

# Allen-Cahn 方程数值稳定性克服

蒋鹏

2025 年 10 月 30 日

## 目录

### 核心问题识别

原代码能量不下降的根本原因：边界发散引发的数值不稳定

- $\log\left(\frac{u}{1-u}\right)$  在  $u \rightarrow 0$  或  $u \rightarrow 1$  时产生无穷大
- 一个点的越界通过拉普拉斯算子传播到整个区域
- 形成“边界发散  $\rightarrow$  迭代发散  $\rightarrow$  能量上升”的恶性循环

### 稳定性改进策略

#### 1. 预防性保护机制

思路：在问题发生前拦截，而非事后处理

`const double EPS = 10-24`

`$u_{\text{safe}} = \max(\text{EPS}, \min(1 - \text{EPS}, u))$`

`$f_{\text{safe}}(u) = \log\left(\frac{u_{\text{safe}}}{1 - u_{\text{safe}}}\right)$`

核心理念：所有可能产生奇异的计算都必须先进行边界保护。

#### 2. 牛顿法的稳健化设计

步长限制：  $|u_{\text{new}} - u| \leq 0.1$

范围保护：  $u_{\text{new}} \in [\text{EPS}, 1 - \text{EPS}]$

导数保护： `if  $|f'(u)| < 10^{-14}$  then 安全更新`

关键洞察：牛顿法在边界区域的导数计算可能失效，需要多重保护。

### 3. 迭代算法的系统稳定性

CG 方法：正确初始残差  $r_0 = b - Ax_0$

ADMM 参数：  $\rho = 10.0$  (增强凸性),  $\tau = 1.8$  (减小步长)

时间推进：每一步后保护  $u^{n+1} \in [\text{EPS}, 1 - \text{EPS}]$

设计原则：每个计算环节都是潜在的误差源，需要独立保护。

## 监控与诊断体系

能量监控：  $E^{n+1} > E^n + 10^{-8} \Rightarrow \text{警告}$

收敛监测：  $\|u_1 - u_2\| < \epsilon, \quad \|u_2^{k+1} - u_2^k\| < \epsilon$

边界验证：  $|u(-1, y) - u(1, y)| < 10^{-10}$

重要经验：数值实验必须有完整的诊断输出，否则就是“盲人摸象”。

## 通用数值稳定性原则

### 1. 层层防御原则

每个计算模块都应该是自保护的独立单元

### 2. 渐进推进原则

小步推进，及时修正，避免误差累积

### 3. 监控反馈原则

实时检测，发现问题立即报警并记录

### 4. 稳健参数原则

选择使算法更稳定而非更快的参数

## 深刻教训

1. 数学正确  $\neq$  数值稳定：即使推导完全正确，数值实现也可能失败
2. 边界是危险的：90% 的数值问题发生在定义域边界
3. 保护要主动：不要等问题出现再处理，要在可能出问题的地方预先保护
4. 诊断是关键：没有详细的监控输出，调试就像在黑暗中寻找开关

## 实践建议

对于类似的非线性偏微分方程数值求解：

1. 先做保护，再求精度
2. 小规模测试，确保稳定后再扩展
3. 保存完整的计算历史用于诊断
4. 参数选择保守一些，稳定优先

总结：这套方法不仅适用于 Allen-Cahn 方程，对于其他具有奇异非线性项或边界敏感问题的数值求解都具有普适指导意义。

数值稳定性不是事后修补，而是从一开始就要融入代码设计的哲学思想。