879115	0.521385
12 月 21 日	0.59673	0.906584	0.86206	0.879055	0.495599

表 6：问题 2 的年平均光学效率及输出功率表

年平均光学效率	年平均余弦效率	年平均阴影遮挡效率	年平均截断效率	年平均输出热功率(MW)	单位面积镜面年平均输出热功率(kW/m ²)
0.591643667	0.86829375	0.893479917	0.874777083	68.24427914	0.572538333

表 7：问题 2 的设计参数表

吸收塔位置坐标	定日镜尺寸(宽×高)	定日镜安装高度(m)	定日镜总面数	定日镜总面积(m ²)
(0,-250)	6×6	4	3311	119196

题中所述对于额定功率，只需要达到即可。而额定功率的变化，一般需要增加总反射面积，因此，对于需要达到更高的单位面积镜面年平均输出热功率，需要减少定日镜的面数。这样，使得总反射面积减少，同时额定功率减少，但是不安装的定日镜对其他定日镜造成的阴影遮挡损失会减少。因此，在计算遍历计算得到一个满足要求的年平均输出功率后，需要减少遮挡他人的定日镜或阴影遮挡损失较高的定日镜。因



此，此处应该是多目标优化模型，使得年平均输出热功率尽量大的情况下，单位面积镜面年平均输出热功率尽量大。

六、问题三的模型建立与求解

6.1 问题三的模型建立

按设计要求，若定日镜尺寸及安装高度可以不同，设计定日镜场的参数，使得在约束条件下，使得单位镜面面积年平均输出热功率尽量大。因此，建立以下优化模型。

1) 决策变量：吸收塔位置坐标、定日镜尺寸（宽度和高度）、安装高度、定日镜数目、定日镜位置。

吸收塔位置坐标： (X_o, Y_o) ；

定日镜尺寸： w_i, v_i ，其中 w_i 为第 i 面定日镜宽度， v_i 为第 i 面定日镜高度；

安装高度： \tilde{h}_i ，表示第 i 面定日镜的安装高度；

定日镜数目： N ；

定日镜位置： $P_i(x_i, y_i, z_i), z_i = h_i$ 。

2) 目标函数： $\max \{\bar{P}_s\} = \max \left\{ \left[\sum_{i=1}^{12} \bar{E}_{field}^i / (12 * \tilde{S}) \right] \right\}$ ，其中 $\tilde{S} = A_i \cdot N$ ， $A_i = w_i \cdot v_i$ 为第 i 面定日镜的面积。关于阴影遮挡损失和反射光线（束）接收判断的变化，见 5.1 2) 目标函数。

3) 约束条件：

- 吸收塔周围 100m 不安装定日镜，圆形区域半径为 350m：

$$(x_i - X_o)^2 + (y_i - Y_o)^2 \geq 100^2, x_i^2 + y_i^2 \leq 350^2$$

- 镜面宽度不小于镜面高度，镜面边长在 2m 至 8m 之间：

$$w_i \geq v_i, 2 \leq w_i \leq 8, 2 \leq v_i \leq 8$$

- 安装高度在 2m 至 6m 之间，安装高度保证水平转轴转动时不触及地面：

$$2 \leq \tilde{h}_i \leq 6, v_i / 2 < \tilde{h}_i$$

- 相邻定日镜底座中心距离比镜面宽度多 5m：

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} > w_i + 5$$

其中， i, j 表示两个相邻的定日镜，由邻接矩阵计算。

- 达到额定功率：

$$\bar{P}_s \cdot \tilde{S} \geq 60$$



$$\max \{\bar{P}_s\} = \max \left\{ \left[\sum_{i=1}^{12} \bar{E}_{field}^i / (12 * \tilde{S}) \right] \right\}$$

$$\begin{cases} (x_i - X_0)^2 + (y_i - Y_0)^2 \geq 100^2 \\ w_i \geq v_i, 2 \leq w_i \leq 8, 2 \leq v_i \leq 8 \\ 2 \leq \tilde{h}_i \leq 6, v_i / 2 < \tilde{h}_i \\ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} > w_i + 5 \\ x_i^2 + y_i^2 \leq 350^2 \\ \bar{P}_s \cdot \tilde{S} \geq 60 \end{cases} \quad (33)$$

