

# 定日镜场优化设计模型

## 摘要

塔式定日镜场是一种利用定日镜和集热器实现太阳能光热发电的系统。它采用大量的定日镜组成阵列，将太阳光聚焦到集热器上，使集热器中的工质被加热并转化为蒸汽，进而驱动发电机产生电能。该题目涉及到新能源电力系统中塔式太阳能光热发电的优化设计问题。通过构建圆形定日镜场，将太阳光通过定日镜反射汇聚到集热器上，实现太阳能向热能的转化，并最终转化为电能。

**针对问题一**，需先求得大气透射率 $\eta_{at}$ ，先求得每个定日镜到集热器中心的欧几里得距离 $d_{HR}$ ，进而根据附件公式 4 求得透射率；通过利用太阳位置算法(SPA)来跟踪太阳，根据附件公式 2 求得法向直接辐射辐照度  $DNI$ (单位: $kW/m^2$ )；依附件公式 1 求得太阳方位角、高度角、大气透射率 $\eta_{at}$ 等与所对应日期和时间对应关系；运用几何关系和向量知识并查阅相关资料推理计算出给定日期定日镜的余弦损失及余弦效率 $\eta_{cos}$ ；为方便计算阴影遮挡效率 $\eta_{sb}$ ，假定塔影造成了 10% 的能量损失；对集热器截断效率采取几何求解思路，依据镜面宽度 $W$ ，镜面高度 $H$ ，集热塔的宽度(直径) $W_r$ ，高度 $H_r$ 等条件进行求解；最后分别带入原函数关系，光学效率各参数之间关系的数学模型，所得结果进一步根据附件公式 3 求得定日镜场的输出热功率  $E_{field}$ 。

**针对问题二**，想要单位镜面面积年平均输出热功率尽量大，应使得  $n_i$  即第*i*面镜子的光学效率尽量大考虑到该问题的结果应该相对有一个固定的区间，使用启发式最优化算法很容易陷入局部最优，应该使用网格搜索算法。由前述的分析，为了最大的平均输出热功率，应该让  $n_i$  越大越好。故在问题 1 确定数据的基础上，分别对每个变量进行调整，观察  $n$  值的变化情况，从而使得搜索空间的范围降低。

**针对问题三**，由于目标都是使得输出热功率尽量大，所以可以沿用问题 2 中确定下来的一些参数。接着针对不同的尺寸与安装高度，建立新的输出热功率计算模型。这里需要考虑一些限制约束条件安装高度必须保证镜面在绕水平转轴旋转时不会触及地面。

**关键词：**定日镜场 布局优化 光学效率

# 一、问题重述

## 1.1 问题背景

“双碳”的目标背景下全球正处于能源结构转型的重要阶段，人类对能源需求的增长和现有能源资源日趋减少的矛盾愈发加剧。构建以新能源为主体的新型电力系统，塔式太阳能光热发电作为一种低碳环保的新型清洁能源技术，定日镜场是塔式光热电站的重要组成部分，不但可以提高对太阳能的利用率，减少较大的阴影和遮挡损失，还为降低投资成本和发电成本提供条件。因此，对定日镜场的优化设计对助力国家能源发展战略具有十分重要的意义。

一种利用定日镜的装置的原理为，在定日镜的反射作用下将太阳直射光以自动聚焦的方式汇聚到安装在镜场中吸收塔顶端上的集热器里，并加热其中的导热介质。并将太阳能以热能形式储存起来，再经过热交换实现由热能向电能的转化。

## 1.2 问题提出

定日镜是塔式太阳能光热发电站收集太阳能的基本组件，定日镜底座由纵向转轴和水平转轴组成，平面反射镜安装在水平转轴上。塔式电站利用大量的定日镜组成阵列，称为定日镜场。控制系统根据太阳的位置实时控制定日镜的法向，使得太阳中心点发出的光线经定日镜中心反射后指向集热器中心。定日镜将太阳光反射汇聚到安装在镜场中吸收塔顶端上的集热器，加热其中的导热介质，并将太阳能以热能形式储存起来，从而能更好的实现其向电能的转化。<sup>[1]</sup>

基于以上背景信息与附件内容，需建立数学模型解决以下问题：

### 问题一：

已知圆形定日镜场位置和大小以及每面定日镜的采光面积，将吸收塔建于该圆形定日镜场中心。控制系统根据太阳的位置实时控制定日镜的法向，使得太阳中心点发出的光线经定日镜中心反射后指向集热器中心。在定日镜场中，太阳方位角与余弦损失成正相关。由于定日镜排列方式的差异，定日镜的背面会产生较大的阴影并造成遮挡损失。建立数学模型，依据附录相关计算公式分别计算阴影遮挡效率、余弦效率、大气透射率、集热器截

断效率、镜面反射率来求出定日镜场的年平均光学效率。在给定的日镜采光面积，需要求法向直接辐射辐照度 DNI，在通过定日镜的光学效率计算输出热功率。

**问题二：**

在给定的定日镜场达到额定电压的条件下，保证定日镜场的额定年平均输出热功率为 60MW。设计好定日镜场相关参数，计算出最大单位镜面面积年平均输出热功率。

**问题三：**

要考虑定日镜场要达到额定功率，在定日镜尺寸和安装高度不同的情况下，设定好吸收塔的位置坐标、各定日镜尺寸、安装高度、位置坐标。通过让定日镜应该始终朝向太阳，就能够更有效地反射太阳光，并随着太阳的运动而调整角度。这可以通过使用跟踪系统或自动控制来完成。确保镜面始终与太阳保持最佳的对准，可以最大限度地减少能量损失，以此来使单位镜面面积年平均输出热功率尽量大。[2]

## 二、问题分析

### 2.1 定日镜光学效率与输出热功率分析

镜场效率包含了镜面的反射率（包括镜面自身的反射率和在大气中对镜面的污染）、镜面与太阳线的夹角导致的余弦损耗、太阳光在传输时的衰减损耗、定日镜间的相互干扰导致的阴影遮挡、太阳光反射到吸热器上的外溢等。

