|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 双波源干涉模拟软件 V1.0   * 使用说明  |  |  | | --- | --- | | 版本 | 1.0 | | 文档状态: | 编辑 | | 作者: | 张素红 李培显 李子敬 王晓颖 王明利 | | 负责人: | 李子敬 | | 创建日期: | 2024年5月26日 | |

修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **修改者** | **描述** |
| 2024-5-26 | V1.0 | 张素红 李培显 李子敬 王晓颖 王明利 | 增添了参数配置功能 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 简介 4](#_Toc167968812)

[1.1 编写目的 4](#_Toc167968813)

[1.2 使用对象 4](#_Toc167968814)

[1.3 产品范围 4](#_Toc167968815)

[2软件概述 5](#_Toc167968816)

[2.1总体框架 5](#_Toc167968817)

[2.2系统架构 7](#_Toc167968818)

[2.3模块描述 8](#_Toc167968819)

[3使用说明 11](#_Toc167968820)

[3.1运行程序 11](#_Toc167968821)

[3.2配置参数 14](#_Toc167968822)

[3.3导出与保存 15](#_Toc167968823)

[3.4观察模拟结果 15](#_Toc167968824)

[3.5注意事项 15](#_Toc167968825)

# 简介

在物理学中，波动干涉是一个重要的概念，它描述了两个或多个波源发出的波在空间某点上叠加时产生的加强或减弱现象。为了帮助学生、教师和研究人员更直观地理解这一概念，我们开发了这款基于Python的波动干涉模拟软件。

在现代教育和科学研究中，可视化工具在提高理解复杂物理现象方面起着至关重要的作用。传统的静态图像或简单的动画无法充分展现波动干涉的动态特性和多样性。因此，开发一款能够实时模拟并展示波动干涉现象的软件显得尤为必要。

本软件结合了NumPy、Matplotlib及其动画模块等先进技术，实现了波动传播、干涉图案的动态演示。用户可以通过自定义参数，如波源位置、波速、相位差等，来探索不同条件下的干涉效果。这种交互式的模拟方式不仅增强了学习的趣味性，还有助于用户更深入地理解干涉原理。

此外，本软件还提供了特定点波动干涉情况的合成振幅分析功能，使用户能够更直观地观察和分析干涉现象。最终，用户可以选择将模拟过程导出为高清视频，便于分享、教学以及进一步的科学研究。

总的来说，这款波动干涉模拟软件是在教育和科学研究领域对可视化工具需求不断增加的背景下产生的。它为用户提供了一个直观、交互式的平台，以更好地理解和探索波动干涉这一复杂而有趣的物理现象。

## 编写目的

本文档为使用说明文档，为产品的使用与维护提供信息基础。

## 使用对象

本文档的使用对象主要为产品测试与使用人员。

## 产品范围

本波动干涉模拟软件的产品范围主要涵盖以下几个方面：

1. **波动干涉模拟**：软件的核心功能是模拟波动干涉现象。这包括能够展示两个或多个波源发出的波在空间中的传播、叠加以及产生的干涉图案。
2. **参数自定义**：用户可以根据需要自定义模拟参数，如波源的位置、波的频率、波速、振幅以及相位差等，以便探索不同条件下的波动干涉效果。
3. **动态演示**：软件能够实时动态地演示波动干涉的过程，帮助用户直观地理解干涉现象的形成和发展。
4. **交互式学习**：提供交互式界面，允许用户通过调整参数即时观察干涉图案的变化，从而增强学习效果和兴趣。
5. **数据可视化**：除了动态的波动干涉演示外，软件还提供数据可视化功能，如波形图、干涉图案的等高线图等，帮助用户更深入地分析干涉现象。
6. **高清视频导出**：用户可以将模拟过程导出为高清视频，便于分享、教学以及科学研究。
7. **用户指南与帮助**：软件包含详细的用户指南和帮助文档，指导用户如何操作软件以及解释软件中的各种功能和参数。

综上所述，本软件的产品范围主要围绕波动干涉的模拟、分析、学习以及结果分享等方面，旨在为用户提供一个全面、易用的工具来研究和理解波动干涉现象

2软件概述

2.1总体框架

软件的总体框架可以分为以下几个主要部分：

**2.1.1初始化与配置**：

* + 设置基础参数，如字体、图形显示属性等。
  + 定义默认参数，如波源位置、波的传播速度、振幅等。
  + 通过用户输入调整这些参数。

**2.1.2数据预处理**：

* + 根据配置参数，初始化x轴、y轴的采样点和网格点。
  + 根据用户输入或默认值设定波源位置、传播速度、相位差等关键参数。

**2.1.3核心计算函数**：

* + wave\_amplitude：计算给定时间、位置和波源下的波的振幅。
  + my\_amplitude：针对单一波源，计算并绘制从波源到目标点的波动情况以及波动方向。
  + total\_amplitude：针对两个波源，计算和绘制它们对同一点产生的波动合成情况。

