

## 气液两相流曲面重构改进算法研究

龚军华<sup>1</sup>, 董静<sup>2</sup>, 陈宇杰<sup>3</sup>, 孙东亮<sup>3</sup>, 宇波<sup>4</sup>, 石国赞<sup>5</sup>, 陈斌<sup>1</sup>

(1. 西安交通大学动力工程多相流国家重点实验室, 710049, 西安; 2. 上海能源建设工程设计研究有限公司, 201300, 上海; 3. 北京石油化工学院机械工程学院, 102617, 北京; 4. 长江大学石油工程学院, 430100, 武汉; 5. 昆仑数智科技有限责任公司, 102266, 北京)

**摘要:** 针对曲面重构(CIR)算法在界面重构及体积分数法(VOF)方程求解过程中,由于采用分段逼近策略导致的经验性、精度损失和界面高度重复计算问题,提出了基于积分的曲面重构(ICIR)算法。该算法通过一系列处理策略,实现基于积分的体积分数求解,避免了对界面的经验性分段以及各分段界面高度的重复求解。测试结果表明:在随机重构 1 000 个圆形界面的静态测试中,相比 CIR 算法,ICIR 算法的重构精度提升了 7.71%,界面方程求解加速比为 6.78;在简单气泡水平移动测试和复杂时间反转单涡问题测试中,ICIR 算法的精度相比 CIR 算法分别提升了 3.96%和 13.09%,计算耗时分别缩短了 40.88%和 68.31%。总体而言,ICIR 算法显著提升了计算效率,并提高了重构界面的精度。

**关键词:** 界面重构;曲面重构;气液两相流

**中图分类号:** O351.2 **文献标志码:** A

**DOI:** 10.7652/xjtuxb202507020 **文章编号:** 0253-987X(2025)07-0202-12

## Research on Improved Algorithm for Surface Reconstruction of Gas-Liquid Two-Phase Flow

GONG Junhua<sup>1</sup>, DONG Jing<sup>2</sup>, CHEN Yujie<sup>3</sup>, SUN Dongliang<sup>3</sup>,  
YU Bo<sup>4</sup>, SHI Guoyun<sup>5</sup>, CHEN Bin<sup>1</sup>

(1. State Key Laboratory of Multiphase Flow in Power Engineering, Xi'an Jiaotong University, Xi'an 710049, China;  
2. Shanghai Energy Construction Engineering Design & Research Co., Ltd., Shanghai 201300, China;  
3. School of Mechanical Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China;  
4. School of Petroleum Engineering, Yangtze University, Wuhan 430100, China;  
5. Kunlun Digital Intelligence Technology Company, Beijing 102266, China)

**Abstract:** To address the issues of empiricism, accuracy loss, and repeated calculation of interface heights caused by the piecewise approximation strategy in interface reconstruction and the volume of fluid equation solving using the circle-based interface reconstruction (CIR) algorithm, an integration circle-based interface reconstruction (ICIR) algorithm is proposed. This algorithm employs a series of processing strategies to achieve volume fraction solutions based on integration, effectively avoiding empirical interface segmentation and repeated solving of interface heights in each segment. Test results indicate that in a static test involving the random reconstruction of 1 000 circular interfaces, the ICIR algorithm improves reconstruction accuracy by 7.71% compared to

the CIR algorithm, with an acceleration ratio of 6.78 for solving the interface equation. In the horizontal motion test of a simple vapor bubble and the complex time-reversed single-vortex test, the ICIR algorithm enhances accuracy by 3.96% and 13.09%, respectively, while computational efficiency improves by 40.88% and 68.31%. Overall, the ICIR algorithm significantly boosts computational efficiency and enhances the accuracy of interface reconstruction.

**Keywords:** interface reconstruction; curved interface reconstruction; gas-liquid two-phase flow

两相流现象广泛存在于自然界和工业生产中,如油气集输管道中的油水两相流<sup>[1-2]</sup>、反应器的汽泡运动<sup>[3]</sup>、地热井筒中蒸汽与水的混合流动<sup>[4]</sup>等。数值模拟是研究两相流现象的主要手段之一。采用界面捕捉方法准确追踪相界面,有助于揭示两相流机理,可为工程实践和生产优化提供指导<sup>[5-6]</sup>。目前较为常见的相界面捕捉方法有标记网格法(MAC)<sup>[7]</sup>、体积分数法(VOF)<sup>[8]</sup>、水平集方法(LS)<sup>[9]</sup>等。其中,VOF方法具备质量守恒特性<sup>[10-11]</sup>,广泛应用于两相流研究<sup>[12-14]</sup>。在基于VOF方法的两相流研究中,需求解VOF输运方程来追踪界面的演变过程,但求解VOF输运方程易出现数值假扩散现象,导致界面随运动逐渐失真<sup>[15]</sup>。为解决这一问题,有学者提出采用几何方法求解VOF输运方程<sup>[16]</sup>。几何方法首先根据体积分分数重构相界面,然后依据界面方程计算VOF通量,再根据VOF通量更新体积分分数<sup>[17]</sup>。相比数值求解方法,几何求解方法重构界面的精度更高。分段线性界面重构(PLIC)算法<sup>[18]</sup>是目前最常用的界面重构方法之一,其采用平面来近似相界面,并通过体积分分数确定平面方程的参数,从而完成界面重构。随后,有学者提出了最小二乘体积分分数界面重构(LVIRA)算法<sup>[19]</sup>、高效最小二乘体积分分数界面重构(ELVIRA)算法<sup>[20]</sup>和分段连续线性界面重构(PCLIC)算法<sup>[21]</sup>等改进算法,用以提高相界面的重构精度。

相界面在表面张力的作用下通常呈现曲面,而传统方法通过平面近似弯曲的相界面,导致界面重构的还原度较低。为提高重构精度,部分学者提出采用曲面来近似相界面。Chorin<sup>[22]</sup>使用圆弧来近似相界面,其中圆弧的半径等于曲率半径,圆弧的圆心坐标以极坐标形式 $(\rho, \theta)$ 表示,并通过迭代方法确定其具体值。Renardy等<sup>[23]</sup>采用包含6个参数的二次方程来近似界面,其参数构成一个6维向量,并通过最小化一个与体积分分数相关的误差函数来确定这些参数的具体值。Scardovelli等<sup>[24]</sup>采用标准圆的一部分来近似相界面,该标准圆方程包含 $a, b, c$ 共3个参数。在进行曲面重构之前,需先通过线性界面重构算法确定初始界面,然后选取部分界面

的中点和端点,利用这些点的坐标通过二次最小二乘法拟合圆方程,从而确定圆形界面的具体位置。Diwakar等<sup>[25]</sup>采用分段抛物线来近似相界面,首先通过PLIC算法获得初始线性界面,然后对初始界面分别进行两次修正,使得最终重构的曲面具有连续性和光滑性。Maity等<sup>[26]</sup>提出的PCIC算法采用分段圆弧来近似相界面,但该方法需要先进行初始PLIC重构,并在此基础上收集初始重构的相界面点,以拟合圆弧方程。对该方法进一步进行C<sup>0</sup>修正,可增强界面的连续性并提高计算精度。

上述曲面重构方法能够提高界面重构的精度,但仍存在一些问题,如在流体体积分分数过大或过小时会导致界面重构失败<sup>[22-24]</sup>,或实施过程复杂且需要额外的线性界面重构步骤<sup>[24-26]</sup>。针对这些问题,Chen等<sup>[27]</sup>提出了一种精度较高且实施简单的曲面重构方法(CIR)。该算法利用圆的一部分近似相界面,其中圆的半径由曲率和符号距离函数决定,圆心位置则通过质量守恒关系确定。然而,CIR算法仍存在不足:其界面重构与VOF通量计算需通过分段策略来确定曲面与网格面围住的不规则流体体积,分段数越多精度越高,但分段数过多会显著降低计算效率。为解决上述问题,本文提出了一种基于积分方法的曲面重构(ICIR)算法。该方法提出了一系列优化策略,直接利用积分方法计算体积分分数,从而改进了CIR算法中界面重构与VOF通量计算的过程,既避免了分段计算引入的误差,又显著提升了计算效率。本研究有望为高精度、高效率的气液两相流数值研究提供技术支持。