由定日镜可通过两轴旋转（高度角-方位角）跟踪方式，在给定镜场经纬度位置及镜场大小、典型日期（每月 21 日）及每日五个时刻、定日镜数量及尺寸、所有定日镜位置（建立镜场坐标系）等的情况下利用太阳位置算法（SPA）跟踪太阳，建立镜场光学效率的数学模型，分析关于各个参数的变化规律，依旧附件公式计算出相关定日镜到集热器中心的距离 $d_{HR}$ 、太阳高度角、太阳方位角、太阳时角、透射率、法向直接辐射辐照度 DNI ( $\text{kW}/\text{m}^2$ )。对于余弦效率的计算求解，查阅相关资料求解其余弦损失即可求得，为方便及计算，阴影遮挡损失和阴影遮挡效率，假设当太阳方位角在 170 至 190 度之间时，塔影造成 10% 的损失能量，简算出集热器接收能量对应的截断效率。据此，我们方可求解出输出热功率。

### 2.2 问题二分析：

问题二是一个优化问题，对定日镜进行优化设计，将定日镜面朝向太阳的角度调整到最佳位置，以最大化接收太阳辐射能量；使用高反射率的镜面材料，以提高定日镜面的反射效率，最大限度地减少能量损失才可以使的单位镜面面积年平均输出尽量大的热功率。在设计定日镜场时，在额定年平均输出热功率的条件下，定日镜的面积越小，则平均输出热功率尽量大。此间需建立单位镜面面积年平均输出热功率优化模型。吸收塔的位置坐标、定日镜尺寸、安装高度、定日镜数目、定日镜位置这几个成为变量，可以用控制变量法逐个优化，然后找到最优单位镜面面积年平均输出热功率。考虑几个变量之间的关系，定日镜的尺寸和排列方式确定后，大致就可以确定定日镜的数目、位置、安装高度等问题。可以使用遗传算法、蚁群算法等优化算法直接求局部最优解。[3]

## 2.3 问题三分析：

改进单位镜面面积年平均输出热功率优化模型。增加了定日镜镜面尺寸、位置、高度为变量，合并问题 2 中的几个变量，此时成为了一个多变量非线性优化问题，可以使用遗传算法、蚁群算法等优化算法直接求局部最优解。定吸收塔的位置坐标同问题 2 相同，通常吸收塔位于定日镜场的几何中心。<sup>[4]</sup>并多目标优化，需要通过优化算法(例如，遗传算法、模拟退火等)来得到额定年平均输出热功率达到 60M, 同时平均输出热功率最大。计算定日镜数目和位置在确定了最优的单个定日镜的面积和安装高度后，计算定日镜的数目和位置。位置的计算可能需要考虑如何最大化光线到达吸收塔的效率。

## 三、模型假设

- 1、假设计算时间年份为 2023 年
- 2、光束是均匀且为平行光
- 3、定日镜场为圆形，定日镜场采用径向交错的布置方案，以中心塔为圆心进行周向均匀布置。为了便于突出太阳位置对阴影遮挡效率的影响，镜场中的定日镜以不发生机械碰撞为原则，建立密集型的仿真镜场
- 4、镜场平面为理想的水平面，且所有定日镜的立柱高度一致。所有定日镜规格一致且为矩形，定日镜均采用高度角-方位角跟踪方式，并假定定日镜镜面为理想平面。
- 5.不考虑天气、空气等对定日镜场的影响

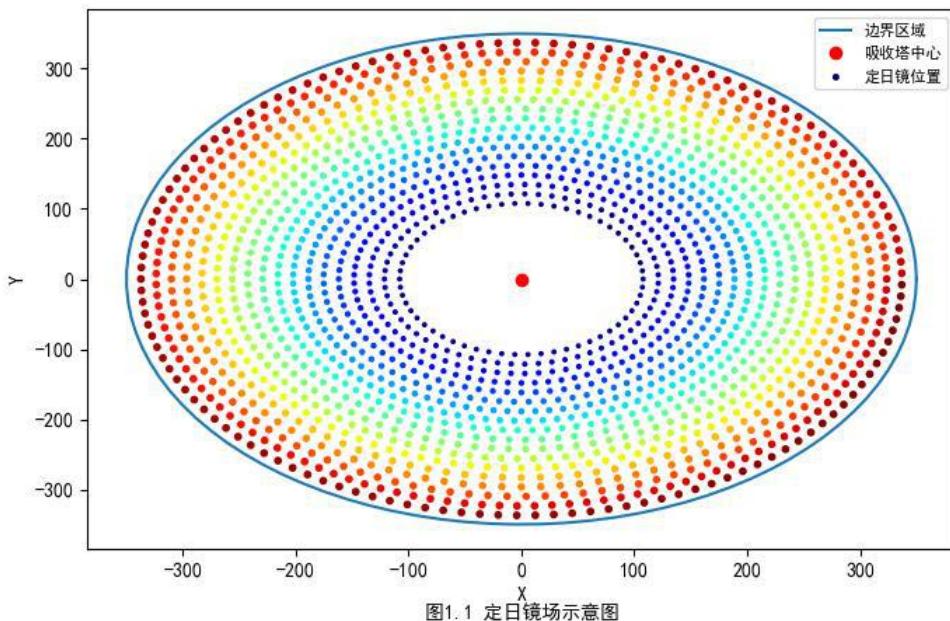
## 四、符号说明

1.  $\alpha_s$ : 太阳高度角 (以度为单位)
2.  $\gamma_s$ : 太阳方位角 (以度为单位)
3.  $\varphi$ : 当地纬度, 北纬为正 ( $\varphi = 39.4^\circ$ )
4.  $\omega$ : 太阳时角,  $\omega = \pi(ST - 12)$ , 其中  $ST$  为当地时间的标准化值
5.  $\delta$ : 太阳赤纬角 (以度为单位)
6.  $D$ : 从春分开始算起的天数 (以天为单位)
7.  $H$ : 海拔高度 (以  $km$  为单位)
8.  $G_0$ : 太阳常数, 表示太阳辐射在地球大气层外的能量流密度
9.  $a$ 、 $b$ 、 $c$ : 太阳辐射模型中的系数
10.  $d_{HR}$ : 水平外大气面直接辐射量
11.  $\theta$  是太阳光线与定日镜法线之间的角度 (以度为单位)
12.  $\vec{S}$  是从太阳到定日镜的单位矢量
13.  $\vec{N}$  是定日镜的单位法线矢量