**2.1.4动画更新函数**：

* + update：在每个时间步上清除之前的图像，重新计算波的振幅，并使用imshow绘制波动干涉图案。
  + 如果用户选择平面化某一点的振幅合成情况，则在同一张图上同时绘制该点的波动分析。

**2.1.5动画与视频导出**：

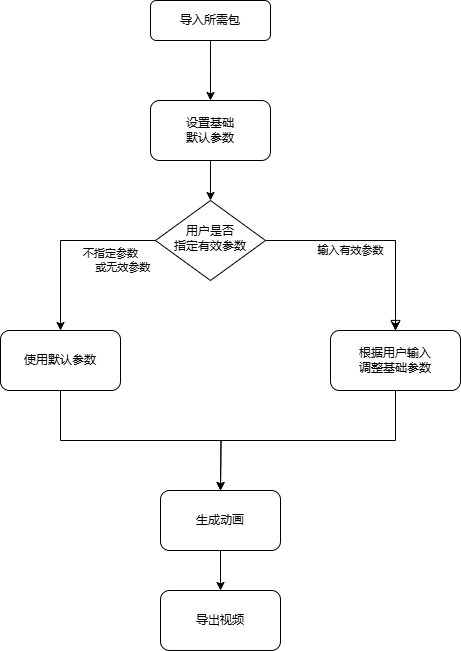
* + 使用matplotlib.animation库创建动画，通过FuncAnimation函数将update函数与图形界面关联起来，实现动画效果。
  + 如果用户选择导出视频，则使用FFMpegWriter将动画保存为MP4格式的视频文件。

**2.1.6程序执行与结束**：

* + 执行完动画显示或视频导出后，程序结束运行。

这个框架大致概括了软件的执行流程和主要功能模块。从初始化配置到核心计算，再到动画的生成与展示，每个步骤都紧密相连，共同实现了波动干涉现象的模拟与可视化。

流程图如下



## 2.2系统架构

本软件是一个基于Python的波动干涉模拟程序，它使用了NumPy、Matplotlib和Matplotlib的Animation等模块。以下是这个软件的系统架构概述：

**2.2.1用户交互层**：

* + 负责与用户进行交互，获取用户输入的配置参数，如波源位置、波的传播速度、相位差等。
  + 提供动画和视频的展示功能。

**2.2.2业务逻辑层**：

* + **参数处理模块**：解析用户输入的配置参数，并设置相应的默认值。
  + **波动计算模块**：根据用户提供的参数和物理公式，计算在给定时间点上每个空间位置的波动振幅。
  + **动画生成模块**：利用Matplotlib的Animation模块，根据波动计算结果生成动画。
  + **视频导出模块**：如果用户选择导出视频，该模块将动画转换为视频文件。

**2.2.3数据层**：

* + 存储和管理程序运行期间所需的所有数据，包括用户配置参数、波动计算结果等。

**2.2.4基础服务层**：

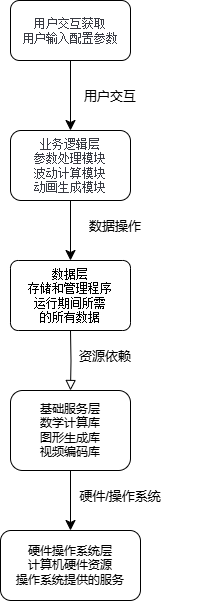
* + **数学计算库**：依赖NumPy进行高效的数学计算，如矩阵运算、三角函数计算等。
  + **图形绘制库**：依赖Matplotlib进行图形和动画的绘制，包括波动图的展示和动画的生成。
  + **视频编码库**：在导出视频时，依赖FFMpeg进行视频的编码和压缩。

