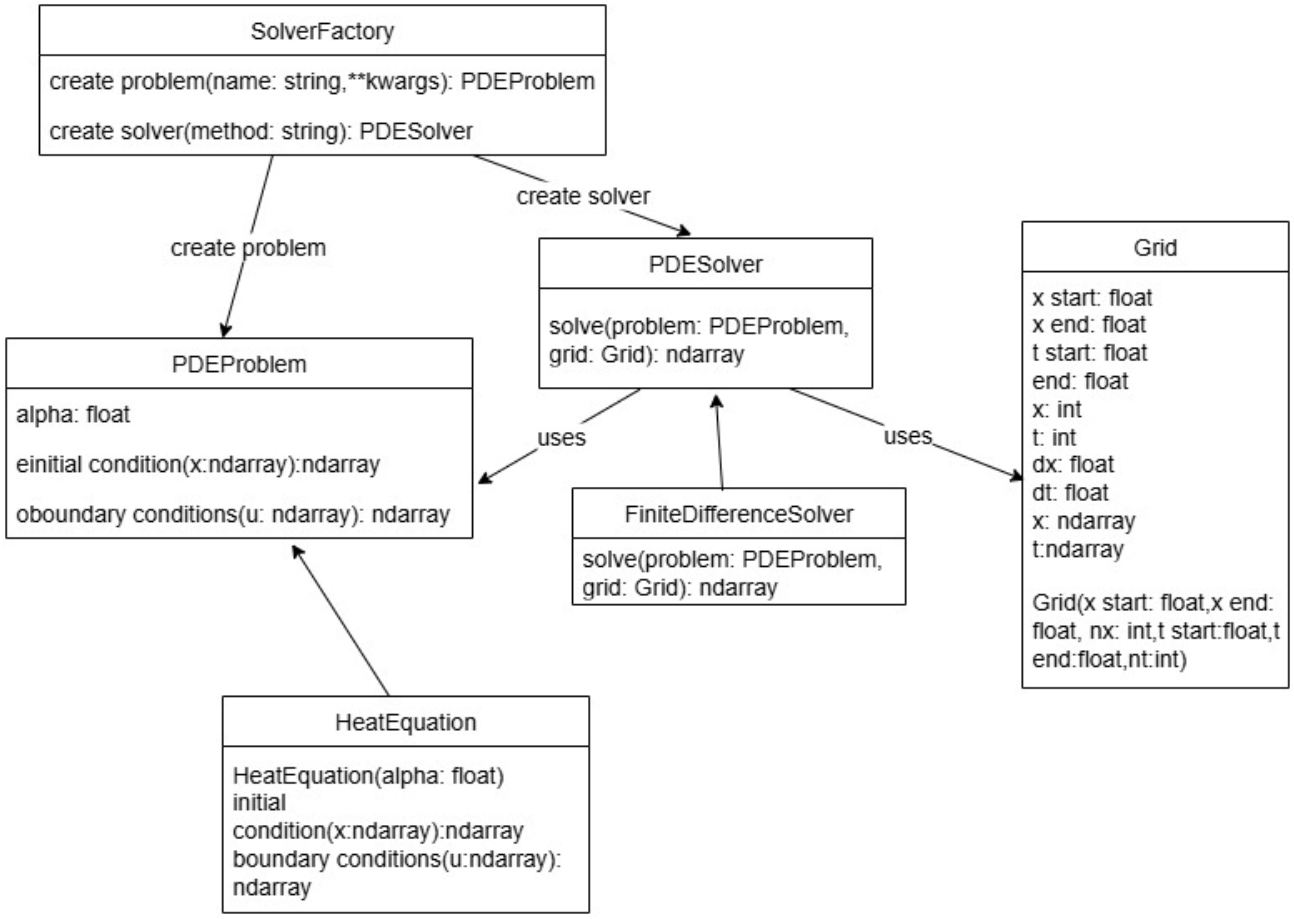

类与成员描述

1. Grid

作用 封装空间与时间离散的所有信息，为求解器提供统一的网格数据。

成员变量

x_start, x_end：空间起、终点

t_start, t_end：时间起、终点

nx, nt：空间、时间节点数

dx, dt：空间步长、时间步长

x (ndarray)：存储均匀离散后的空间节点

t (ndarray)：存储均匀离散后的时间节点

成员函数

__init__(self, x_start, x_end, nx, t_start, t_end, nt) 负责计算并初始化上述所有变量。

2. PDEProblem (抽象基类)

作用 定义 PDE 问题的统一接口：物理参数、初值与边界条件。

成员变量

alpha：扩散系数（物理参数）

成员函数（抽象）

initial_condition(self, x: ndarray) -> ndarray 根据空间节点返回初始场分布。

boundary_conditions(self, u: ndarray) -> ndarray 对解数组 u 应用边界条件，返回处

理后结果。

3. HeatEquation (PDEProblem 子类)

作用 一维热传导方程的具体实现。

继承自 PDEProblem

重写成员函数

initial_condition(self, x) 实现为高斯分布：$u(x,0) = \exp(-100(x-0.5)^2)$

boundary_conditions(self, u) 两端固定为 0：u[0]=u[-1]=0

4. PDESolver (抽象基类)

作用 定义通用求解器接口，支持多种数值方法。

成员函数（抽象）

solve(self, problem: PDEProblem, grid: Grid) -> ndarray 接受一个 PDEProblem 实例和一个 Grid 实例，返回数值解矩阵。

5. FiniteDifferenceSolver (PDESolver 子类)

作用 基于显式有限差分法，求解一维热传导方程。

继承

继承自 PDESolver

成员函数

solve(self, problem, grid)

计算稳定性参数 $r = \alpha \frac{dt}{dx^2}$，抛出不满足 r≤0.5 的异常

初始化解矩阵 u[0,:]

在时间步循环中按显式差分格式更新内部节点

每步后调用 problem.boundary_conditions

6. SolverFactory

作用 工厂类，负责根据字符串标识动态实例化问题或求解器，解耦创建逻辑。

成员函数（静态）

create_problem(name: str, **kwargs) -> PDEProblem 根据 name 返回对应 PDEProblem 实例

create_solver(method: str) -> PDESolver 根据 method 返回对应 PDESolver 实例

类之间的继承关系

HeatEquation 从 PDEProblem 继承，重写初值与边界条件

FiniteDifferenceSolver 从 PDESolver 继承，实现具体的 solve 方法

设计范式与原则

抽象基类

用 PDEProblem 与 PDESolver 定义公共接口，实现对不同 PDE 问题和不同数值方法的统一调用。

工厂模式 (Factory Pattern)

SolverFactory 静态方法屏蔽具体类名，保证调用端只依赖接口，不直引用实现类，符合**依赖倒置原则**。