arXiv:2602.07817v1 [math.NA] 8 Feb 2026

# 大规模并行自适应八叉树网格上的守恒场自适应网格细化（AMR）方案

Kumar Saurab h[a]，Makrand A. Khanwale[b]，*，Masado Ishii[c]，[d]，Hari Sundar[c]，[e]，Baskar Ganapathysubramanian[a]，*

[a]爱荷华州立大学机械工程系，艾姆斯，爱荷华州[b]斯坦福大学机械工程系，斯坦福，加利福尼亚州[c]犹他大学 *Kahlert* 计算学院，盐湖城，犹他州[d]理化学研究所计算科学中心，日本兵库县[e]塔夫茨大学计算机科学系，梅德福，马萨诸塞州

**摘要**

自适应网格细化（AMR）通过选择性地细化和粗化网格，被广泛用于高效解析随时间变化的偏微分方程（PDE）中的局部特征。然而，在长时间跨度的模拟中，反复进行网格间插值可能会导致守恒量发生系统性漂移，尤其是在采用连续基函数的变分离散化中。对于连续伽辽金（CG）离散化，细化时从父单元到子单元的插值天然具有守恒性，但基于注入的标准子单元到父单元粗化插值通常并不守恒。

我们提出了一种简单且可扩展的场守恒粗化算子，适用于基于八叉树的并行自适应网格细化。该方法通过先计算求积点处守恒的粗单元值，再通过$L^2$投影（质量矩阵求解）恢复粗化节点自由度，从而在粗化过程中确保离散全局守恒；这一过程还可同时控制$L_2$误差。我们在质量守恒相场模型上评估了该方法，包括Cahn–Hilliard方程和 Cahn–Hilliard–Navier–Stokes系统，并根据守恒误差、解的质量和计算成本，将其与注入法进行比较。

## 1. 引言

质量守恒是许多多物理场模拟的基本要求，在长时域计算中尤为关键，因为每个时间步的微小缺陷都可能累积并造成明显漂移。许多研究致力于推导具有内在质量守恒性质的控制方程，例如用于模拟相分离现象的 Cahn–Hilliard方程 [1, 2, 3, 4], ，或用于界面追踪方法的 Cahn–Hilliard–Navier–Stokes [5, 6, 7, 8] 以及守恒型水平集方法 [9, 10, 11]。此外，还有一些研究致力于开发能够准确模拟这些偏微分方程的数值格式，并证明在固定网格上，这些方程的基本不变量（如质量）能够保持在数值精度范围内 [7, 12, 13, 14, 15, 16]。

在这些偏微分方程的实际模拟中，自适应网格细化（AMR）通常至关重要：它在演化界面或陡峭梯度附近集中分辨率，同时保持可控的计算成本。在各种 AMR 策略中，平衡自适应八叉树网格因其算法简洁性、细化/粗化的严格局部性以及在大量核心数上的可扩展性而得到广泛应用 [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]。在八叉树自适应网格细化中，自适应过程通过两个基本操作进行：细化，即将选定的八分体细分；以及粗化，即合并同级八分体。这两种操作都需要在不同分辨率的网格之间传递离散场。从一个网格 ($\mathscr{M}_1$) 到另一个网格 ($\mathscr{M}_2$) 的场传递通常称为网格间传递，可表示为

$$\underline{\mathbf{T}}(\phi_{\mathscr{M}_1}) = \phi_{\mathscr{M}_2}, \tag{1}$$

*通讯作者电子邮箱： maksbh@iastate.edu(Kumar Saurabh), khanwale@stanford.edu(Makrand A. Khanwale),masado.ishii@riken.jp (Masado Ishii), hari.sundar@tufts.edu (Hari Sundar), baskarg@iastate.edu(Baskar Ganapathysubramanian)

其中， $\phi_{\mathscr{M}_1}$ 和 $\phi_{\mathscr{M}_2}$ 分别表示 $\mathscr{M}_1$ 和 $\mathscr{M}_2$ 上的离散场表示。对于守恒标量 $\phi$，守恒传递满足

$$\left[\int_\Omega \phi \, dV\right]_{\mathscr{M}_1} = \left[\int_\Omega \phi \, dV\right]_{\mathscr{M}_2}. \tag{2}$$

一个更强的要求（我们的方法能够满足）是，方程 2 对任意子域 $\Omega_k \subset \Omega$都成立。

尽管细化和粗化在自适应网格细化中都很常见，但它们在连续伽辽金（CG）离散化的守恒方面发挥着根本不同的作用。在连续伽辽金中，细化传递可以通过基函数插值从父单元传至子单元，并且对于所表示场的积分量而言，这种传递天然具有守恒性。相比之下，标准的基于注入的粗化传递（从子单元传至父单元）会移除与粗网格节点不重合的细网格自由度，从而可能导致全局不变量发生系统性漂移。在较长的时间范围内——这正是许多多物理场问题所关注的情形——即使底层偏微分方程和时间积分器具有守恒性，这种漂移仍可能主导误差预算。

本研究针对一种具体且广泛应用的情形，旨在解决上述失效问题：采用单层级自适应的并行八叉树自适应网格细化。我们表明，对于八叉树网格上的 CG 有限元离散，非守恒粗化是全局质量漂移的主要来源，并提出一种简单且可扩展的场守恒粗化算子，以保证自适应网格细化循环中的离散全局守恒。为便于说明，我们的演示和数值算例采用线性和二次拉格朗日单元；不过，只要传递和投影所用的求积与所选阶数相一致，所提出的粗化过程适用于具有任意多项式阶拉格朗日基函数的任意CG形式（例如，$Q_p/P_p$，$p \geq 1$）。

八叉树自适应网格细化分为两个互斥阶段：

1. **细化：** $\mathscr{M}_2$ 中的每个单元要么处于比 $\mathscr{M}_1$更细的层级，要么处于相同层级。
2. **粗化：** $\mathscr{M}_2$ 中的每个单元要么处于比 $\mathscr{M}_1$更粗的层级，要么处于相同层级。

在**细化** 阶段，每个八分体都会被标记为 REFINE 或 NO CHANGE；而在**粗化** 阶段，则会被标记为 COARSEN 或 NO CHANGE。我们将每个自适应周期中的细化和粗化限制在单个层级内（在一维/二维/三维中分别拆分/合并 2/4/8 个八分体），这在八叉树自适应网格细化实现中是标准做法 [20, 19, 18, 26]。

### *1.1.* 八叉树自适应网格细化网格间传递的守恒性质

我们简要分析细化和粗化阶段，以说明为什么守恒式粗化对于长时间模拟至关重要。

**细化是守恒的（父到子传递）：** 在细化阶段，网格间算子 $\mathbf{T}_R$ 使用连续 Galerkin 离散的基函数，将数值从父单元插值到其子单元（也称为父到子传递，即 P2C）。[1] 对于线性单元，这意味着细化引入的中点值与线性表示一致，因此所表示的场的积分得以保留。

例如，考虑一个一维线性函数 $g(x) = \alpha x + \beta$。曲线在 $[x_1, x_2]$ 区间下的面积为

$$A = \int_{x_1}^{x_2} g(x)\,dx = \frac{\alpha(x_2^2 - x_1^2)}{2} + \beta(x_2 - x_1). \tag{3}$$

在一维中，细化通过引入 $x' = (x_1 + x_2)/2$将父单元划分开。由于我们使用线性基函数进行插值，$g(x') = \alpha(x_1 + x_2)/2 + \beta$，并且

$$\begin{aligned} A &= \int_{x_1}^{x_2} g(x)\,dx = \int_{x_1}^{x'} g(x)\,dx + \int_{x'}^{x_2} g(x)\,dx \\ &= \frac{\alpha(x_2^2 - x_1^2)}{2} + \beta(x_2 - x_1), \end{aligned}$$

这与公式（3）相同。因此，细化阶段保持了场变量的积分。图1说明了$g(x) = |\cos(2\pi x)| + 10$的这一行为。在算法上，细化阶段传递是局部的：新的节点值由父单元自由度计算得到。实现细节请参阅 Saurabh 等人 [26] 。

[1] I在八叉树自适应网格细化中，较粗单元称为较细单元的父单元，每个经过细化的单元称为一个子单元。

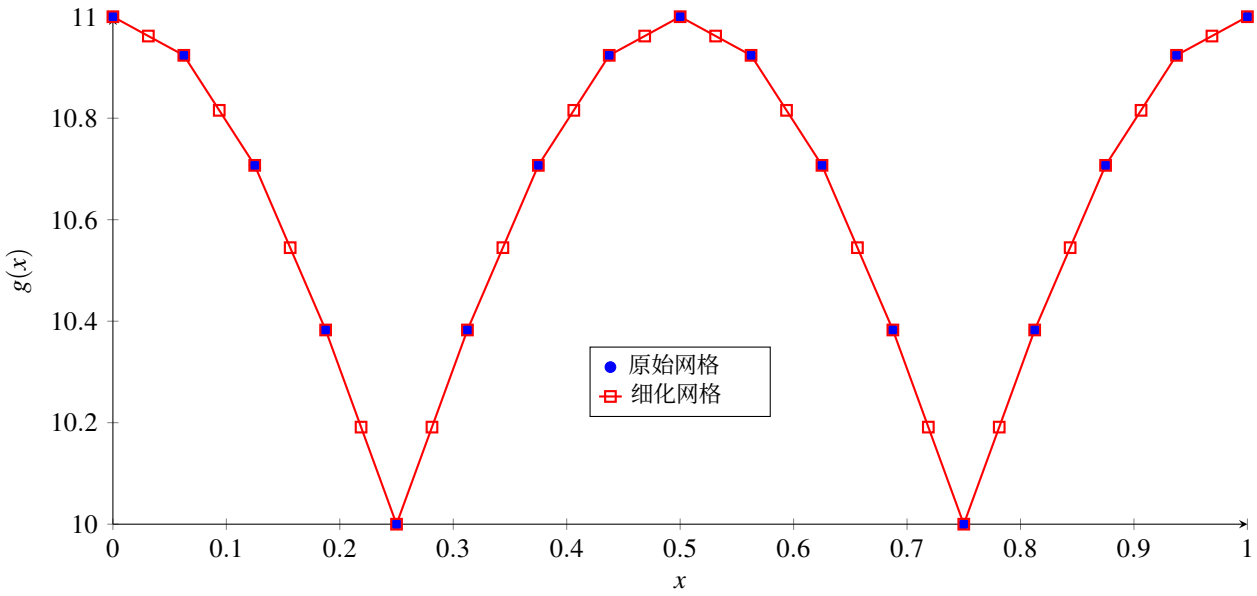



**图1. 细化：**表示细化情形的图。红色标记表示添加到细化网格中的额外点。 $\left[\int_\Omega g(x)\,d\Omega\right]_{\mathscr{M}_O} = \left[\int_\Omega g(x)\,d\Omega\right]_{\mathscr{M}_R} = 10.6284$，其中 $\mathscr{M}_O$ 是原始网格， $\mathscr{M}_R$ 是细化网格。

**基于注入的粗化并不守恒（子代到父代传递）：**粗化传递算子$\mathbf{T}_C$ 更为微妙。常用的先进方法采用基于注入的方法，其中与父节点不重合的子单元自由度会从表示中剔除（亦称子代到父代传递，即 C2P 传递）。该过程通常不守恒所表示场的积分量。

考虑定义在 $[x_1,x_3]$ 上的分段线性函数，其断点位于 $x_2$，即可看出：

$$g(x) = \begin{cases} \alpha_1 x + \beta_1 & x \in [x_1,x_2], \\ \alpha_2 x + \beta_2 & x \in [x_2,x_3]. \end{cases} \tag{4}$$

粗化通过移除内部节点 $x_2$，将两个子单元合并为一个父单元。注入通过一条新直线连接 $(x_1, g(x_1))$ 和 $(x_3, g(x_3))$

$$g_C(x) = \alpha_C x + \beta_C, \qquad x \in [x_1,x_3],$$

使得 $g_C(x_1) = g(x_1)$ 且 $g_C(x_3) = g(x_3)$，其中

$$\alpha_C = \frac{\alpha_2 x_3 + \beta_2 - \alpha_1 x_1 - \beta_1}{x_3 - x_1},$$
$$\beta_C = \frac{(\alpha_1 - \alpha_2) x_1 x_3 + \beta_1 x_3 - \beta_2 x_1}{x_3 - x_1}.$$

一般而言， $\int$x3x1 g(x)dx $\neq \int$x3x1 $g_C$(x)dx，且仅当 $\alpha_1 = \alpha_2$和 $\beta_1 = \beta_2$（即两个子单元位于同一条直线上）时等号成立。图 2 展示了对同一测试函数 g(x) $= |\cos(2\pi x)| + 10$ 而言的这一效应，其中阴影区域表示基于注入的粗化所造成的积分量损失（或增加）。

**为什么 CG 需要全局方法：**连续 Galerkin（CG）与局部离散方法（例如有限体积法或间断伽辽金（DG））的一个关键区别在于，CG 基函数的支撑跨越单元边界，具有全局性。在有限体积法或 DG 方法中，粗化时可以通过守恒地对单元值取平均，或在局部模板内将质量重新分配给相邻单元，从而在局部重新分配质量。相比之下，CG 的单元间表示具有连续性，这意味着，为了在粗化位置实现守恒而改变节点值，必然会影响多个单元所表示的场。因此，在粗化过程中消除自由度时，仅靠局部操作无法恢复 CG 的全局守恒；相反，这需要一种涉及所有自由度的机制（例如通过全局约束或投影）。

