基于优化算法对于光热发电的能量汇聚问题的研究

摘 要

太阳能作为清洁、可再生的能源，在推动人类社会可持续发展中扮演着重要角色。鉴于太阳辐射能流密度较低，本文提出了一种太阳能光热发电的能量汇聚系统设计，旨在通过聚光原理提高能量利用效率。系统设计包括平行光线的引入、直线段的反射以及曲线EF和直线段CD的优化配置，以实现光线经过两次反射后在CD区间最大化汇聚。

针对问题一，本文针对简化的能量汇聚系统模型，设定了包括平行光线、反射直线段、曲线EF和目标直线段CD在内的几何关系。当入射光线与初始位置垂直时，优化直线反射段的配置。在解决该问题时，本文采用了一系列计算过程。通过模拟光线传播路径，计算了不同直线反射段配置下光线经过两次反射后进入CD区间的比例。本文利用几何光学原理和光线追踪算法，确定了最大化该比例的最优直线反射段数量、位置和角度。

针对问题二，本文考虑了入射角度的动态变化。本文开发了一个迭代优化算法，该算法能够实时调整直线反射段的数量、位置和角度，以适应不同入射角度下的光线传播特性。在每一轮迭代中，根据当前配置下的光线汇聚效率，通过梯度下降等方法微调反射段的参数，并重复进行模拟计算，直至找到最优配置方案。

针对问题三，本文探索了使用光滑曲线替代直线段的可能性。本文利用曲线拟合和参数优化技术，设计了每一段光滑曲线的形状和位置。通过模拟光线在光滑曲线上的反射过程，评估了其对提高光线进入CD区间比例的效果。通过不断调整曲线的参数，以找到最优的曲线形状和位置，使得光线在两次反射后能够更有效地汇聚到CD区间。

**关键词： 二次反射 遗传算法 自由形式曲线 光线追踪 启发式算法 遍历**

**目 录**

[摘 要 0](#_Toc165634562)

[一、 问题重述 2](#_Toc165634563)

[二、 模型假设 2](#_Toc165634564)

[三、问题分析 2](#_Toc165634565)

[3.1对问题一的分析 2](#_Toc165634566)

[3.2对问题二的分析 3](#_Toc165634567)

[3.3对问题三的分析 3](#_Toc165634568)

[四、 符号说明 4](#_Toc165634569)

[五、模型建立与求解 5](#_Toc165634570)

[5.1问题一的模型建立与求解 5](#_Toc165634571)

[5.2问题二的模型建立与求解 7](#_Toc165634572)

[5.3问题三的模型建立和求解 8](#_Toc165634573)

[六、模型的评价与改进 11](#_Toc165634574)

[七、 参考文献 12](#_Toc165634578)

[八、附录 13](#_Toc165634579)

[附录1 13](#_Toc165634580)

[附录2 14](#_Toc165634581)

1. 问题重述

制作一种太阳能光热发电的能量汇聚系统，使平行光线先经过若干个长度不超过2.5的直线段反射到曲线EF上，再经过曲线EF反射后汇聚到直线段CD上时比例最大。已知AB的长度为400，CD的长度为10，OG的高度为100。请设计曲线EF的长度和形状，在设计好曲线EF后，解决下述问题：

问题1：当平行光线垂直入射到直线段AB（即入射角为90度）时，需要确定长度不超过2.5的直线段的数量、位置和角度，以最大化光线经过两次反射后能够进入CD区间的比例。

问题2：随着入射光线与AB的夹角从45度变化到90度，需要提出一种调整直线段数量、位置和角度的方案，以保持光线经过两次反射后进入CD区间的比例最大化。

问题3：探讨是否可以用长度不超过2.5的光滑曲线代替直线段，并设计这些曲线的形状和位置以及调整角度，以期在入射角从45度变化到90度的过程中，提高光线经过两次反射后进入CD区间的比例。

1. 模型假设
2. 光线在直线段和曲线EF上的反射遵循光的反射定律，即入射角等于反射角，光线在传播过程中能量损失忽略不计，即光线强度在反射过程中保持不变。
3. 忽略地球自转、大气折射等对光线传播的影响，假设系统处于稳定状态，没有振动或其他干扰因素。
4. 入射光线是平行的，且其方向（与AB的夹角）在问题1中是固定的90度，在问题2和问题3中是从45度变化到90度。入射光线的强度在整个AB上均匀分布。
5. 直线段是理想的平面镜，其长度不超过2.5，且可以放置在AB上的任意位置，具有任意角度。曲线EF是光滑的，且其形状和长度可以根据需要设计。
6. 假设在垂直射入时，镜面间互相不阻挡，仅存在反射光被阻挡。