**2.2.5硬件/操作系统层**：

* + 软件运行在用户的计算机上，依赖于计算机的硬件资源和操作系统提供的服务，如CPU计算能力、内存、存储空间以及操作系统的文件管理、进程调度等功能。

这个系统架构是围绕用户交互、业务逻辑处理、数据管理和基础服务支持四个核心部分构建的，确保了软件的高效运行和用户友好性。

系统架构图如下



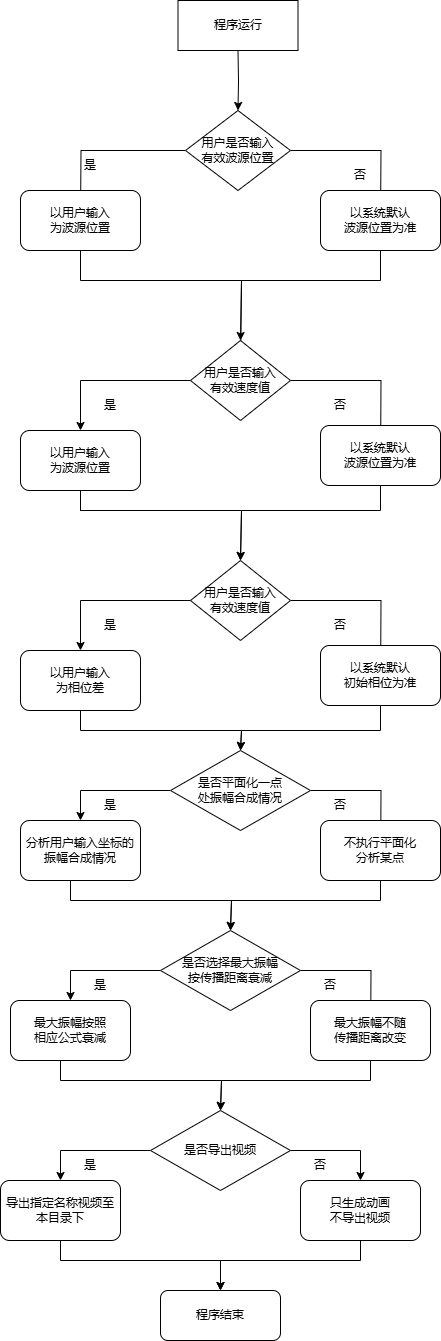
## 2.3模块描述

本软件是一个用于模拟和展示两波源干涉现象的Python程序，它结合了NumPy、Matplotlib及其动画模块来实现动态的波动干涉效果。以下是对程序主要模块的描述：

**2.3.1参数配置模块**：

* + 负责与用户交互，获取用户输入的配置参数，如波源位置、波的传播速度、相位差等。
  + 提供默认参数，以便在用户未输入特定参数时使用。

参数配置模块流程图：



**2.3.2波动计算模块**：

* + 定义了一个函数wave\_amplitude，用于计算在给定时间t、位置(x, y)和波源位置下波的振幅。
  + 考虑了波的传播速度、相位差以及振幅随距离衰减的因素（可选）。

核心算法：

计算一波源引起的振幅情况：wave\_amplitude：

np.where(t > distance / v, 1.3\*np.cos(w \* np.pi \* (distance / v - t)+np.pi/2+add\_angle\*np.pi), 0)

计算两波源的合成振幅情况：

Z = k\*(wave\_amplitude(X, Y, t, source1)+wave\_amplitude(X, Y, t, source2))

其中k是用来调整最大振幅的参数，用于模拟最大振幅随距离降低，由各点到两波源的距离而确定（与其成负相关）

波动计算模块流程图：



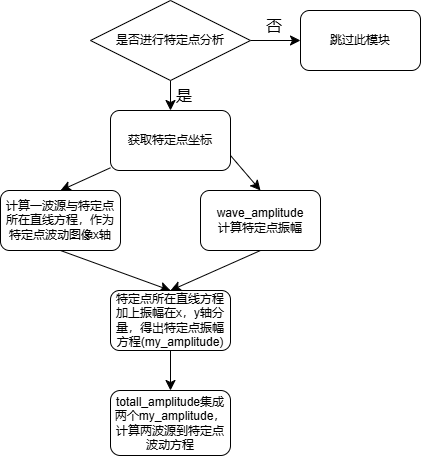
**2.3.3波动可视化模块**：

* + 利用Matplotlib库创建一个动画，动态展示两波源干涉的过程。
  + 在每个时间步，通过update函数更新波动图像，并显示当前时间步的干涉图样。
  + 使用imshow函数以彩色图像的形式展示波的振幅分布，其中蓝色代表正振幅，红色代表负振幅。