6.2 问题三模型求解

同二问对于每块板的安装高度和尺寸大小可以不同，我们考虑使用第二问的结果进行进一步优化，远离吸收塔的最外围定日镜进行安装高度升高为 6m，其他保持不变。最粗优化，然后通过更细的优化，即增加北方的定日镜高度，而减小南方定日镜的安装高度，使得各个定日镜被太阳照射时造成的阴影遮挡损失尽量小。类似于如下图的排列方式，使得前镜的阴影投射后镜的镜面下。

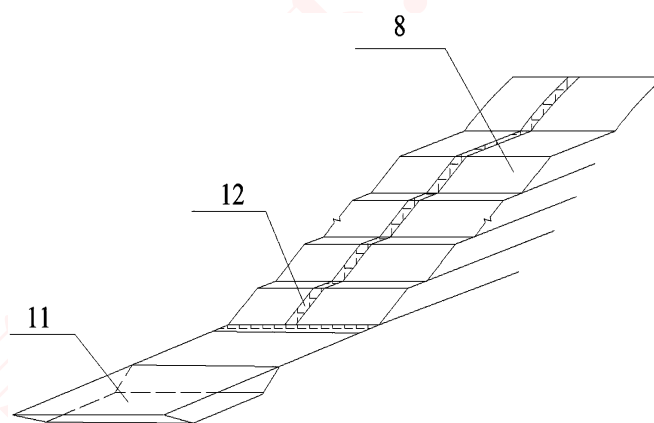


图 11：斜坡式排列

而本文，通过增长最外圈的定日镜高度，提高单位面积镜面平均输出热功率。

表 8：问题 3 的每月 21 日平均光学效率及输出功率

日期	平均光学效率	平均余弦效率	平均阴影遮挡效率	平均截断效率	单位面积镜面平均输出热功率(kW/m ²)
1 月 21 日	0.501243	0.837242	0.868675	0.799767	0.446412
2 月 21 日	0.502673	0.835808	0.891524	0.778916	0.493947
3 月 21 日	0.500066	0.830134	0.905623	0.768468	0.517516
4 月 21 日	0.492968	0.818982	0.908644	0.765327	0.527452
5 月 21 日	0.489248	0.807669	0.91011	0.768716	0.531351
6 月 21 日	0.487423	0.802654	0.91105	0.769805	0.551611
7 月 21 日	0.489293	0.807824	0.91007	0.768672	0.546321



8月21日	0.493269	0.819588	0.908398	0.765471	0.537681
9月21日	0.500524	0.830561	0.905477	0.768889	0.516838
10月21日	0.502911	0.836184	0.888973	0.781334	0.490196
11月21日	0.500548	0.837219	0.866258	0.801138	0.462046
12月21日	0.496971	0.836919	0.856283	0.806654	0.452938

表 9：问题 3 的年平均光学效率及输出功率表

年平均光学效率	年平均余弦效率	年平均阴影遮挡效率	年平均截断效率	年平均输出热功率 (MW)	单位面积镜面年平均输出热功率 (kW/m ²)
0.496428083	0.825065333	0.894257083	0.778596417	60.336111	0.506192417

表 10：问题 3 的设计参数表

吸收塔位置坐标	定日镜尺寸 (宽×高)	定日镜安装高度 (m)	定日镜总面数	定日镜总面积 (m ²)
(0,-250)	/	/	3311	119196

七、模型的评价和推广

7.1 模型的优点

在计算时，简化计算，并考虑了锥形束的入射、反射和阴影遮挡损失。

7.2 模型的缺点

在遍历时，取的步长对结果影响大，由于计算量的问题，取得步长过大，使得结果不够精确。

7.3 模型的推广

可将模型推广到定日镜场的优化设计和安排策略。

八、参考文献

- [1] 张平等，太阳能塔式光热镜场光学效率计算方法[J]，技术与市场，2021，28(6):5-8.
- [2] 杜宇航等，塔式光热电站定日镜不同聚焦策略的影响分析[J]，动力工程学报，2020，40(5):426-432.
- [3] <https://baike.baidu.com/item/%E5%A4%AA%E9%98%B3%E9%AB%98%E5%BA%A6%E8%A7%92/1563831>
- [4] <https://baike.baidu.com/item/%E5%A4%AA%E9%98%B3%E6%96%B9%E4%BD%8D%E8%A7%92/2772684>
- [5] 孙浩. 基于混合策略鲸鱼优化算法的定日镜场布局研究及优化[D]. 兰州交通大学, 2023.