## 五、模型建立与求解

### 4.1 问题一的模型建立与求解

#### 4.1.1 定日镜到集热器中心的距离



依据附件给定数据，建立图 1.1，未发现有异常点，进而可继续求解。吸收塔位于中央，计算每个定日镜到集热器中心的距离 $d_{HR}$ ，由集热器中心的坐标是 $(x_{HR}, y_{HR}, h)$ ，可以使用欧氏距离公式计算每个镜面中心到集热器中心的距离 $d_{HR}$ （数据与后续计算透射率一并记录）：

$$d_{HR} = \sqrt{(x - x_{HR})^2 + (y - y_{HR})^2 + h^2} \quad (1)$$

其中  $(x_{HR}, y_{HR}) = (0,0)$  为集热塔底部中心， $h$  为集热器中心高度： $h = 80 - 8/2$

#### 4.1.2 给定时刻的太阳时角、高度角和方位角

依附录所给公式 1, 计算 12 个月, 每日的五个时刻(9:00、10:30、12:00、13:30、15:00)的太阳高度角和方位角。为方便计算, 对时间数据进行预处理, 进行时间标准化, 依据公式 1, 可进行求解。

太阳高度角  $\alpha_s$ :

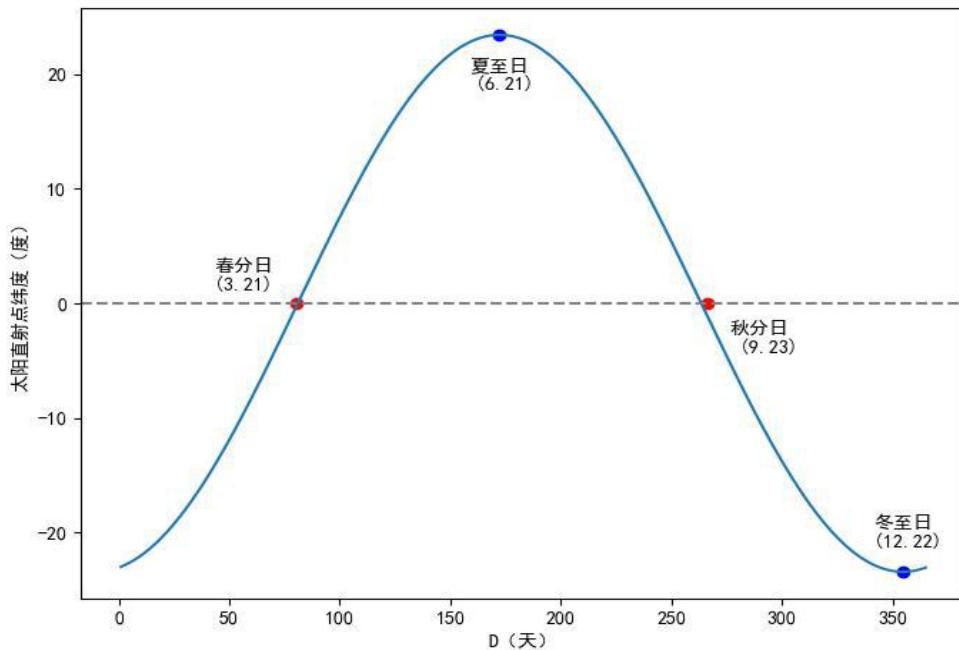
$$\sin\alpha_s = \cos\delta \cos\varphi \cos\omega + \sin\delta \sin\varphi \quad (2)$$

太阳方位角  $\gamma_s$ :

$$\cos\gamma_s = \frac{\sin\delta - \sin\alpha_s \sin\varphi}{\cos\alpha_s \cos\varphi} \quad (3)$$

其中,  $\varphi$  表示当地纬度及北纬  $39.4^\circ$ , 北纬为正;  $\omega$  表示太阳时角,  $\omega = \pi(ST - 12)$ , 其中  $ST$  为当地时间 (标准化后的时间);  $\delta$  表示太阳赤纬角:

$$\sin\delta = \sin\left(\frac{2\pi D}{365}\right) \sin\left(\frac{2\pi}{23.45}\right) \quad (4)$$



其中 $D$ 是从春分开始算起的天数，例如，2023年的春分日是3月21日(图1.2)，则4月8日对应 $D=18$ 。算得典型日期对应的D放到了days\_diff.xlsx。

通过以上公式建立，对于太阳时角 $w$ 进行求解，再进一步求解太阳高度角与太阳方位角，最后对所求数据导出，以便后续计算。

#### 4.1.3 法向直接辐射辐照度 DNI(kW/m<sup>2</sup>)

根据公式2，计算法向直接辐射辐照度 DNI(kW/m<sup>2</sup>)，H=3 代表海拔高度是3000m

$$\begin{cases} DNI = G_0 \cdot (a + b \cdot e^{\frac{-c}{\sin \alpha}}) \\ a = 0.4237 - 0.00821(6 - H)^2 \\ b = 0.5055 + 0.00595(6.5 - H)^2 \\ c = 0.2711 + 0.01858(2.5 - H)^2 \\ H = 3 \\ G_0 = 1.366 \end{cases} \quad (5)$$

表 4.1 典型日时间点信息

日期	标准化时间	正常时间	太阳高度角	太阳方位角	太阳时角	DNI
1 月	9	9:00	17.43091546	135.7753923	135	0.792539684
	10.5	10:30	27.20616975	156.1129528	157.5	0.910091145
	12	12:00	30.83375539	179.9999988	180	0.939184036
	13.5	13:30	27.20616975	156.1129528	202.5	0.910091145
	15	15:00	17.43091546	135.7753923	225	0.792539684
2	9	9:00	24.83077923	130.024897	135	0.887571441
	10.5	10:30	35.7727981	152.3820963	157.5	0.970800037
	12	12:00	39.96981306	179.9999988	180	0.992177192

月

13.5	13:30	35.7727981	152.3820963	202.5	0.970800037
15	15:00	24.83077923	130.024897	225	0.887571441

• • •

6

月	12	12:00	74.04792869	180	180	1.01478034
	13.5	13:30	65.16599791	123.2873526	202.5	0.954822013
	15	15:00	48.92526902	99.13181029	225	0.973887683

9	9:00	46.98556033	103.2271359	135	1.027361103
10.5	10:30	62.68508886	128.4445599	157.5	1.04185403