**2.3.4特定点分析模块**：

* + 如果用户选择对特定点的波动情况进行分析，该模块将绘制从波源到该点的波动传播路径以及波动在该点的叠加情况。
  + 通过my\_amplitude和total\_amplitude函数实现，以不同颜色的余弦波形展示不同波源对该点的影响。

求特征点在平面的波动方程算法：



**2.3.5视频导出模块**：

* + 如果用户选择导出动画为视频，该模块将负责将动画保存为MP4格式的视频文件。
  + 使用FFMpegWriter来设置视频的帧率、艺术家信息和比特率等参数。

**2.3.6主程序控制流**：

* + 负责整合上述模块，按照用户配置运行模拟并展示结果。
  + 在程序结束时显示动画窗口，并在用户选择导出视频时保存视频文件。

综上所述，该程序通过参数配置、波动计算、可视化展示、特定点分析和视频导出等模块，提供了一个全面的波动干涉现象模拟和展示工具。

3使用说明

3.1运行程序

**3.1.1下载了zip文件**

首先，确保已经下载并解压了本软件的zip文件。之后，切换到解压目录，运行其exe文件即可。

操作流程图如下



**3.1.2只下载了python脚本**

若只有本程序python脚本，则要确保已经安装了正确的Python环境以及必要的库（如numpy、matplotlib等）。之后，运行本Python脚本

配置python环境

Python及所用库的版本信息：

Python：Python 3.8.10

必要库：

numpy 1.24.4

matplotlib 3.7.5

**Python下载**

Python的官方下载网址是<https://www.python.org/downloads/>

在官网找到Python 3.8.10并下载

注意在安装python时在自定义选项中勾选“添加到PATH”选项

**必要库下载**

在下载python后，同时摁下win+r输入cmd打开命令行，依次执行下列命令(每次只输入一行命令后回车，待下载完毕后再执行下一条命令)