## 1 基于积分方法的曲面重构算法

### 1.1 CIR算法简介

CIR算法的基本思想是利用圆的一部分近似相界面,以实现曲面重构<sup>[17, 27]</sup>,如图1所示。CIR算法在界面重构时的核心步骤就在于确定圆的半径和圆心位置。圆的方程为

$$(x' - x_0)^2 + (y' - y_0)^2 = r^2 \quad (1)$$

式中: $x'$ 和 $y'$ 为圆上任意一点的 $x$ 轴和 $y$ 轴坐标;

$x_o$  和  $y_o$  分别为圆心的  $x$  和  $y$  轴坐标;  $r$  为圆的半径。

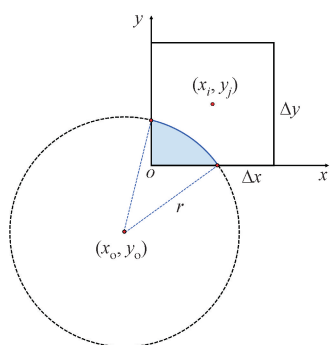


图1 利用圆的一部分近似相界面的示意图

Fig. 1 Schematic diagram of approximating the phase interface using part of a circle

CIR 算法借鉴了体积分数与水平集耦合 (VOSET) 方法的思想, 通过多次迭代过程来提高圆心和半径的计算精度。VOSET 算法是一种融合 VOF 方法与 LS 方法的界面重构算法, 其核心思想如下: 首先, 借助体积分数计算相界面法向, 以此重构初始界面; 接着, 基于该重构界面计算符号距离函数。鉴于符号距离函数所计算的相界面法向和曲率等物理量精度较高, 因此利用符号距离函数计算这些物理量。随后, 根据这些物理量进一步重构界面, 并更新符号距离函数。经过多次迭代后, 可得到高精度的相界面。该方法有效整合了 VOF 方法和 LS 方法的优势, 提高了重构界面的精度。

#### 1.1.1 圆半径的确定

CIR 算法通过式(2)来确定圆的半径, 公式为

$$r = \begin{cases} \frac{1.0}{\nabla \cdot (\nabla \phi / |\nabla \phi|)}, & \text{第1次迭代} \\ \frac{1.0}{\nabla \cdot (\nabla \phi / |\nabla \phi|)} + \phi, & \text{其他迭代} \end{cases} \quad (2)$$

式中:  $\phi$  为体积分数;  $\phi$  为当前网格单元节点处的符号距离函数。

#### 1.1.2 圆心位置的确定

CIR 算法通过一种简单的方法来确定圆心位置。如图2所示, 首先根据计算得到的相界面法向分量  $n_x$  和  $n_y$  确定圆心所在的直线位置; 然后根据圆形界面与网格单元的相交关系, 确定圆心在直线上的变动范围  $[L_{\min}, L_{\max}]$ ; 接着在该范围内, 利用二分法不断迭代以确定圆心位置, 然后采用分块方法计算圆形界面与网格单元相交区域的体积分数, 直至计算值与网格单元的实际体积分数之差小于阈值  $\epsilon$  (具体取值将在后文介绍), 即确定圆心位置。

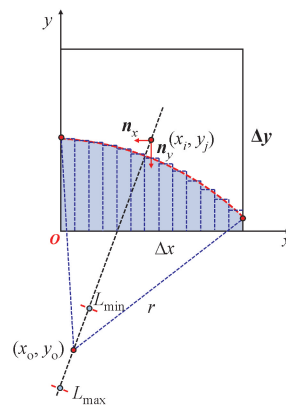


图2 基于二分法确定圆心位置的示意图

Fig. 2 Schematic diagram of determining the circle center using dichotomy

$L_{\min}$  和  $L_{\max}$  的计算公式分别为

$$L_{\min} = r - \sqrt{\Delta x^2 + \Delta y^2} \quad (3)$$

$$L_{\max} = r + \sqrt{\Delta x^2 + \Delta y^2} \quad (4)$$

式中:  $\Delta x$  和  $\Delta y$  分别为网格单元的宽度和高度。

然而, 如图2所示的分块计算体积分数的方法存在以下问题。首先是计算精度问题, 如图3所示。CIR 算法使用矩形小块的体积 ( $V_3 + V_4$ ) 近似真实主相流体的体积, 而实际主相流体的体积应为  $V_2 + V_4$ , 因此两者之间存在体积误差  $V_1$ 。为减小误差, CIR 算法通常采用较多的小块, 使得每个小块内的重构界面尽量接近直线, 从而减小体积误差  $V_1$ 。

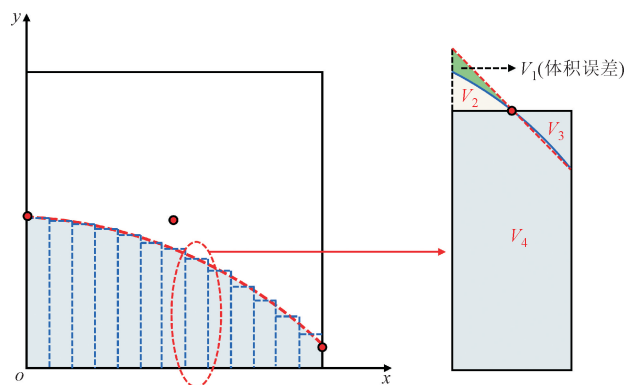


图3 使用分块方法计算体积分数的误差示意图

Fig. 3 Schematic diagram of the error in volume fraction calculation by using the segmented method

其次是计算效率问题。当分块较多时, 体积分数的计算耗时显著增加。CIR 算法通过多次迭代确定圆形界面的半径和圆心位置。每次迭代均需采用分块方法计算体积分数。在此过程中, 需多次求解界面方程来计算每个小块的高度。单个网格单元的

界面重构需经历多次迭代,较多的分块会显著增加计算量,从而降低界面重构效率。

因此,分块数的选取需要在计算精度和效率之间权衡。尽管可以根据经验方法选择合适的分块数目,但不同问题对分块数的需求各不相同,难以通过经验建立统一的选取标准,这在一定程度上限制了算法的适用性。

## 1.2 基于积分思想的体积分数计算方法

针对 CIR 算法分块计算体积分数方法存在的精度和效率问题,本文提出了一种基于积分思想的体积分数计算方法。如图 4 所示,以网格单元的左下角点为原点建立局部坐标系,重构圆形界面的半径为  $r$ ,圆心为  $(x_o, y_o)$ 。圆形界面与网格单元边界相交于点 1  $(x_1, y_1)$  和点 2  $(x_2, y_2)$ ,两交点与圆心连线与  $x$  轴正方向的夹角分别为  $\theta_1$  和  $\theta_2$ 。网格单元内的主相体积可分为矩形区域  $A_1$  和不规则区域  $A_2$ ,其中  $A_2$  的一边为重构的圆形界面。

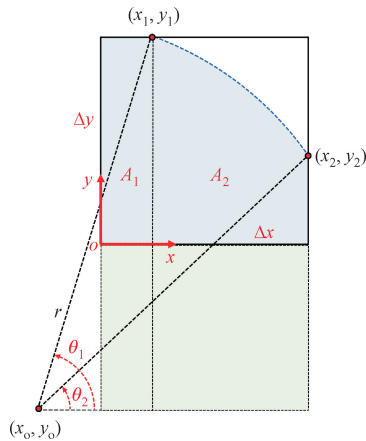


图 4 基于积分方法的体积分数计算示意图

Fig. 4 Schematic diagram of volume fraction calculation based on the integral method