7

月	12	12:00	70.68941689	180	180	1.027361103
	13.5	13:30	62.68508886	128.4445599	202.5	0.973887683
	15	15:00	46.98556033	103.2271359	225	1.019909118

• • •

1

1

月	12	12:00	30.10047279	180	180	1.019909118
---	----	-------	-------------	-----	-----	-------------

13.5	13:30	26.51434492	156.3854189	202.5	1.025881445
15	15:00	16.829322	136.2141296	225	1.060899872
9	9:00	14.40452957	137.9491957	135	1.070928079
10.5	10:30	23.73117266	157.44804	157.5	1.060899872
1 2 月	12	12:00	27.15575341	180	1.025881445
13.5	13:30	23.73117266	157.44804	202.5	1.019700896
15	15:00	14.40452957	137.9491957	225	1.057117685

#### 4.1.4 大气透射率

计算光学效率将用到,  $d_{HR}$ 前面以求出, 光学效率 $\eta_{at}$ :

$$\eta_{at} = 0.99321 - 0.0001176d_{HR} + 1.97 \times 10^{-8} \cdot d_{HR}^2 \quad (d_{HR} \leq 1000) \quad (6)$$

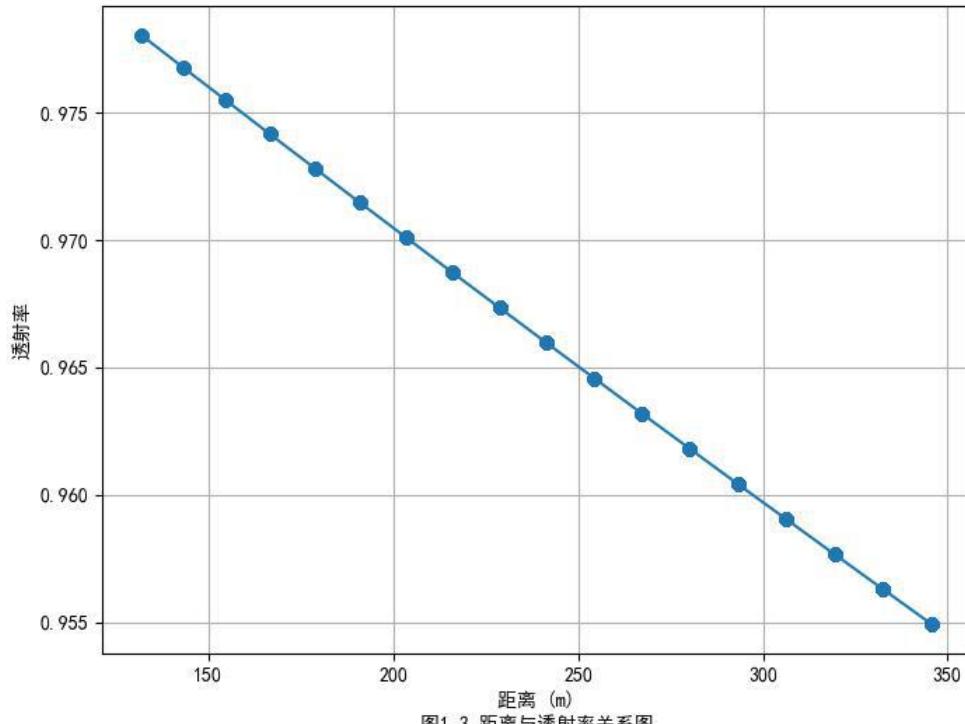


图1.3 距离与透射率关系图

#### 4.1.5 余弦效率

余弦效率描述了太阳光线与定日镜的法线之间的角度，即反射到塔上的光线与定日镜法线之间的角度。建立三维坐标系如图 1.3 所示，其中 x 轴表示正东方向，y 轴表示正南方向，z 轴表示天顶轴（与图 1.1 相同）。在该坐标系中， $QO$  代表接收塔， $Q$  代表吸热器入口面中心， $B$  点表示镜场中某个定日镜的中心位置，其坐标为  $(x_i, y_i, z_i)$ 。 $\vec{S}$  表示  $B$  点处太阳入射光线的方向向量， $\vec{R}$  表示  $B$  点处太阳反射光线的方向向量， $\vec{H}$  表示  $B$  点处定日镜的法线向量。

入射角的大小取决于太阳的位置，而太阳的位置主要由太阳的时角  $\omega$ 、当地的地理纬度  $\varphi$  和太阳的赤纬角  $\delta$  决定。