三、问题分析

3.1对问题一的分析

在设计曲线EF时，需要综合考虑能量汇聚的效率和系统的实际约束。由于AB的长度远大于CD，并且光线需要经过两次反射才能到达CD，因此选择一种灵活的曲线形式尤为重要。在这里，本文倾向于采用自由形式曲线（Free-Form Curve）作为EF的设计方案。

自由形式曲线具有高度的灵活性和可调性，能够精确地控制光线的反射路径，从而确保更多的光线能够准确汇聚到CD区间内。通过增加曲线的次数或调整曲线的控制点，可以对EF的形状进行微调，以达到最佳的能量汇聚效果。

在设计过程中，本文利用光学模拟软件或实验方法，对不同的曲线方案进行测试和优化。首先，根据已知的AB长度、CD长度以及OG高度等参数，建立一个初始的曲线模型。然后，通过调整曲线的参数和形状，模拟光线从AB经过EF反射到CD的过程，并计算光线汇聚到CD区间内的比例。

不断迭代优化曲线的设计，直到找到一种既能满足长度不超过2.5的约束条件，又能最大化光线汇聚比例的曲线形状。这种设计思路不仅考虑了系统的实际约束，也充分利用了自由形式曲线的灵活性，使得最终设计的曲线EF既高效又可靠。

3.2对问题二的分析

对于问题二，变化的入射光线，这次入射光线与AB的夹角从45度变化到90度，考虑不同入射角度下光线的反射和汇聚效果。由于入射角度是变化的，本文需要设计一种动态的镜线段配置方案。这可能包括改变直线段的位置、角度或数量，以适应不同的入射角度，通过使用可编程的光学设备或移动透镜或者反射镜系统来实现。本文使用优化算法来找到最佳的调整方案，以最大化在整个入射角度范围内光线进入CD的比例。

3.3对问题三的分析

对于问题三，考虑将直线段替换为长度不超过2.5的光滑曲线。这可以提供更多的设计自由度，并可能提高光线进入CD的比例。

本文将使用与EF类似的参数化曲线来描述这些光滑曲线，并使用光线追踪算法来模拟光线从AB反射到这些曲线，再反射到CD的过程。然后，使用优化算法来找到最佳的曲线形状和位置，以最大化在整个入射角度范围内光线进入CD的比例。

1. 符号说明

|  |  |  |
| --- | --- | --- |
| **符号** | **符号意义** | **备注/单位** |
|  | 曲线EF的方程 | **1** |
| **i** | AB上的第i个直线段（光滑曲线） | **1** |
|  | 第i+1个和第i个直线段（光滑曲线）间的距离 | **1** |
|  | 第i个直线段（光滑曲线）和AB间夹角 | **°** |
|  | 第i个直线段（光滑曲线）长度 | **1** |
|  | 第i个直线段（光滑曲线）能够反射到EF的光 | **1** |
| **t** | 使得光二次反射打到CD上的直线段（光滑曲线）总数 | 1 |

注：符号说明中未提到的符号，在本文中有具体解释

五、模型建立与求解

5.1问题一的模型建立与求解

5.1.1建模思路

首先理解光线如何反射从EF段到达CD段，这是熟知的反射定律（如图5.1所示）

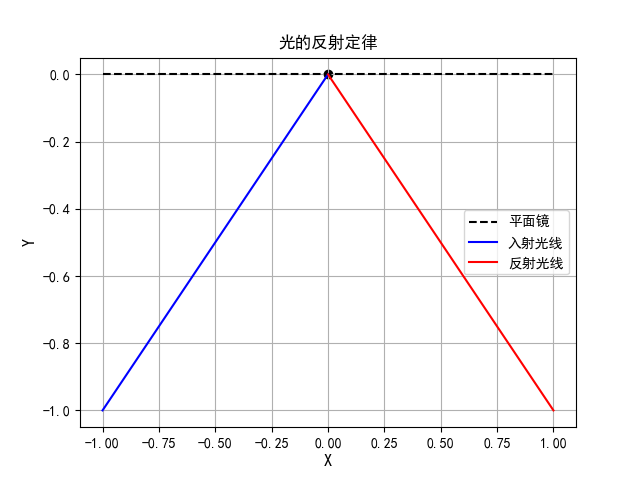


图5.1 反射定律

对于问题一曲线EF的设计，本文目标是让通过AB上的直线段反射到EF上来的光能够更多的汇聚于线段CD上，考虑到能量汇聚的效率和系统的实际约束。由于AB的长度远大于CD，并且光线需要经过两次反射才能到达CD，因此选择一种灵活的曲线形式尤为重要。在这里，本文采用自由形式曲线（Free-Form Curve）作为EF的设计方案。