矩形区域  $A_1$  的体积分数为

$$\varphi_1 = \frac{(x_1 - 0)\Delta y}{\Delta x \Delta y} = \frac{x_1}{\Delta x} \quad (5)$$

虽然区域  $A_2$  为不规则区域,但考虑到其一条边为圆弧,因此可通过积分方法计算其体积分数。具体的计算公式如下

$$\varphi_2 = \frac{\int_{x_1}^{x_2} \sqrt{r^2 - (x - x_o)^2} dx - (x_2 - x_1)(0 - y_o)}{\Delta x \Delta y} = \frac{r^2 \left( \frac{1}{4} \sin 2\theta - \frac{1}{2} \theta \right) \Big|_{\theta_1}^{\theta_2} - (x_2 - x_1)(0 - y_o)}{\Delta x \Delta y}$$

(6)

式中:  $\theta_1 = \arccos\left(\frac{x_1 - x_o}{r}\right)$ ;  $\theta_2 = \arccos\left(\frac{x_2 - x_o}{r}\right)$ 。

## 1.3 基于积分思想的体积分数计算流程

按照上述方法,可以高效且精确地计算网格单元内部的主相体积分数。为进一步提高该算法在不同工况下的适用性,本节详细介绍 ICIR 算法界面重构的关键实施细节,并提出相应的处理策略。

### 1.3.1 简化相交类型

在实际计算中,圆形界面与网格单元可能存在多种相交类型,如图 5 所示。

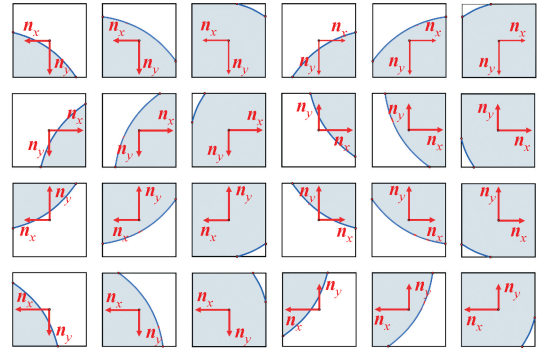


图 5 圆形界面与网格单元不同相交类型示意图

Fig. 5 Schematic diagram of different intersection types between a circular interface and grid cells

为了简化处理,可根据法向分量  $n_x$  和  $n_y$  的相对大小确定圆形界面的朝向,并通过旋转变换统一调整为朝下方向,如图 6 所示。这种方法可将众多相交类型归纳为图 7 所示的 6 种标准类型,从而简化需要处理的相交类型。

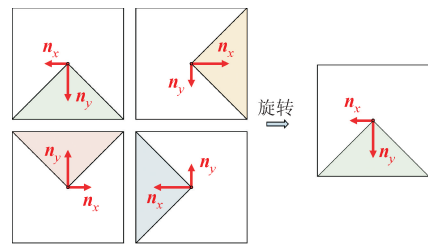
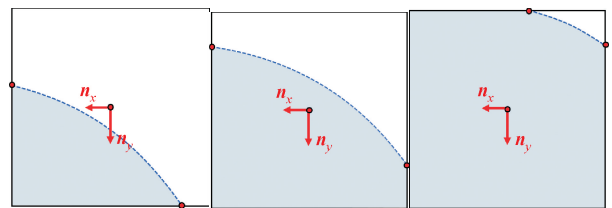


图 6 圆形界面朝向变换示意图

Fig. 6 Schematic diagram of orientation transformation of a circular interface



(a) 类型 1

(b) 类型 2

(c) 类型 3



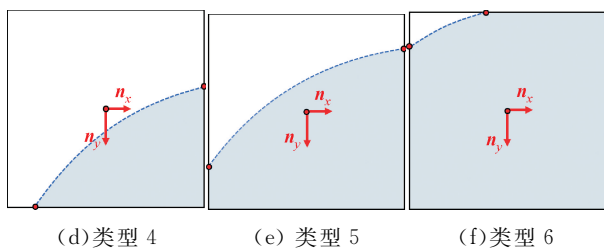


图 7 6 种标准相交类型

Fig. 7 6 standard intersection types

### 1.3.2 通用体积分数计算方法

经过变换后,仅需计算 6 种标准相交类型的网格单元体积分数。然而,不同相交类型的网格单元体积分数仍需采用差异化的计算过程。为简化这一过程,本文提出了一种通用的体积分数计算方法。如图 8 所示,不同的标准相交类型被统一抽象为通用计算类型。对于标准相交类型 1,  $A_1$  和  $A_3$  的体积分数为 0,而  $A_2$  则采用前述基于积分方法(式(6)) 计算不规则区域的体积分数。其余相交类型网格的体积分数也可采用相似的方法进行计算。

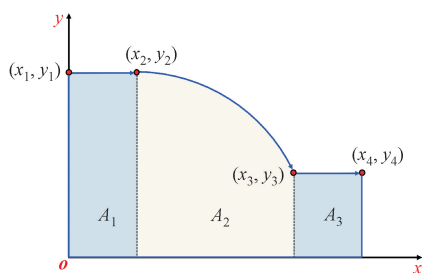


图 8 通用体积分数计算类型

Fig. 8 General types for calculating volume fractions

通用体积分数的计算公式如下

$$\varphi = \sum_{i=1}^m [\lambda \varphi_1(x_i, x_{i+1}) + (1-\lambda) \varphi_2(x_i, x_{i+1})] \quad (7)$$

式中:  $\varphi_1$  和  $\varphi_2$  分别为式(5) 和式(6) 计算的体积分数;  $m$  为圆形界面与网格单元交点的总数;  $\lambda$  为取值为 0 或 1 的指示变量。当计算区域为规则矩形区域时,  $\lambda = 1$ , 采用式(5) 计算体积分数; 当计算区域为不规则区域时,  $\lambda = 0$ , 采用式(6) 计算体积分数。

在实际计算过程中,可能会出现图 9 所示的两种特殊情况,均可通过通用计算类型和相应公式处理。

对于特殊类型 1, 通用计算类型中  $A_1$  和  $A_3$  的体积分数为 0,  $A_2$  的体积分数可直接用式(6) 计算。对于特殊类型 2, 通用计算类型中  $A_1$  和  $A_3$  的体积分数同样为 0, 但存在 3 个  $A_2$  区域, 其体积分数均可通过式(6) 计算。然而, 特殊类型 2 的结果中包含超出网格单元边界的体积分数部分, 可用以下公式修正

$$\varphi'(x_2, x_3) = \min(\varphi(x_2, x_3), (x_3 - x_2) \Delta y) \quad (8)$$

式中:  $\varphi'(x_2, x_3)$  为修正后的体积分数;  $\varphi(x_2, x_3)$  为通过式(6) 计算出来的体积分数。

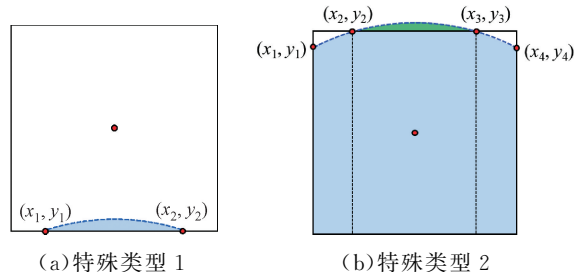


图 9 特殊相交类型

Fig. 9 Special intersection types

最终得到的通用体积分数计算公式如下

$$\varphi = \sum_{i=1}^m \min(\lambda \varphi^1(x_i, x_{i+1}) + (1-\lambda) \varphi^2(x_i, x_{i+1}), (x_{i+1} - x_i) \Delta y) \quad (9)$$

