

# PDE 求解框架-<https://github.com/mjx20060910/PDE->

## 一、类的成员变量，成员函数及继承关系

类名	成员变量	成员函数	继承关系
Mesh	start: float (网格起始坐标)  h: float (网格步长) -  N: int (网格节点数)  x_vec: array (网格坐标向量)	init(): 初始化网格参数  str(): 返回网格对象的字符串描述	无 (基类)
Field	val: array (存储场变量的数值 (继承 Mesh 的 start、h、N、x_vec))	init(): 初始化网格和场值-  str(): 返回场对象的字符串描述	继承自 Mesh
Source	val: array (存储源项的数值) (继承 Mesh 的 start、h、N、x_vec)	init(): 初始化网格和源项值-  make_sin (omega): 生成正弦波源项- make_gaussian (pos, sigma): 生成高斯分布源项  str(): 返回源项对象的字符串描述	继承自 Mesh
PDE	u: Field (方程的解)  f: Source (方程的源项) (继承 Mesh 的 start、h、N、x_vec)	init(): 初始化网格、解场和源项  build_diff_matrix (): 构建差分矩阵 (待子类实现)  solve (): 求解方程 (待子类实现)  save_fig (): 保存结果图像 (待子类实现)  str(): 返回方程对象的字符串描述	继承自 Mesh
LaplaceEq	-A: matrix (拉普拉斯方程的差分矩阵) (继承 PDE 的 u、f、start、h、N、x_vec)	init(): 初始化网格和差分矩阵  make_equation (f): 类方法, 从源项创建方程实例  build_diff_matrix (): 构建	继承自 PDE

		拉普拉斯方程的差分矩阵- make_sin (omega): 调用源项 的正弦生成方法- make_gaussian (pos, sigma): 调用源项的高斯生成方法- solve (): 求解拉普拉斯方程- <b>str()</b> : 返回拉普拉斯方程对象 的字符串描述	
<b>HeatEq</b>	t_end: float (模拟结束时间) - dt: float (时间步长) (继承 PDE 的 u、f、start、h、N、x_vec)	<b>init()</b> : 初始化网格、时间参 数  build_diff_matrix (): 构建 热传导方程的差分矩阵  solve (): 求解热传导方程 (显式 / 隐式方法  <b>str()</b> : 返回热传导方程对象的 字符串描述	继承自 PDE
<b>PDEFactory</b>	无	create_pde (pde_type, *args, **kwargs): 根据类型 字符串创建对应的 PDE 子类实 例	无

## 二、设计范式应用说明

该项目实现了一个基于面向对象设计的偏微分方程 (PDE) 求解框架，主要用于数值求解一维的拉普拉斯方程和热传导方程。采用了模块化设计，通过继承和多态实现代码复用，并使用工厂模式简化对象。

在 PDE 求解框架中，工厂模式通过 PDEFactory 类中导入 LaplaceEq 和 HeatEq 类实现了对对象创建的封装和抽象，使客户端代码可以通过统一接口获取不同类型的 PDE 求解器，无需关心具体实现类的构造细节，轻松扩展支持新的方程类型，这种设计使框架具有良好的可维护性和扩展性

## 三、Uml 图