附录

本文采用 Python 进行编程。

问题一

```
import numpy as np
import pandas as pd
from math import cos,sin,acos,asin,pi,exp,sqrt
import time

# 导入题目的附件 并添加 Z 值
P = pd.read_excel(r'附件.xlsx').values
Dis = np.load('distance.npy')

# D_0 = [306,337,0,31,61,92,122,153,184,214,245,275]
ST_0 = [9,10.5,12,13.5,15]

D_0=[337,0,31,61,92];
# ST_0 = [9]
delta_t = 1.5
x_c=0; y_c=0; L=W=6;
H0 = 80;H1=8;H = 4;
HR = 3.5

def F(L,W,H,D_0,ST_0,delta_t,H0,H1,HR,Dis,p1,x_c,y_c):
    N = (W/delta_t+1)**2*4
    yita_ref = 0.92
    S = L*W

    new_column = np.array([H for i in range(len(p1))]) # 安装高度 4m
    P = np.column_stack((p1, new_column))

    Yita =np.zeros([len(ST_0),len(P)]);
    Yita_trunc =np.zeros([len(ST_0),len(P)])
    Yita_at =np.zeros([len(ST_0),len(P)]);
```



```

Yita_cos=np.zeros([len(ST_0),len(P)]);
Yita_sb =np.zeros([len(ST_0),len(P)])
E_0 = np.zeros([len(D_0),len(ST_0)])

Y1=np.zeros([len(D_0),len(ST_0)]);
Y2=np.zeros([len(D_0),len(ST_0)]);
Y3=np.zeros([len(D_0),len(ST_0)]);
Y4=np.zeros([len(D_0),len(ST_0)]);

DF = np.zeros([len(D_0),5])

for di,D in enumerate(D_0):
    for sti,ST in enumerate(ST_0):
        # 计算太阳位置 以及相关参数
        fai = 39.4*pi/180;
        delta = asin(sin(2*pi*D/365)*sin(2*pi*23.45/360))
        w = (pi/12)*(ST-12)

        sin_as = cos(delta)*cos(fai)*cos(w)+sin(delta)*sin(fai) # 太阳高度角
        cos_as = sqrt(1-sin_as**2)

        cos_rs = (sin(delta)-sin_as*sin(fai))/(sqrt(1-sin_as**2)*cos(fai)) # 太阳方位角
        if abs(cos_rs)>1:
            cos_rs = cos_rs/abs(cos_rs)

        if ST <=12:
            sin_rs = sqrt(1-cos_rs**2)
        else:
            sin_rs = -sqrt(1-cos_rs**2)

        a = 0.4237 - 0.00821*(6-3)**2
        b = 0.5055 + 0.00595*(6.5-3)**2
        c = 0.2711 + 0.01858*(2.5-3)**2
        DNI = 1.366*(a+b*exp(-c/sin_as))