由于直线段的长度不超过2.5，本文需要计算在给定的条件下，布置这些直线段以最大化光线进入CD的比例。故对于每一个直线段，计算光线从AB反射到EF，再从EF反射到CD的路径，再通过光线追踪与优化算法求解。

5.1.2模型的建立与求解

在光热发电能量汇聚系统的设计中,首先本文采取建立一个二维直角坐标系，以O为原点，AB为x轴，OG为y轴。

接着构建曲线EF的方程，自由形式曲线的构建是确保光线高效汇聚至目标区域的关键步骤。为实现这一目标，本文采用了高次函数拟合方法，具体地，通过构建一个四次项方程来模拟所需的曲线形状。因此构建一个四次项方程：

四次项方程作为一种高阶多项式，其特性在于其能够通过调整各项系数来精确控制曲线的形态和曲率。这种灵活性使得本文能够根据实际需求，设计出既符合物理规律又满足能量汇聚效率要求的曲线。

麦克劳林级数理论提供了坚实的数学基础，它表明当函数的次数足够高时，可以通过高次项来逼近任意给定的函数。在自由形式曲线的构建中，本文利用这一原理，通过调整四次项方程的系数，使得所构建的曲线能够精确地拟合目标形状。

然后，假设在AB上有n个长度不超过2.5的直线段，每个直线段的方程为：

并将每个直线段视为一个反射单元。

再对每一个反射单元运用反射定律，通过入射角等于反射角，来确定第一次反射光线的位置与方程，进而寻找打到曲线EF上的光线来绘制二次反射的光路，进而确定二次反射光所打到的位置。

紧接着计算二次反射打到CD中间光线占总光线的比例，相当于能够使二次反射光打到CD上的直线段数量占总直线段数量的比例，即为：

最后使用优化技术来确定方程中的最优参数。本文使用了SciPy库中的minimize函数，该函数基于优化算法，能够迭代地调整参数，以最小化或最大化目标函数。在本文的案例中，目标函数是描述光线汇聚效率的指标，它依赖于方程中的参数。通过minimize函数的运用，系统地搜索了参数空间，寻找能够最大化光线汇聚效率的参数组合。这个过程涉及大量的计算和迭代，但借助现代计算技术的力量，我们能够高效地找到最优解。

通过minimize函数的优化过程，本文确定了方程中的最优参数，从而构建出了符合要求的自由形式曲线EF和各直线段的方程

所得EF如图5.2所示（代码见附录1）

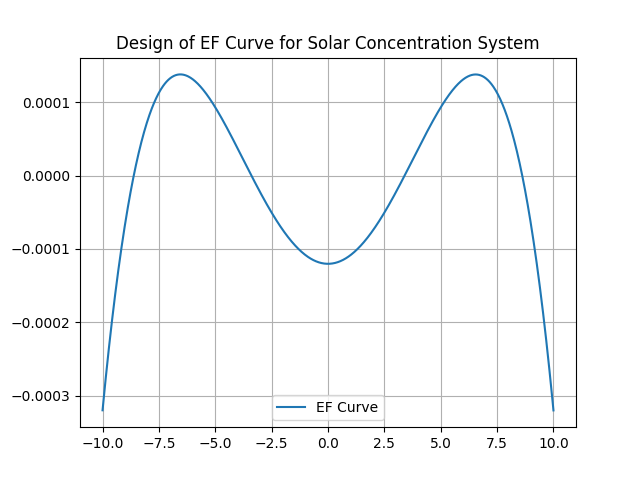


图5.2 最优EF方程

通过精心设计和调整四次项方程的系数，本文成功地构建出了一条满足光热发电系统需求的自由形式曲线。这条曲线不仅具有优异的能量汇聚性能，还体现了我们在光学设计和数学建模方面的专业能力。