### 1.3.3 交点求解与排序

上述讨论基于交点坐标和顺序已知的前提。但在实际计算中,交点坐标需通过额外求解获得,其顺序通常为乱序。因此,在计算体积分数之前需要对交点进行排序。具体步骤如下:首先,在已知圆形界面的半径和圆心位置的情况下,求解圆形界面与网格单元各边界的交点坐标;随后按照  $x$  坐标从小到大进行排序;接着根据区域类型选择合适的体积分数计算公式,矩形区域用式(5),不规则区域用式(6);最后,将各部分体积分数相加,得到网格单元的最终体积分数。

## 2 基于积分方法的 VOF 通量计算方法

CIR 算法计算网格单元流出的 VOF 通量的过程如图 10 所示。其中,  $u_l$ 、 $u_r$  分别为左侧、右侧边界中心处速度在  $x$  方向的分量;  $u_b$ 、 $u_t$  分别为底部、顶部中心处速度在  $y$  方向上的分量;  $\Delta t$  为时间步长。

CIR 算法在计算 VOF 通量时采用与界面重构过程中相同的分块计算方法。然而,这种方法在计算 VOF 通量时仍存在精度和效率方面的问题。在使用分块方法计算右侧边界流出的 VOF 通量的过程中,可能产生如图 11 所示的两种误差类型:①对流出范围内的小块,计算的流出 VOF 通量与实际流出的 VOF 通量之间的误差为  $V_1$  与  $V_3$  的差;②对位于流出范围边界处的小块,计算的流出 VOF 通量与实际流出的 VOF 通量的误差为  $V_5$  与  $V_6$  的差。

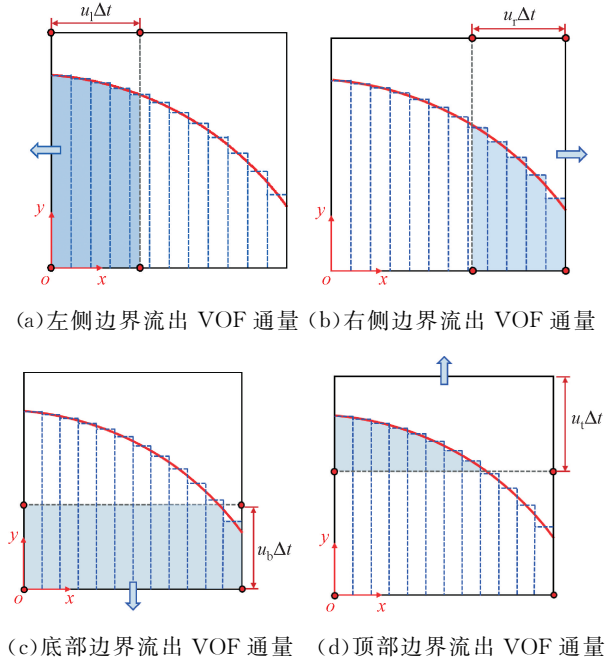


图 10 CIR 算法计算 VOF 通量示意图

Fig. 10 Schematic diagram of the CIR algorithm for calculating VOF fluxes

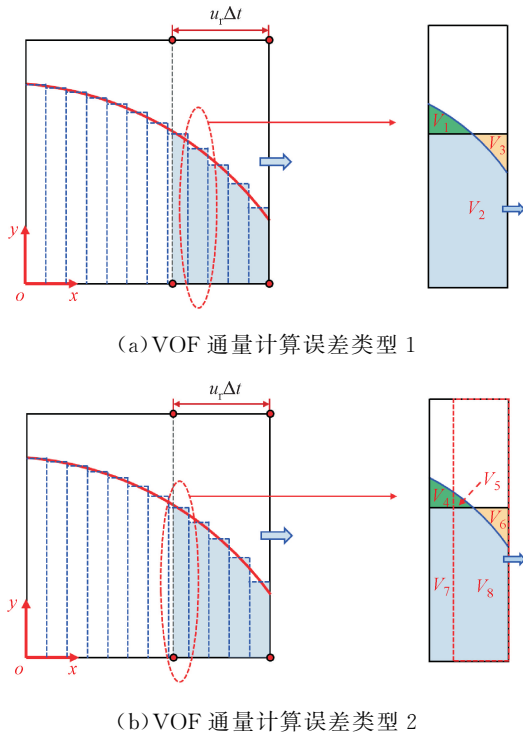


图 11 VOF 通量计算误差的示意图

Fig. 11 Schematic diagram of the error in VOF flux calculation

对于其他边界处的流出 VOF 通量,同样会出现类似的误差问题。虽然增加分块数能够减少误差,但同时显著增加计算量,降低计算效率。为此,

本文提出了一种基于积分方法的计算策略,用于高效且精确地计算 VOF 通量。

考虑到流出范围的不同,网格单元左侧边界流出 VOF 通量可分为图 12 所示的 3 种情况。第 1 种情况下,流出区域为规则矩形,其体积可采用矩形区域体积分数计算公式(式(5))直接计算。对于第 2 种情况,流动范围的右边界与圆形界面相交,形成一个不规则区域。该不规则区域可进一步拆分为矩形区域和一个小的不规则区域,其中不规则部分的体积分数可基于积分方法计算(式(6))。对于第 3 种情况,尽管流动范围的右边界未与圆形界面存在相交,但其体积分数计算方法与第 2 种情况一致。针对以上 3 种情况,本文提出了一种通用的网格单元左侧流出 VOF 通量计算公式,如下所示

$$\Delta\varphi_l = \sum_{i=1}^m \epsilon_i (\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1})) \quad (10)$$

式中:  $\epsilon$  为流出范围指示变量,取值为 0 或 1。当  $\epsilon = 0$  时,表示当前区域不在流出范围内;当  $\epsilon = 1$  时,表示当前区域在流出范围内。

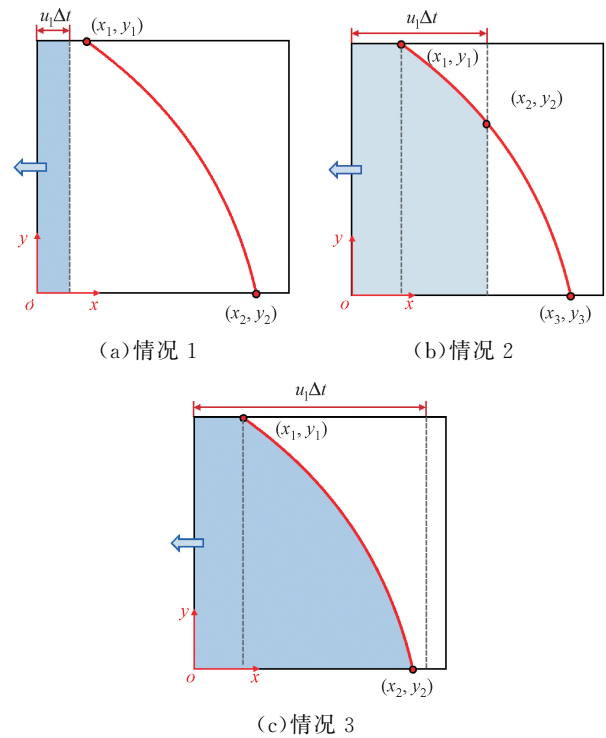


图 12 不同情况下网格单元左侧流出的 VOF 通量

Fig. 12 Outflow VOF fluxes from the left side of the grid cell for different scenarios

网格单元右侧边界流出 VOF 通量的计算方法与左侧边界类似,其通用计算公式如下

$$\Delta\varphi_r = \sum_{i=1}^m \epsilon_i (\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1})) \quad (11)$$

网格单元底部流出 VOF 通量的示意图如图 13 所示。由于底部边界流体流出区域的大小不同,可能出现多种情况。这些情况下 VOF 流量均可采用通用计算公式如下

$$\Delta\varphi_b = \sum_{i=1}^m \min(\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1}), (x_{i+1} - x_i) |u_b| \Delta t) \quad (12)$$

