

专题研究笔记：三个 Note 的数学思想与代码整合分析

Week 4 手写笔记 \leftrightarrow RoomHeating 代码库映射

Group Project — Room Heating System Control

2026 年 2 月 19 日

目录

I 三个 Note 的数学思想系统梳理	3
1 Note 1: 从 ODE 到 PDE 的解析推导	3
1.1 ODE 解析解	3
1.2 Bang-Bang 振荡周期分析	4
1.3 1D PDE 稳态解析解	4
1.4 2D PDE 稳态分析	5
2 Note 2: 有限差分离散格式	5
2.1 1D FDM: 从绝热到 Robin BC	6
2.2 2D FDM: 完整边界处理	6
3 Note 3: 四分量评价体系	7
3.1 J_E : 能耗	7
3.2 J_C : 舒适度——从点到场	7
3.3 J_S : 开关惩罚	8
3.4 J_L : 位置相关指标（两个子分量）	8
II 代码映射分析: 已实现 vs 未实现	8
4 全景映射表	9
5 已实现部分的详细对照	9
5.1 ODE 振荡周期: Note 1 \leftrightarrow <code>ode_model.py</code>	9
5.2 2D FDM: Note 2 \leftrightarrow <code>pde_2d_model.py</code>	10
III Gap 分析与整合方案	10
6 Gap 1: ODE 解析解验证工具	11

7 Gap 2: 1D 稳态解析验证	11
8 Gap 3: 空间舒适度 J_C (1D/2D)	11
9 Gap 4: 空间均匀性 J_{L_1} (时间积分版)	12
10 Gap 5: 传感器代表性 J_{L_2}	13
11 Gap 6: 四分量加权评价函数	13
IV 整合路线图	14
12 优先级排序	14
13 与场景系统的整合	15
14 总结: 三个 Note 的思想在代码中的当前覆盖率	15
15 自检清单	15

Part I

三个 Note 的数学思想系统梳理

1 Note 1: 从 ODE 到 PDE 的解析推导

Note 1 共 6 页，建立了从 ODE（零维）到 1D PDE 再到 2D PDE 的完整解析框架。其核心价值在于：为数值仿真提供解析基准解（analytical benchmark），用于验证代码正确性。

1.1 ODE 解析解

Newton 冷却定律的精确解 [Note 1, p.1]

起始方程：

$$\frac{dT}{dt} = -k(T - T_a) + u(t) \quad (1)$$

当 $u(t) = U_t$ （分段常数）时，用积分因子法 e^{kt} 得：

$$\frac{d}{dt}[e^{kt}T(t)] = e^{kt}[kT_a + U_t] \quad (2)$$

$$\Rightarrow T(t) = T_a + \frac{U_t}{k} + e^{-k(t-t_0)} \left(T(t_0) - T_a - \frac{U_t}{k} \right) \quad (3)$$

ODE 解析解 — 两种工况

关机 ($U_t = 0$)：

$$T_{\text{off}}(t) = T_a + (T(t_0) - T_a)e^{-k(t-t_0)} \quad (4)$$

温度指数衰减至室外温度 T_a 。

开机 ($U_t = U_{\max}$)：

$$T_{\text{on}}(t) = T_a + \frac{U_{\max}}{k} + \left(T(t_0) - T_a - \frac{U_{\max}}{k} \right) e^{-k(t-t_0)} \quad (5)$$

温度指数趋近稳态 $T_{\text{ss}} = T_a + U_{\max}/k$ 。

为什么解析解重要？

数值方法（如 `solve_ivp` 的 RK45）给出的是近似解。解析解提供了**精确基准**——若数值解与解析解的误差超过求解器容差，则代码有 bug。这是验证（verification）与确认（validation）的第一步。

1.2 Bang-Bang 振荡周期分析

升温/降温时间推导 [Note 1, p.1-2]

设 Bang-Bang 控制器在 $[T_L, T_H] = [T_{\text{set}} - \delta, T_{\text{set}} + \delta]$ 之间切换。

升温阶段：从 T_L 加热到 T_H , 令 $T(0) = T_L$, $U_t = U_{\max}$:

$$T_H = T_a + \frac{U_{\max}}{k} + \left(T_L - T_a - \frac{U_{\max}}{k}\right)e^{-kt_{\text{on}}} \quad (6)$$

$$\Rightarrow t_{\text{on}} = \frac{1}{k} \ln \frac{T_L - (T_a + U_{\max}/k)}{T_H - (T_a + U_{\max}/k)} = \frac{1}{k} \ln \frac{T_{\text{ss}} - T_L}{T_{\text{ss}} - T_H} \quad (7)$$