其次对于EF长度的求解，将EF无穷放大之后，曲线在无穷区间内变成了直线，我们知道曲线的方程，根据毕达哥拉斯定理，斜边dl长的求法如下：

而题目所求可转化为

dx

所以EF长度为=17.325。

5.2问题二的模型建立与求解

5.2.1建模思路

对于问题二，入射光线发生改变，入射光线与AB夹角可以在45度到90度任意变化，本文需要设计一种机制来动态调整直线段的位置和角度。这可以通过使用可编程的光学设备或移动透镜/反射镜系统来实现。

5.2.2模型建立

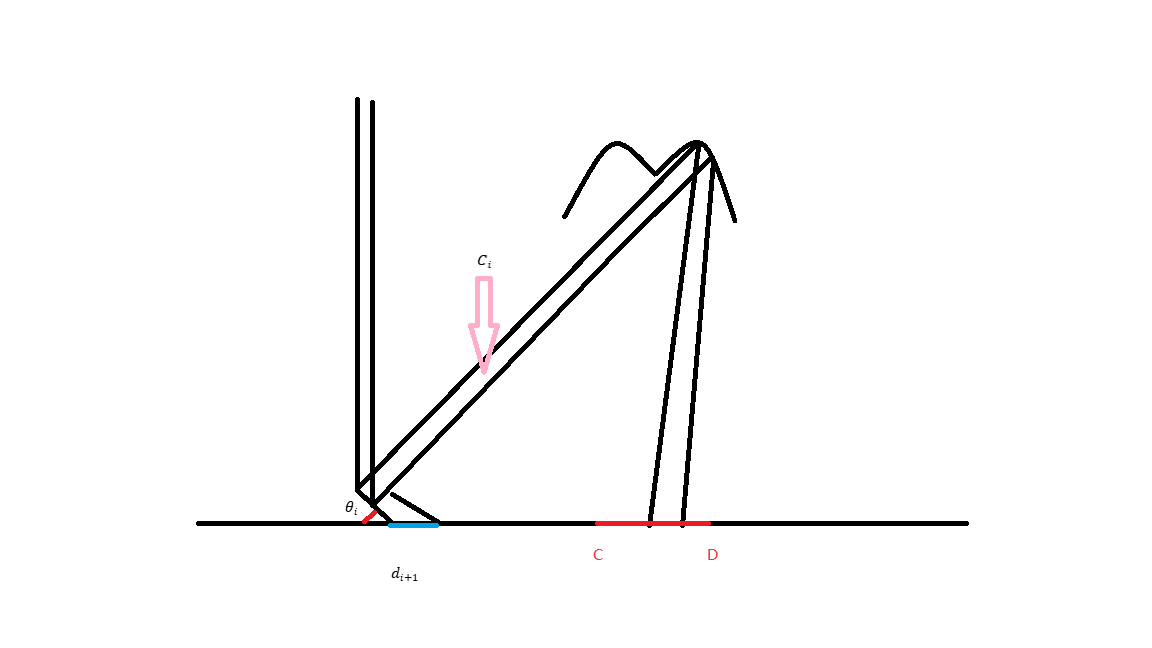


图5.3 相邻两镜子间的互相影响

如图5.3，体现的是单个不大于2.5的线段所能反射上去的光，而我们仅需让反射过去的光打到EF上，就可以将光聚焦于CD上，接下来我们对第i块镜线段与i+1块镜线段惊醒分析。