pip install numpy==1.24.4 -i https://pypi.tuna.tsinghua.edu.cn/simple

pip install matplotlib==3.7.5 -i https://pypi.tuna.tsinghua.edu.cn/simple

**检验是否配置成功**

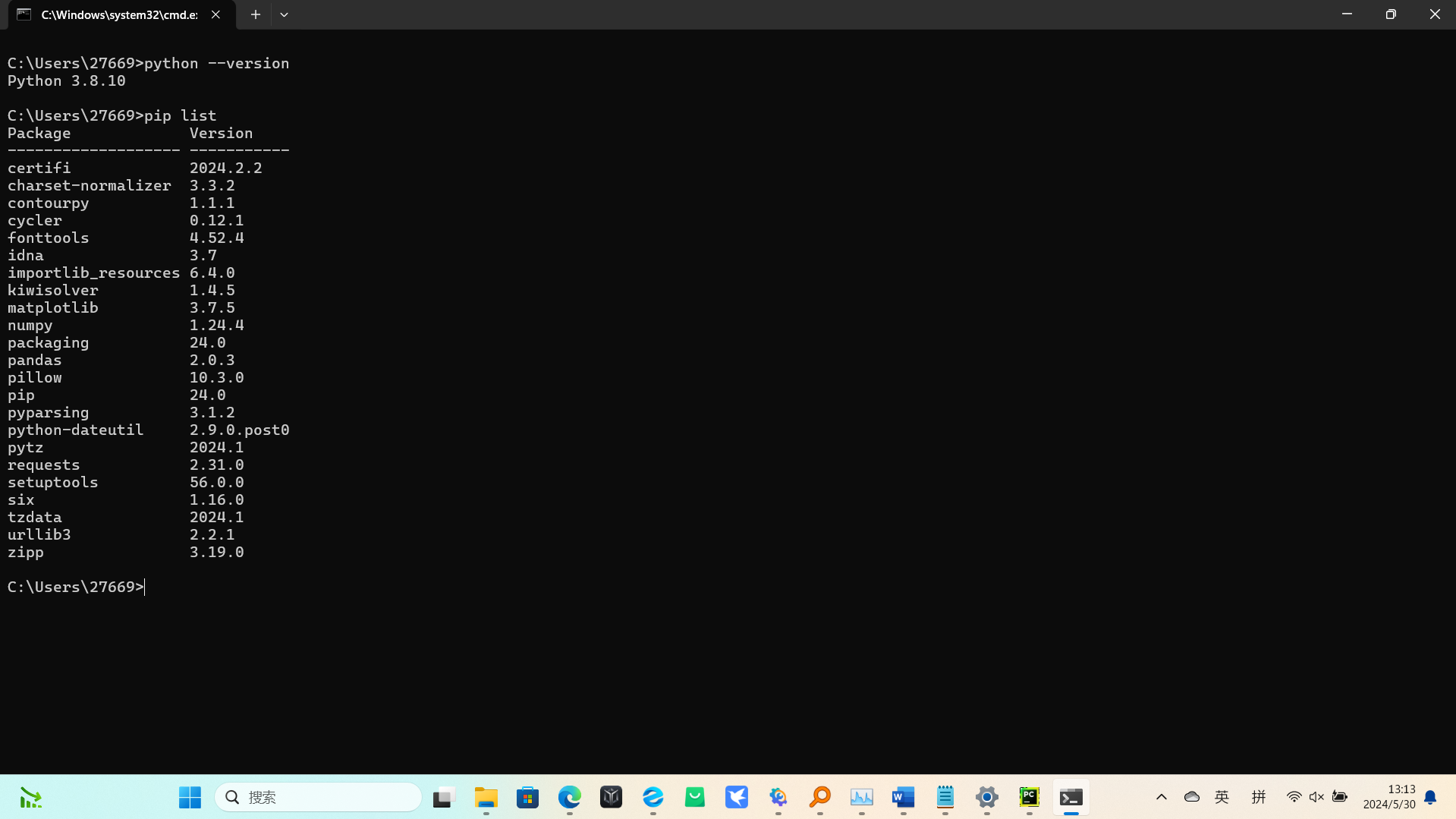
在cmd中分别输入以下两条指令并运行

python --version

pip show

若显示python以及pip相关版本和库的信息（如下图），则表示配置成功。

否则则需检查上述步骤是否有遗漏并再次配置python环境。



运行python脚本

这时我们已经完成了python环境的配置，运行python脚本方法如下:

**获取脚本绝对路径**

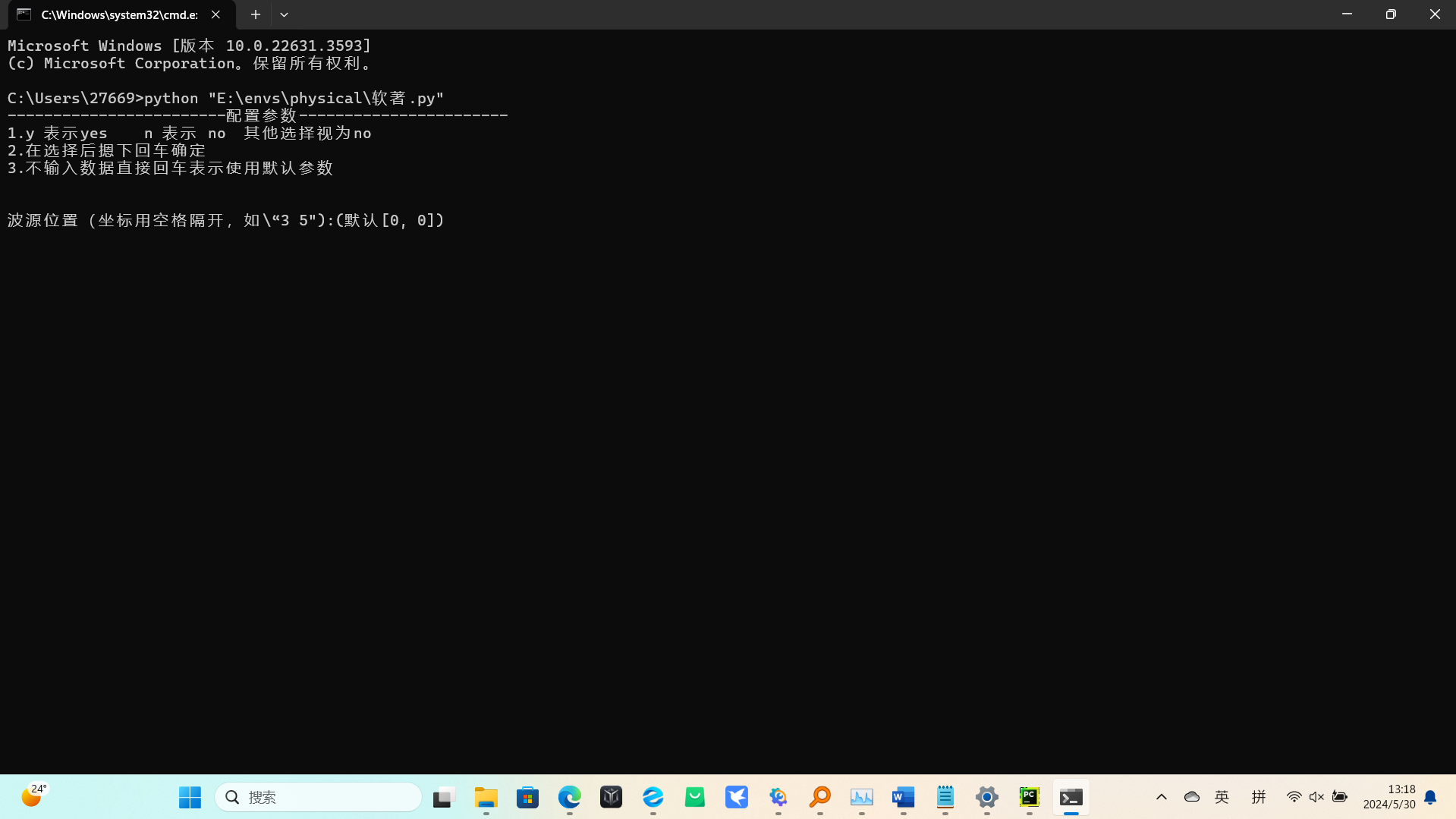
找到并右键单击脚本文件，选择“复制文件地址”