降温阶段：从 T_H 自然冷却到 T_L , $U_t = 0$:

$$T_L = T_a + (T_H - T_a)e^{-kt_{\text{off}}} \quad (8)$$

$$\Rightarrow t_{\text{off}} = \frac{1}{k} \ln \frac{T_H - T_a}{T_L - T_a} \quad (9)$$

Bang-Bang 关键指标 [Note 1, p.2]

$$\text{周期 } P = t_{\text{on}} + t_{\text{off}} \quad (10)$$

$$\text{开关次数 } N_{\text{sw}} = \frac{2}{P} \times T_{\text{total}} \quad (11)$$

$$\text{能耗 } E = U_{\max} \cdot t_{\text{on}} \quad (\text{每周期}) \quad (12)$$

$$\text{总能耗 } E_{\text{total}} = \frac{T_{\text{total}}}{P} \cdot U_{\max} \cdot t_{\text{on}} + N_{\text{sw}} \cdot \epsilon \quad (13)$$

其中 ϵ 为每次开关的附加能耗（设备启停损耗）。

Zeno 现象： $\delta \rightarrow 0$ 的病态 [Note 1, p.2]

当 $\delta \rightarrow 0$ 时, $T_H \rightarrow T_L \rightarrow T_{\text{set}}$, 有

$$t_{\text{on}}, t_{\text{off}} \rightarrow \frac{1}{k} \ln(1) = 0$$

开关频率趋于无穷大——**Zeno 现象** Zhang et al., 2001。物理上加热器无法无限快切换, 数值上求解器会陷入无穷多事件。解决方案: 引入滞回带 $\delta > 0$ (hysteresis) Goebel et al., 2012。

1.3 1D PDE 稳态解析解

点热源 1D 稳态解 [Note 1, p.3-4]

热方程: $\partial T / \partial t = \alpha \partial^2 T / \partial x^2 + u(x, t)$ 。

点热源模型: $u(x, t) = q(t) \cdot \delta(x - x_0)$, 功率 $P = q(t)$ 。

稳态时 $\partial T / \partial t = 0$, 稳态方程:

$$k_1 T''(x) + P \delta(x - x_0) = 0 \quad (14)$$

边界条件: $T'(0) = 0$ (左侧绝热), $-k_1 T'(L) = h(T(L) - T_a)$ (右侧 Robin)。

在 $[0, x_0]$ 和 $(x_0, L]$ 段分别有 $T'' = 0$, 即 T 为线性函数。通过在 x_0 处积分跳跃条件 $T'(x_0^+) - T'(x_0^-) = -P/k_1$ 以及温度连续性 $T(x_0^-) = T(x_0^+)$, 解得:

$$T(x) = \begin{cases} T_a + \frac{P}{h} + \frac{P}{k_1}(L - x_0), & 0 \leq x \leq x_0 \\ T_a + \frac{P}{h} + \frac{P}{k_1}(L - x), & x_0 < x \leq L \end{cases} \quad (15)$$

1D 稳态温度分布的物理直觉

- 左侧 (绝热壁到热源): 温度恒定 ($T' = 0$ 段), 因为热量只向右流。
- 右侧 (热源到对流壁): 温度线性递减, 斜率 $= P/k_1$ 。
- 右壁温度 $T(L) = T_a + P/h$: 由 Robin BC 决定, h 越大越接近 T_a 。
- 热源越靠左 (x_0 小), 右侧线性段越长, 壁面温度越高。

1.4 2D PDE 稳态分析

2D Green's Function 方法 [Note 1, p.5–6]

2D 稳态方程:

$$k_1 \Delta T + P \delta(x - x_0) \delta(y - y_0) = 0, \quad -k_1 \partial_n T = h(T - T_a) \text{ on } \partial\Omega \quad (16)$$

令 $\theta = T - T_a$, 定义 Green 函数 $G(\vec{r}, \vec{r}_0)$ 满足:

$$k_1 \Delta G + \delta(\vec{r} - \vec{r}_0) = 0, \quad -k_1 \partial_n G = hG \text{ on } \partial\Omega \quad (17)$$