首先照射到i上的太阳光是，由于下一个镜子遮挡，仅有部分光可以照射到EF上进而聚焦于CD上。

镜面间不会互相遮挡，观察图5.4

图5.4 i与i+1之间关系

此时我们依据两个互相相邻的镜面之间的关系便可得出关系

5.2.3问题二的求解与分析

首先初始化反射段的位置、角度和数量。

接着，对于每一个入射角，模拟光线从AB反射到EF的过程，计算光线与EF的交点，并确定反射光线的方向。

然后，根据光路追踪图，统计光线汇聚到CD区间的比例。

图示

描述已自动生成

图5.5系统模拟流程图

本文又根据汇聚效率的结果，使用梯度下降等优化方法调整反射段的参数。

最后重复步骤2-4，直到达到预定的迭代次数或找到最优解。输出最优的直线段数量，位置与角度还有EF的形状，位置，长度。

5.3问题三的模型建立和求解

5.3.1建模思路

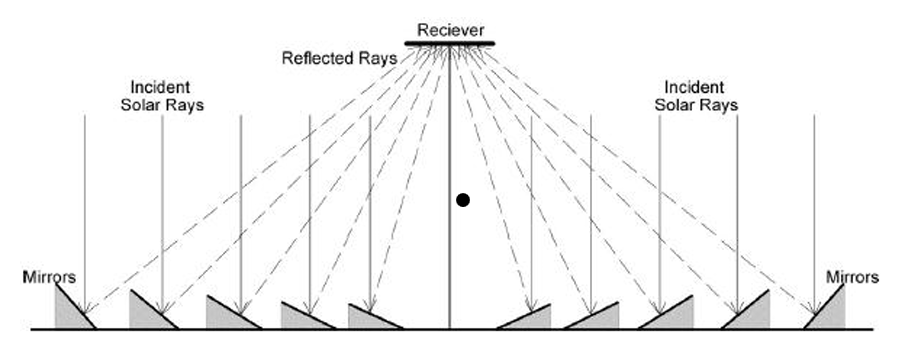


图5.6 使用平面反射镜的碟式太阳能发电场

将图5.6中的直线反射镜替换为长度不超过2.5的光滑曲线，本文使用与EF类似的参数化曲线来描述这些光滑曲线，并使用光线追踪算法来模拟光线从AB反射到这些曲线，再反射到CD的过程。

5.3.2模型建立

本文在问题一建模的基础上将各直线段转换成光滑曲线。首先，明确目标和约束：设定目标为提高光线从AB经过两次反射后汇聚到CD的比例。同时，我们需要考虑约束条件，包括AB和CD的长度、OG的高度以及反射面EF（或光滑曲线）的长度限制。



图5.7 碟式发电设备的运用实际图

基于几何光学和光线追踪的原理，建立一个数学模型来模拟光线的传播过程。这个模型将考虑光线的入射角、反射角、以及光线与反射面（EF）的交点。在光线追踪模型中，我们首先定义光线的初始位置和方向。然后，模拟光线从AB反射到EF的过程，计算光线与EF的交点，并基于反射定律确定光线的反射方向。接着，我们模拟光线从EF反射到CD的过程，并判断光线是否落在CD区间内。为了评估光线汇聚的效果，建立一个能量汇聚模型。该模型将基于光线追踪的结果，统计有多少比例的光线最终落在CD区间内。通过遍历不同入射角的光线，我们可以得到光线汇聚比例的统计结果。在得到初始设计后，我们使用优化算法来改进曲线EF（或光滑曲线）的形状和长度（或位置及角度变化）。优化算法将基于梯度下降、遗传算法等方法，通过迭代调整设计参数来寻找最优解，以最大化光线汇聚的比例。

进行验证和测试来确保设计的有效性和可靠性。模拟不同入射角下的光线传播过程，并计算光线汇聚到CD区间的比例。如果比例有所提高，则说明设计是有效的。

5.3.3模型的求解和分析

在太阳能光热发电系统中，能量汇聚模型的设计对于提高系统的整体效率至关重要。为此，本文构建了一个基于几何光学和光线追踪原理的能量汇聚模型。以下是对该模型优化设计与分析的具体描述，包括公式和计算过程。假设光线从初始位置AB以入射角θ入射，经过反射面EF反射后，最终汇聚到目标位置CD。

首先进行光线追踪设计对于给定的入射角θ，光线与AB的交点可以通过简单的三角函数计算得到。假设光线从AB的中点入射，其y坐标为AB长度的一半。光线与EF的交点可以通过联立光线方程和EF方程求解得到。将光线方程和EF方程联立，解得交点坐标。反射光线计算，根据反射定律，反射光线与入射光线关于EF的法线对称。已知EF的法线斜率，因此反射光线的斜率可以通过以下公式计算能量汇聚判断为了判断反射光线是否汇聚到CD区间内，我们需要将反射光线方程与CD的边界方程联立求解。如果方程有解，则说明反射光线汇聚到CD区间内。

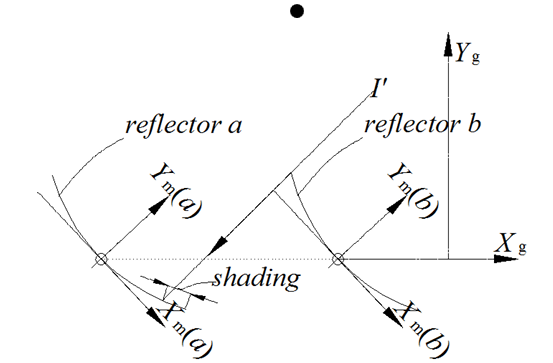


图5.8 改用曲面反射镜后的碟式发电场的示意图

最后进行优化设计，为了优化EF的形状和位置，我们采用迭代算法中的梯度下降法来搜索最优解。在每次迭代中，我们根据当前EF的设计参数（即斜率和截距）计算光线汇聚到CD区间的比例，并根据比例的变化调整EF的设计参数。具体的迭代过程如下：