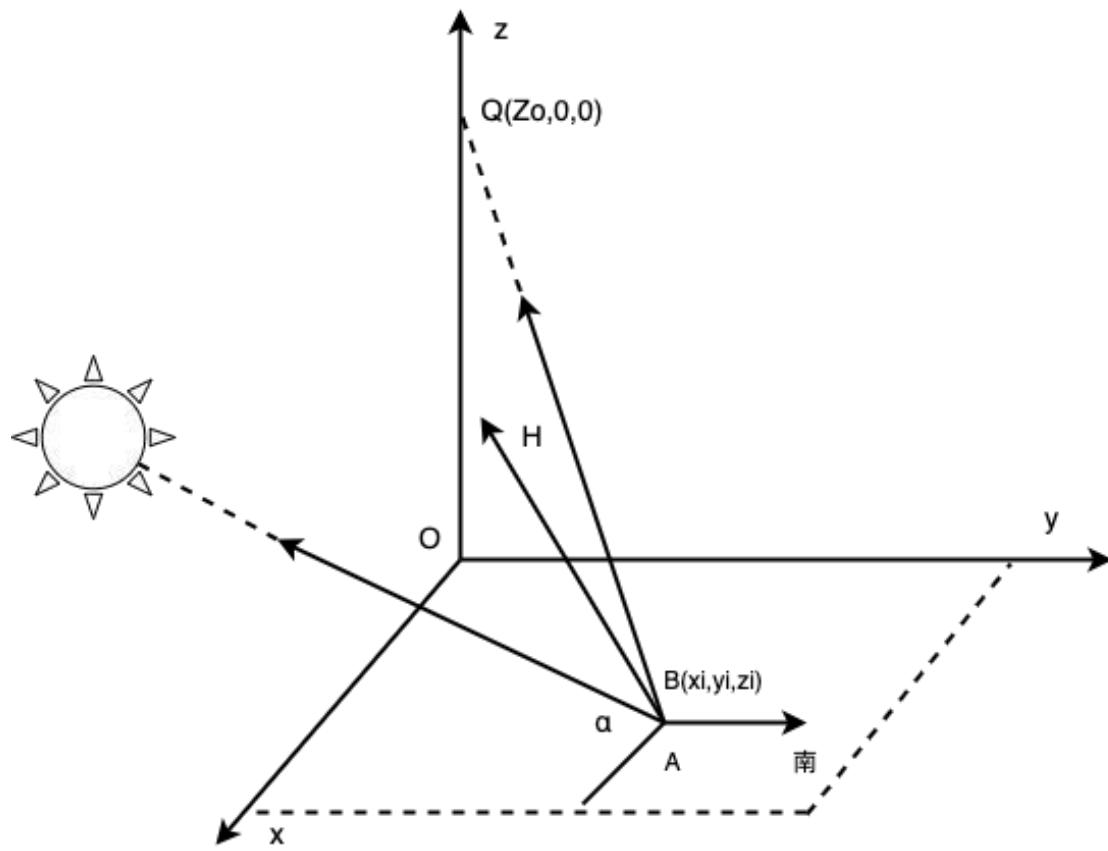


图1.4 太阳入射角示意图

查询相关资料，由以下公式描述：

$$\cos(\theta) = \vec{S} \cdot \vec{N} \quad (7)$$

其中：

$\theta$  是太阳光线与定日镜法线之间的角度。

$\vec{S}$  是从太阳到定日镜的单位矢量。

$\vec{N}$  是定日镜的单位法线矢量。

因此，余弦效率可以描述为：

$$\eta_{cos} = \cos(\theta) \quad (8)$$

$$a_s = \sin^{-1}(\cos\varphi\cos\delta\cos\omega + \sin\varphi\sin\delta) \quad (9)$$

$$\sin A = -\frac{\cos\delta\sin\omega}{\cos a_s} \quad (10)$$

太阳入射光线的方向向量  $\vec{S}$  为:

$$\vec{S} = (\cos\alpha_s \sin A, \cos\alpha_s \cos A, \sin\alpha_s) \quad (11)$$

太阳反射光线的方向向量  $\vec{R}$  为:

$$\vec{R} = \left( \frac{-x_i}{(x_i^2 + y_i^2 + (z_0 - z_1)^2)^{\frac{1}{2}}}, \frac{-y_i}{(x_i^2 + y_i^2 + (z_0 - z_1)^2)^{\frac{1}{2}}}, \frac{(z_0 - z_i)}{(x_i^2 + y_i^2 + (z_0 - z_1)^2)^{\frac{1}{2}}} \right) \quad (12)$$

式中,  $a_s$  为太阳的高度角,  $A$  为太阳的方位角 (4.1.2 已计算)。

利用  $\cos 2\theta_i = \vec{S}^* \vec{R}$ , 并结合公式(11) 和式(12) 可得到:

$$\cos 2\theta_i = \frac{-x_i \cos\alpha_s \sin A - y_i \cos\alpha_s \cos A + (z_0 - z_1) \sin\alpha_s}{(x_i^2 + y_i^2 + (z_0 - z_1)^2)^{\frac{1}{2}}} \quad (13)$$

#### 4.1.6 阴影遮挡效率

计算阴影遮挡损失和阴影遮挡效率, 也就是吸收塔的阴影遮挡到定日镜。阴影遮挡效率是由定日镜阴影效率和遮挡效率的乘积获得的。阴影损失是指目标定日镜的入射光线被周围定日镜遮挡而造成的能力损失, 而遮挡损失指目标定日镜的反射光线被周围定日镜遮挡而造成的损失。在计算阴影遮挡效率时, 首先需要确定可能对目标定日镜产生阴影损失和遮挡损失的其他定日镜在二维平面上的范围, 也称为目标范围。然后, 在目标定日镜的阴影目标范围内搜索除目标定日镜之外的其他定日镜 (问题定日镜), 并确定对目标定日镜造成实际阴影损失的定日镜进行阴影效率的计算。接着, 计算经过阴影损失后定日镜实际接受光线的坐标点, 并在遮挡目标范围内寻找真正对目标定日镜造成遮挡损失的问题定日镜进行遮挡效率的计算。最后, 将目标定日镜的阴影效率与遮挡效率相乘, 得到阴影遮挡效率。

这里简化了问题，假设当太阳方位角在 170 至 190 度之间时，塔影造成 10% 的能量损失。

#### 4.1.7 集热器接收能量

集热器接收能量，根据题目提供的镜面反射率，假设吸收效率是 92%

#### 4.1.8 集热器截断效率

吸热器溢出损失是指定日镜场反射的太阳辐射未能达到吸热器表面而溢出到外界环境导致的能量损失。[6]该损失的大小主要受到以下因素的影响：定日镜光斑的形状和大小、定日镜的跟踪误差和形变误差，以及吸热器受光面的大小。

吸热器截断效率则表示吸热器表面接收到的能量与定日镜反射到吸热器表面的全部能量之比。该指标用于衡量吸热器的光吸收能力，其数值越高表示吸热器能够更有效地捕获和利用太阳辐射能。

$$\eta_{trunc} = \frac{\text{集热器接收能量}}{\text{镜面全反射能量} - \text{阴影遮挡损失能量}} \quad (14)$$

镜面宽度为  $W$ ，镜面高度为  $H$ ，集热塔的宽度（直径）为  $Wr$ ，高度为  $Hr$ 。

如果  $W \leq Wr$  且  $H \leq Hr$ ，那么全部的光束都打在了集热塔吸热器上， $\eta_{trunc} = 1$ 。

如果  $W > Wr$  或  $H > Hr$ ，那么光束的部分被截断。这种情况下，截断效率可以通过以下公式计算：

$$\eta_{trunc} = \frac{\min(W, Wr) \times \min(H, H_r)}{W \times H} \quad (15)$$

#### 4.1.9 定日镜的光学效率

为进行计算光学效率，上述内容已提前计算出光学效率的所有未知参数，依据定日镜光学效率公式：

$$\eta = \eta_{sb} \times \eta_{cos} \times \eta_{at} \times \eta_{trunc} \times \eta_{ref} \quad (16)$$