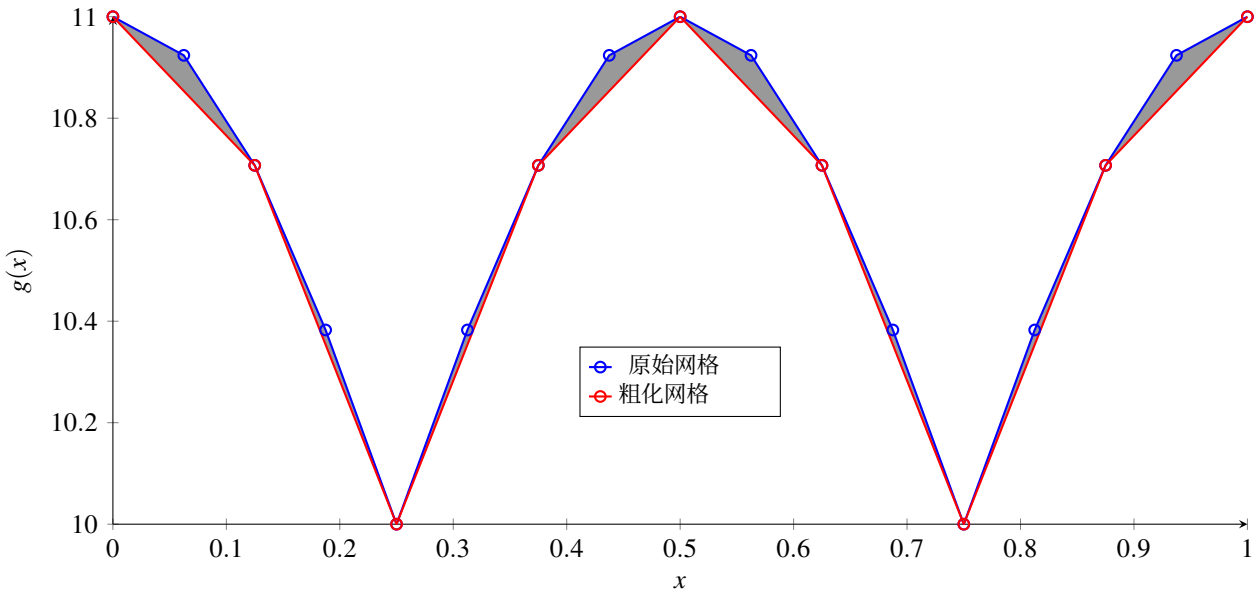



**图 2. 注入式粗化**：展示通过注入进行粗化过程的图示。灰色阴影区域表示两个网格之间差异所对应的区域。 $\left[\int_\Omega g(x)\,d\Omega\right]_{\mathscr{M}_O} = 10.6284$，而 $\left[\int_\Omega g(x)\,d\Omega\right]_{\mathscr{M}_C} = 10.6036$，其中 $\mathscr{M}_O$ 为原始网格，$\mathscr{M}_C$ 为粗化网格。

## 2. 数值过程

我们考虑一个标量场 $g(\mathbf{x})$，它由平衡自适应八叉树网格上的连续 Galerkin（CG）有限元空间表示。我们的目标是将该场从细网格传递到单层粗化后得到的网格（在一维/二维/三维中合并 2/4/8 同级八分体），同时确保积分量（例如质量）的离散全局守恒。关键难点在于，在标准的基于注入的子代到父代传递中，与粗网格不重合的自由度（DOF）会被丢弃，这通常会在反复进行自适应网格细化循环后造成守恒漂移。因此，我们构造了一种适用于八叉树自适应网格细化的守恒粗化算子，它（i）将信息守恒地传递到粗单元求积点，并且（ii）通过$L^2$ 投影（质量矩阵求解）重构粗网格节点自由度。

尽管我们使用线性拉格朗日单元和常用的高斯–勒让德模板来说明该方法，但只要用于传递和投影的求积规则与所选多项式阶一致，该过程适用于具有任意多项式次数（$P_p/Q_p, p \geq 1$）的拉格朗日基函数的任何连续 Galerkin 离散。

### *2.1.* 求积点处的离散守恒约束

设一个粗（父）单元由 $\mathrm{N_{elem},}_f$ 个细（子）单元粗化而成，其中 $\mathrm{N_{elem},}_f = 2/4/8$ 分别对应一维/二维/三维。令 $\{(\mathrm{x\,f\,i,\,j,w\,f\,i})\}^{N_{\mathrm{ip}}}{}_{,f\mathrm{i}=1}$ 表示第 j 个细单元上的求积点和权重，令 $\{(\mathrm{x}_k^c,\mathrm{w}_k^c)\}^{N_{\mathrm{ip}}}{}_{,c\mathrm{k}=1}$ 表示由此得到的粗单元上的求积点和权重。以 $\mathrm{g\,f\,(x\,f\,i,\,j)}$ 表示在细求积点处求值的细网格场，以 $\mathrm{g}_c(\mathrm{x}_k^c)$ 表示待确定的（未知）粗单元求积值。

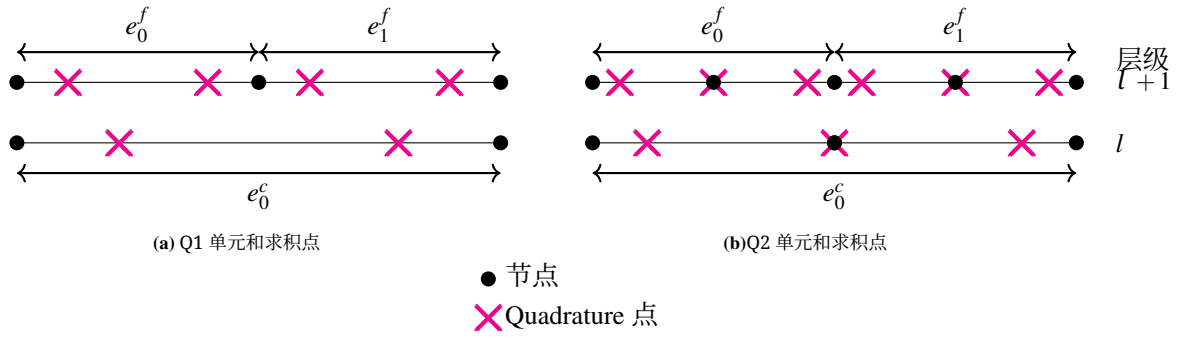
在离散情形下，父单元上积分量的全局守恒要求子单元贡献之和等于粗单元贡献（达到数值精度）：

$$\sum_{j=1}^{N_{\mathrm{elem},f}} \sum_{i=1}^{N_{\mathrm{ip},f}} w_i^f\, g^f\left(\mathbf{x}_{i,j}^f\right) = \sum_{k=1}^{N_{\mathrm{ip},c}} w_k^c\, g_c(\mathbf{x}_k^c)\,. \tag{5}$$

这本质上是将式 2应用于单个粗（父）单元的离散表述。

**求积选择与映射：**在粗化步骤中，式（5）的左侧完全由细网格表示确定。要计算右侧，我们必须在粗单元上选择一种求积规则，并确定细单元与粗网格求积点之间的映射。图 3展示了标准高斯-勒让德

一维中 $Q_1$ 和 $Q_2$ 单元的高斯-勒让德求积点和节点。我们希望将细单元（记为 单元 细 0 和 单元 细 1）的求积点处的值插值到粗单元的求积点。为实现这一目标，我们通过将细单元局部投影到粗单元上，构造一个限制算子$L^2$。




**(a)** Q1 单元和求积点　　**(b)**Q2 单元和求积点

**图 3.** 图　展示一维中两个细化层级下（a）Q1 和（b）Q2 单元的高斯求积点和节点　.

*2.2.* 一维局部 $L^2$ 投影示例

考虑一个一维定义域，其中有两个细单元（e f 0 和 e f 1），它们将粗化为一个粗单元（$e_0^c$）。设父参考单元为 $[-1,1]$，两个子参考单元为 $[-1,0]$和 $[0,1]$。假设细单元和粗单元的多项式空间均为 $Q_p$，其中 $p = 1, 2$。数值积分采用求积，细单元和粗单元分别使用 $N_{ip,f}$和 $N_{ip,c}$ 个求积点。图 3 示意了 $Q_1$和 $Q_2$单元的高斯-勒让德求积点，其中 $N_{ip,c} = N_{ip,f} = (p+1)$。

**粗表示：**我们将父单元上的粗网格场表示为：

$$g_c(x) = \sum_{j=1}^{N_{ip,c}} U_j N_j(x) \tag{6}$$

其中，$N_j(x)$ 是拉格朗日基函数，$U_j$ 是粗单元节点处的节点值。我们首先考虑标准高斯–勒让德求积这一典型情形，其中 $N_{ip,c} = N_{ip,f} = p+1$ 个求积点。此外，多项式的构造方式使节点定义在求积点处：$N_j(\mathbf{x}_k^c) = \delta_{jk}$。因此，粗自由度就是场在求积点处的值，即 $g_c(\mathbf{x}_{k,i}^c) = U_j$。我们有意采用这种选择，以便在下文定义的局部 $L^2$ 投影中得到对角质量矩阵。

**局部$L^2$ 投影定义：**给定子单元上的细单元值$g^f$，我们希望计算粗单元系数$U_j$，使细单元表示与粗单元表示之间的$L^2$ 误差最小。数学上可表述为：

$$\{U_j\}_{j=1}^{N_{ip,c}} = \arg\min_{\{U_j\}} \int_K \Big(g_f(x) - g_c(x)\Big)^2 dx, \tag{7}$$

其中$K$ 是父单元，$g^f$ 表示两个子单元上的（分片）细表示，$g_c$ 表示$K$上的粗表示。式Eq. (7) 中的最小化问题可表述为伽辽金正交条件：

$$\int_K \big(g_f(x) - g_c(x)\big)\, v(x)\, dx = 0 \qquad \forall v \in Q_p(K). \tag{8}$$

选择 $v = N_i$并使用 $g_c$ 在 公式（6） 中的离散表示，可得以下局部质量矩阵方程组

$$\sum_{j=1}^{N_{ip,c}} M_{ij} U_j = b_i, \qquad M_{ij} = \int_K N_i(x) N_j(x)\, dx, \qquad b_i = \int_K g_f(x)\, N_i(x)\, dx. \tag{9}$$

为了从子单元数据中计算$b_i$，我们将父单元划分为左、右子单元$K = K_L \cup K_R$，并通过以下方式将每个子单元的参考坐标$r \in [-1,1]$映射到父单元坐标$x \in [-1,1]$

$$x_L(r) = \tfrac{1}{2}(r-1), \qquad x_R(r) = \tfrac{1}{2}(r+1), \qquad dx = \tfrac{1}{2}\, dr. \tag{10}$$

右端项$b_i$ 可通过对每个子单元上的值求和来计算，其中节点/权重为 $\{(r_q, w_q)\}^{N_{\mathrm{ip},f}}_{q=1}$ ，从而得到

$$b_i \approx \frac{1}{2}\sum_{q=1}^{N_{\mathrm{ip},f}} w_q\, g^f_{L,q}\, N_i\big(x_L(r_q)\big) + \frac{1}{2}\sum_{q=1}^{N_{\mathrm{ip},f}} w_q\, g^f_{R,q}\, N_i\big(x_R(r_q)\big), \tag{11}$$

其中，$g^f{}_{L,q}$和 $g^f{}_{R,q}$分别表示左右子单元上第$q$个高斯点处的细网格值。最后，质量矩阵 $M_{ij}$ 可计算为：

$$M_{ij} \approx \sum_{k=1}^{N_{\mathrm{ip},c}} w_k\, N_i(\mathbf{x}^c_k) N_j(\mathbf{x}^c_k) = w_i\, \delta_{ij}, \tag{12}$$

我们注意到，在求积点处具有拉格朗日性质的基函数选择会产生对角质量矩阵。结合公式（11）和公式（12），无需求解线性系统即可直接计算粗单元系数。

**一维的限制矩阵形式：**上述过程可以用矩阵–向量形式简洁地表示。将两个子单元上的细网格值收集到一个向量中

$$g^f := \big[g^f_{L,1}, \ldots, g^f_{L,N_{\mathrm{ip},f}},\, g^f_{R,1}, \ldots, g^f_{R,N_{\mathrm{ip},f}}\big]^T \in \mathbb{R}^{2\mathrm{N}_{\mathrm{ip},f}}.$$

随后，投影限制可写成矩阵–向量乘积

$$g_c = R^{1D}\, g^f, \qquad g_c = [g_{c_1}, \ldots, g_{c_{N_{\mathrm{ip},c}}}]^T \in \mathbb{R}^{N_{\mathrm{ip},c}}, \tag{13}$$

其中，限制矩阵 $R^{1D} \in \mathbb{R}^{N_{\mathrm{ip},c} \times 2N_{\mathrm{ip}f}}$ 的元素为

$$R^{1D}_{i,(c,q)} = \frac{1}{w_i}\left(\frac{1}{2} w_q\right) N_i\big(x_c(r_q)\big), \qquad c \in \{L, R\},\ q = 1, \ldots, N_{\mathrm{ip},f}. \tag{14}$$

这里，$x_L(r) = \frac{1}{2}(r-1)$ 和 $x_R(r) = \frac{1}{2}(r+1)$ 是子单元到父单元映射，因子 12 是雅可比 $dx = \frac{1}{2}dr$。