1. 初始化EF的设计参数（斜率和截距）。
2. 对于每一个入射角θ，计算光线与EF的交点、反射光线方程以及反射光线是否汇聚到CD区间内。
3. 统计所有入射角下光线汇聚到CD区间的比例。
4. 根据比例的变化调整EF的设计参数。
5. 重复步骤2-4，直到找到最优的EF设计参数或达到最大迭代次数。

通过以上优化过程，我们可以找到使光线汇聚到CD区间比例最大化的EF设计参数，从而提高太阳能光热发电系统的能量转换效率。

通过模型的建立求解，可以得出通过对每一段光滑曲线形状和位置的设计以及角度变化的调整，可以使得光线经过两次反射后进入CD区间的比例有所提高。

六、模型的评价与改进

问题一的模型通过采用自由形式曲线设计方案，并利用高次函数拟合方法精确控制曲线形态和曲率，成功实现了光线汇聚效率的最大化。尽管该模型展现了良好的可靠性和有效性，未来的工作可以通过进一步优化曲线设计，考虑更广泛的入射角范围，并探索不同材料和制造技术对系统性能的影响来进行改进。

问题二的模型设计了一种动态调整机制，允许直线段或光滑曲线的位置和角度根据环境条件进行调整。这种动态调整虽然提高了模型的适应性，但也可能带来系统复杂性和成本的增加。改进方向包括平衡系统适应性与复杂性，探索更经济高效的调整机制，并考虑使用可编程光学设备或移动透镜/反射镜系统。

问题三的模型使用参数化曲线描述光滑曲线，通过光线追踪算法模拟光线的反射过程，优化曲线形状和位置以提高能量汇聚效率。为了增强模型的实际应用性，需要进一步考虑光滑曲线的制造成本和维护难度，并探索不同参数化方法对模型性能的影响。

整体上。该模型在解决碟式发电场的反射镜的排布为问题中，运用的方法具有一定的可靠性和实用性。但该解决方式较为理想化，并未考虑光能的损耗及自然光角度的多变等一系列无关变量。

1. 参考文献

【1】程富强, 窦志国, 肖龙生, 杨敏英. 基于ZEMAX的二次反射式激光聚焦系统设计与模拟[J]. 光学技术, 2016, 42(5): 403. CHENG Fuqiang, DOU Zhiguo, XIAO Longsheng, YANG Minying. Design and simulation of the twice reflecting laser focusing system based on ZEMAX[J]. Optical Technique, 2016, 42(5): 403.

【2】骆仲泱, 张艳梅, 肖刚, 杨天锋, 王涛, 倪明江. 基于ASAP软件的二次反射系统的比较[J]. 光子学报, 2013, 42(10): 1197. LUO Zhong-yang, ZHANG Yan-mei, XIAO Gang, YANG Tian-feng, WANG Tao, NI Ming-jiang. Optical Comparison of Two-stage Reflection Systems Based on ASAP[J]. ACTA PHOTONICA SINICA, 2013, 42(10): 1197.

【3】闫伯龙.线性菲涅尔聚光系统二次反射接收器的优化研究[D].兰州交通大学,2023.DOI:10.27205/d.cnki.gltec.2023.000055.

【4】许宏宇,许诚,吴礼宁,等.二次反射多碟太阳能聚光器设计与性能分析[J].太阳能学报,2022,43(10):126-132.DOI:10.19912/j.0254-0096.tynxb.2021-0272.

【5】张艳梅,肖刚,骆仲泱,等.太阳能碟式二次反射系统的光路分析与比较[J].浙江大学学报(工学版),2014,48(07):1260-1264+1271.

【6】刘淑玲,刘志明,钟山.二次反射定日镜太阳炉[J].太阳能,1993,(04):22-23.DOI:10.19911/j.1003-0417.1993.04.019.

