Allen-Cahn 方程数值格式设计

蒋鹏

2025 年 11 月 4 日

目录

1 问题描述

我们欲求解下述方程

$$\frac{\partial u}{\partial t} - \varepsilon^2 \Delta u + f(u) = 0; \tag{1}$$

求解区域为$\Omega = [-1, 1]^2$。初值条件为$u(x, y, 0) = u_0(x, y)$。其中$f(u)$是非线性$f(u) = \log(\frac{u}{1-u}) + \theta(1 - 2u)$.

定义$g(u) = \log(\frac{u}{1-u})$,$G(u) = u\log u + (1-u)\log(1-u)$,$F(u) = u\log u + (1-u)\log(1-u) + \theta(u - u^2)$，则$F'(u) = f(u)$，$G'(u) = g(u)$。

2 数值算法

2.1 凸分裂格式

利用时间向后 Euler，空间五点差分格式

$$\frac{U^{n+1}-U^n}{dt}-\varepsilon^2\Delta_h U^{n+1}+f(U^{n+1})=0; \tag{2}$$

做一步凸分裂，得到格式

$$\frac{U^{n+1}-U^n}{dt}-\varepsilon^2\Delta_h U^{n+1}+g(U^{n+1})+\theta(1-2U^n)=0; \tag{3}$$

2.2 基于泛函的 ADMM 格式

定义泛函

$$J(U)=\int_\Omega(\frac{\|U-U^n\|^2}{2dt}+\frac{1}{2}\varepsilon^2|\nabla U|^2+G(U)+\theta<U,1-2U^n>)dx; \tag{4}$$

于是格式等价于

$$U^{n+1}=\arg\min_{U\in V_h} J(U); \tag{5}$$

定义

$$F_1(U)=\int_\Omega(\frac{\|U-U^n\|^2}{2dt}+\frac{1}{2}\varepsilon^2|\nabla U|^2)dx \tag{6}$$

和

$$F_2(U)=\int_\Omega(G(U)+\theta<U,1-2U^n>)dx \tag{7}$$

此时，原格式等价于

$$\begin{cases}\min_{U_1,U_2} F_1(U_1)+F_2(U_2)\\ s.t.\quad U_1=U_2;\end{cases} \tag{8}$$

令 Y 为 Lagrange 乘子，设置二次罚函数

$$L(U_1,U_2;Y)=F_1(U_1)+F_2(U_2)+<Y,U_1-U_2>+\frac{\rho}{2}\|U_1-U_2\|^2; \tag{9}$$

得到 ADMM 格式

$$\begin{cases}U_1^{k+1}=\arg\min_{U_1} L(U_1,U_2^k;Y^k)\\ U_2^{k+1}=\arg\min_{U_2} L(U_1^{k+1},U_2;Y^k)\\ Y^{k+1}=Y^k+\tau\rho(U_1^{k+1}-U_2^{k+1})\end{cases} \tag{10}$$

第一个子问题有显式解：

$$(\frac{1}{dt}-\varepsilon^2\Delta_h+\rho)U_1^{k+1}=\frac{1}{dt}u^n+\rho U_2^k-Y^k; \tag{11}$$

于是可以用 CG 或 FFT 进行求解，数值实验发现 FFT 更高效。

第二个子问题的解满足

$$\log(\frac{U_2^{k+1}}{1-U_2^{k+1}})+\theta(1-2U^n)-Y^k-\rho(U_1^{k+1}-U_2^{k+1})=0; \tag{12}$$

注意此方程可以视为 $Nx * Ny$ 个一维问题，在每个网格点上，

$$\log(\frac{u_2^{k+1}}{1-u_2^{k+1}}) + \theta(1-2u^n) - y^k - \rho(u_1^{k+1} - u_2^{k+1}) = 0; \tag{13}$$

可以用一维 Newton 法进行求根。
罚函数因子 ρ 进行自适应动态调节。

3 高维 Newton 迭代

仍然基于凸分裂格式，视其为高维非线性方程求根问题：在这里令

$$b = \frac{1}{dt}U^n - \theta(1-2U^n) \tag{14}$$

和矩阵

$$A = \frac{1}{dt} - \varepsilon^2\Delta_h \tag{15}$$

于是凸分裂格式等价于求非线性方程 $AU^{n+1} + g(U^{n+1}) - b = 0$ 的根。

Newton 迭代格式

$$(A + g'(U_k))(U_{k+1} - U_k) = -AU_k - g(U_k) + b \tag{16}$$

即

$$(A + g'(U_k))U_{k+1} = g'(U_k)U_k - g(U_k) + b \tag{17}$$

注意 A 是对称正定矩阵，$g'(U_k)$ 是对角元均为正数的对角矩阵，进而该方程可以用 CG 进行求解。

4 数值算例

4.1 算例一：随机初值

取初值函数

$$u_0(x,y) = 0.01 + 0.98 * ((\text{double})\text{rand}()/(\text{RAND\_MAX} + 1.0)) \tag{18}$$

和参数 $\theta = 4.0$.

4.2 算例二：光滑初值

取初值函数

$$u_0(x,y) = \frac{1 + \sin 2\pi x \sin 2\pi y}{3} \tag{19}$$

和参数 $\theta = 4.0$.

算法	判断收敛阈值	cpu_time(s)	主迭代次数 (次)	能量	残量
truncated_admm	1e-6	17.521	admm:71	1.23249	2.04158e-05
non_cg_admm	1e-6	18.912	admm:26	1.23249	3.66484e-05
non_fft_admm	1e-6	8.907	admm:27	1.23249	2.91842e-05
newton	1e-8	2.963	newton:5	1.23249	1.85867e-05

表 1: 随机初值数值结果

算法	判断收敛阈值	cpu_time(s)	主迭代次数 (次)	能量	残量
truncated_admm	1e-6	15.496	admm:82	0.595444	4.07057e-05
non_cg_admm	1e-6	19.68	admm:41	0.595444	3.30783e-05
non_fft_admm	1e-6	10.935	admm:42	0.595444	2.18749e-05
newton		21.845	newton:18	0.595444	1.02206e-05

表 2: 光滑初值数值结果

5 实验现象

- 对于算例二，newton 法的 CG 收敛很慢，原因是正定矩阵条件数相比 admm 的 CG 太大，而 admm 的 CG 有 ρ 进行调节。
- 相同参数下，串行 newton 可能出现 CG 迭代不收敛的情况，并行版本不会，可能是因为并行版本不同的浮点数运算方法导致避免了某些数值误差

6 算法优势

猜测：在引入并行计算的模式下，由于 ADMM 的 U_2 子问题天然并行，受网格尺度影响更小，在更小的网格尺度下会体现出加速优势，不同时间步长下可以通过调整罚函数因子改进运算速度

结果记录: 取 $dt = 1e10$，$Nx = Ny = 1024$，计算一个时间步，并行化程序用 16 个计算核心：注意当阈值取 $1e-8$ 时能量下降才一致

- 随机初值
- 光滑初值