其中  $\eta_{ref}$  为镜面反射率，此处选定为常数 0.92，所有最终求解结果填入对应表 4.2

表 4.2 问题每月 1 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 <sup>2</sup> )
1月 21 日	0.723558	0.885723	0.98	0.904444	0.86888914
2月 21 日	0.761094	0.931672	0.98	0.940444	0.94178403
3月 21 日	0.778027	0.952446	0.98	0.940444	0.99400121
4月 21 日	0.783806	0.959474	0.98	0.940444	1.00887032
5月 21 日	0.779268	0.953919	0.98	0.940444	1.04443197
6月 21 日	0.776791	0.951032	0.98	0.940444	1.04889814
7月 21 日	0.774314	0.951032	0.98	0.940444	1.04427488
8月 21 日	0.771837	0.948145	0.98	0.940444	1.02742998
9月 21 日	0.769361	0.945259	0.98	0.940444	0.99173116
10月 21 日	0.766884	0.942372	0.98	0.940444	0.93390839
11月 21 日	0.764407	0.936599	0.98	0.940444	0.86135329
12月 21 日	0.76193	0.933712	0.98	0.940444	0.82779425

#### 4.1.10 输出热功率

定日镜输出热功率是指使用定日镜系统收集太阳辐射并将其转化为热能的能力。求解公式为：

$$E_{\text{field}} = DNI \cdot \sum_{i=1}^n \times A_i \eta_i \quad (17)$$

其中：DNI 为法向直接辐射辐照度;N 为定日镜总数(单位:面)，N = 1745；Ai 为第 i 面定日镜采光面积(单位:m<sup>2</sup>)，为常数 6 × 6；ηi 为第 i 面镜子的光学效率。

将已知和以求得的所有未知量带入上述公式(18)，最终求解得出输出热功率并填入对应表 4.3

表 4.3 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	面平均输出热功率 (kW/m <sup>2</sup> )	年平均输出热功率 (MW)
0.767606417	0.94094875	0.98	0.937444	0.966113897	56.1968561

## 六、评价与改进

### 6.1 模型的优点

1. 该模型利用太阳位置算法（SPA）跟踪太阳，建立镜场光学效率的模型，为了方便及计算，阴影遮挡损失和阴影遮挡效率，假设当太阳方位角在 170-190 度之间时，塔影造成 10% 的能量损失。
2. 该模型将定日镜面朝向太阳的角度调整到最佳位置，以最大化接受太阳辐射能量。

### 6.2 模型的缺点

1. 求解太阳反射角假定太阳的光束是平行的，对于余弦效率误差较大。
2. 效率建模在精度方面还存在一定误差
3. 优化目标是较为单一。镜场优化研究中大都是以效率为目标进行优化，但在实际应用中，镜场的投资巨大，所以可将光学效率与镜场的投资成本、并等因素结合，以整体的经济性来综合优化镜场布局，从而达到效率最高。

### 6.3 模型的改进

本文对定日镜的光照的计算效率转换过程作了理论分析，并构建了数学模型，通过对光学效率的优化计算在塔式光热电站的运行过程当中，降低光学效率。在考虑定日镜场的布置时应当加入电站实际的占地面积限制、地形影响等因素，结合镜场效率，加以考虑南北镜场余弦效率不同，进一步对定日镜场布置进行优化。

## 参考文献

- [1]杜宇航.塔式光热电站机组性能模型建模及运行特性研究.2020.上海发电设备成套设计研究院,MA thesis.
- [2]曾季川.塔式太阳能吸热器热效率评估方法研究.2021.浙江大学,MA thesis.
- [3]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.2022.兰州交通大学,MA thesis.
- [4]阚斌.古浪10万千瓦槽式光热电站系统方案研究.2018.兰州理工大学,MA thesis.
- [5]刘建兴.塔式光热电站光学效率建模仿真及定日镜场优化布置.2022.兰州交通大学,MA thesis.
- [6]牛一森.基于太阳形状的直接辐射精细测量.2021.华北电力大学(北京),MA thesis.
- [7] <https://www.rili.com.cn/24jieqi/xiazhi/106650.html> (日历查询)

## 附 录

附录 1、python 代码

01\_定日镜场示意图（异常点检测）.py

02\_d\_HR.py

03\_太阳高度角和方位角.py

04\_太阳直射点回归图.py

05\_天数计算.py

06\_DNI.py

07\_入射角.py

08\_余弦损失.py

09\_距离与透射率关系图.py

10\_能量计算.py

附录 2、表格数据

days\_diff.xlsx

太阳角度.xlsx

太阳角度和 DNI\_计算后.xlsx

太阳角度和正常时间.xlsx

太阳角度数据.xlsx

透射率.xlsx

python具体代码：

### 01\_定日镜场示意图（异常点检测）.py

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
# 设置忽略警告
import warnings

warnings.filterwarnings('ignore')
# 设置matplotlib正常显示中文和负号
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 读取Excel文件中的数据
data = pd.read_excel('./A/附件.xlsx')
x = data['x坐标 (m)']
y = data['y坐标 (m)']

radius = 350 # 圆形的半径

# 圆心坐标
center_x = 0
center_y = 0

# 计算每个点到圆心的距离并归一化
distances = np.sqrt((x - center_x) ** 2 + (y - center_y) ** 2)
normalized_distances = distances / radius

# 生成渐变的颜色值数组
colors = np.linspace(0, 1, len(x))

# 创建一个新的图形
plt.figure(figsize=(8, 5)) # 8表示宽度, 6表示高度

# 绘制圆形
theta = np.linspace(0, 2 * np.pi, 100)
x_circle = radius * np.cos(theta)
y_circle = radius * np.sin(theta)
plt.plot(x_circle, y_circle, label='边界区域')

# 绘制圆心
plt.plot(center_x, center_y, 'ro', label='吸收塔中心')

# 绘制分布点, 大小和颜色渐变展示
plt.scatter(x, y, c=colors, cmap='jet', label='定日镜位置', s=10*normalized_distances)
```

```

# 设置图形的标题和坐标轴标签
# 手动放置标题在底部略微往下移动位置
plt.text(0.5, -0.12, '图1.1 定日镜场示意图', horizontalalignment='center',
verticalalignment='center', transform=plt.gca().transAxes)
plt.xlabel('X')
plt.ylabel('Y')