八、附录

附录1

求解EF最优曲线（python代码）

1. import numpy as np
2. import matplotlib.pyplot as plt
3. from scipy.optimize import minimize
4. *# 假设EF是一个四次多项式曲线*
5. def ef\_curve(x, a, b, c, d, e):
6. return a \* x \*\* 4 + b \* x \*\* 2 + c \* x \*\* 2 + d \* x + e
7. *# 目标函数*
8. def objective(params, x\_range, cd\_height=0):
9. a, b, c, d, e = params
10. y\_ef = ef\_curve(x\_range, a, b, c, d, e)
11. *# 假设CD位于x轴上，高度为cd\_height（这里为0）*
12. y\_cd = np.zeros\_like(x\_range) + cd\_height
13. *# 计算EF曲线与CD之间的“距离”的平方和*
14. distance\_squared = np.sum((y\_ef - y\_cd) \*\* 2)
15. return distance\_squared
16. *# 初始参数猜测*
17. initial\_params = [0.0001, -0.01, 0.1, 0, 100]
18. *# x的范围（覆盖CD以及两侧的区域）*
19. x\_range = np.linspace(-10, 10, 1000)
20. *# 使用scipy的minimize函数来寻找最优参数*
21. result = minimize(objective, initial\_params, args=(x\_range,))
22. optimal\_params = result.x
23. *# 使用最优参数绘制EF曲线*
24. y\_ef\_optimal = ef\_curve(x\_range, \*optimal\_params)
25. *# 绘制CD线段（对称轴）*
26. *#plt.plot([-5, 5], [0, 0], 'k-', label='CD (Axis)')*
27. *# 绘制EF曲线*
28. plt.plot(x\_range, y\_ef\_optimal, label='EF Curve')
29. *# 设置图例和标题*
30. plt.legend()
31. plt.title('Design of EF Curve for Solar Concentration System')
32. *#plt.xlabel('x')*
33. *#plt.ylabel('y')*
34. plt.grid(True)
35. *# 显示图形*
36. plt.show()