```



```

A0=np.array([x_c,y_c,H0]) # 集热器中心
Ls = np.array([-cos_as*sin_rs, -cos_as*cos_rs, -sin_as]) # 入射
s = np.array([1,0,0])

for i in range(len(P)):
    # A 镜中点 反射向量
    Di = P[i]; Lr = A0-Di
    nl = -Ls + Lr/np.linalg.norm(Lr) # A 的法向量
    nl = nl/np.linalg.norm(nl)

    beta1 = asin(nl.dot([0,0,1])/np.linalg.norm(nl)) # 俯仰角

    n0 = np.array([nl[0],nl[1],0])
    if nl[1]>=0:
        alpha1 = acos(n0.dot(s)/np.linalg.norm(n0)) # 方位角
    else:
        alpha1 = -acos(n0.dot(s)/np.linalg.norm(n0))

    # A 的旋转矩阵
    Ta = np.array([
        [cos(alpha1)*cos(pi/2-beta1),-sin(alpha1),cos(alpha1)*sin(pi/2-beta1)],
        [sin(alpha1)*cos(pi/2-beta1),cos(alpha1),sin(alpha1)*sin(pi/2-beta1) ],
        [-sin(pi/2-beta1),0,cos(pi/2-beta1)]
    ])

    light=0; empty=0;
    barr_tower=0; barr_s=0; barr_r=0
    # 遍历每一个点
    for dx in np.arange(-W/2,W/2+0.1,delta_t):
        for dy in np.arange(-L/2,L/2+0.1,delta_t):

            Dxy = np.array([dx,dy,0]) # A 镜上的某个点 在 A 镜坐标系
            Di_d =Ta.dot(Dxy) + Di # A 镜上的点 转置到地面坐标系

```



线

```

##### 遍历每一个入射光线圆锥的光线 #####
# for the1 in np.arange(0.001,0.00465,0.002):
the1 = 0.002
for the2 in np.arange(0,2*pi,pi/2):
    if _barr=0

# g 是入射光线在主光线锥体系的坐标
g = np.array([sin(the1)*cos(the2),
              sin(the1)*sin(the2),
              cos(the1)])

# 根据入射主光线，计算主光线锥体系的旋转矩阵 Ls:入射的太阳主光

v = pi/2 -acos(Ls.dot(np.array([0,0,1]))/np.linalg.norm(Ls))
nl_g0 = np.array([Ls[0],Ls[1],0])
if Ls[1]>=0:
    u = acos(nl_g0.dot(s)/np.linalg.norm(nl_g0))
else:
    u = -acos(nl_g0.dot(s)/np.linalg.norm(nl_g0))

T_s = np.array([
    [cos(u)*cos(pi/2-v),-sin(u),cos(u)*sin(pi/2-v)],
    [sin(u)*cos(pi/2-v),cos(u),sin(u)*sin(pi/2-v) ],
    [-sin(pi/2-v),0,cos(pi/2-v)]
])
g_d = T_s.dot(g) # 转置到地面坐标系 g:入射光锥中的某条入射线
g_r = g_d - 2*g_d.dot(nl)*(nl) # 对应的反射向量 # nl:A 的法向量
##### 一、判断入射光线是否被塔遮挡 #####
a,b,c = g_d;x0,y0,z0 = Di_d
delta_tower = 4*(a*(x0-x_c)+b*(y0-y_c))**2-4*(a**2+b**2)*((x0-x_c)**2+(y0-y_c)**2-HR**2)
if delta_tower>=0:
    t1 = (-2*(a*(x0-x_c)+b*(y0-y_c))+sqrt(delta_tower))/(2*(a**2+b**2))
    t2 = (-2*(a*(x0-x_c)+b*(y0-y_c))-sqrt(delta_tower))/(2*(a**2+b**2))

```



```

if min(t1*c+z0,t2*c+z0)<= (H0 +H1/2) and min(t1*c+z0,t2*c+z0)>=0 :
    barr_tower += 1
    continue

```

二、判断入射光线是否被其他光镜遮挡
#####

```

indices_in_circle = np.where(Dis[:,i] == 1)[0] # 取周围半径
for j in indices_in_circle: #len(P)
    if i==j:
        continue
    # B 的中心坐标 alpha,beta 直接调取提前计算的
    B = P[j]; Lrb = A0-B
    nlb = -Ls + Lrb/np.linalg.norm(Lrb) # 法向量
    # beta2, alpha2= all_alpha_beta[j,:]

    beta2 = asin(nlb.dot([0,0,1])/np.linalg.norm(nlb)) # 俯仰角

    n0b = np.array([nlb[0],nlb[1],0]);
    if nlb[1]>=0:
        alpha2 = acos(n0b.dot(s)/np.linalg.norm(n0b)) # 方位角
    else:
        alpha2 = -acos(n0b.dot(s)/np.linalg.norm(n0b))

    Tb = np.array([
        [cos(alpha2)*cos(pi/2-beta2),-sin(alpha2),cos(alpha2)*sin(pi/2-beta2)],
        [sin(alpha2)*cos(pi/2-beta2),cos(alpha2),sin(alpha2)*sin(pi/2-beta2) ],
        [-sin(pi/2-beta2),0,cos(pi/2-beta2)]
    ])

    Di_b = Tb.T.dot(Di_d-B) # A 镜上的点 从地面坐标系 → B 镜坐标系
    g_b = Tb.T.dot(g_d) # A 入射光线 从地面坐标系 → B 镜坐标系

    t = -Di_b[2]/g_b[2] # 计算入线光线的在 B 镜坐标系的交点