# 添加图例
plt.legend(loc='upper right', fontsize=8)

# 显示图形
plt.show()

```

02\_d\_HR.py

```

import pandas as pd
import numpy as np

# 读取Excel数据
data = pd.read_excel('A/附件.xlsx')

# 获取x坐标列和y坐标列的数据
x_coordinates = data['x坐标 (m)'].values
y_coordinates = data['y坐标 (m)'].values
h = 76
# 收集塔（集热塔）中心的坐标
tower_center = (0, 0, 76)

# 存储计算结果的列表
results = []

# 计算每个镜面中心到集热器中心的距离，并计算对应的大气透射率
for x, y in zip(x_coordinates, y_coordinates):
    # 计算镜面中心到集热器中心距离
    distance = np.sqrt((x - tower_center[0]) ** 2 + (y - tower_center[1]) ** 2 + h**2)

    # 计算大气透射率
    if distance <= 1000:
        transmittance = 0.99321 - 0.0001176 * distance + 1.97 * 10**(-8) * distance ** 2
    else:
        transmittance = None # 设为默认值

    # 将结果添加到列表中
    results.append([x, y, distance, transmittance])

# 创建DataFrame对象
df_results = pd.DataFrame(results, columns=['x坐标 (m)', 'y坐标 (m)', '距离 (m)', '透射率'])

```

```
# 导出为Excel文件
df_results.to_excel('透射率.xlsx', index=False)
```

## 03\_太阳高度角和方位角.py

```
import math
import pandas as pd

def day_from_equinox(month, day):
    # 春分是第80天
    offset = [0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334]
    return offset[month-1] + day - 80

def solar_declination(day):
    # 计算太阳赤纬
    return math.degrees(math.asin(math.sin(math.radians(23.45)) *
math.sin(math.radians(0.9863 * (day - 199)))))

def solar_hour_angle(time):
    # 计算太阳时角
    return (time - 12) * 15

def solar_altitude_angle(declination, latitude, hour_angle):
    # 计算太阳高度角
    sin_alpha = math.sin(math.radians(declination)) * math.sin(math.radians(latitude))
+ math.cos(math.radians(declination)) * math.cos(math.radians(latitude)) *
math.cos(math.radians(hour_angle))
    return math.degrees(math.asin(sin_alpha))

def solar_azimuth(declination, latitude, hour_angle):
    # 计算太阳方位角
    cos_alpha = math.cos(math.radians(declination)) * math.cos(math.radians(latitude))
* math.sin(math.radians(hour_angle))
    sin_delta = math.sin(math.radians(declination))
    cos_delta = math.cos(math.radians(declination)) * math.sin(math.radians(latitude))
- math.sin(math.radians(declination)) * math.cos(math.radians(latitude)) *
math.cos(math.radians(hour_angle))
    gamma = math.degrees(math.atan2(cos_alpha, cos_delta))
    return (gamma + 360) % 360

def calculate_solar_angles():
    ST = [9, 10.5, 12, 13.5, 15]
    months = range(1, 13)
    days = [21] * 12

    latitude = 39.4
```

```

data = []
for month, day in zip(months, days):
    D = day_from_equinox(month, day)
    declination = solar_declination(D)

    for time in ST:
        hour_angle = solar_hour_angle(time)
        altitude = solar_altitude_angle(declination, latitude, hour_angle)
        azimuth = solar_azimuth(declination, latitude, hour_angle)

        data.append([month, day, time, altitude, azimuth, hour_angle])

# 将数据导出到Excel表格的第4/5/6列
df = pd.DataFrame(data, columns=['月份', '日期', '时间', '太阳高度角（°）', '太阳方位角（°）', '太阳时角（°）'])
df.to_excel('太阳角度数据.xlsx', index=False)

print("数据已导出到 Excel 文件 '太阳角度数据.xlsx'")

# 示例调用
calculate_solar_angles()

```

## 04\_太阳直射点回归图.py

```

import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
# 设置忽略警告
import warnings

warnings.filterwarnings('ignore')
# 设置matplotlib正常显示中文和负号
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 设置起始日期和结束日期
start_date = datetime(2023, 1, 1)
end_date = datetime(2024, 1, 1)

# 创建一个空列表存储太阳直射点纬度
subSolar_latitude = []

# 生成每一天的日期
current_date = start_date
while current_date < end_date:
    # 计算当前日期的太阳直射点纬度，这里使用了一个简化的计算公式
    day_of_year = (current_date - datetime(current_date.year, 1, 1)).days + 1

```

```

    subsolar_latitude.append(23.45 * np.sin(np.radians(360 * (day_of_year - 81) /
365)))

    # 增加一天
    current_date += timedelta(days=1)

# 绘制太阳直射点纬度
plt.figure(figsize=(8,6))
plt.plot(range(1, 366), subsolar_latitude)
plt.xlabel('D (天)')
plt.ylabel('太阳直射点纬度 (度)')
plt.text(0.5, -0.11, '图1.2 2023年太阳直射点回归运动图', horizontalalignment='center',
verticalalignment='center', transform=plt.gca().transAxes)

# 标记春分日和秋分日
spring_equinox = datetime(start_date.year, 3, 21)
autumn_equinox = datetime(start_date.year, 9, 23)
spring_equinox_day = (spring_equinox - datetime(start_date.year, 1, 1)).days + 1
autumn_equinox_day = (autumn_equinox - datetime(start_date.year, 1, 1)).days + 1
plt.scatter(spring_equinox_day, 0, color='red', marker='o')
plt.scatter(autumn_equinox_day, 0, color='red', marker='o')
plt.annotate('春分日\n(3.21)', xy=(spring_equinox_day, 0), xytext=(spring_equinox_day -
10, 2.5),
            ha='right', va='center')
plt.annotate('秋分日\n(9.23)', xy=(autumn_equinox_day, 0), xytext=(autumn_equinox_day +
10, -3),
            ha='left', va='center')

# 标记夏至日和冬至日
summer_solstice = datetime(start_date.year, 6, 21)
winter_solstice = datetime(start_date.year, 12, 21)
summer_solstice_day = (summer_solstice - datetime(start_date.year, 1, 1)).days + 1
winter_solstice_day = (winter_solstice - datetime(start_date.year, 1, 1)).days + 1
plt.scatter(summer_solstice_day, 23.45, color='blue', marker='o')
plt.scatter(winter_solstice_day, -23.45, color='blue', marker='o')
plt.annotate('夏至日\n(6.21)', xy=(summer_solstice_day, 23.45), xytext=
(summer_solstice_day, 23.45 - 5),
            ha='center', va='bottom')
plt.annotate('冬至日\n(12.22)', xy=(winter_solstice_day, -23.45), xytext=
(winter_solstice_day, -23.45 + 5),
            ha='center', va='top')