则稳态解 $T(x, y) = T_a + P G(\vec{r}, \vec{r}_0)$ 。

结论: 矩形域 + 四面 Robin BC 的 Green 函数没有简洁闭式解, 需要用特征函数展开 (双重 Fourier 级数) 或直接数值求解。

这解释了为什么代码中采用有限差分方法而非解析求解 2D 情况。

2 Note 2: 有限差分离散格式

Note 2 共 4 页, 给出了 1D 和 2D 热方程的完整有限差分离散格式, 是代码实现的直接数学基础。

2.1 1D FDM: 从绝热到 Robin BC

1D 统一离散格式 [Note 2, p.3]

引入参数 γ 统一处理两种边界：

$$\gamma = \begin{cases} 0, & \text{绝热 (insulated)} \\ \frac{2h\Delta x}{k}, & \text{非绝热 (Robin)} \end{cases}$$

左边界 ($i = 0$, 散热 + 热源)：

$$T_0^{n+1} = T_0^n + \frac{\alpha\Delta t}{(\Delta x)^2} [2T_1^n - 2T_0^n - \gamma(T_0^n - T_a)] + \Delta t u_0^n \quad (18)$$

内部节点 ($i = 1, \dots, N-1$, 纯传导)：

$$T_i^{n+1} = T_i^n + \frac{\alpha\Delta t}{(\Delta x)^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) + \Delta t u_i^n \quad (19)$$

右边界 ($i = N$, 散热)：

$$T_N^{n+1} = T_N^n + \frac{\alpha\Delta t}{(\Delta x)^2} [2T_{N-1}^n - 2T_N^n - \gamma(T_N^n - T_a)] + \Delta t u_N^n \quad (20)$$

γ 参数的物理直觉

$\gamma = 2h\Delta x/k$ 本质是 **Biot** 数的离散版本——衡量对流散热 vs 导热的相对强度。

- $\gamma = 0$: 墙壁完全绝热, 边界等价于 Neumann $\partial T / \partial n = 0$ 。
- $\gamma \gg 1$: 强对流散热, 边界温度趋近 T_a (接近 Dirichlet)。
- 窗户 $\gamma_{\text{win}} \approx 5\gamma_{\text{wall}}$: 窗户散热远强于普通墙壁。

2.2 2D FDM: 完整边界处理

2D 有限差分——四边界更新公式 [Note 2, p.4]

内部节点 (五点格式, $\Delta x = \Delta y$)：

$$T_{i,j}^{n+1} = T_{i,j}^n + \frac{\alpha\Delta t}{(\Delta x)^2} (T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n) + \Delta t u_{i,j} \quad (21)$$

四个边界的通用格式 (以左边界 $i = 0$ 为例)：

$$T_{0,j}^{n+1} = T_{0,j}^n + \frac{\alpha\Delta t}{(\Delta x)^2} [2T_{1,j}^n + T_{0,j+1}^n + T_{0,j-1}^n - 4T_{0,j}^n - \gamma_{\text{left}}(T_{0,j}^n - T_a)] + \Delta t u_{0,j} \quad (22)$$

其中 $\gamma_{\text{left}} = 2h_{\text{left}}\Delta x/k$ 。其他三个边界完全类似。

Note 2 的关键创新: γ 可以逐网格点变化

Note 2 的推导天然支持 h 沿墙壁变化——不同网格点用不同 γ 值。这正是代码中**分段式 Robin BC** (per-wall h 数组) 的数学基础:

- 普通墙段: $h_j = 0.5 \Rightarrow \gamma_j = 2 \times 0.5 \times \Delta x/k$
- 窗户段: $h_j = 2.5 \Rightarrow \gamma_j = 2 \times 2.5 \times \Delta x/k$
- 关门段: $h_j = 0 \Rightarrow \gamma_j = 0$ (绝热)
- 开门段: $h_j = 10 \Rightarrow \gamma_j$ 很大 (近似 Dirichlet)

3 Note 3: 四分量评价体系

Note 3 共 2 页, 提出了一套系统化的多目标评价框架, 将控制系统的性能评估分解为四个正交维度。

四分量加权评价函数 [Note 3, p.1]

$$J = \alpha J_E + \beta J_C + \gamma J_S + \delta J_L \quad (23)$$

其中各分量含义如下:

符号	名称	含义
J_E	能耗 (Energy)	加热器消耗的总能量
J_C	舒适度 (Comfort)	温度偏离设定值的程度
J_S	开关代价 (Switch)	加热器开关次数的惩罚
J_L	位置指标 (Location)	温度空间分布和传感器代表性

权重 $\alpha, \beta, \gamma, \delta \geq 0$ 可根据不同评估需求调节。

3.1 J_E : 能耗

$$J_E = \int_0^T u(t) dt \quad (24)$$

对于 Bang-Bang 控制器, $J_E = U_{\max} \cdot t_{\text{on}}$ (每周期) [Note 1, Eq.(12)]。

3.2 J_C : 舒适度——从点到场

J_C 的三级定义 [Note 3, p.1]

Note 3 的核心贡献之一是将舒适度指标从 ODE 推广到 PDE:

ODE (零维, 整间房一个温度):

$$J_{C,\text{ODE}} = \int_0^T (T(t) - T_{\text{set}})^2 dt \quad (25)$$

1D PDE (温度沿空间分布):

$$J_{C,1D} = \int_0^T \int_{\Omega} (T(x, t) - T_{\text{set}})^2 dx dt \quad (26)$$

2D PDE (温度场):

$$J_{C,2D} = \int_0^T \iint_{\Omega} (T(x, y, t) - T_{\text{set}})^2 dx dy dt \quad (27)$$

为什么需要空间积分?

恒温器只测量一个点。当温度场不均匀时(窗户附近冷、加热器附近热),恒温器读数达到 T_{set} 并不代表全房间舒适。 $J_{C,2D}$ 捕捉了所有位置的偏差,是更公平的舒适度评判。

3.3 J_S : 开关惩罚

$$J_S = \gamma \cdot N_{\text{switch}} \quad (28)$$

这直接对应 Zeno 分析的动机 [Note 1, p.2]: 防止高频切换导致设备损耗和数值困难。

3.4 J_L : 位置相关指标 (两个子分量)

J_L 的两个子分量 [Note 3, p.2]

(1) 空间温度均匀性 J_{L_1} :

$$1D: J_{L_1} = \int_0^T \int_{\Omega} (T(x, t) - \bar{T}(t))^2 dx dt$$

$$2D: J_{L_1} = \int_0^T \iint_{\Omega} (T(x, y, t) - \bar{T}(t))^2 dx dy dt$$

其中 $\bar{T}(t)$ 为该时刻的空间平均温度。

(2) 传感器代表性 J_{L_2} :

$$J_{L_2} = \int_0^T (T(x_s, y_s, t) - \bar{T}(t))^2 dt \quad (29)$$

衡量恒温器读数 $T(x_s, y_s, t)$ 与房间均温 $\bar{T}(t)$ 的偏差。

J_{L_2} 的实际意义

J_{L_2} 回答了核心问题: **恒温器放在哪里最有代表性?** 如果 J_{L_2} 很大, 说明恒温器位置“看到”的温度与全房间均温差距大, 控制器会基于错误信息做出决策——导致部分区域过热、部分区域过冷。

Part II

代码映射分析：已实现 vs 未实现

4 全景映射表

三个 Note ↔ 代码库映射总览					
来源	数学思想	对应代码	状态	备注	
<i>Note 1: 解析推导</i>					
N1.1	ODE 解析解	—	Gap	可用于验证	simulate_ode()
N1.2	振荡周期公式	oscillation_period_estimation()	Done	公式完全一致	
N1.3	Zeno 分析	BangBangNoHysteresis	Done	含实验脚本	
N1.4	1D 稳态解析解	—	Gap	可验证	pde_1d_model
N1.5	2D Green's function	2D FDM 数值求解	Done	确认需数值方法	
<i>Note 2: 有限差分格式</i>					
N2.1	1D FDM + Robin BC	pde_1d_model.py	Done	Method of Lines	
N2.2	γ 参数统一框架	per-wall h arrays	Done	等价实现	
N2.3	2D FDM 五点格式	pde_2d_model.py:rhs()	Done	ghost-point BC	
N2.4	2D 四边界格式	pde_2d_model.py L.160–213	Done	含角点处理	
<i>Note 3: 评价体系</i>					
N3.1	J_E 能耗	energy_consumption()	Done	含开关代价选项	
N3.2	J_C 舒适度 (ODE)	temperature_rmse()	Done	RMSE 形式	
N3.3	J_C 舒适度 (1D/2D 空间)	—	Gap	缺空间积分版	
N3.4	J_S 开关惩罚	switching_count()	Done		
N3.5	J_{L_1} 空间均匀性 (时间积分)	T_nonuniformity(仅终态)	Partial	缺时间积分	
N3.6	J_{L_2} 传感器代表性	—	Gap	完全未实现	
N3.7	四分量加权 J	unified_cost(仅 Q, R)	Partial	形式不同	