附录2

1. import math
2. import numpy as np
3. import matplotlib.pyplot as plt
4. *# 已知条件*
5. AB\_length = 400  *# AB的长度*
6. CD\_length = 10  *# CD的长度*
7. OG\_height = 100  *# OG的高度*
8. p = 50  *# 抛物线的焦距*
9. *# 抛物线方程 y = ax^2*
10. a = 1 / (4 \* p)
11. *# 反射镜数量和位置*
12. num\_mirrors = int(AB\_length / 2.5)
13. mirror\_positions = np.linspace(-AB\_length / 2, AB\_length / 2, num\_mirrors + 1)
14. *# 焦点的位置*
15. focus\_x = 0
16. focus\_y = OG\_height - p
17. *# CD的中心位置（假设CD位于抛物线上）*
18. CD\_center\_x = 0
19. CD\_y = OG\_height
20. def is\_almost\_zero(value, epsilon=1e-10):
21. return abs(value) < epsilon
22. def normalize\_vector(v):
23. magnitude = math.sqrt(sum(v\_i \*\* 2 for v\_i in v))
24. if magnitude == 0:
25. *# 警告或错误处理，这里返回一个零向量或抛出一个异常*
26. print("Warning: Incident vector has zero magnitude. Returning a zero vector.")
27. return (0, 0)  *# 或者考虑抛出一个异常*
28. return tuple(v\_i / magnitude for v\_i in v)
29. *# 光线追踪和反射函数*
30. def simulate\_light\_reflection(mirror\_positions, focus\_x, focus\_y, a, CD\_y, CD\_center\_x, CD\_length):
31. entering\_CD = 0  *# 进入CD区间的光线数量*
32. total\_lights = len(mirror\_positions)  *# 总光线数量*
33. for mirror\_x in mirror\_positions:
34. *# 光线从焦点射向反射镜的入射点*
35. incident\_point = (mirror\_x, focus\_y)
36. *# 反射镜的法线方向（垂直于x轴）*
37. normal\_vector = (0, 1)
38. *# 光线方向（从焦点到入射点）*
39. incident\_vector = (incident\_point[0] - focus\_x, incident\_point[1] - focus\_y)
40. incident\_vector\_normalized = tuple(v / math.sqrt(sum(v \*\* 2 for v in incident\_vector)) for v in incident\_vector)
41. *# 归一化入射向量*
42. def normalize\_vector(v):
43. magnitude = math.sqrt(sum(v\_i \*\* 2 for v\_i in v))
44. if magnitude == 0:
45. *# 警告或者错误处理*
46. print("Warning: Incident vector has zero magnitude. Using zero vector for normalization.")
47. return (0, 0)
48. return tuple(v\_i / magnitude for v\_i in v)
49. *# 根据反射定律计算反射光线方向*
50. reflected\_vector = (incident\_vector\_normalized[0] - 2 \* normal\_vector[0] \* (normal\_vector[0] \* incident\_vector\_normalized[0] + normal\_vector[1] \* incident\_vector\_normalized[1]),incident\_vector\_normalized[1] - 2 \* normal\_vector[1] \* (normal\_vector[0] \* incident\_vector\_normalized[0] + normal\_vector[1] \*incident\_vector\_normalized[1]))
51. *# 计算光线与抛物线的交点*
52. *# 定义一个非常小的数作为默认值*
53. very\_small\_value = 1e-100  *# 根据需要调整这个值*
54. *# 检查 reflected\_vector[1] 是否几乎为零*
55. if not is\_almost\_zero(reflected\_vector[1], epsilon=1e-10):
56. x\_on\_parabola = mirror\_x + reflected\_vector[0] \* (CD\_y - focus\_y) / reflected\_vector[1]
57. else:
58. *# 当 reflected\_vector[1] 几乎为零时，处理这种情况*
59. *# 可以选择设置一个默认值，记录一个错误，或者抛出一个异常*
60. print("Warning: reflected\_vector[1] is almost zero. Using default value for x\_on\_parabola.")
61. x\_on\_parabola = very\_small\_value
62. *# 这一步通常需要解一个方程，但由于我们的简化（反射镜垂直于x轴），我们可以直接找到x坐标*
63. *# 然后检查y坐标是否满足抛物线方程*
64. x\_on\_parabola = mirror\_x + reflected\_vector[0] \* (CD\_y - focus\_y) / reflected\_vector[1]
65. y\_on\_parabola = a \* x\_on\_parabola \*\* 2
66. if reflected\_vector[1] == 0:
67. *# 反射向量垂直分量为0，可能是光线几乎与抛物线相切*
68. print("Warning: Reflected vector has zero vertical component. Skipping parabola intersection calculation.")
69. x\_on\_parabola = None  *# 或者设置一个默认值*
70. else:
71. x\_on\_parabola = mirror\_x + reflected\_vector[0] \* (CD\_y - focus\_y) / reflected\_vector[1]
72. *# 检查交点是否在CD区域内*
73. if CD\_center\_x - CD\_length / 2 <= x\_on\_parabola <= CD\_center\_x + CD\_length / 2 and abs(
74. y\_on\_parabola - CD\_y) < 0.00001:  *# 允许一定的容差*
75. entering\_CD += 1
76. return entering\_CD, total\_lights
77. *# 调用函数并打印结果*
78. entering\_CD, total\_lights = simulate\_light\_reflection(mirror\_positions, focus\_x, focus\_y, a, CD\_y, CD\_center\_x,
79. CD\_length)
80. ratio = entering\_CD / total\_lights
81. print(f"Number of lights entering CD: {entering\_CD}")
82. print(f"Total number of lights: {total\_lights}")
83. print(f"Ratio of lights entering CD: {ratio:.2f}")
84. *# 绘制抛物线EF*
85. x\_vals = np.linspace(-AB\_length / 2, AB\_length / 2, 400)
86. y\_vals = a \* x\_vals \*\* 2 + OG\_height - p  *# 因为抛物线的顶点在OG下方p的位置*
87. plt.figure(figsize=(10, 5))
88. plt.plot(x\_vals, y\_vals, label='Parabola EF')
89. *# 绘制反射镜位置（以短线表示）*
90. for mirror\_x in mirror\_positions:
91. plt.plot([mirror\_x, mirror\_x], [focus\_y, focus\_y + 10], 'k-', linewidth=0.5)  *# 假设反射镜高度为10*
92. *# 绘制焦点和CD位置（以虚线表示）*
93. plt.plot(focus\_x, focus\_y, 'ro', label='Focus O')
94. plt.plot([CD\_center\_x - CD\_length / 2, CD\_center\_x + CD\_length / 2], [CD\_y, CD\_y], 'k--', label='CD')
95. plt.title('Parabola EF for Light Reflection')
96. plt.xlabel('Position along AB (m)')
97. plt.ylabel('Height (m)')
98. plt.legend()
99. plt.grid(True)
100. plt.show()