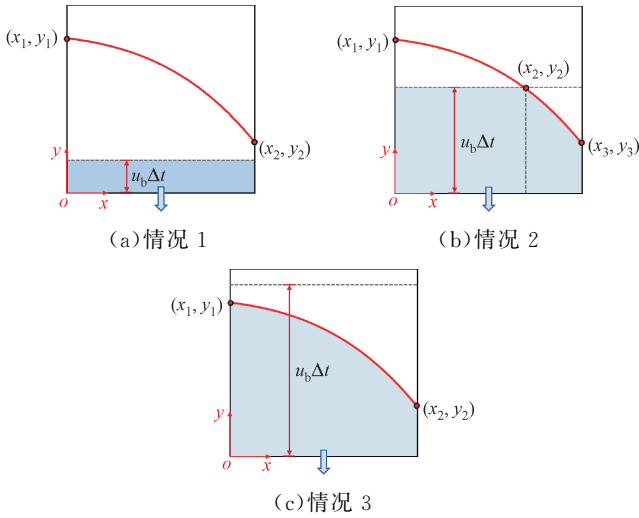


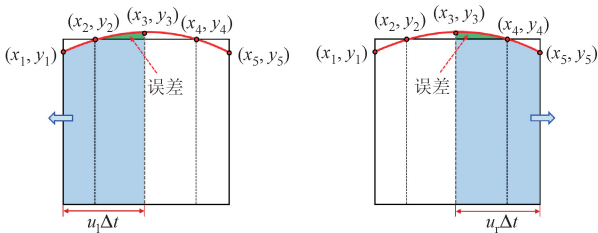
图 13 不同情况下网格单元底部流出的 VOF 流量

Fig. 13 Outflow VOF fluxes from the bottom of the grid cell for different scenarios

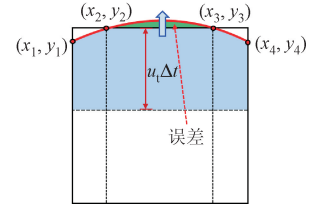
类似地,顶部边界流出 VOF 通量的通用计算公式如下

$$\Delta\varphi_t = \sum_{i=1}^m \max(\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1}) - (x_{i+1} - x_i) (\Delta y - |u_t| \Delta t), 0) \quad (13)$$

如图 14 所示,具有特殊相交类型 2 的网格单元,存在超出网格单元的一部分流体,在计算具有这类网格单元 VOF 流量时,超出的流体会使得计算的左侧边界流出 VOF 流量,右侧边界流出 VOF 流量以及顶部边界流出 VOF 流量产生误差,因此也需要对这些 VOF 流量的计算公式进行修正。



(a) 左侧流出流量计算误差 (b) 右侧流出流量计算误差



(c) 顶部流出流量计算误差

图 14 特殊相交类型计算 VOF 流量的误差

Fig. 14 Errors in calculating VOF fluxes for special intersection types

修正后的计算公式如下

$$\Delta\varphi'_l = \sum_{i=1}^m \min(\epsilon_i (\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1})), (x_{i+1} - x_i) \Delta y) \quad (14)$$

$$\Delta\varphi'_r = \sum_{i=1}^m \min(\epsilon_i (\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1})), (x_{i+1} - x_i) \Delta y) \quad (15)$$

$$\Delta\varphi'_t = \sum_{i=1}^m \min(\max(\lambda_i \varphi^1(x_i, x_{i+1}) + (1 - \lambda_i) \varphi^2(x_i, x_{i+1}) - (x_{i+1} - x_i) (\Delta y - |u_t| \Delta t), 0), (x_{i+1} - x_i) |u_t| \Delta t) \quad (16)$$

计算出从网格单元 4 个边界流出的 VOF 流量后,可以得到当前网格单元下一时刻的体积分数,即

$$\varphi^{t+\Delta t} = \varphi^t + \Delta\varphi'_b - \Delta\varphi'_l + \Delta\varphi'_r - \Delta\varphi'_t \quad (17)$$

式中:  $t$  为当前时刻;  $\varphi^{t+\Delta t}$  为下一时刻网格单元的体积分数;  $\varphi^t$  为当前已知时刻网格单元的体积分数;  $\Delta\varphi'$  为修正后的体积分数。

### 3 案例测试

本节将通过经典的静态界面重构测试算例和动态界面捕捉测试算例,对本文提出的 ICIR 算法在界面重构及 VOF 流量计算方面的精度和效率进行全面评估。

#### 3.1 静态界面重构

随机界面重构测试算例常用于评估界面重构算法的精度<sup>[28]</sup>。在测试中,算法需在计算域  $1.0 \times 1.0$  内重构不同形状的界面。重构界面的精度可通过  $L^1$  误差衡量,其计算公式如下

$$L^1 = \frac{1}{l} \iint |F^{\text{ex}}(x, y) - F^{\text{re}}(x, y)| dx dy \quad (18)$$

式中:  $l$  为界面长度;  $F^{\text{ex}}(x, y)$  为精确相界面的特征函数;  $F^{\text{re}}(x, y)$  为重构界面的特征函数。上述公式中的积分项等于重构界面与精确界面之间的误差面积,因此  $L^1$  误差能够衡量重构界面与精确相界面之

间的距离误差。重构界面与精确界面越接近,  $L^1$  误差越小, 算法重构界面的精度越高。

### 3.1.1 随机圆形界面重构

在前文介绍重构算法时提到, 阈值  $\epsilon$  可作为判断计算体积分数与真实体积分数是否相等的标准。在不同的阈值  $\epsilon$  下, 重构圆形界面的  $L^1$  误差如表 1 所示。随着阈值  $\epsilon$  的减小, 重构界面的  $L^1$  误差逐渐降低, 但误差减少的幅度也逐步减小。当阈值降至  $1.0 \times 10^{-8}$  时, 误差不再变化。因此, 后续计算中将阈值固定为  $1.0 \times 10^{-8}$ 。

表 1 不同阈值  $\epsilon$  下重构圆形界面  $L^1$  误差

Table 1  $L^1$  error of different threshold values of  $\epsilon$  for reconstructing circular interfaces

| 阈值                   | $L^1$ 误差              | 变化率/%  |
|----------------------|-----------------------|--------|
| $1.0 \times 10^{-4}$ | $3.62 \times 10^{-7}$ |        |
| $1.0 \times 10^{-5}$ | $8.21 \times 10^{-8}$ | -77.35 |
| $1.0 \times 10^{-6}$ | $7.16 \times 10^{-8}$ | -12.71 |
| $1.0 \times 10^{-7}$ | $7.12 \times 10^{-8}$ | -0.65  |
| $1.0 \times 10^{-8}$ | $7.11 \times 10^{-8}$ | -0.05  |
| $1.0 \times 10^{-9}$ | $7.11 \times 10^{-8}$ | 0.00   |

在随机圆形界面重构过程中, 圆形界面的圆心位置在  $(0.5, 0.5)$  附近变化, 半径在  $0.20 \sim 0.25$  变化。采用 CIR 算法进行界面重构时, 分块数的变化会对重构界面的精度产生影响。图 15 展示了不同分块数下 CIR 算法重构 1 000 个圆形界面的平均  $L^1$  误差, 其中  $P$  代表分块数。总体而言, 随着网格加密, CIR 算法的平均  $L^1$  误差显著降低, 重构精度逐步提高。在粗网格条件下, 不同分块数的 CIR 算法表现出相近的  $L^1$  误差。这是因为此时需要界面重构的网格单元数较少, 增加分块数对整体精度的提升作用有限。然而, 随着网格变密, 需要界面重构的网格单元数逐渐增多, 增加分块数对精度的提升效果变得显著。当网格数为  $80^2$  时, 将 CIR 算法的分块数从 20 增加至 100, 算法误差降低了 56.7%; 而当网格数为  $320^2$  时, 在相同条件下, 误差降低幅度进一步提升至 88.5%。不过, 随着分块数的逐渐增多, CIR 算法精度的提升效果会逐渐下降。当 CIR 算法的分块数从 80 增加至 100 时, 精度的提升效果便不是很明显。为平衡 CIR 算法的精度与计算量, 后续测试中将 CIR 算法的分块数设定为 80。