```




```

x_b = g_b[0]*t + Di_b[0]
y_b = g_b[1]*t + Di_b[1]

D0 = np.array([x_b,y_b,0])
D0 = Tb.dot(D0)+B    # 交点转到地面

if abs(x_b)<=W/2 and abs(y_b)<=L/2 and D0[2]>Di_d[2]: # 如果被遮挡
了,就直接算下一条入射的线
    barr_s += 1
    if_barr=1
    break

##### 三、对应的反射 #####
g_r = g_d - 2*g_d.dot(nl)*(nl) # 对应的反射向量 # nl:A 的法向量
g_r_b = Tb.T.dot(g_r)    # 转置到 b 镜面坐标系

t = -Di_b[2]/g_r_b[2]    # 计算反射光线的交点
x_b = g_r_b[0]*t + Di_b[0]
y_b = g_r_b[1]*t + Di_b[1]

D0 = np.array([x_b,y_b,0])
D0 = Tb.dot(D0)+B    # 交点转到地面
if abs(x_b)<=W/2 and abs(y_b)<=L/2 and D0[2]>Di_d[2]:
    barr_r += 1
    if_barr =1
    break

##### 四、是否吸收 #####
if if_barr == 0:
    a,b,c = g_r,x0,y0,z0 = Di_d
    delta_recieve = 4*(a*(x0-x_c)+b*(y0-y_c))**2-4*(a**2+b**2)*((x0-
x_c)**2+(y0-y_c)**2-HR**2)
    if delta_recieve>= 0:
        t1 = ( -2*(a*(x0-x_c)+b*(y0-y_c))+sqrt(delta_recieve) )/
(2*(a**2+b**2))

```



```

t2 = ( -2*(a*(x0-x_c)+b*(y0-y_c))-sqrt(delta_recieve) )/
(2*(a**2+b**2))
if min(t1*c+z0,t2*c+z0)<= (H0+H1/2) and
min(t1*c+z0,t2*c+z0)>=(H0-H1/2):
    light += 1
else:
    empty += 1

# 这里是这个时间点,计算第 i 个面板的数值,并记录,每个列表是 1745 的长度
yita_sb = 1 - (barr_r + barr_s + barr_tower)/N;
yita_cos = abs(Ls.dot(-nl)/np.linalg.norm(Ls));
HR0 = np.linalg.norm(Lr);
yita_at = 0.99321 - 0.0001176*HR0 + 1.97e-8*(HR0**2);

if N - barr_s - barr_r - barr_tower == 0:
    yita_trunc = 1
else:
    yita_trunc = (light)/(N - barr_s - barr_r - barr_tower)

yita = yita_sb*yita_cos*yita_at*yita_trunc*yita_ref;

Yita_sb[sti,i] = yita_sb
Yita_cos[sti,i] = yita_cos
Yita_at[sti,i] = yita_at
Yita_trunc[sti,i] = yita_trunc
Yita[sti,i] = yita

# 这里是在 D,ST 的循环里, 计算第 sti 个时间点
E_0[di,sti] = DNI*sum(S*Yita[sti,:])

Y1[di,sti] = np.mean(Yita[sti,:])
Y2[di,sti] = np.mean(Yita_cos[sti,:])
Y3[di,sti] = np.mean(Yita_sb[sti,:])
Y4[di,sti] = np.mean(Yita_trunc[sti,:])