**运行文件**

打开cmd，输入并运行以下指令（fileroad换为复制的绝对路径）即可成功运行python脚本

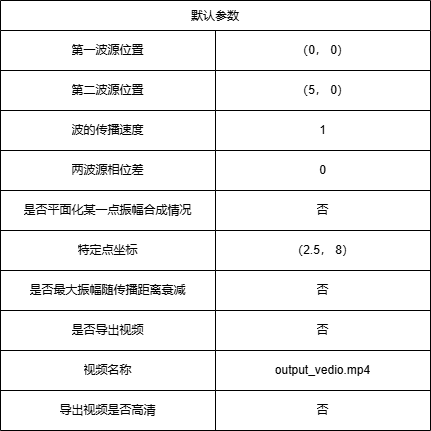
python fileroda

正确运行后如下图：



**3.2配置参数**

程序运行后，会提示用户配置一系列参数。这些参数包括波源位置、波的传播速度、两波源的相位差等。用户可以根据需要输入相应的值。如果不输入或输入无效，程序将使用默认值。



1. **波源位置**：用户需要输入两个波源的位置坐标，格式为“x y”（例如，“3 5”表示横坐标为3，纵坐标为5的位置）。如果不输入或输入无效，则分别默认为(0, 0)和(5, 0)。
2. **波的传播速度**：用户可以输入波的传播速度值。默认为1。
3. **两波源相位差**：用户可以输入两个波源之间的相位差，以π\*rad为单位。默认为0。
4. **是否平面化某一点振幅合成情况**：如果用户希望观察空间中某一点的振幅随时间的变化情况，可以选择“是”并输入该点的坐标。默认为“否”。若用户选择“是”，可继续指定被分析点的坐标，格式为“x y”（例如，“3 5”表示横坐标为3，纵坐标为5的位置）。如果不输入或输入无效，则默认为（2.5， 8）
5. **是否振幅随距离衰减**：用户可以选择振幅是否随传播距离衰减。默认为“否”。
6. **是否导出视频**：如果用户希望将模拟结果导出为视频文件，可以选择“是”并输入视频文件的名称。默认为“否”。如果选择导出高清视频，则视频质量更高但文件大小也会相应增加。

****3.3导出与保存****

如果用户选择了导出视频选项，配置参数结束后，程序将自动生成一个以指定名称加mp4后缀的视频文件并保存在程序所在的目录下（这可能要花费一分钟）。用户可以根据需要查看或分享该文件。

****3.4观察模拟结果****

配置完成并生成视频后，程序将开始模拟并显示干涉现象。用户可以观察到波在空间中的传播、叠加以及干涉条纹的形成过程。如果选择了平面化某一点的振幅合成情况，则还会在该点处绘制出振幅随时间变化的曲线图。

3.5注意事项

* 确保在运行程序之前已经正确安装了所有必要的Python库和依赖项。
* 输入参数时请遵循程序提示的格式要求以避免错误或异常情况的发生。
* 根据计算机性能和资源情况适当调整模拟的精度和范围以获得最佳的观察效果。
* 导出视频时请注意文件大小和磁盘空间占用情况以避免不必要的麻烦或数据丢失风险