统计：13 个思想点中，7 个已完整实现，2 个部分实现，4 个为 Gap。

5 已实现部分的详细对照

5.1 ODE 振荡周期：Note 1 ↔ ode_model.py

Note 1 推导的振荡周期公式 [p.2]：

$$t_{\text{on}} = \frac{1}{k} \ln \frac{T_{ss} - T_L}{T_{ss} - T_H}, \quad t_{\text{off}} = \frac{1}{k} \ln \frac{T_H - T_a}{T_L - T_a}$$

代码 ode_model.py 第 158–176 行：

```

1 def oscillation_period_estimate(k=K_COOL, T_a=T_AMBIENT, T_set=20.0,
2                                 U_max=U_MAX, delta=0.5):
3     T_low = T_set - delta
4     T_high = T_set + delta
5     T_ss = T_a + U_max / k
6     t_heat = (1.0 / k) * np.log((T_ss - T_low) / (T_ss - T_high))
7     t_cool = (1.0 / k) * np.log((T_high - T_a) / (T_low - T_a))
8     return t_heat + t_cool

```

完全一致。代码直接实现了 Note 1 的公式。

5.2 2D FDM: Note 2 \leftrightarrow pde_2d_model.py

Note 2 的左边界更新公式 ($i = 0$):

$$T_{0,j}^{n+1} = T_{0,j}^n + \frac{\alpha \Delta t}{(\Delta x)^2} [2T_{1,j}^n + T_{0,j+1}^n + T_{0,j-1}^n - 4T_{0,j}^n - \gamma_{\text{left}}(T_{0,j}^n - T_a)] + \Delta t u_{0,j}$$

代码 pde_2d_model.py 第 168–171 行 (West wall):

```

1 dTdt[0, 1:-1] = alpha * (
2     (2*T[1,1:-1] - 2*T[0,1:-1]
3      - 2*dx*self.h_west[1:-1]*(T[0,1:-1] - self.T_a)) / dx**2
4      + (T[0,2:] - 2*T[0,1:-1] + T[0,:-2]) / dy**2
5 )

```

等价关系: 代码中 $2*dx*h_west$ 对应 Note 2 的 $\gamma_{\text{left}} = 2h\Delta x/k$ 中的分子部分 (代码中 α 已含 k 因子)。两者在数学上完全等价, 只是代码用 Method of Lines (连续时间、离散空间), 而 Note 2 用全离散格式 (前向 Euler 时间步进)。

Method of Lines vs 全离散的区别

- **Note 2:** 空间和时间都离散化 (前向 Euler), 简单直观但有 CFL 稳定性限制。
- **代码:** 仅离散空间, 时间交给 solve_ivp (自适应 RK45), 更稳定、精度更高、不需要手动选 Δt 。

两者的**空间离散**完全一致, 差别仅在时间积分方案。

Part III

Gap 分析与整合方案

6 Gap 1: ODE 解析解验证工具

当前状况：数值解缺乏解析基准

- ✓ **优势：**`simulate_ode()` 使用成熟的 `solve_ivp(RK45)`，数值精度高。
- ✗ **局限：**没有与 Note 1 解析解对照的验证测试。如果未来修改 ODE 模型参数或求解器设置，无法自动检测回归错误。
- ? **未解决：**解析解仅在 $u(t)$ 分段常数时成立，对 PID/LQR 等连续控制不适用。
- **整合方案：**在 `tests/test_models.py` 中添加解析解验证测试。

建议实现——在 `test_models.py` 中添加：

```

1 def test_ode_analytical_vs_numerical():
2     """Verify numerical ODE matches Note 1 analytical solution."""
3     k, Ta, U = K_COOL, T_AMBIENT, U_MAX
4     T0 = T_INITIAL
5     # Analytical: constant heating
6     T_analytical = lambda t: Ta + U/k + (T0 - Ta - U/k)*np.exp(-k*t)
7     # Numerical
8     t_num, T_num = simulate_ode(lambda t, T: U, T0=T0, t_end=60)
9     T_ana = T_analytical(t_num)
10    assert np.max(np.abs(T_num - T_ana)) < 0.01  # < 0.01 deg