```



```

# 已经计算完一个具体时间点
DF[di,0] = np.mean(Y1[di,:])
DF[di,1] = np.mean(Y2[di,:])
DF[di,2] = np.mean(Y3[di,:])
DF[di,3] = np.mean(Y4[di,:])
DF[di,4] = sum(E_0[di,:])/(len(P)*S)/len(ST_0)

# print(DF)

return DF

start_time = time.time()

Wp = F(L,W,H,D_0,ST_0,delta_t,H0,H1,HR,Dis,P,x_c,y_c)
print(Wp)

# 记录程序运行时间
end_time = time.time()
elapsed_time_seconds = end_time - start_time
elapsed_time_minutes = round(elapsed_time_seconds / 60, 2)
print(f'代码运行时间: {elapsed_time_minutes}分钟')

```

问题二

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
from math import cos,sin,acos,asin,pi,exp,sqrt,floor
import time
from Function import F

# 导入题目的附件 并添加 Z 值
p1 = pd.read_excel(r'D:\数学建模学习\2023 国赛\数据文件\
r6_raw.xlsx',header=None).values

HR = 3.5; H = 4; L = W = 6

```



```

H0 = 80; H1=8;

# 导入每块板 在某时刻的 俯仰角与方位角
# D_0 = [306,337,0,31,61,92,122,153,184,214,245,275]
# ST_0 = [9,10.5,12,13.5,15]
D_0=[275]; ST_0=[12]

# 旋转角度
for gama in np.linspace(0,pi/3+0.01,4):
    TT = np.array([
        [cos(gama),-sin(gama)],
        [sin(gama),cos(gama)]
    ])
    p1 = TT.dot(p1.T).T
    p1 = p1

    # Dis0 = np.ones([len(p1),len(p1)])
    # for i in range(len(p1)):
    #     for j in range(len(p1)):
    #         distance = np.linalg.norm(p1[i,:]-p1[j,:])
    #         if distance<20:
    #             Dis0[i,j] =1
    #         else:
    #             Dis0[i,j] =0

# np.save('D:/数学建模学习/2023 国赛/数据文件/dist_tran_r6_raw', Dis0)
Dis0 = np.load('D:/数学建模学习/2023 国赛/数据文件/dist_tran_r6_raw.npy')

xy_len = range(-250,251,50);delta_t =1
EP=np.zeros([len(xy_len),len(xy_len)])
# x_c=0; y_c=-100
for m,x_c in enumerate(xy_len):
    for n, y_c in enumerate(xy_len):
        # x_c=0; y_c=250
        if x_c**2+ y_c**2 > 250**2:

```



```

        continue

    EP0=0
    Pi = [];
    # 判断 r=100
    for i in range(len(p1)):
        dis = sqrt((p1[i,0]-x_c)**2+(p1[i,1]-y_c)**2)
        if dis >= 100:
            Pi.append(i)
    p1 = p1[Pi]

    # 生成新的连接矩阵
    Dis1 = np.ones([len(Pi),len(Pi)])
    for i in range(len(Pi)):
        for j in range(len(Pi)):
            Dis1[i,j] = Dis0[Pi[i],Pi[j]]

    m_n=floor(0.1*len(p1))
    for k in range(1,3):
        random_numbers = random.sample(list(range(len(p1))), m_n)
        p11 = p1[random_numbers]
        Dis = np.ones([m_n,m_n])
        for i in range(m_n):
            for j in range(m_n):
                Dis[i,j] = Dis0[random_numbers[i],random_numbers[j]]

        Wp = F(L,W,H,D_0,ST_0,delta_t,H0,H1,HR,Dis,p11,x_c,y_c)
        EP0 += np.mean(Wp)*W*L*len(p1)/1000

    print(x_c,y_c,EP0/3)
    EP[m,n] = EP0

```

```

import numpy as np
import pandas as pd

```



```

import matplotlib.pyplot as plt
import random
from math import cos,sin,acos,asin,pi,exp,sqrt,floor
import time
from Function_2 import F