### *2.3.* 扩展到任意积分阶数

某些应用需要更多的积分点（$\geq (p+1)$），以获得更高精度（例如，[27]）。上述构造可以推广到任意积分阶数，并允许细网格单元和粗网格单元使用不同数量的求积点。局部 $L^2$ 投影框架保持不变：我们仍通过求解局部质量矩阵方程组 公式（9）来计算粗网格的求积值，其中右端项 $b_i$ 通过采用适当的求积规则，对所有子单元的贡献求和得到。唯一的区别是，质量矩阵 $M_{ij}$ 不再像 公式（11）中那样是对角矩阵，因此我们必须求解线性系统以获得粗网格的求积值。我们还注意到，对于给定的基函数和求积规则，这些量可以预先计算，并且不依赖于实际的细网格数据，而取决于求积点的位置和权重。限制矩阵形式 公式（13）–公式（14） 仍然有效，只需修改矩阵元素，以考虑细网格单元和粗网格单元中求积点数量的不同。本文其余部分将重点讨论在 $(p+1)$ 个求积点下，对 $Q^p$ 单元使用标准高斯–勒让德求积的情况。

### *2.4.* 扩展到多维

一维限制矩阵$R^{1D}$ 定义于 式 (14) 可通过张量积扩展到多维。例如，在二维中，限制矩阵为

$$R^{2D} = R^{1D} \otimes R^{1D}, \tag{15}$$

其中 $\otimes$表示克罗内克积。$R^{2D}$ 的元素可明确写为

$$R^{2D}_{(i_1,i_2),(c_1,q_1,c_2,q_2)} = R^{1D}_{i_1,(c_1,q_1)}\, R^{1D}_{i_2,(c_2,q_2)}, \tag{16}$$

其中 $i_1, i_2 = 1$，…，$N_{\mathrm{ip},\ c}$、$c_1, c_2 \in \{L,\ R\}$以及 $q_1, q_2 = 1$，…，$N_{\mathrm{ip},\ f}$。多维限制矩阵则为

$$g_c = R^{2D}\, g^f, \tag{17}$$

其中，$g^f$ 按字典序收集了所有子单元上的细网格值。需要指出的是，我们无需显式构造多维限制矩阵；相反，可以在限制操作过程中即时应用克罗内克结构。

*2.5.* 通过全局 $L^2$ 投影恢复粗网格节点自由度

上述得到的值 $g_c(\mathbf{x}_j^c)$位于粗单元的求积点上。为获得粗网格节点自由度，我们计算 $L^2$ 投影，将$g_c$ 投影到粗网格连续 Galerkin 有限元空间中。为组装式（20）中的质量矩阵和右端项，积分采用与定义粗网格求积点值 $g_c$ (式（14）)时相同的求积规则。

令$V_h \subset L^2(\Omega)$ 表示由拉格朗日基函数 $\{N^a\}_{a=1}^{n_h}$（次数为 $p \geq 1$）构成的粗网格连续伽辽金有限元空间）。我们寻求$g^h \in V_h$，使其相对于求积点场 $L^2$ 误差的 $g_c$：

$$g^h = \arg\min_{v^h \in V_h} \frac{1}{2} \|v^h - g_c\|^2_{L^2(\Omega)}. \tag{18}$$

相应的变分条件为：求 $g^h \in V_h$，使得

$$(g^h, w^h) = (g_c, w^h) \qquad \forall w^h \in V_h, \tag{19}$$

其中 $(\cdot,\cdot)$ 表示标准的 $L^2$ 内积。令 $g^h(\mathbf{x}) = \sum_{a=1}^{n_h} G_a N_a(\mathbf{x})$，即可得到熟知的质量矩阵系统

$$\mathbf{M}\mathbf{G} = \mathbf{b}, \qquad M_{ab} = (N_a, N_b), \qquad b_a = (g_c, N_a), \tag{20}$$

求解该系统可得到粗网格上的节点系数 $\mathbf{G} = \{G_a\}$。在实际应用中，右端项 $(g_c,N_a)$ 使用粗网格求积点及其对应的值 $g_c(\mathbf{x}_j^c)$ 进行组装。需要注意的是，尽管质量矩阵 $\mathbf{M}$ 的维度仅取决于粗网格自由度的数量，但组装 $\mathbf{M}$ 和 $\mathbf{b}$时的积分必须使用与计算粗网格点值 $g_c$时相同的求积规则。

**解释：** 整个过程可以理解为对守恒量进行全局重新分配：粗网格节点值在局部可能不同于注入所得的值，但积分约束方程（5）按构造即可满足。这是预期的，因为表示这些量的基函数是全局且连续的。 图 4展示了将$g(x) = |\cos(2\pi x)| + 10$从细网格传递到粗网格时的这一效果：局部节点值有所不同，而全局积分则保持在数值精度范围内不变。类似地，图 5 比较了标准的基于注入的粗化与针对二次基函数提出的守恒粗化。基于注入的粗化会导致明显的质量漂移，而所提出的守恒格式则能将质量守恒保持在数值精度范围内。

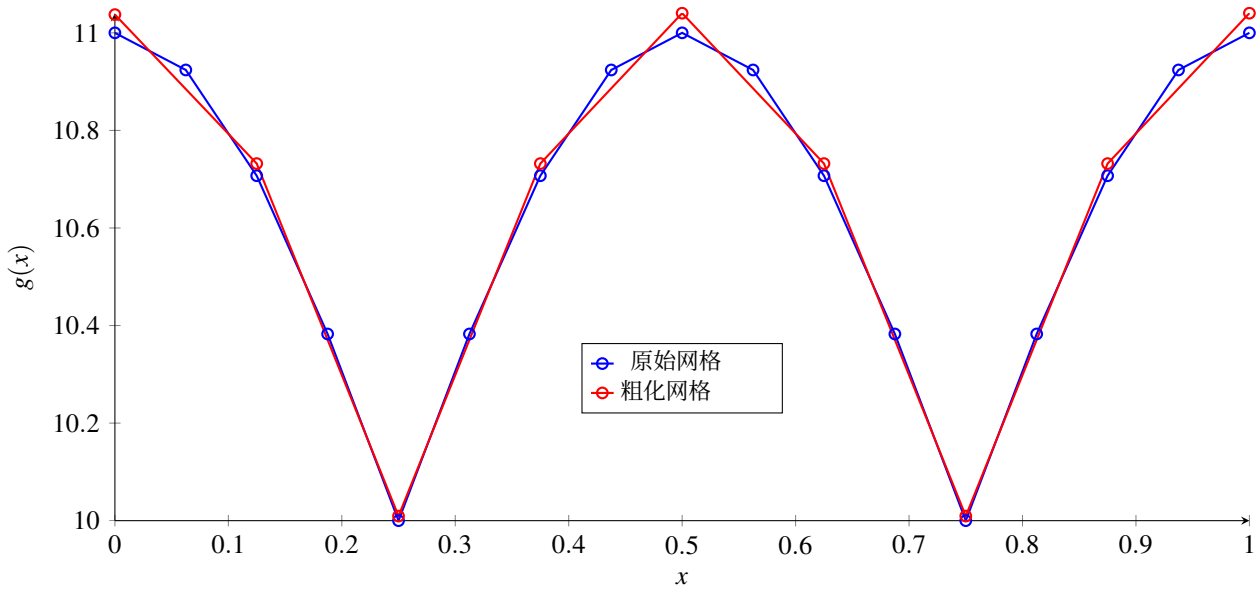
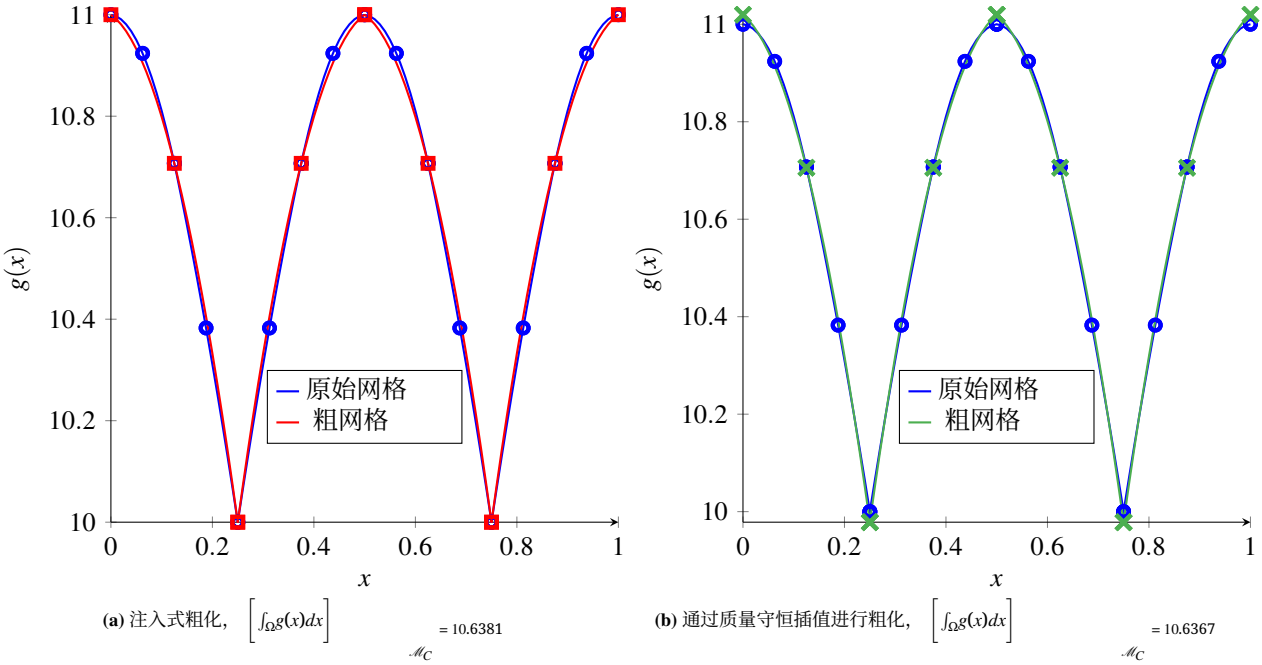



**图 4. 质量守恒插值：**示意图展示了将解从细网格投影到粗网格时 $\phi$ 的重新分布，使得 $\int_\Omega g(x)d\Omega$ 保持不变。 $\left[\int_\Omega g(x)\, d\Omega\right]_{\mathscr{M}} o = \left[\int_\Omega g(x)\, d\Omega\right]_{\mathscr{M}} c = 10.6284$，其中 $\mathscr{M}_O$ 为原始网格， $\mathscr{M}_C$ 为粗化网格。




**(a)** 注入式粗化，$\left[\int_\Omega g(x)dx\right]_{\mathscr{M}_C} = 10.6381$

**(b)** 通过质量守恒插值进行粗化，$\left[\int_\Omega g(x)dx\right]_{\mathscr{M}_C} = 10.6367$

**图 5. 使用二次基函数进行插值**：该图比较了使用 (a) 注入和 (b) 质量守恒插值，将定义在含有 8 个二次单元的细网格上的函数 $g(x) = |\cos(2\pi x)| + 10$ 转移到含有 4 个二次单元的粗网格的情况。原细化网格上函数在定义域 $\Omega = [0,1]$ 上的积分（=10.6367）通过所提出的质量守恒插值方案得以保留，精度达到数值精度；而注入则导致了明显偏差。

## 3. 算法

本节将介绍在自适应网格细化的粗化阶段执行质量守恒插值所需的算法步骤，该插值基于第 2 节概述的数值过程。请注意，细化阶段和粗化阶段互斥；此处我们仅关注粗化阶段，即当前标记的单元仅包含粗化或 不变 标志。要对单元进行粗化，特定父单元的所有子单元都必须满足以下两个条件：

1. 所有子单元必须处于同一层级。
2. 所有子单元都必须带有粗化标志。

图 6 概述了粗化所需的步骤。被归类为需要粗化的单元，其主要步骤如下：

1. **高斯点插值：** 使用相应的基函数，对每个单元的节点值进行插值，得到高斯点处的值。2. **传递到较粗单元：** 获得所有子单元的高斯点处的值后，需要将这些值传递到较粗的父单元。为此，我们按照 $L^2$ 中所述，对细网格到粗网格进行局部算法 1投影。首先，将所有子单元的高斯点处的值存入同一个向量。子单元按莫顿顺序排列，而每个单元内的积分点则按字典序排列。接着，进行局部$L^2$ 投影，将值从细网格投影到粗网格，从而求得较粗单元的高斯点处的值。图 6展示了这一过程在二维情况下的示意图。3. $L^2$ **投影：** 完成上一步后，我们便得到较粗单元的高斯点处的值。为确定节点处的值，我们按照 $L^2$ 中所述，进行从高斯点到节点的全局第 2 节投影。