# 添加赤道的虚线
plt.axhline(0, color='gray', linestyle='--')

plt.grid(False)
plt.show()

```

```

from datetime import datetime
from openpyxl import Workbook

def calculate_days_diff(date):
    target_date = datetime(date.year, 3, 21)
    days_diff = (date - target_date).days
    return days_diff

# 给定的日期列表
dates = [
    datetime(2023, 1, 21),
    datetime(2023, 2, 21),
    datetime(2023, 3, 21),
    datetime(2023, 4, 21),
    datetime(2023, 5, 21),
    datetime(2023, 6, 21),
    datetime(2023, 7, 21),
    datetime(2023, 8, 21),
    datetime(2023, 9, 21),
    datetime(2023, 10, 21),
    datetime(2023, 11, 21),
    datetime(2023, 12, 21)
]

# 创建一个新的Excel工作簿
workbook = Workbook()
sheet = workbook.active

# 设置表头
sheet["A1"] = "日期"
sheet["B1"] = "天数差值"

# 将数据写入工作表
for row, date in enumerate(dates, start=2):
    days_diff = calculate_days_diff(date)
    sheet[f"A{row}"] = date.strftime("%Y-%m-%d")
    sheet[f"B{row}"] = days_diff

# 保存Excel文件
workbook.save("days_diff.xlsx")

```

06\_DNI.py

```

import math
import pandas as pd

# 计算DNI

```

```

H = 3 # km
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

G0 = 1.366 # kW/m2

# 读取Excel文件
df = pd.read_excel("太阳角度.xlsx")

# 定义计算DNI的函数
def compute_dni(alpha_s):
    G0 = 1.366 # kW/m2
    return G0 * (a + b * math.exp(-c / math.sin(math.radians(alpha_s)))))

# 遍历每行计算DNI
dni_values = []
for index, row in df.iterrows():
    alpha_s_deg = row["太阳高度角 (度)"]
    dni = compute_dni(alpha_s_deg)
    dni_values.append(dni)

# 将DNI值添加到DataFrame中
df["DNI (kW/m2)"] = dni_values

# 保存更新后的DataFrame到Excel文件
df.to_excel("太阳角度和DNI_计算后.xlsx", index=False, engine='openpyxl')

```

## 07\_入射角.py

```

import math
import pandas as pd

omegas = pd.read_excel("太阳角度.xlsx", usecols=[5])
omegas = omegas.values.tolist()
print(omegas)

def calculate_inclination_angle(xi, yi, zi, z0, z1, alpha_s, sin_A):
    numerator = -xi * math.cos(alpha_s) * sin_A - yi * math.cos(alpha_s) *
    math.cos(alpha_s) + (z0 - z1) * math.sin(alpha_s)
    denominator = math.sqrt(xi**2 + yi**2 + (z0 - z1)**2)
    cos_2theta_i = numerator / denominator
    return math.acos(cos_2theta_i) / 2

# 输入参数

```

```

xi = 107.250 # B点x坐标
yi = 11.664 # B点y坐标
zi = 0.0 # B点z坐标
z0 = 80 # O点到Q点的距离
z1 = 80-8 # Q点到B点的高度差
phi = math.radians(39.4) # φ为地理纬度
nd = 21 # 计算日期在一年中的天数 (1月21日)
delta = math.radians(23.45) * math.sin(math.radians(360 * (284 + nd) / 365)) # δ为太阳赤纬角
omega = math.radians(135) # ω为太阳时角
alpha_s = math.asin(math.cos(phi) * math.cos(delta) * math.cos(omega) + math.sin(phi) * math.sin(delta)) # α_s为太阳高度角
sin_A = -math.cos(delta) * math.sin(omega) / math.cos(alpha_s) # sin_A为太阳方位角的负值

# 计算太阳入射角
theta_i = calculate_inclination_angle(xi, yi, zi, z0, z1, alpha_s, sin_A)
theta_i_deg = math.degrees(theta_i)

# 输出结果
print("太阳入射角: ", theta_i_deg, "度")

```

## 08\_余弦损失.py

```

import numpy as np

# 定义太阳和定日镜的位置及方向
sun_position = np.array([0, 1, 0]) # 例如, 太阳在北方
heliostat_normal = np.array([0, 0, 1]) # 例如, 定日镜面朝上

# 计算余弦效率
cosine_efficiency = np.dot(sun_position, heliostat_normal)
cosine_loss = 1 - cosine_efficiency

# 输出
print(f"Cosine Efficiency: {cosine_efficiency:.2f}")
print(f"Cosine Loss: {cosine_loss:.2f}")

```

## 09\_距离与透射率关系图.py

```

import pandas as pd
import matplotlib.pyplot as plt
# 设置忽略警告
import warnings

warnings.filterwarnings('ignore')
# 设置matplotlib正常显示中文和负号
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

```

```

# 读取Excel数据表
df = pd.read_excel('透射率.xlsx')

# 获取距离和透射率的数据列
distance = df['距离 (m)']
transmittance = df['透射率']

# 绘制关系曲线
plt.figure(figsize=(8,6))
plt.plot(distance, transmittance, marker='o', linestyle='--')
plt.xlabel('距离 (m)')
plt.ylabel('透射率')
plt.text(0.5, -0.11, '图1.3 距离与透射率关系图', horizontalalignment='center',
verticalalignment='center', transform=plt.gca().transAxes)

plt.grid(True)
plt.show()

```

## 10\_能量计算.py

```

import numpy as np

def calculate_E_field(DNI, Ai, eta):
    """
    计算字段电能量 (E_field) 的函数

    参数:
    DNI: 太阳直射辐照度
    Ai: 定日镜的面积 (可能是一个数组)
    eta: 定日镜的效率 (可能是一个数组)

    返回值:
    E_field: 字段电能量
    """
    E_field = DNI * np.sum(Ai * eta)
    return E_field
Ai=6*6

```