不同界面重构算法在随机圆形界面重构测试中的平均  $L^1$  误差如表 2 所示。可以看出, 相较于平面

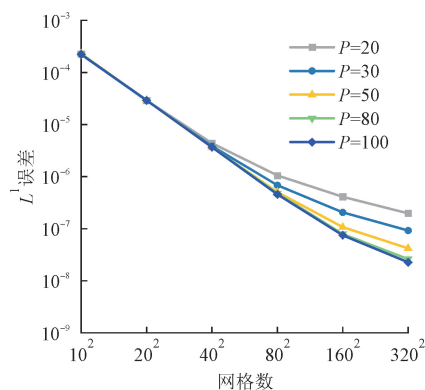


图 15 不同分块数下 CIR 算法重构界面的  $L^1$  误差

Fig. 15  $L^1$  error of the CIR algorithm for reconstructing interfaces with different number of pieces

重构方法(如 PLIC 和 ELVIRA 算法), 曲面重构方法在所有测试网格数下均表现出更低的  $L^1$  误差, 具有更高的界面重构精度。此外, 与 CIR 算法相比, 基于积分的 ICIR 算法能够进一步提高界面重构的精度, 且随着网格逐渐加密, 精度提升效果越加显著。当网格数从  $10^2$  加密至  $320^2$  时, ICIR 算法的  $L^1$  误差降低比例从 0.53% 上升至 27.15%。在所有测试的网格数下, ICIR 算法相较于 CIR 算法的平均精度提升 7.71%。

界面长度在模拟液滴蒸发等涉及质量传递的计算中至关重要<sup>[21]</sup>。因此, 也可通过重构界面长度与精确界面长度之间的相对误差来评估重构算法的精度。不同算法重构圆形界面的长度相对误差如表 3 所示。与 PLIC 和 PCLIC 两种线性重构算法相比, CIR 和 ICIR 算法的重构界面长度相对误差更小, 并且随着网格分辨率的提高, 两种算法的长度相对误差进一步降低。相比 CIR 算法, ICIR 算法的长度相对误差更小, 总体而言, 不同网格数下的平均重构圆形界面长度误差降低了 26.8%。

在 CIR 算法和 ICIR 算法进行静态界面重构的过程中, 求解界面方程是核心步骤, 包括确定圆形界面的半径和圆心位置。其中, 半径的求解耗时可以忽略不计, 而圆心位置的确定占据了绝大部分计算时间。CIR 算法和 ICIR 算法在不同网格数下求解界面方程的耗时以及 ICIR 算法的加速比如表 4 所示。结果显示, 在所有网格数下, ICIR 算法的耗时均显著低于 CIR 算法, 其求解加速比始终保持在 8 以上, 平均加速比达为 6.78。这种性能提升得益于 ICIR 算法提出的快速积分方法, 该方法能够高效计算体积分数, 从而显著缩短界面方程的求解耗时。



表 2 不同算法重构随机圆形界面的平均  $L^1$  误差Table 2 Average  $L^1$  error of different algorithms for reconstructing random circular interfaces

| 算法   |        | $L^1$ 误差              |                       |                       |                       |                       |                       |
|------|--------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|      |        | $10^2$                | $20^2$                | $40^2$                | $80^2$                | $160^2$               | $320^2$               |
| 平面重构 | PLIC   | $1.37 \times 10^{-3}$ | $3.60 \times 10^{-4}$ | $1.20 \times 10^{-4}$ | $5.04 \times 10^{-5}$ | $2.32 \times 10^{-5}$ | $1.12 \times 10^{-5}$ |
|      | ELVIRA | $2.63 \times 10^{-3}$ | $8.34 \times 10^{-4}$ | $2.59 \times 10^{-4}$ | $7.39 \times 10^{-5}$ | $1.97 \times 10^{-5}$ | $5.13 \times 10^{-6}$ |
| 曲面重构 | CIR    | $2.25 \times 10^{-4}$ | $2.94 \times 10^{-5}$ | $3.70 \times 10^{-6}$ | $4.74 \times 10^{-7}$ | $7.89 \times 10^{-8}$ | $2.60 \times 10^{-8}$ |
|      | ICIR   | $2.24 \times 10^{-4}$ | $2.87 \times 10^{-5}$ | $3.63 \times 10^{-6}$ | $4.63 \times 10^{-7}$ | $6.94 \times 10^{-8}$ | $1.90 \times 10^{-8}$ |

表 3 不同算法重构圆形界面长度相对误差

Table 3 Relative errors in reconstructed circular interface lengths for different algorithms

| 算法    | 相对误差/% |        |        |         |         |
|-------|--------|--------|--------|---------|---------|
|       | $20^2$ | $40^2$ | $80^2$ | $160^2$ | $320^2$ |
| PLIC  | 1.440  | 0.940  | 0.910  | 0.890   | 0.900   |
| PCLIC | 2.610  | 1.850  | 1.260  | 0.760   | 0.440   |
| CIR   | 0.149  | 0.036  | 0.014  | 0.007   | 0.005   |
| ICIR  | 0.124  | 0.034  | 0.011  | 0.005   | 0.002   |

表 4 CIR 算法和 ICIR 算法求解界面方程耗时

Table 4 Computational time for solving the interface equation using the CIR and ICIR algorithms

| 算法   | 耗时/s   |        |        |        |         |         |
|------|--------|--------|--------|--------|---------|---------|
|      | $10^2$ | $20^2$ | $40^2$ | $80^2$ | $160^2$ | $320^2$ |
| CIR  | 0.42   | 0.89   | 1.86   | 2.64   | 8.45    | 14.46   |
| ICIR | 0.06   | 0.15   | 0.20   | 0.43   | 1.23    | 2.55    |
| 加速比  | 6.74   | 6.05   | 9.25   | 6.14   | 6.85    | 5.67    |

### 3.1.2 随机椭圆界面重构

在随机椭圆界面重构过程中,椭圆的中心固定在  $(0.5, 0.5)$  位置,其朝向随机变化。椭圆界面可描述为

表 5 不同算法重构随机椭圆界面的平均  $L^1$  误差Table 5 Average  $L^1$  error of different algorithms for reconstructing random elliptical interfaces

| 算法               | $L^1$ 误差              |                       |                       |                       |                       |                       |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|                  | $10^2$                | $20^2$                | $40^2$                | $80^2$                | $160^2$               | $320^2$               |
| ELVIRA           | $2.97 \times 10^{-3}$ | $5.54 \times 10^{-4}$ | $1.11 \times 10^{-4}$ | $2.57 \times 10^{-5}$ | $6.25 \times 10^{-6}$ | $1.56 \times 10^{-6}$ |
| PCIC(无 $C^0$ 修正) | $1.81 \times 10^{-3}$ | $2.67 \times 10^{-4}$ | $3.83 \times 10^{-5}$ | $4.92 \times 10^{-6}$ | $6.25 \times 10^{-7}$ | $8.24 \times 10^{-8}$ |
| PCIC(有 $C^0$ 修正) | $1.23 \times 10^{-3}$ | $1.11 \times 10^{-4}$ | $1.26 \times 10^{-5}$ | $1.39 \times 10^{-6}$ | $1.65 \times 10^{-7}$ | $2.14 \times 10^{-8}$ |
| CIR              | $1.80 \times 10^{-3}$ | $2.18 \times 10^{-4}$ | $2.64 \times 10^{-5}$ | $3.26 \times 10^{-6}$ | $4.22 \times 10^{-7}$ | $6.41 \times 10^{-8}$ |
| ICIR             | $1.76 \times 10^{-3}$ | $2.15 \times 10^{-4}$ | $2.63 \times 10^{-5}$ | $3.21 \times 10^{-6}$ | $4.11 \times 10^{-7}$ | $5.71 \times 10^{-8}$ |