接下来，我们将详细说明执行质量守恒插值所需的算法和数据结构。首先，为细化前（$\mathscr{M}_1$）和细化后

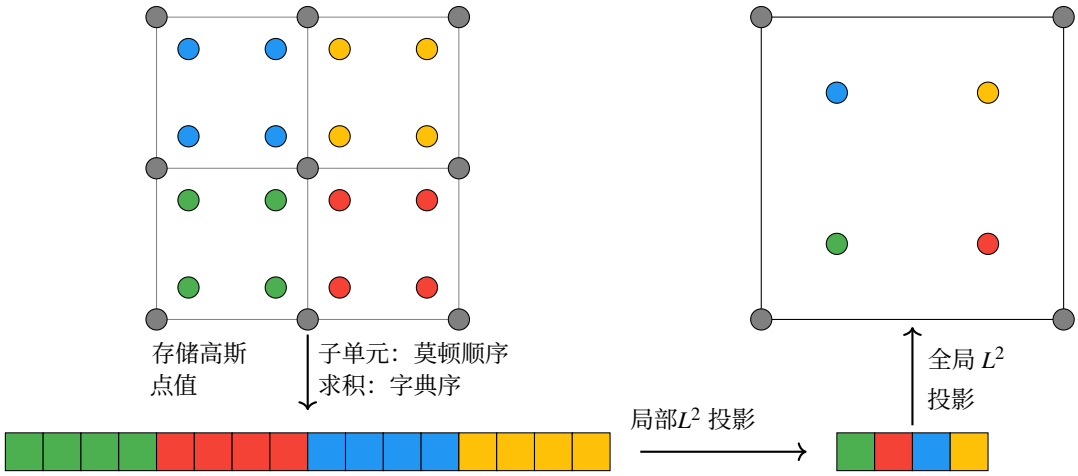



**图 6.** 插值格式：细网格中的单元（左图）粗化一次后，得到右图所示的单个单元。我们首先将所有高斯点处的值（二维中为 4 个，三维中为 8 个）存储在一个向量中。子单元按莫顿顺序排列，而每个单元内的积分点则按字典序排列。随后，使用局部 $L^2$ 投影（如算法 1所述）对这些高斯点处的值进行插值。最后，我们执行从高斯点到节点的 $L^2$ 投影，得到粗网格节点处的最终值。

的 AMR 步骤（$\mathscr{M}_2$）中的网格分别分配两个大小为 elem size $\mathscr{M}_2$ngp 的向量。这里，elem size 表示单元数量，ngp 表示用于评估场变量的高斯点数量。我们假设（不失一般性）每个单元使用相同数量的高斯点。基本上，$N_{ip,\,c} = N_{ip,\,f} = \text{ngp} = (p+1)^{dim}$。

我们遍历原始网格中的每个高斯点 $\mathscr{M}_1$，并将其对应的值存储在向量中。对于未经历粗化阶段的单元，其关联值会被复制到网格对应的向量 $\mathscr{M}_2$中。但如果单元经历了粗化，我们就按照$L^2$ 投影的描述，从细网格到粗网格计算局部投影，以算法 1 所述的方法，将值从 $\mathscr{M}_1$ 向量传递过来。此操作完全是局部的，对每个粗化单元而言彼此独立，仅涉及子单元及其在细单元中的对应条目。随后，将局部投影的结果存储在粗单元向量中的适当位置。这一步可视为在单元级传递值（单元级网格间传递）。获得较粗网格的高斯点处的值后 $\mathscr{M}_2$，我们继续通过求逆质量矩阵，执行从高斯点到节点的$L^2$ 投影，如第 2 节所述。

## 4. 结果

本节将所提出的场守恒八叉树粗化算子与标准的基于注入的子代到父代传递进行比较。首先，我们进行人工解测试，以验证收敛性和守恒性质，详见第 4.1 节。接下来，我们分别在第 4.2 节和第 4.3 节中，评估这些方案在由 Cahn–Hilliard 方程和 Cahn–Hilliard–Navier–Stokes 方程所描述的相分离现象中的表现。我们使用多项式自由能和 Flory–Huggins 自由能，研究由 Cahn–Hilliard（CH）方程描述的旋节线分解，并进一步将评估扩展到一个具有代表性的 Cahn–Hilliard–Navier–Stokes（CHNS）多相流测试案例。整个过程中，网格均采用平衡八叉树上的单层级细化/粗化进行自适应调整，因此不同运行结果之间的差异主要可归因于粗化传递算子。

### *4.1.* 收敛误差：制造解法

为验证所提出的守恒粗化算子的精度，我们首先使用制造解法（MMS）进行收敛性研究。我们考虑扩散方程：

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (\kappa \nabla \phi) \tag{21}$$

在定义域 $\Omega = [0,1]^2$ 上，具有齐次诺伊曼边界条件：

$$\nabla \phi \cdot \hat{n} = 0 \quad \text{on } \partial\Omega, \tag{22}$$

**算法 1** 单个粗化单元上从细网格到粗网格的局部限制[2]

**要求：** 维度$dim \in \{2,3\}$，每个维度上的积分点数$n$
**要求：** 一维限制矩阵$R^{1D}$
**要求：** 一个子单元上的高斯点值 $c$，层级为 $l+1$：$g_f$ [f itg],f itg $= 0,\ldots,N_{ip,\ f} - 1$
**要求：** 子单元编号：child $\in \{0,\ldots,2^{dim}-1\}$ ▷ 莫顿顺序中的子单元索引
**确保：** 父单元上的粗网格采样值：$U$[c dof],c dof $= 0,\ldots,N_{ip,c} - 1$
$(c_x,c_y,c_z) \leftarrow$ 莫顿解码(child,dim) ▷算法3
$U \leftarrow 0$ ▷ 将粗网格高斯点值初始化为零
**对于** c itg $= 0$ 至 $N_{ip,c} - 1$ **执行** ▷ 字典序排列的粗点高斯点值
    $(I_x,I_y,I_z) \leftarrow$ 字典序解码(c itg,dim,n) ▷算法 2
    acc $\leftarrow 0$
    **对** f itg $= 0$ 至 $N_{ip,f} - 1$ **执行** ▷ 按字典序排列的细点高斯点
        $\ell = (\ell_x,\ \ell_y,\ \ell_z) \leftarrow$ 字典序解码(f itg,dim,n) ▷算法 2
        w $\leftarrow 1$
        **对** d $\in$ {x,y,z} **执行**
            $i_d \leftarrow c_d n + \ell_d$
            w $\leftarrow$ w $\cdot R^{1D}$ [I d] [i d] ▷ 克罗内克积
        acc $\leftarrow$ acc $+ w \cdot g^f$ [f itg]
    $U$[c itg] $\leftarrow U$[c itg] + acc
**返回** $U$

**算法 2**字典序解码：将字典序索引解码为多维坐标

要求：字典序索引 lex idx 要求：维度
dim $\in \{2,3\}$要求：每个维度的点数：n
确保：坐标 ($I_x$, $I_y$, $I_z$) 1: 如果 dim $= 2$ 则
2: $I_x \leftarrow$lex idx mod n 3: $I_y \leftarrow \lfloor$lex idx/
n$\rfloor$4: $I_z \leftarrow 0$5: 否则如果 dim $= 3$ 则 6: $I_x \leftarrow$
lex idx mod n 7:$\div I_y \leftarrow \lfloor$(lex idx/n)$\rfloor$
mod n 8: $I_z \leftarrow \lfloor$lex idx/$n^2\rfloor$9: 返回 ($I_x$, $I_y$,
$I_z$)

其中 $\kappa$ 为扩散系数。我们选择的人工解为：

$$\phi(x,y,t) = 1 + A\cos(2\pi x)\cos(2\pi y)e^{-\kappa 8\pi^2 t} \tag{23}$$

参数为 $A = 0.1$ 和 $\kappa = 0.03$。上述偏微分方程的解满足质量守恒，即 $\int_\Omega \phi d(\Omega)$ 随时间保持不变，并等于初始质量 $\int_\Omega \phi(t = 0)d\Omega = 1$。

为对偏微分方程进行数值模拟，我们采用二阶 Crank-Nicholson 时间积分，线性基函数的时间步为 $\Delta t = 0.01$，二次基函数的时间步为 $\Delta t = 0.001$。我们为二次基函数选用较小的时间步，以确保时间误差相较于空间误差可以忽略。最终时间设为 $t_f = 1.0$。

为进行收敛性研究，我们考虑不同八叉树层级下的一系列均匀细化网格。在每一步中，我们计算解 $\nabla\phi$ 的局部梯度，并根据以下准则调整网格：

$$\eta_e = \int_{\Omega_e} \|\nabla\phi\|_{L_2(e)} \mathrm{d}\Omega_e < \tau, \tag{24}$$

其中，$\tau$ 是用户定义的网格相关参数，其随网格尺寸 h 缩放。表 1 报告了不同层级下的取值。我们选择 $\eta_e$ 作为一种简单的启发式方法，该方法基于在求积点处计算的导数。满足方程（24）的单元将被粗化，而其他单元则标记为不变。此外，在每次迭代中，我们将粗化

**算法3** 莫顿解码：将莫顿顺序（Z序）索引解码为子坐标

**输入：** 莫顿顺序中的子单元索引 child $\in \{0,\ldots,2^{dim}-1\}$
**输入：** 维度 $dim \in \{2,3\}$
**输出：** 子坐标 $(c_x,\ c_y,\ c_z) \in \{0,1\}^{dim}$

$c_x \leftarrow$ 子单元 & 1 ▷ 提取位 0
$c_y \leftarrow$ (子单元 & 2) $\gg 1$ ▷ 提取位 1
**如果** 维度 $= 3$ **则**
    $c_z \leftarrow$ (子单元 & 4) $\gg 2$ ▷ Extract bit 2
**否则**
    $c_z \leftarrow 0$
**返回** $(c_x,c_y,c_z)$

按方程（24）排序，并将取值最低的单元粗化，以此选取总单元中固定百分比（10%）的单元。在每次模拟中，我们考虑层级为$l$或$l-1$的八叉树单元。若层级为$l$的八分体满足上述条件，则将其粗化至层级$l-1$；而已处于层级$l-1$的八分体保持不变。这样可确保我们执行足够多的粗化操作，以评估所提出的格式。在整个模拟过程中，我们在每个时间步都执行此自适应过程。

图 7 显示了 $L_2$ 在 $\phi$ 最终时刻 $t_f = 1.0$ 线性基函数和二次基函数的误差。我们没有观察到两种插值方案的收敛率有显著差异。两种方案均呈现出预期的最优收敛率：线性基函数约为 2，二次基函数约为 3。我们观察到两种方案的误差幅值略有差异：对于线性基函数，所提出的守恒粗化算子产生的误差略低；对于二次基函数，误差则几乎相同。这表明，所提出的守恒粗化算子在网格自适应过程中不会引入额外误差，同时能够保持数值解的预期精度。

| 层级 | $\tau$ | 线性 | | 二次 | |
|---|---|---|---|---|---|
| | | 注入 | 守恒 | 注入 | 守恒 |
| 5 | $1\times10^{-2}$ | $3.72\times10^{-7}$ | $2.08\times10^{-13}$ | $1.71\times10^{-10}$ | $3.60\times10^{-13}$ |
| 6 | $5\times10^{-3}$ | $1.97\times10^{-9}$ | $8.84\times10^{-13}$ | $9.75\times10^{-12}$ | $1.16\times10^{-12}$ |
| 7 | $2.5\times10^{-3}$ | $6.39\times10^{-10}$ | $3.83\times10^{-13}$ | $4.00\times10^{-12}$ | $9.75\times10^{-13}$ |
| 8 | $1.25\times10^{-3}$ | $1.79\times10^{-10}$ | $2.75\times10^{-13}$ | – | – |

**表 1.** 采用注入和守恒粗化方案时，MMS 测试算例在最终时刻 $t_f = 1.0$ 不同八叉树层级下的质量漂移。

我们最终计算 $\phi$ 中最终时刻 $t_f = 1.0$ 的质量漂移。质量漂移定义为：

$$\Delta\phi = \left| \int_\Omega \phi(t = t_f)d\Omega - \int_\Omega \phi(t = 0)d\Omega \right|$$

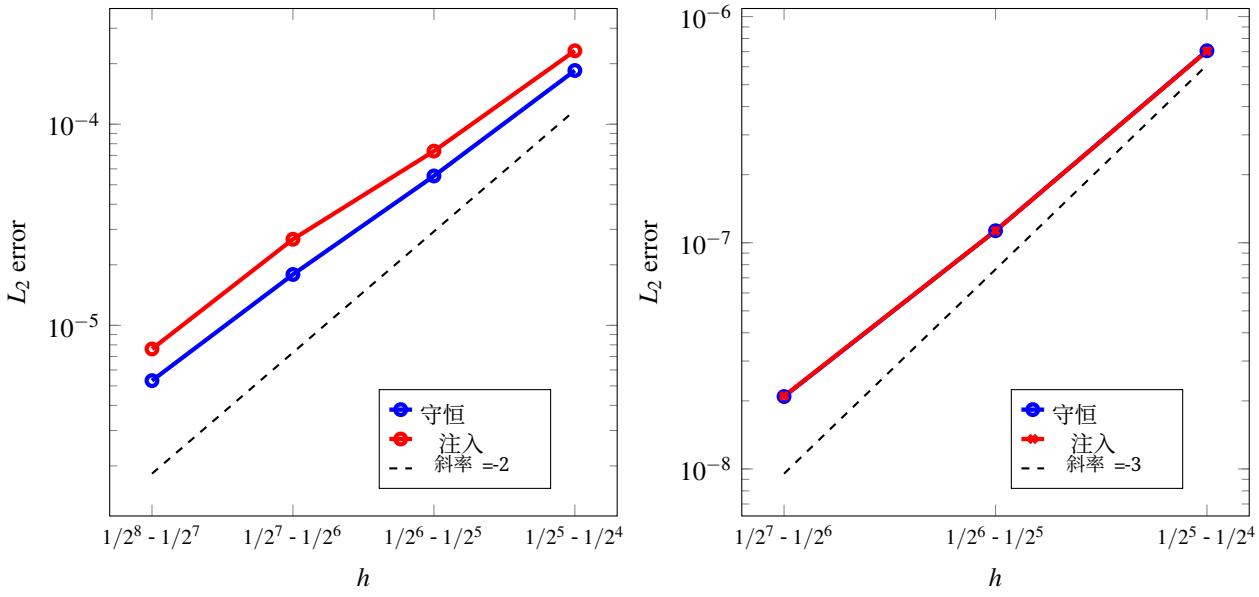
这种质量漂移可归因于网格自适应过程中执行的粗化操作。在没有进行任何自适应的均匀网格上，数值解的守恒误差仅达机器精度 $\int\phi$。然而，执行粗化时，基于注入的方案无法保证质量守恒，因此会产生非零质量漂移。表 1报告了不同八叉树层级下最终时刻 $t_f = 1.0$ 的质量漂移。我们观察到，质量漂移随网格细化而减小，这表明基于注入的方案在网格细化后精度更高。相比之下，所提出的守恒粗化算子可确保每一步粗化操作都严格满足质量守恒，从而避免质量随时间漂移。