# 导入题目的附件 并添加 Z 值
p1 = pd.read_excel('r6_raw.xlsx',header=None).values
Dis0 = np.load('dist_tran_r6_raw.npy')

HR = 3.5; H = 4; L = W = 6
H0 = 80; H1=8;
delta_t = 1.5 # 步长

# 导入每块板 在某时刻的 俯仰角与方位角
# D_0 = [306,337,0,31,61,92,122,153,184,214,245,275]
ST_0 = [9,10.5,12,13.5,15]
D_0 = [306]
# ST_0 = [9]
time.time()

x_c=0; y_c=-250
Pi=[]
start_time = time.time()

# 判断 r=100
for i in range(len(p1)):
    dis = sqrt((p1[i,0]-x_c)**2+(p1[i,1]-y_c)**2)
    if dis >= 100:
        Pi.append(i)
p1 = p1[Pi]

Dis1 = np.ones([len(Pi),len(Pi)])
# 生成新的连接矩阵

```




```

for i in range(len(Pi)):
    for j in range(len(Pi)):
        Dis1[i,j] = Dis0[Pi[i],Pi[j]]

m_n=floor(0.3*len(p1))
random_numbers = random.sample(list(range(len(p1))), m_n)
p11 = p1[random_numbers]
Dis = np.ones([m_n,m_n])
for i in range(m_n):
    for j in range(m_n):
        Dis[i,j] = Dis0[random_numbers[i],random_numbers[j]]

Wp = F(L,W,H,D_0,ST_0,delta_t,H0,H1,HR,Dis,p11,x_c,y_c)

print(Wp)
# 记录程序运行时间
end_time = time.time()
elapsed_time_seconds = end_time - start_time
elapsed_time_minutes = round(elapsed_time_seconds / 60, 2)
print(f'代码运行时间: {elapsed_time_minutes}分钟')

# 0.24874209*36*3311

```

Matlab 程序

```

clc; clear;

k = 2;          % 计数
R = 350;
r = 11;         % 宽+5
n = fix(2*R/r); % 一排的点数

% 第一排第一个点
W = [-R, R];
W1 = W;

```



```

% 计算第一排
for i = 1:n
    W(k, 1) = W (k-1, 1)+r;
    W(k, 2) = W (1, 2);
    k = k+1;
end

% 第二排第一个点
W(k, 1) = -R+r/2;
W(k, 2) = R-r*sqrt(3)/2;
W2 = W(k, :);
k = k+1;

% 计算第二排
for i = 1:n
    W(k, 1) = W (k-1, 1)+r;
    W(k, 2) = W (k-1, 2);
    k = k+1;
end

n1 = fix(2*R/(r*sqrt(3)/2))+1;
% 从第三排开始
for i = 3:n1
    if mod(i,2) == 1 % 奇数行
        % 先写该行的第一个点
        W(k, 1) = W1(1, 1);
        W(k, 2) = W1(1, 2)-sqrt(3)*r*(i-1)/2;
        k = k+1;

        for j = 1:n
            W(k, 1) = W (k-1, 1)+r;
            W(k, 2) = W (k-1, 2);
            k = k+1;
        end
    end
end

```



```

else % 偶数行
    % 先写该行的第一个点
    W(k, 1) = W2(1, 1);
    W(k, 2) = W2(1, 2)-sqrt(3)*r*(i-2)/2;
    k = k+1;

    for j = 1:n
        W(k, 1) = W (k-1, 1)+r;
        W(k, 2) = W (k-1, 2);
        k = k+1;
    end
end
end

% 计算范围内点
a = 0;
for i = 1:length(W)
    if W(i,1)^2+W(i,2)^2 <= (350-(r-5)/2)^2 %& W(i,1)^2+W(i,2)^2 >= 100^2
        a = a+1;
        W_e(a,:) = W(i,:);
    end
end

scatter(W_e(:,1),W_e(:,2),'r')

T = [cos(pi/4), -sin(pi/4);
     sin(pi/4), cos(pi/4)
];

WW = (T*W)';