### 3.2 动态界面捕捉

本节通过气泡水平移动和时间反转单涡问题两

$$\frac{(x' - 0.5)^2}{a^2} + \frac{(y' - 0.5)^2}{b^2} = 1 \quad (19)$$

式中:  $a^2 = 0.12$ ;  $b^2 = 0.02$ 。

在随机椭圆界面重构测试中,不同界面重构算法的平均  $L^1$  误差如表 5 所示。与线性重构算法 ELVIRA 相比,PCIC(无  $C^0$  修正)、PCIC(有  $C^0$  修正)、CIR 和 ICIR 算法的误差较小,表明曲面重构有助于提高界面重构精度。CIR 和 ICIR 算法的误差介于 PCIC(无  $C^0$  修正)和 PCIC(有  $C^0$  修正)之间。 $C^0$  修正需要先判断相界面不连续性的类型,并针对不同类型采用相应方法强制合并不连续点。然而,此过程使得体积分数不守恒,因此需适当调整界面参数以满足体积分数守恒条件。 $C^0$  修正增强了界面连续性,提高了算法精度,但也增加了计算复杂度。相比 CIR 算法,ICIR 算法的  $L^1$  误差更低,不同网格数下的平均重构椭圆界面  $L^1$  误差减少了 3.08%。总体而言,与 PCIC 算法相比,CIR 和 ICIR 算法无需初始 PLIC 重构及  $C^0$  修正,计算过程更简便,且精度较高。

总体而言,在静态界面重构测试中,与 CIR 算法相比,ICIR 算法表现出更高的重构精度和更少的界面方程求解耗时。

个算例,验证所提 ICIR 算法在动态界面捕捉过程中的精度与效率。在这两个测试算例中,采用形变

误差  $D_e$  作为评估算法精度的指标,计算公式如下

$$D_e = \frac{\sum_{i,j} |\varphi_{i,j} - \varphi_{i,j}^0|}{\sum_{i,j} \varphi_{i,j}^0} \quad (20)$$

式中:  $\varphi_{i,j}$  为计算得到的网格单元  $(i, j)$  的体积分数;  $\varphi_{i,j}^0$  为网格单元  $(i, j)$  的精确体积分数。

### 3.2.1 气泡水平移动

在  $1.0 \times 1.0$  的计算域中, 归一化半径为 0.20、初始位置为  $(0.3, 0.5)$  的气泡在水平恒定流场的作用下, 以  $1.0 \text{ m/s}$  的速度向右移动。经过  $0.4 \text{ s}$  后, 汽

泡移动至  $(0.7, 0.5)$ 。不同网格数下, VOSET-PLIC 算法、CIR 算法和 ICIR 算法的形变误差如表 6 所示。在水平移动气泡过程中, CIR 算法和 ICIR 算法相比于 VOSET-PLIC 算法表现出更低的形变误差。这主要得益于曲面重构算法能够更精确地重构界面, 从而提高了 VOF 通量计算和体积分数计算的精度, 进而提升了气泡捕捉精度。此外, 与 CIR 算法相比, ICIR 算法在不同网格数下均表现出更低的形变误差, 精度平均提升了  $3.96\%$ 。

表 6 水平移动气泡的形变误差

Table 6 Deformation error of bubble during horizontal translation

| 算法         | $D_e$                 |                       |                       |                       |                       |
|------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|            | $25^2$                | $50^2$                | $100^2$               | $200^2$               | $250^2$               |
| VOSET-PLIC | $9.92 \times 10^{-3}$ | $2.74 \times 10^{-3}$ | $1.09 \times 10^{-3}$ | $3.88 \times 10^{-4}$ | $2.63 \times 10^{-4}$ |
| CIR        | $2.33 \times 10^{-4}$ | $2.04 \times 10^{-4}$ | $4.83 \times 10^{-5}$ | $3.58 \times 10^{-5}$ | $3.12 \times 10^{-5}$ |
| ICIR       | $2.07 \times 10^{-4}$ | $2.04 \times 10^{-4}$ | $4.77 \times 10^{-5}$ | $3.42 \times 10^{-5}$ | $3.02 \times 10^{-5}$ |

CIR 算法和 ICIR 算法在气泡水平移动过程中的耗时如表 7 所示。ICIR 算法的耗时显著低于 CIR 算法, 这是因为基于积分方法的计算过程能够更高效快速地完成界面重构和 VOF 通量计算, 从而减少整体计算时间。平均而言, ICIR 算法的计算耗时相较于 CIR 算法缩短了  $40.88\%$ 。

表 7 水平移动气泡耗时

Table 7 Computational time for horizontal bubble translation

| 算法   | 耗时/s   |        |         |         |         |
|------|--------|--------|---------|---------|---------|
|      | $25^2$ | $50^2$ | $100^2$ | $200^2$ | $250^2$ |
| CIR  | 0.086  | 0.305  | 1.066   | 3.982   | 5.694   |
| ICIR | 0.072  | 0.168  | 0.563   | 1.905   | 3.198   |

在气泡水平移动问题中, 与 CIR 算法相比, ICIR 算法不仅表现出更高的界面捕捉精度, 还具有更高的计算效率。

### 3.2.2 时间反转单涡问题

在计算域  $1.0 \times 1.0$  内  $(0.5, 0.75)$  位置处, 存在一个半径为 0.15 的圆形气泡。计算域内还存在一个随时间变化的涡场, 其速度分量与时间的关系如下

$$u_x(x, y, t) = -\sin(2\pi y)\sin^2(\pi x)\cos(\pi t/T) \quad (21)$$

$$u_y(x, y, t) = \sin(2\pi x)\sin^2(\pi y)\cos(\pi t/T) \quad (22)$$

式中:  $x$  和  $y$  为网格单元中心的坐标;  $u_x$  和  $u_y$  分别  $x$  方向和  $y$  方向的速度分量;  $T$  为涡场变化的周期, 取为  $8.0 \text{ s}$ 。

气泡在时间反转涡场的作用下发生变形, 经过一个周期后, 气泡回到初始位置并恢复初始形状。界面重构算法的精度越高, 气泡恢复后的形状越接近初始状态, 即变形率越小。因此, 该算例可有效评估界面重构算法在复杂流场中的性能。

VOSET-PLIC 算法、CIR 算法和 ICIR 算法在不同网格数下捕捉气泡的形变误差如表 8 所示。在不同的网格数下, CIR 和 ICIR 这两种曲面重构算法捕捉气泡的变形率均低于平面重构算法 VOSET-PLIC, 这表明曲面重构算法在复杂流场中对气泡捕捉的精度高于平面重构算法。此外, 相较于 CIR 算法, ICIR 算法在气泡捕捉中的形变误差进一步降低, 平均精度提升了  $13.09\%$ 。

CIR 算法和 ICIR 算法捕捉气泡的耗时如图 16 所示。ICIR 算法在不同网格数下的耗时均显著低于 CIR 算法, 计算耗时分别缩短了  $63.26\%$ 、 $69.29\%$ 、 $69.84\%$  和  $70.85\%$ 。平均而言, ICIR 算法的计算耗时较 CIR 算法缩短了  $68.31\%$ 。

对于复杂的时间反转单涡问题, ICIR 算法相比 CIR 算法不仅具有更高的计算精度, 还能显著缩短计算耗时。因此, 相较于 CIR 算法, ICIR 算法在简单和复杂问题中的计算精度和效率优势都得到了验证。

表8 时间反转单涡问题中气泡的形变误差

Table 8 Deformation errors of bubbles in the time-reversal single-vortex problem

| 算法         | $D_e$  |         |         |         |
|------------|--------|---------|---------|---------|
|            | $50^2$ | $100^2$ | $150^2$ | $200^2$ |
| VOSET-PLIC | 0.405  | 0.075   | 0.040   | 0.027   |
| CIR        | 0.241  | 0.040   | 0.021   | 0.012   |
| ICIR       | 0.223  | 0.032   | 0.019   | 0.010   |