#### *4.2. Cahn–Hilliard方程*

我们从 Cahn–Hilliard方程开始。这是一个用于模拟二元混合物相分离的刚性四阶非线性偏微分方程：

$$\frac{\partial\phi}{\partial t} = \nabla\cdot\left(M(\phi)\nabla\left(\frac{\partial f}{\partial\phi} - \varepsilon^2\nabla^2\phi\right)\right), \tag{25}$$

其中 $\phi(\mathbf{x},t)$ 为浓度场，$M(\phi)$ 为迁移率，$f(\phi)$ 为混合自由能，$\varepsilon$控制弥散界面厚度。




**(a)** 线性基函数 **(b)**二次基函数

**图 7. MMS误差：**比较线性和二次基函数的$L_2$ 误差的图。每次运行包含两个不同分辨率的网格，两者相差一个层级，如$h$的范围所示。两种粗化方案均表现出近似最优的收敛率；其中，所提出的质量守恒方案在线性基函数下的误差略低，在二次基函数下的误差则相当。

我们关注旋节线分解，其中初始充分混合的状态会自发分离为各组分富集的区域。为量化热力学状态，我们使用金兹堡–朗道能量泛函

$$E(\phi) = \int_\Omega \left( f(\phi) + \frac{1}{2}\varepsilon^2 |\nabla \phi|^2 \right) d\Omega, \tag{26}$$

whe其中，第一项是体相自由能，第二项是界面（梯度）贡献 .

CH动力学具有两个关键性质，我们用它们来评估自适应网格细化下的数值行为：

1. 总质量守恒：$\int_\Omega \phi \, d\Omega$ 在时间上保持恒定。

2. 能量单调衰减：$E(\phi)$ 随时间单调递减。

**自由能模型：** 我们考虑 (i) 多项式双阱自由能

$$f(\phi) = \frac{1}{4}(1-\phi^2)^2, \tag{27}$$

f或满足 $\phi \in [-1,1]$，以及 (ii) 对数型 Flory–Huggins 自由能（见 第 4.2.4 节 ）.

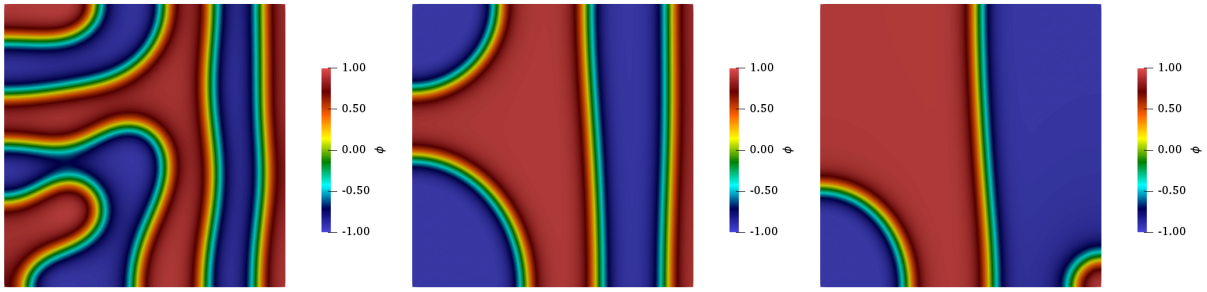
**数值离散：** 为避免直接离散四阶算子，我们采用经典的分裂策略 [29]，通过引入化学势 $\mu$:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \nabla \cdot \big( M \nabla \mu \big), \\ \mu &= \frac{\partial f}{\partial \phi} - \varepsilon^2 \nabla^2 \phi. \end{aligned} \tag{28}$$

空间离散采用线性拉格朗日基函数的连续伽辽金有限元，时间积分采用后向 Euler 法。

**八叉树自适应网格细化策略：** 准确解析弥散界面对于捕捉正确的相分离动力学至关重要。因此，我们采用两级八叉树自适应网格细化策略，包括精细的界面 分辨率$\mathscr{R}_i$ 和粗略的体相 分辨率 $\mathscr{R}_b$ ($\mathscr{R}_i > \mathscr{R}_b$)。[2] 满足 $|\phi| \le \delta$ 的单元被标记为界面单元并保持

[2]层级为$l$ 的八叉树单元对应于 $\Delta x = \Delta y = \Delta z = L/2^l$，其中$L$ 是定义域的区域长度。




**(a)** $t = 0.1$ **(b)** $t = 1.0$ **(c)** $t = 3.0$

**图 8.** 均匀随机混合物在 $\phi_0 = 0$附近受到扰动时，相场变量 $\phi$ 随时间的演化。

在最精细层级 $\mathscr{R}_i$，其中 $\delta$ 是一个用于控制界面带宽的用户参数。[3] 在八叉树上执行平衡的单层细化/粗化。

所有模拟均使用大规模并行框架 `Dendrite-kt`，该框架使用 `Dendro-kt`生成八叉树网格，并使用 `PETSc` 进行线性代数运算。若干多物理场应用的实现细节、扩展性分析和验证结果见 [27, 26, 18, 28]。Saurabh 等人 [26]所述的网格间传递算子构成了基于注入方案的基准，并为第 3 节提出的守恒粗化算子奠定了基础。

*4.2.1.* 二维旋节线分解（多项式自由能）与线性基函数

我们考虑 $\Omega = [0,1]^2$，其初始条件为 $\phi(\mathbf{x},\ 0) = \phi_0 + \eta(\mathbf{x})$，其中 $\phi_0 = 0$，且 $\eta$ 为均值为零、最大振幅为 0。1 的均匀随机扰动。我们设定 $\varepsilon^2 = 0.001$，并采用恒定迁移率 $M = 1$。时间步为 $\Delta t = 5\times10^{-4}$。我们选择体相细化 $\mathscr{R}_b = 4$ 和界面细化 $\mathscr{R}_i = 8$，并改变 $\delta \in [0.90, 0.98]$以研究界面带宽的敏感性。

图8 展示了界面演化：初始混合态分离为两个相；随着时间推移，其中一相形成岛状结构，随后发生颈缩断裂并合并。 图9 显示八叉树网格在层级 $\mathscr{R}_i = 8$ 上解析界面，同时将体相保持在层级 $\mathscr{R}_b = 4$。

图10 比较了不同粗化方案的守恒性和能量行为。我们通过以下方式量化质量漂移：

$$\Delta m(t) = \int_\Omega \phi(\mathbf{x},t)\,d\Omega \;-\; \int_\Omega \phi(\mathbf{x},0)\,d\Omega. \tag{29}$$

如图图10a所示，所提出的格式对所有 $\delta$都将质量保持到数值精度，而注入则产生随时间累积的非零漂移。总能量$E(\phi)$ 对两种格式均表现出几乎相同的单调衰减（图10b），表明守恒粗化不会降低物理耗散行为。

为了单独考察效 应对粗化传递对能量一致性的影响，我们还计算绝对能量失配 h 在每次粗化 事件：

$$\Delta E(t) = \left|E(\phi)_{\mathscr{M}_1} - E(\phi)_{\mathscr{M}_2}\right|, \tag{30}$$

其中 $\mathscr{M}_1$ 与 $\mathscr{M}_2$ 分别表示紧邻粗化之前和之后的网格。图10c表明，所提出的格式得到的 $\Delta E$ 始终比注入小近两个数量级，反映出粗化下传递场的保真度有所提高。

我们进一步计算了两种方案随时间变化的单元数量（图1），发现单元数量几乎相同，这表明守恒性和能量行为的差异并非由网格尺寸不同造成。较高的 $\delta$ 会导致更守恒的粗化，因此无论采用何种粗化方案，单元数量都相对较多。总体而言，两种方案产生的网格自适应模式相似，但与基于标准注入的方法相比，所提出的守恒粗化算子能更好地保持质量和能量一致性。

[3]例如，$\delta = 1$ 将整个定义域细化到 $\mathscr{R}_i$，而 $\delta = 0$ 将定义域保持在 $\mathscr{R}_b$。

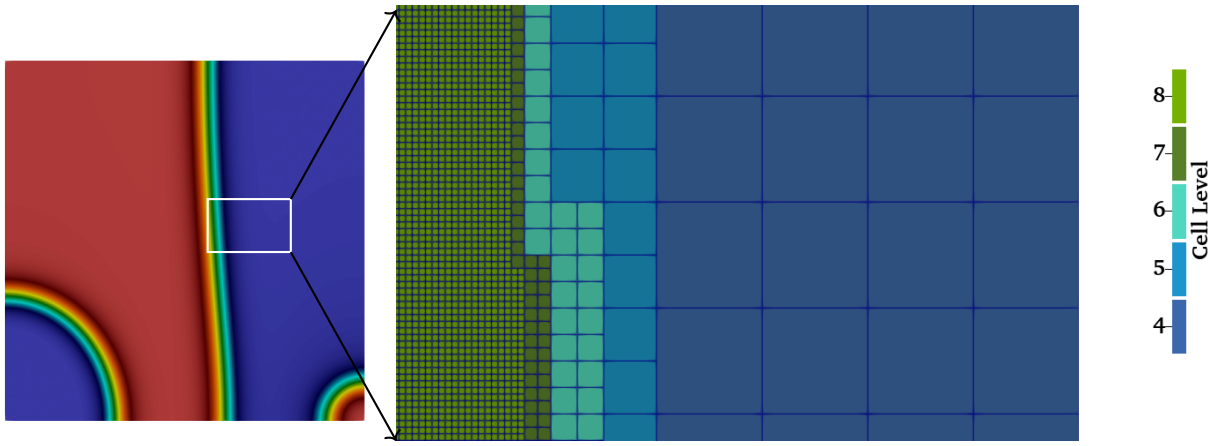



**图9**。 放大的网格视图，显示细化情况

*4.2.2.* 二次基函数下的二维旋节线分解（多项式自由能）

我们进一步测试了采用二次基函数进行空间离散的所提出的格式。我们考虑与第 4.2.1 节中相同的设置，其中 $\Omega = [0,1]^2$、 $\varepsilon^2 = 0.001$、$M = 1$、 $\Delta t = 5 \times 10^{-4}$、 $\mathscr{R}_b = 3$、 $\mathscr{R}_i = 7$以及 $\delta = 0.90$。图11显示了与线性基函数情形一致的结果。所提出的格式将质量保持在数值精度范围内（图11a），而注入表现出显著的质量漂移。各格式的总能量衰减相近（图11b），但对于所提出的方法，粗化引起的能量失配 $\Delta E$ 显著更小（图11c），表明其在粗化过程中能够实现更准确的传递。

*4.2.3.* 三维旋节线分解（多项式自由能）

我们将第 4.2.1 节扩展 $\Omega = [0,1]^3$，参数值保持不变，但将最细的界面细化设为 $\mathscr{R}_i = 7$（其中 $\mathscr{R}_b = 4$），以控制自由度总数。我们考虑 $\delta = 0.90$。图12展示了界面演化：早期形成小岛状结构（图12a），随后粗化为较大的相区（图12c）。

图13显示了与二维情形一致的趋势。所提出的格式将质量保持在数值精度范围内（图13a），而注入会导致显著的质量损失，且损失随时间累积。不同格式的全局能量衰减趋势相近，但所提出方法的粗化引起的能量失配 $\Delta E$ 显著较小（图13c），表明其在粗化过程中的传递更为准确。

*4.2.4. Flory–Huggins* 自由能

接下来，我们使用对数型 Flory–Huggins 形式，测试自由能模型的稳健性

$$f(\phi) = A\left(\phi \log(\phi) + (1-\phi)\log(1-\phi)\right) + \chi_{12}\,\phi(1-\phi) + \beta\left(\frac{1}{\phi} + \frac{1}{1-\phi}\right), \tag{31}$$

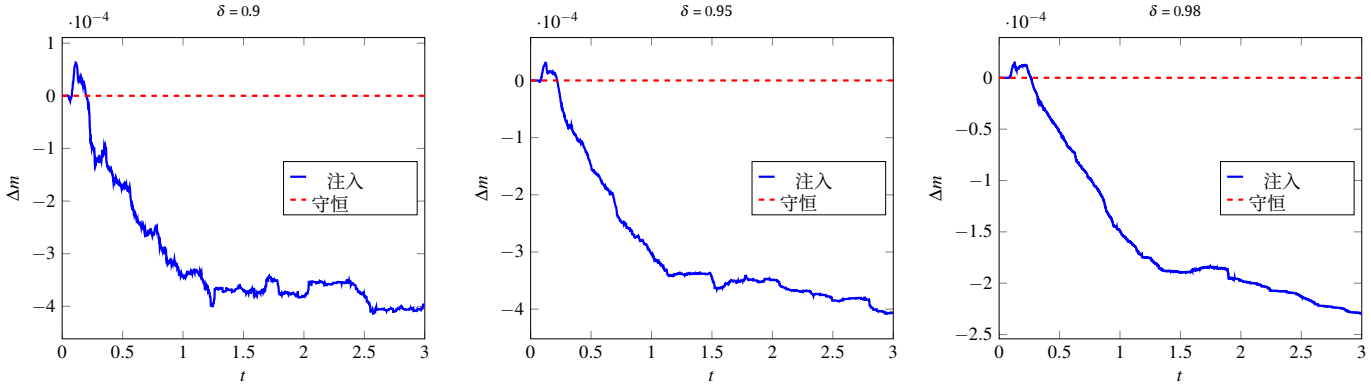
其中 $\chi_{12}$ 是 Flory–Huggins 相互作用参数， $\beta$ 是松弛参数。这里， $\phi \in [0,1]$，其中 $\phi = 0$ 和$\phi = 1$ 表示纯相（与 式（27）相对）。