```

7 Gap 2: 1D 稳态解析验证

1D PDE 缺乏稳态基准解

- ✓ **优势：**Note 1 给出了点热源下的精确稳态解。
- ✗ **局限：**代码中 `pde_1d_model.py` 无稳态对比测试。
- **整合方案：**运行 1D 仿真足够长时间 ($t \rightarrow \infty$)，将数值稳态与解析解 $T(x) = T_a + P/h + P/k_1 \cdot \max(L - x, L - x_0)$ 对比。

8 Gap 3: 空间舒适度 J_C (1D/2D)

这是 Note 3 最重要的未实现思想。

Gap: `metrics.py` 的 RMSE 只用恒温器点温度

当前 `temperature_rmse(t, T, T_set)` 的 `T` 参数是 **恒温器处的温度时间序列** (标量函数)，对应 Note 3 的 $J_{C,ODE}$ 。

但对 1D/2D 场景，真正的舒适度应该是全空间积分：

$$J_{C,2D} = \int_0^T \iint_{\Omega} (T(x, y, t) - T_{set})^2 dx dy dt$$

这会捕捉到恒温器“看不到”的冷区（如窗户旁、L 形死角）。

建议实现——在 metrics.py 中添加：

```

1 def spatial_comfort(t, T_field, T_set, dx, dy, mask=None):
2     """
3         J_C for 2D: integral of (T(x,y,t) - T_set)^2 over space and time.
4         T_field: shape (nx, ny, nt)
5     """
6     err2 = (T_field - T_set)**2 # (nx, ny, nt)
7     if mask is not None:
8         err2[~mask[:, :, None].repeat(err2.shape[2], axis=2)] = 0
9     # spatial integral at each time step
10    spatial_integral = np.sum(err2, axis=(0, 1)) * dx * dy # (nt, )
11    # time integral
12    return np.trapz(spatial_integral, t)

```

9 Gap 4：空间均匀性 J_{L_1} (时间积分版)

当前 $T_{nonuniformity}$ 仅用终态快照

- ✓ **优势：** run_scenarios.py 的 compute_metrics() 计算了终态温度场的标准差。
- ✗ **局限：** 只看最后一帧。在动态过程中（如开门通风 S4），温度不均匀性随时间剧烈变化，终态 std 无法反映这一过程。
- **整合方案：** 实现 Note 3 的时间积分版 J_{L_1} 。

建议实现：

```

1 def spatial_uniformity(t, T_field, dx, dy, mask=None):
2     """
3         J_L1: time-integrated spatial variance.
4         T_field: (nx, ny, nt)
5     """
6     if mask is not None:
7         n_active = mask.sum()
8         T_mean = np.array([T_field[:, :, i][mask].mean()
9                            for i in range(len(t))])
10        var_t = np.array([(np.mean((T_field[:, :, i][mask] - T_mean[i])**2)
11                          for i in range(len(t)))]
12
13    else:
        T_mean = np.mean(T_field, axis=(0, 1)) # (nt, )

```

```

14     var_t = np.mean((T_field - T_mean[None, None, :])**2,
15                      axis=(0,1)) # (nt, )
16     return np.trapz(var_t, t)

```

10 Gap 5: 传感器代表性 J_{L_2}

这是完全未实现的指标，也是 Note 3 最具洞察力的贡献。

J_{L_2} 回答的核心问题：恒温器该放哪？

$$J_{L_2} = \int_0^T (T(x_s, y_s, t) - \bar{T}(t))^2 dt \quad (30)$$

- J_{L_2} 小 \Rightarrow 恒温器读数接近均温，控制决策合理。
- J_{L_2} 大 \Rightarrow 恒温器位置不具代表性，控制器被“欺骗”。

应用场景：

1. 扫描不同 (x_s, y_s) ，找 $\arg \min J_{L_2} \Rightarrow$ 最优恒温器位置。
2. 比较不同场景的 $\min J_{L_2} \Rightarrow$ 哪种房间形状更难放恒温器。
3. 多恒温器方案 (S8): 用 $\bar{T}_{\text{sensor}} = \text{mean}/\min(T_{s_1}, T_{s_2}, T_{s_3})$ 替换单点。