% 计算点距离
for i = 1:length(W)
    for j = i:length(W)

```



```

    D(i,j) = sqrt((WW(i,1)-WW(j,1))^2+(WW(i,2)-WW(j,2))^2);
    D(j,i) = D(i,j);
end
end

```

问题三

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
from math import cos,sin,acos,asin,pi,exp,sqrt,floor
import time
from Function_3 import F

p1 = pd.read_excel(r'D:\数学建模学习\2023 国赛\数据文件\
r6_raw.xlsx',header=None).values
Dis0 = np.load('D:/数学建模学习/2023 国赛/数据文件/dist_tran_r6_raw.npy')

R_dis = [np.linalg.norm(i) for i in p1]

new_column = np.array([4 for i in range(len(p1))]) # 安装高度 4m
P = np.column_stack((p1, new_column))

# 筛选北边的最外层
r_1 = 340; r_2 = 351
Ai=[]
for i in range(len(P)):
    if R_dis[i] >= r_1 and p1[i,1]>0:
        Ai.append(i)
    P[i,2] = P[i,2] + 2

```



```
x_c=0; y_c=-250
```

```
Pi=[]
```

```
for i in range(len(P)):
```

```
    dis = sqrt((p1[i,0]-x_c)**2+(p1[i,1]-y_c)**2)
```

```
    if dis >= 100:
```

```
        Pi.append(i)
```

```
PP = P[Pi]
```

```
## 计算新的邻接矩阵，计算一次后即存储
```

```
# Dis0 = np.ones([len(PP),len(PP)])
```

```
# for i in range(len(PP)):
```

```
#     for j in range(len(PP)):
```

```
#         distance = np.linalg.norm(PP[i,:]-PP[j,:])
```

```
#         if distance<20:
```

```
#             Dis0[i,j] =1
```

```
#         else:
```

```
#             Dis0[i,j] =0
```

```
# np.save('D:/数学建模学习/2023 国赛/数据文件/dist_tran_r6_raw_3', Dis0)
```

```
np.save('D:/数学建模学习/2023 国赛/数据文件/position_3',PP)
```

```
Dis0 = np.load('D:/数学建模学习/2023 国赛/数据文件/dist_tran_r6_raw_3.npy')
```

```
###
```

```
D_0=[306]; ST_0=[9]
```

```
HR = 3.5;H = 4; L = W = 6
```

```
H0 = 80; H1=8; delta_t = 3 # 步长
```

```
PW = F(L,W,H,D_0,ST_0,delta_t,H0,H1,HR,Dis0,PP,x_c,y_c)
```

```
print( np.mean(PW)*W*L*len(PP)/1000 )
```

```
###
```

```
plt.scatter(p1[:,0], p1[:,1],c='b')
```

```
plt.scatter(p1[Ai,0],p1[Ai,1],c='r')
```



```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
from math import cos,sin,acos,asin,pi,exp,sqrt,floor
import time
from Function_3 import F

# 导入题目的附件 并添加 Z 值
p = np.load('D:/数学建模学习/2023 国赛/数据文件/position_3.npy')
Dis0 = np.load('D:/数学建模学习/2023 国赛/数据文件/dist_tran_r6_raw_3.npy')

HR = 3.5;H = 4; L = W = 6
H0 = 80; H1=8; delta_t = 3 # 步长

# 导入每块板 在某时刻的 俯仰角与方位角
D_0 = [306,337,0,31,61,92,122,153,184,214,245,275]
ST_0 = [9,10.5,12,13.5,15]
# D_0 = [306]
# ST_0 = [9]
x_c=-250; y_c=-150

m_n=floor(0.3*len(p))
random_numbers = random.sample(list(range(len(p))), m_n)
p11 = p[random_numbers]
Dis = np.ones([m_n,m_n])
for i in range(m_n):
    for j in range(m_n):
        Dis[i,j] = Dis0[random_numbers[i],random_numbers[j]]

```




```
start_time= time.time()
PW = F(L,W,H,D_0,ST_0,delta_t,H0,H1,HR,Dis,p11,x_c,y_c)

end_time = time.time()
elapsed_time_seconds = end_time - start_time
elapsed_time_minutes = round(elapsed_time_seconds / 60, 2)
print(f'代码运行时间： {elapsed_time_minutes}分钟')

print(PW)
# print(np.mean(PW)*W*L*len(p)/1000 )
```