我们采用 $A = 1$, $\chi_{12} = 3.0$, $\beta = 0.01$ 和 $\varepsilon^2 = 10^{-3}$。初始条件为均值 0.5、波动幅度为 0.01 的随机混合物。取 $\Delta t = 5 \times 10^{-4}$，并将界面带定义为 $\delta \in (0.30, 0.70)$。与前面一样，$\mathscr{R}_b = 4$，其中二维时为 $\mathscr{R}_i = 8$，三维时为 $\mathscr{R}_i = 7$。

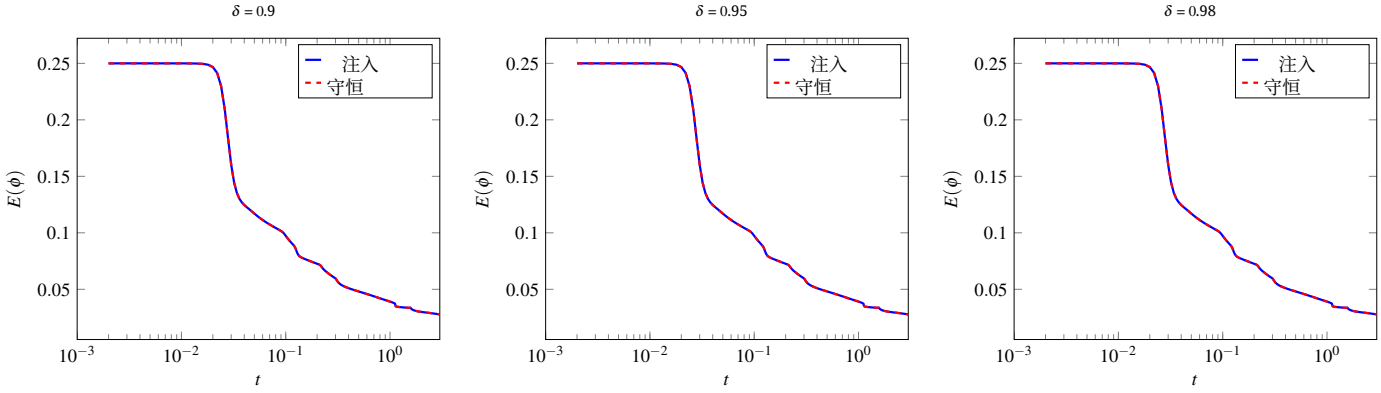
图14 和 图15 分别展示了二维和三维的结果。与多项式情形一样，定性结论相同：所提出的格式将质量守恒至数值精度，而注入会引起明显的质量波动/漂移（图14a，图15a）。不同格式的总能量衰减情况相近（图14b，图15b），但所提出的方法产生的粗化引起的能量失配显著较小 $\Delta E$ （图14c，图15c）。这些结果表明，守恒粗化算子对于不同的自由能模型均具有良好的稳健性。

*4.3. Cahn–Hilliard Navier–Stokes*（*CHNS*）

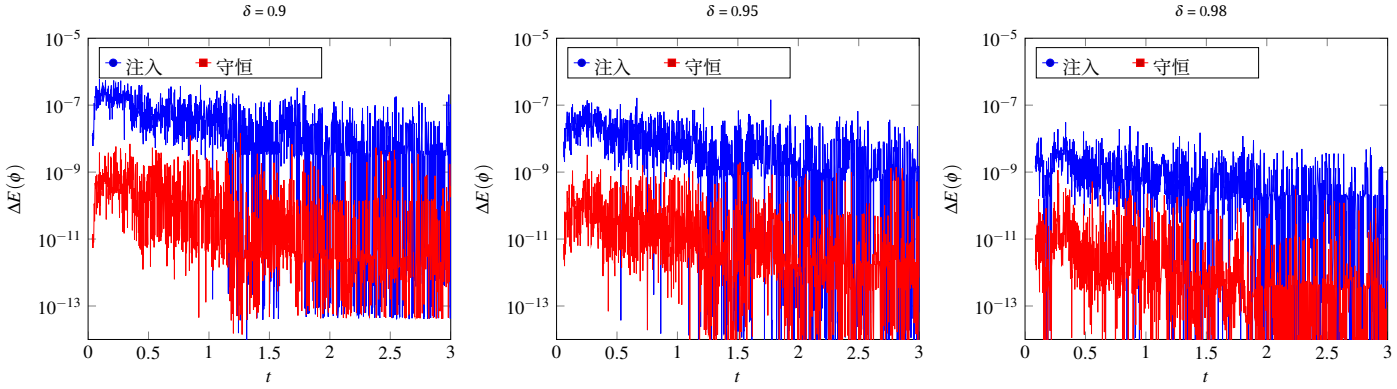
我们进一步扩展了守恒格式，利用 CHNS 偏微分方程（PDEs）组模拟两相流。CHNS系统提供了一个计算上易于处理的模型，能够有效捕捉两相流中的界面动力学。通过采用 Cahn-Hilliard方程跟踪界面，我们获得了若干



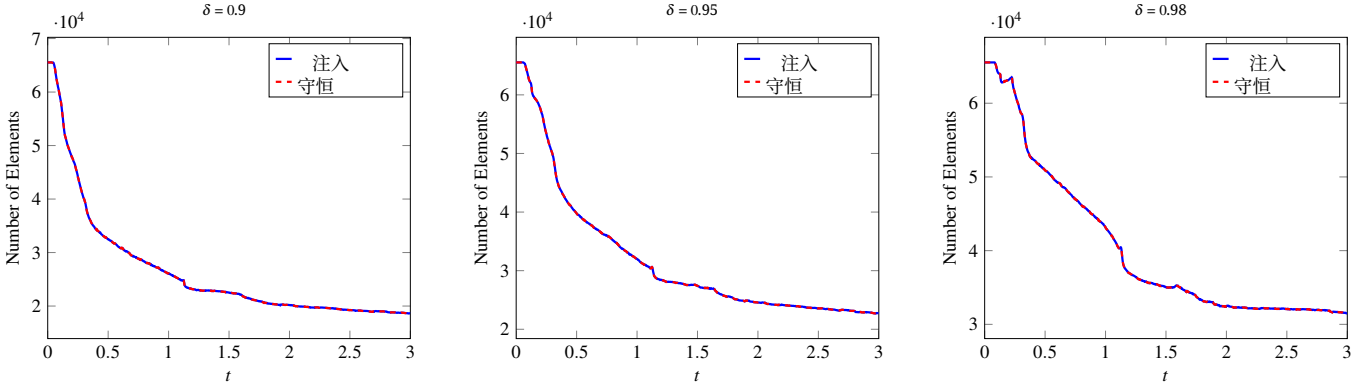

**(a)**质量差异图




**(b)** 能量衰减图

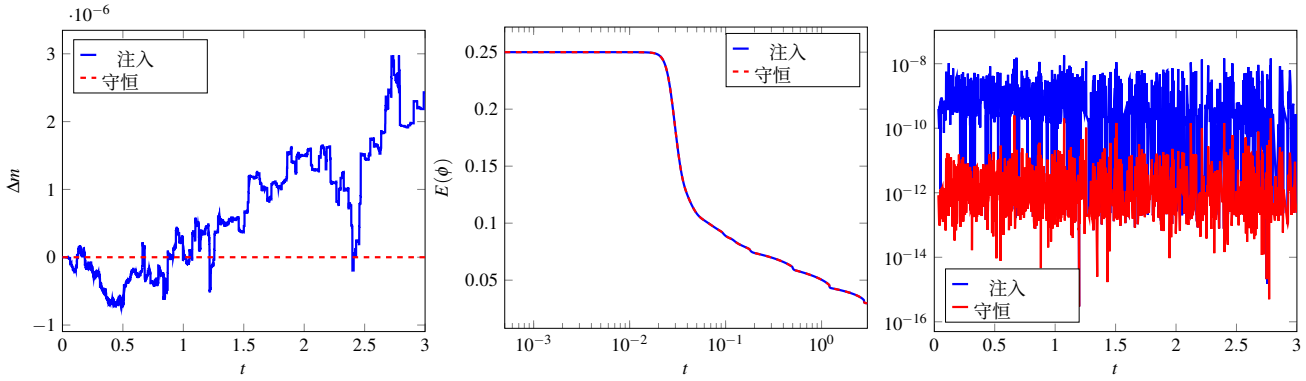



**(c)**能量差异图



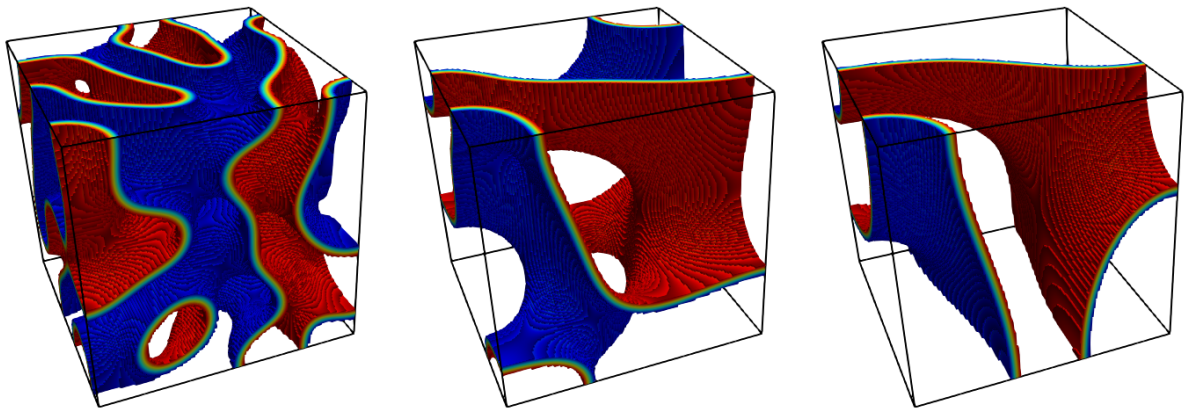

**(d)** 两种格式的单元数量比较

**图10.** 线性基函数和多项式自由能条件下不同插值方案的能量与质量守恒比较。



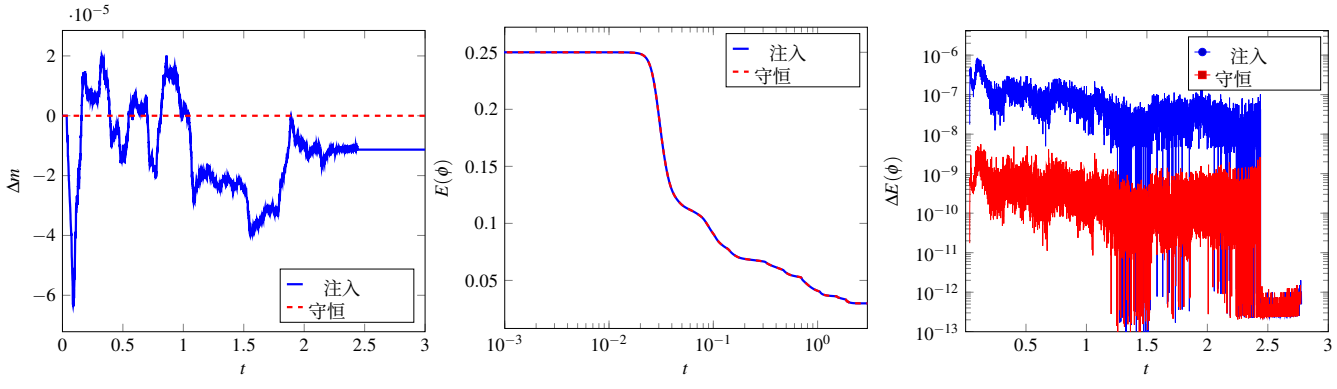

**(a)** 质量差异图 **(b)**能量演化 **(c)** 能量差异

**图11.** 不同插值方案在二次基函数和多项式自由能下的能量与质量守恒比较。



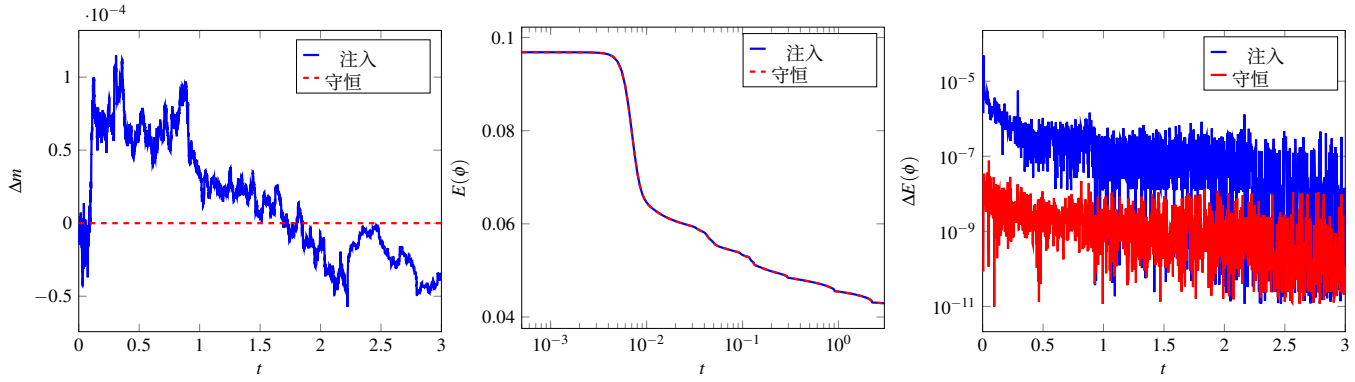
**(a)** $t = 0.1$ **(b)** $t = 1.0$ **(c)** $t = 3.0$