建议实现：

```

1 def sensor_representativeness(t, T_sensor, T_field, mask=None):
2     """
3         J_L2: how well does the thermostat represent room average?
4         T_sensor: (nt,) - thermostat reading
5         T_field: (nx, ny, nt) - full temperature field
6     """
7     if mask is not None:
8         T_mean = np.array([T_field[:, :, i][mask].mean()
9                            for i in range(len(t))])
10    else:
11        T_mean = np.mean(T_field, axis=(0,1))
12    return np.trapz((T_sensor - T_mean)**2, t)

```

11 Gap 6: 四分量加权评价函数

当前 `unified_cost` 与 Note 3 框架不一致

- ✓ **优势:** `unified_cost()` 实现了 LQR 风格的 $J = \int [Q(T - T_{\text{set}})^2 + Ru^2] dt$, 适合 LQR/Pontryagin 控制器的评估。
- ✗ **局限:** Note 3 的框架 $J = \alpha J_E + \beta J_C + \gamma J_S + \delta J_L$ 是更通用的多目标评价，将能耗、舒适度、开关代价、空间指标正交分解。`unified_cost` 将能耗和舒适度混在 Ru^2 中，无法单独调节。

→ 整合方案：新增 `weighted_cost()` 函数，保留原 `unified_cost`（向后兼容）。

建议实现：

```

1 def weighted_cost(J_E, J_C, J_S, J_L,
2                     alpha=1.0, beta=1.0, gamma=0.1, delta=0.5):
3     """
4     Note: four-component evaluation:
5     J = alpha * J_E + beta * J_C + gamma * J_S + delta * J_L
6     """
7     return alpha * J_E + beta * J_C + gamma * J_S + delta * J_L

```

Part IV

整合路线图

12 优先级排序



13 与场景系统的整合

新指标在各场景中的预期贡献

场景	J_C 空间版	J_{L_1} 均匀性	J_{L_2} 传感器	预期洞察
S1 基线	基准	基准	基准	对照组
S2 窗户	↑窗边冷区	↑↑不均匀	↑看不到冷区	窗户效应量化
S3 窗户对比	对比三组	双层窗改善	大窗更难	隔热价值
S4 开门	↑↑门开时飙升	↑↑动态变化	↑门边冷流	通风影响
S5 长窄	↑远端冷	↑↑距离效应	↑↑位置敏感	几何效应
S6 L 形	↑↑死角	↑↑↑	↑↑死角看不到	最难控制
S7 多热源	↓改善	↓↓改善	—	分布加热优势
S8 多传感器	—	—	↓↓改善	多点感知优势

14 总结：三个 Note 的思想在代码中的当前覆盖率

整体评估

- **Note 1** (解析推导) 覆盖率 **60%**: 振荡周期和 Zeno 分析已实现, 但解析解未用于验证 (这是一个容易弥补的高价值 Gap)。
- **Note 2** (FDM 格式) 覆盖率 **100%**: 所有离散格式都已正确实现于 `pde_1d_model.py` 和 `pde_2d_model.py`。
- **Note 3** (评价体系) 覆盖率 **40%**: J_E 和 J_S 已实现, 但最具区分力的 J_C (空间版)、 J_{L_1} 、 J_{L_2} 和四分量框架均未实现。

最大整合机会在 Note 3——实现空间评价指标将直接回答项目核心问题“恒温器放在哪里最合理?”, 并为不同场景的定量对比提供数学基础。

15 自检清单

- 能否复述 Note 1 的 ODE 解析解及其推导?
- 能否解释 Note 2 的 γ 参数如何统一 insulated 和 Robin BC?
- 能否说清 Note 3 的 J_C (点版本 vs 空间版本) 的区别?
- 能否解释 J_{L_2} (传感器代表性) 为什么对项目核心问题至关重要?
- 是否理解代码中 Method of Lines 与 Note 2 全离散格式的等价性?
- 是否清楚 6 个 Gap 的优先级及实现方案?

参考文献

Goebel, Rafal et al. (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

Zhang, Jun et al. (2001). “Zeno hybrid systems”. In: *International Journal of Robust and Nonlinear Control* 11.5, pp. 435–451.