**图12. 三维 情况：** 均匀随机混合物在 $\phi_0 = 0$ 附近受到扰动时，相场变量 $\phi$ 随时间的演化 .



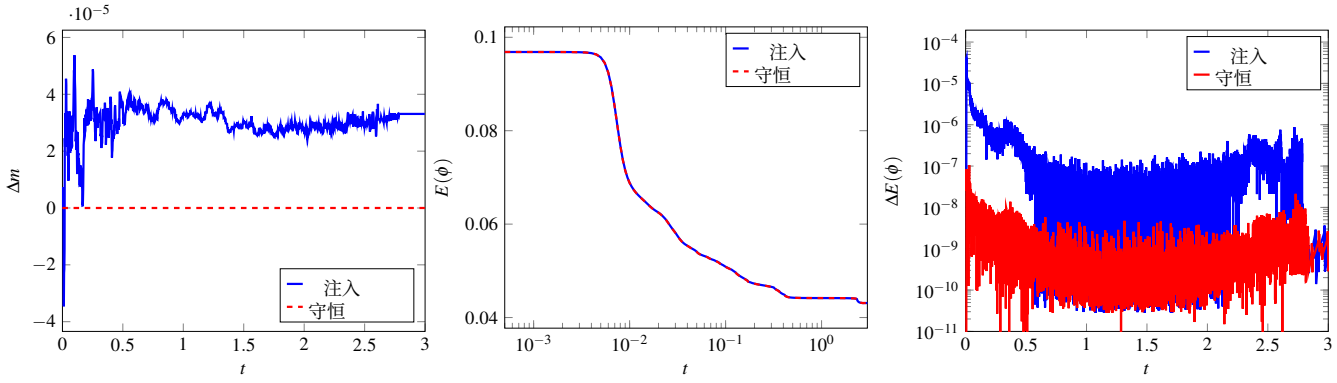

**(a)** 质量差异图 **(b)**能量演化 **(c)** 能量差异

**图13。** 三维 Cahn-Hillard 方程采用多项式自由能时的插值格式对比。




**(a)**质量差异图 **(b)** 能量演化 **(c)** 能量差异

**图14.** 二维 Cahn-Hillard 方程与 Flory-Huggins 自由能的插值格式比较。




**(a)** 质量差异图 **(b)**能量演化 **(c)** 能量差异

**图15.** 采用 Flory-Huggins 自由能的三维 Cahn-Hillard 方程插值格式对比。

其优势包括质量守恒、热力学一致性，以及基于自由能的表面张力描述，并由非平衡热力学中成熟的理论提供支持 [30]。与 Cahn–Hilliard 类似，CHNS系统具有相同的鲜明特征，即质量守恒（$\int_\Omega \phi$ 保持不变）以及总能量单调递减。

**控制方程：** 令 $\phi$ 为追踪各相位置的相场变量，其值在 +1 和-1之间平滑变化，特征弥散界面厚度为 $\varepsilon$；令$v_i$ 为两相混合物速度的$i^{th}$ 分量。 $\eta_+$ 和 $\rho_+$ （$\eta_-$ 和 $\rho_-$）分别表示 +1 相流体（-1相流体）的黏度和密度。热力学一致的CHNS方程 [12, 31, 32, 33] 写作如下：

i 动量方程：

$$\frac{\partial(\rho(\phi)v_i)}{\partial t}+\frac{\partial(\rho(\phi)v_iv_j)}{\partial x_j}+\frac{1}{Pe}\frac{\partial(J_jv_i)}{\partial x_j}+\frac{Cn}{We}\frac{\partial}{\partial x_j}\left(\frac{\partial\phi}{\partial x_i}\frac{\partial\phi}{\partial x_j}\right)$$
$$+\frac{1}{We}\frac{\partial p}{\partial x_i}-\frac{1}{Re}\frac{\partial}{\partial x_j}\left(\eta(\phi)\frac{\partial v_i}{\partial x_j}\right)-\frac{\rho(\phi)\hat{g}_i}{Fr}=0,$$
$$\text{where,}\quad J_i=\frac{(\rho_--\rho_+)}{2\,\rho_+Cn}m(\phi)\frac{\partial\mu}{\partial x_i},$$

ii 无散性与连续性：

$$\frac{\partial v_i}{\partial x_i}=0,\quad \frac{\partial\rho(\phi)}{\partial t}+\frac{\partial(\rho(\phi)v_i)}{\partial x_i}+\frac{1}{Pe}\frac{\partial J_i}{\partial x_i}=0,$$

iii Cahn–Hillard方程

$$\frac{\partial\phi}{\partial t}+\frac{\partial(v_i\phi)}{\partial x_i}-\frac{1}{PeCn}\frac{\partial}{\partial x_i}\left(m(\phi)\frac{\partial\mu}{\partial x_i}\right)=0$$
$$\mu=\psi'(\phi)-Cn^2\frac{\partial}{\partial x_i}\left(\frac{\partial\phi}{\partial x_i}\right)$$

其中， $\rho(\phi)$ 是由 $(\rho_+-\rho_-/2\rho_+)\phi+(\rho_++\rho_-/2\rho_+)$ 给出的无量纲混合物密度，无量纲混合物黏度 $\eta(\phi)$ 则由 $(\eta_+-\eta_-/2\eta_+)\phi+(\eta_++\eta_-/2\eta_+)$ 给出。

无量纲参数如下：佩克莱数 Pe $=$ urL²rmr$\sigma$；雷诺数 Re $=$ urLrvr；韦伯数 We $=$ $^{\rho}$ru²rLr$\sigma$；Cah n数 Cn $=$ $^{\varepsilon}$Lr；以及弗劳德数 Fr $=$ u²rgLr，其中 u$^r$ 和 L$_r$ 分别表示参考速度和参考长度。迁移率 m($\phi$) 设为常数 1。

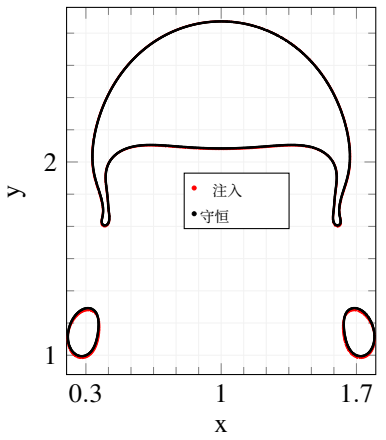
多项研究致力于开发稳健的数值方法，以保持 CHNS系统的上述性质。然而，以往的研究结果表明，将这些方法与自适应网格细化（AMR）结合使用时，质量守恒存在偏差。我们采用 Khanwale 等人 [12] 提出的基于投影的离散格式来模拟该系统。模拟中结合使用二阶半隐式时间步进和线性基函数，以实现二阶空间精度。

我们将二维气泡上升作为基准问题 [34, 35, 13, 12]，并将结果与我们提出的质量守恒插值方案进行比较。表 2 展示了模拟所用的参数。与 Cahn-Hillard 算例类似，AMR 过程由两个细化层级定义：体相（$\mathscr{R}_b$）和界面（$\mathscr{R}_i$），其值分别设为 6 和 11。我们将 $\delta$ 的值设为 0.95，作为细化阈值。随着界面移动，靠近界面的单元会被细化，而远离界面的单元则会被粗化。

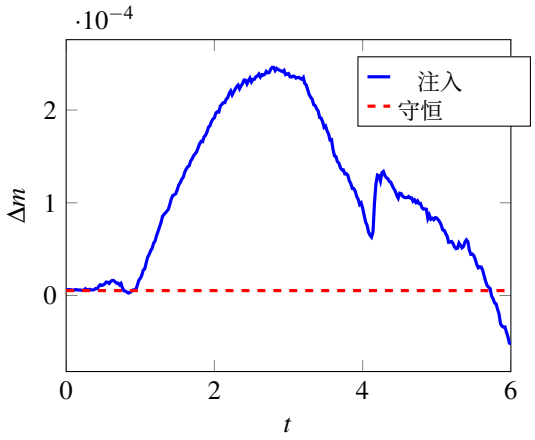
| $\rho_+$ | $\rho_-$ | $\mu_+$ | $\mu_-$ | $\rho+/\rho-$ | $\nu_+/\nu_-$ | $g$ | $\sigma$ | $Ar$ | $We$ | $Fr$ | $Cn$ | $Pe$ | $\Delta t$ | $\mathscr{R}_b$ | $\mathscr{R}_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 1.0 | 10 | 0.1 | 1000 | 100 | 0.98 | 1.96 | 35 | 125 | 1.0 | 0.01 | 3333.33 | 0.0025 | 6 | 11 |

**表2**。 表格展示了 Khanwale 等人 [12] 所研究的气泡上升基准问题中的物理参数和模拟参数。

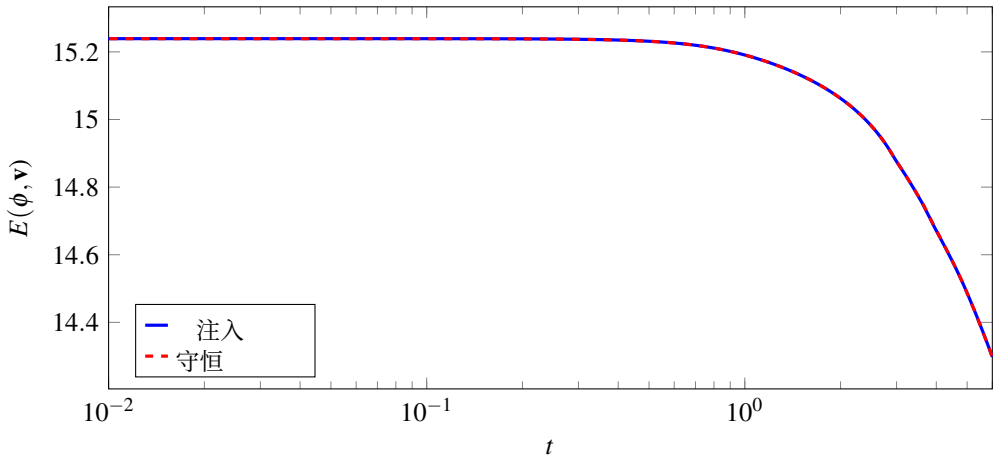
图16展示了模拟结果。值得注意的是，只有变量 $\phi$ 采用质量守恒插值，而其他变量(**v**、$p$ $\mu$) 在粗化阶段采用注入方案进行插值。与我们提出的质量守恒方案相比，气泡的形状没有明显差异（图16a）。然而，所提出的格式能够有效保持系统的总质量。相比之下，基于注入的粗化过程会导致约 $\mathscr{O}(10^{-4})$(图16b)的质量偏差，而所提出的格式则实现了质量守恒，如近乎平坦的曲线所示。此外，根据公式（32）定义的系统总能量会随时间单调衰减，两种插值方案之间没有显著差异（图16c）。




**(a)** 在 $t = 4.2$ 时的气泡形状。




**(b)**质量守恒。




**(c)** 能量衰减。

**图16.** 二维单个上升液滴测试案例 2：图中对比了第 4.3 节中测试案例所采用的所提出的质量守恒插值与基于注入的插值。

$$E_{\text{tot}}(\mathbf{v},\phi,t) = \int_\Omega \frac{1}{2}\rho(\phi)\,\|\mathbf{v}\|^2\,\mathrm{d}\mathbf{x} \\ \frac{1}{CnWe}\int_\Omega \left(\psi(\phi) + \frac{Cn^2}{2}\,\|\nabla\phi\|^2 + \frac{1}{Fr}\rho(\phi)y\right)\mathrm{d}\mathbf{x}. \tag{32}$$

### *4.4.* 计算成本与并行可扩展性

我们在 DENDRITE-KT 代码框架中实现了整个过程。在先前的工作 [26, 27, 36, 12, 7, 28] 中，我们展示了整个框架截至 $\mathcal{O}(114K)$ 的可扩展性结果。执行所提出的格式还需要以下两个额外步骤：

1. 执行从细网格到粗网格的局部 $L^2$ 投影，如 算法 1 所述，在高斯点处将数值从细单元传递到粗网格单元。
2. 求逆常系数全局质量矩阵。

粗化阶段首先根据原始八叉树构建代理树，以确保要粗化的父单元的所有叶单元都位于同一处理器上。无论采用哪种网格间传递方案，都必须执行此重新分配步骤，Saurabh 等人 [26] 对此进行了详细说明。这样做是为了确保后续

粗化操作可在本地执行，无需进程间通信。建立代理树后，我们执行上述两个步骤。

第一部分完全是局部操作，针对每个粗化单元独立进行，仅涉及其子单元以及细单元中对应的条目。该步骤不会引入任何额外的进程间通信。更多详情请参阅 Saurabh 等人论文的 II.C 节。 [26]该方法利用了八叉树上已有的算法，并已在我们之前的工作中证明具有良好的可扩展性。第二部分依赖于质量矩阵的最优求逆，可将其视为标准的线性系统求逆。总体而言，所提方法的额外开销相对很小，且主要由 CHNS 中的界面追踪、重新网格化和压力投影求解成本决定。

在我们的实验中，我们发现，所提方法带来的额外成本约占 Cahn–Hilliard 方程求解每个时间步总计算成本的 10%，而在二维 CHNS 模拟中约占 7%。对于更大规模的问题，预计这部分开销还会进一步降低，因为与整体模拟相比，质量矩阵求逆的成本将变得不那么显著，而整体模拟的成本主要由压力投影和界面追踪决定。

## 5. 结论

我们提出了一种简单且可扩展的守恒场粗化算子，用于大规模并行的平衡八叉树网格上的自适应网格细化（AMR）。核心发现是，对于连续 Galerkin（CG）离散化，父到子细化传递天然具有守恒性，而传统的基于注入的子到父粗化传递通常不具备守恒性。在反复进行自适应网格细化循环时，这种非守恒粗化会导致全局守恒量发生系统性漂移（或振荡），在长时间跨度的模拟中尤为棘手。

为消除这一缺陷，我们提出了一种守恒粗化过程，通过以下步骤在粗化期间确保离散全局守恒：（i）在求积点处使用局部$L^2$ 投影，计算守恒的粗单元值；（ii）通过全局$L^2$ 投影恢复粗网格节点自由度，即在粗网格上求解质量矩阵。与注入相比，该方法需要额外进行一次线性求解，但此算子在分布式内存架构上具有很强的可扩展性，并且能够高效地集成到现有的八叉树自适应网格细化框架中。

我们在质量守恒相场模型上验证了该方法，包括 Cahn–Hilliard方程和耦合的卡恩-希利亚德-纳维-斯托克斯（CHNS）系统。在二维和三维测试以及多种自由能模型中，所提出的格式在自适应网格细化下将质量保持在数值精度范围内，而注入则表现出明显的质量漂移。重要的是，守恒粗化在实现这一点的同时，并未损害预期的能量行为；与注入相比，它将粗化引起的能量失配降低了近两个数量级。

尽管本研究重点关注质量守恒，但通过修改粗化过程中施加的守恒矩，该框架也可扩展至约束其他积分量的离散守恒（例如物种、电荷或特定应用不变量） [37]。未来工作将探讨：（i）同时约束多个守恒量；（ii）将其扩展到采用任意次数拉格朗日基函数的高阶连续Galerkin离散；以及（iii）评估守恒约束对更广泛的多物理场应用中的精度、稳定性和求解器性能的影响。总体而言，所提出的守恒粗化算子为提升八叉树自适应网格细化模拟的稳健性和可靠性提供了一种实用方法，尤其适用于长期守恒至关重要的场景。

## 6. 致谢

BG 和 HS 感谢美国国家科学基金会 2513871 的资助。我们感谢美国国家科学基金会 2323716 和美国国家科学基金会 2053760 的资助。

## 参考文献

[1] C. M. Elliott，《相分离动力学的 Cahn-Hilliard 模型》，《相变问题的数学模型》（1989）35–73。[2] M. Gameiro、K. Mischaikow、T. Wanner，《Cahn–Hilliard 相分离理论中的图案复杂度演化》，《材料学报》53（2005）693–704。

[3] T. Speck、J. Bialk´e、A. M. Menzel、H. L¨owen, 《活性布朗粒子相分离的有效 Cahn-Hilliard 方程》, 《物理评论快报》 112（2014）218304。[4] J. F. Blowey、C. M. Elliott, 《非光滑自由能部分的相分离 Cahn–Hilliard 梯度理论，第一部分：数学分析》, 《欧洲应用数学杂志》2（1991）233–280。[5] D. Han、X. Wang, 《Cahn–Hilliard–Navier–Stokes 方程的二阶时间、唯一可解且无条件稳定数值格式》, 《计算物理学杂志》290（2015）139–156。[6] A. E. Diegel、C. Wang、X. Wang、S. M. Wise, 《Cahn–Hilliard–Navier–Stokes 系统二阶精度有限元方法的收敛性分析与误差估计》, 《数值数学》137（2017）495–534。[7] M. A. Khanwale、A. D. Lofquist、H. Sundar、J. A. Rossmanith、B. Ganapathysubramanian, 《在并行自适应八叉树网格上，使用热力学一致且能量稳定的 Cahn-Hilliard Navier-Stokes 方程模拟两相流》, 《计算物理学杂志》（2020）109674。[8] F. Boyer、C. Lapuerta、S. Minjeaud、B. Piar、M. Quintard, 《用于模拟三相流的 Cahn–Hilliard/Navier–Stokes 模型》, 《多孔介质中的传输》 82（2010）463–483。[9] E. Olsson、G. Kreiss, 《用于两相流的守恒水平集方法》, 《计算物理学杂志》210（2005）225–246。[10] E. Olsson、G. Kreiss、S. Zahedi, 《用于两相流的守恒水平集方法 II》, 《计算物理学杂志》225（2007）785–807。[11] S. Van der Pijl、A. Segal、C. Vuik、P. Wesseling, 《用于多相流建模的质量守恒水平集方法》, 《流体数值方法国际期刊》47（2005）339–361。[12] M. A. Khanwale、K. Saurabh、M. Ishii、H. Sundar、J. A. Rossmanith、B. Ganapathysubramanian, 《一种用于自适应八叉树网格上 Cahn-Hilliard Navier-Stokes 方程的基于投影的半隐式时间步进方法》, 《计算物理学杂志》475（2023）111874。[13] M. A. Khanwale、K. Saurabh、M. Fernando、V. M. Calo、H. Sundar、J. A. Rossmanith、B. Ganapathysubramanian, 《求解 Cahn-Hilliard Navier-Stokes 方程的全耦合框架：自适应八叉树网格上的二阶能量稳定数值方法》, 《计算物理通讯》280（2022）108501。[14] O. Wodo、B. Ganapathysubramanian, 《高效求解 Cahn–Hilliard 方程：自适应隐式时间格式、网格敏感性分析与三维等周问题》, 《计算物理学杂志》230（2011）6037–6060。[15] A.-K. Tornberg、B. Engquist, 《一种用于多相流应用的基于有限元的水平集方法》, 《科学计算与可视化》3（2000）93–101。[16] E. Marchandise、J.-F. Remacle, 《一种采用不连续水平集方法求解两相不可压缩流的稳定化有限元方法》, 《计算物理学杂志》219（2006）780–800。[17] T. Tu、D. R. O'Hallaron、O. Ghattas, 《面向万亿级应用的可扩展并行八叉树网格生成》, 载于：SC'05：2005年 ACM/IEEE 超级计算大会论文集，IEEE，2005，页4–4。[18] M. Ishii、M. Fernando、K. Saurabh、B. Khara、B. Ganapathysubramanian、H. Sundar, 《求解时空中的偏微分方程：基于四维树的自适应、无网格与无矩阵方法》, 载于：国际高性能计算、网络、存储与分析会议论文集，2019，页1–61。[19] M. Fernando、D. Neilsen、H. Lim、E. Hirschmann、H. Sundar, 《双黑洞中等质量比旋近的大规模并行模拟》, 《SIAM科学计算杂志》41（2019）C97–C138。[20] H. Sundar、R. S. Sampath、G. Biros, 《并行线性八叉树的自底向上构造与 2:1 平衡细化》, 《SIAM科学计算杂志》30（2008）2675–2708。[21] P. Bastian、M. Blatt、A. Dedner、C. Engwer、R. Kl¨ofkorn、M. Ohlberger、O. Sander, 《用于并行自适应科学计算的通用网格接口。第一部分：抽象框架》, 《计算》 82（2008）103–119。[22] D. M. Greaves、A. Borthwick, 《基于层次树的有限元网格生成》, 《国际工程数值方法杂志》45（1999）447–471。[23] M. Bader, 《空间填充曲线：导论及其在科学计算中的应用》, 第9卷，Springer Science & Business Media，2012。[24] C. Burstedde、L. C. Wilcox、O. Ghattas, `p4est`: 《八叉树森林上并行自适应网格细化的可扩展算法》, 《SIAM科学计算杂志》33（2011）1103–1133。doi:10.1137/100791634。[25] S. Popinet, 《Gerris：一种用于复杂几何中不可压缩欧拉方程的基于树的自适应求解器》, 《计算物理学杂志》190（2003）572–600。[26] K. Saurabh、M. Ishii、M. A. Khanwale、H. Sundar、B. Ganapathysubramanian, 《面向下一代多相流模拟的可扩展自适应算法》, 载于：2023年 IEEE 国际并行与分布式处理研讨会（IPDPS），IEEE，2023，页590–601。

[27] K. Saurabh, B. Gao, M. Fernando, S. Xu, M. A. Khanwale, B. Khara, M.-C. Hsu, A. Krishnamurthy, H. Sundar, B. Ganapathysubramanian，采用浸入式等几何分析的自适应八叉树网格进行工业尺度大涡模拟，《计算机与应用数学》97 (2021) 28–44。[28] K. Saurabh, M. Ishii, M. Fernando, B. Gao, K. Tan, M.-C. Hsu, A. Krishnamurthy, H. Sundar, B. Ganapathysubramanian，任意区域中的可扩展自适应偏微分方程求解器，载于：《国际高性能计算、网络、存储与分析会议论文集》，2021，第1–15页。[29] C. M. Elliott, D. A. French, F. Milner，Cahn–Hilliard方程的二阶分裂方法，《数值数学》54 (1989) 575–590。[30] D. Jacqmin，连续介质表面张力方法的能量方法，载于：第34届航空航天科学会议与展览，1996，第858页。[31] J. Shen, X. Yang，两相不可压缩流相场模型的解耦能量稳定格式，《SIAM数值分析杂志》53 (2015) 279–296。[32] D. M. Anderson, G. B. McFadden, A. A. Wheeler，流体力学中的弥散界面方法，《流体力学年度评论》30 (1998) 139–165。[33] X. Feng，两相流Navier–Stokes–Cahn–Hilliard弥散界面模型的全离散有限元近似，《SIAM数值分析杂志》44 (2006) 1049–1072。[34] S. Hysing，二维气泡动力学的定量基准计算，《流体数值方法国际期刊》60 (2009) 1259–1288。[35] S. Aland, A. Voigt，二维气泡动力学弥散界面模型的基准计算，《流体数值方法国际期刊》69 (2012) 747–761。[36] M. A. Khanwale, K. Saurabh, M. Ishii, H. Sundar, B. Ganapathysubramanian，采用网格与界面细化的Cahn-Hilliard Navier-Stokes方法研究主射流雾化中的破碎动力学（2022）。网址：https://arxiv.org/abs/2209.13142v1。arXiv:2209.13142。[37] J. Baiges, M. Chiumenti, C. A. Moreira, M. Cervera, R. Codina，增材制造过程数值模拟的自适应有限元策略，《增材制造》37 (2021) 101650。

## Ap附录 A. 一维单元的限制矩阵 元

本附录给出第 4 节中使用的一维 Q1 和 Q2 单元的限制矩阵，用于在粗化操作期间根据细单元求积值构造粗单元求积值。对于 Q1 和 Q2 单元，我们均采用标准高斯-勒让德求积，因此每个粗单元和细单元分别有 $N_{\mathrm{ip},c} = N_{\mathrm{ip},f} = 2$ 和$N_{\mathrm{ip},c} = N_{\mathrm{ip},f} = 3$个求积点。由此得到的、大小为$N_{\mathrm{ip},c} \times 2N_{\mathrm{ip},f}$的限制矩阵如下所示。

### 附录 *A.1.* 一维 *Q1* 单元的限制矩阵

一维中采用标准高斯-勒让德求积的 Q1 单元的限制矩阵（$N_{\mathrm{ip},c} = N_{\mathrm{ip},f} = 2$）为：

$$R_{1D} = \begin{pmatrix} 0.5915063509461096 & 0.3415063509461096 & 0.1584936490538904 & -0.09150635094610965 \\ -0.09150635094610965 & 0.1584936490538904 & 0.3415063509461096 & 0.5915063509461096 \end{pmatrix} \tag{A.1}$$

### 附录 *A.2.* 一维 *Q2* 单元的限制矩阵

一维中，采用标准 Gauss–Legendre 求积、具有 $N_{\mathrm{ip},c} = N_{\mathrm{ip},f} = 3$ 个求积点的 Q2 单元的限制矩阵为：

$$右_{1D} = \begin{pmatrix} 0.614415278851 & 0.424865556414 & 0.041666666667 & -0.031081945517 & -0.091532223080 & 0.041666666667 \\ -0.097551215948 & 0.291666666667 & 0.305884549282 & 0.305884549282 & 0.291666666667 & -0.097551215948 \\ 0.041666666667 & -0.091532223080 & -0.031081945517 & 0.041666666667 & 0.424865556414 & 0.614415278851 \end{pmatrix}$$

(A.2)

上述限制矩阵可用于公式（13），在一维粗化操作过程中根据细单元求积值计算粗单元求积值。更高维度的扩展可通过沿各维度对一维限制矩阵进行张量积来实现（算法 1）。