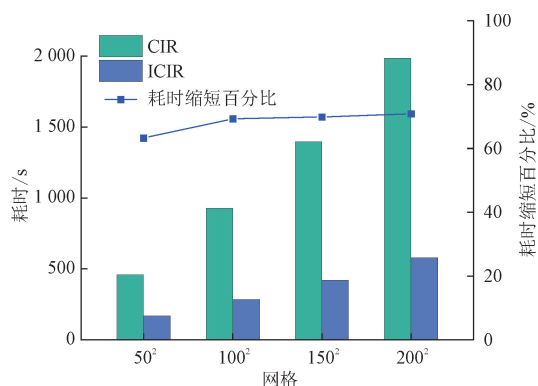


图16 时间反演单涡问题中两种算法的耗时以及ICIR算法的耗时缩短百分比

Fig. 16 Computational time of two algorithms and the percentage reduction in computation time of the ICIR algorithm in the time-reversal single-vortex problem

## 4 结 论

(1)对于静态界面重构问题,与平面重构算法相比,曲面重构算法能够显著提升重构界面的精度。不同网格数下,基于曲面重构的CIR算法以及ICIR算法的界面重构精度均高于基于平面重构的PLIC算法和ELVIRA算法。相较于CIR算法,ICIR算法具有更高的界面重构精度和计算效率。相较于采用80个分块数的CIR算法,ICIR算法在不同网格数下的界面重构平均精度提升了7.71%,界面方程求解加速比达到6.78。

(2)对于动态捕捉问题,曲面重构算法相较于平面重构算法,仍然具有更高的界面捕捉精度。在水平流场和时间反转单涡流场中的气泡动态捕捉测试算例中,CIR算法和ICIR算法捕捉气泡的精度均高于VOSET-PLIC算法。此外,相较于CIR算法,ICIR算法可实现简单水平流场和复杂的时间反转单涡流场中实现更准确的界面重构,在不同网格数下的平均计算精度分别提高了3.96%和13.09%,平均计算耗

时分别缩短了40.88%和68.31%。

## 参考文献:

- [1] 伍坤一,张露露,林宇,等. 四川某气田集输管线积液影响规律研究[J]. 西南石油大学学报(自然科学版), 2024, 46(1): 136-146.  
WU Kunyi, ZHANG Lulu, LIN Yu, et al. A study on the law of liquid accumulation influence in southern Sichuan wet gas pipeline [J]. Journal of Southwest Petroleum University (Science & Technology Edition), 2024, 46(1): 136-146.
- [2] 闵元,李晓平,宫敬,等. 气液两相流管道内腐蚀失效概率计算方法[J]. 油气储运, 2018, 37(12): 1349-1356.  
MIN Yuan, LI Xiaoping, GONG Jing, et al. A calculation method for the failure probability of gas-liquid two-phase flow pipeline caused by internal corrosion [J]. Oil & Gas Storage and Transportation, 2018, 37(12): 1349-1356.
- [3] 韩宇,刘志军,王云峰,等. T型微通道反应器内气液两相流动机制及影响因素[J]. 力学学报, 2019, 51(2): 441-449.  
HAN Yu, LIU Zhijun, WANG Yunfeng, et al. Gas-liquid two-phase flow regimes and impact factors in T-junction microreactor [J]. Chinese Journal of Theoretical and Applied Mechanics, 2019, 51(2): 441-449.
- [4] MUBAROK M H, ZARROUK S J, CATER J E. Two-phase flow measurement of geothermal fluid using orifice plate: field testing and CFD validation [J]. Renewable Energy, 2019, 134: 927-946.
- [5] 宋云超. 气液两相流动相界面追踪方法及液滴撞击壁面运动机制的研究[D]. 北京: 北京交通大学, 2012.
- [6] BALZ R, VON ROTZ B, SEDARSKY D. In-nozzle flow and spray characteristics of large two-stroke marine diesel fuel injectors [J]. Applied Thermal Engineering, 2020, 180: 115809.
- [7] MCKEE S, TOMÉ M F, FERREIRA V G, et al. The MAC method [J]. Computers & Fluids, 2008, 37(8): 907-930.
- [8] HIRT C W, NICHOLS B D. Volume of fluid (VOF) method for the dynamics of free boundaries [J]. Journal of Computational Physics, 1981, 39(1): 201-225.
- [9] SUSSMAN M, FATEMI E, SMERKA P, et al. An improved level set method for incompressible two-phase flows [J]. Computers & Fluids, 1998, 27(5/6): 663-680.
- [10] MULBAH C, KANG Can, MAO Ning, et al. A review of VOF methods for simulating bubble dynamics [J]. Progress in Nuclear Energy, 2022, 154: 104478.

- [11] YOKOI K. Efficient implementation of THINC scheme: a simple and practical smoothed VOF algorithm [J]. *Journal of Computational Physics*, 2007, 226(2): 1985-2002.
- [12] BORTOLIN S, DA RIVA E, DEL COL D. Condensation in a square minichannel: application of the VOF method [J]. *Heat Transfer Engineering*, 2013, 35(2): 193-203.
- [13] POZZETTI G, PETERS B. A multiscale DEM-VOF method for the simulation of three-phase flows [J]. *International Journal of Multiphase Flow*, 2018, 99: 186-204.
- [14] LI Tong, WANG Shiping, LI Shuai, et al. Numerical investigation of an underwater explosion bubble based on FVM and VOF [J]. *Applied Ocean Research*, 2018, 74: 49-58.
- [15] 张漫漫, 孙姣, 陈文义. 一种基于几何重构的 Youngs-VOF 耦合水平集追踪方法 [J]. *力学学报*, 2019, 51(3): 775-786.
- ZHANG Manman, SUN Jiao, CHEN Wenyi. An interface tracking method of coupled Youngs-VOF and level set based on geometric reconstruction [J]. *Chinese Journal of Theoretical and Applied Mechanics*, 2019, 51(3): 775-786.
- [16] 孟巍, 梅桂林. 基于机器学习的二维流体 VOF 方法 [J]. *中国造船*, 2024, 65(3): 244-253.
- MENG Wei, MEI Guilin. A machine learning-based VOF method for 2D flowfield [J]. *Shipbuilding of China*, 2024, 65(3): 244-253.
- [17] CHEN Yujie, LU Wei, SUN Dongliang, et al. A horizontal refined piecewise curve interface reconstruction (HOPCIR) algorithm for reconstructing the vapor-liquid interface [J]. *International Journal of Multiphase Flow*, 2024, 178: 104905.
- [18] RIDER W J, KOTHE D B. Reconstructing volume tracking [J]. *Journal of Computational Physics*, 1998, 141(2): 112-152.
- [19] IZADI A, MOHEBBI A, FEILI MONFARED A E. Combining lattice Boltzmann and smoothed profile methods for calculating the interface normal vectors and its application for simulating dissolution phenomenon [J]. *Physics of Fluids*, 2023, 35(2): 023335.
- [20] ITO K, KUNUGI T, OHNO S, et al. A high-precision calculation method for interface normal and curvature on an unstructured grid [J]. *Journal of Computational Physics*, 2014, 273: 38-53.
- [21] VIGNESH T G, BAKSHI S. Noniterative interface reconstruction algorithms for volume of fluid method [J]. *International Journal for Numerical Methods in Fluids*, 2013, 73(1): 1-18.
- [22] CHORIN A J. Curvature and solidification [J]. *Journal of Computational Physics*, 1985, 57(3): 472-490.
- [23] RENARDY Y, RENARDY M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method [J]. *Journal of Computational Physics*, 2002, 183(2): 400-421.
- [24] SCARDOVELLI R, ZALESKI S. Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection [J]. *International Journal for Numerical Methods in Fluids*, 2003, 41(3): 251-274.
- [25] DIWAKAR S V, DAS S K, SUNDARARAJAN T. A quadratic spline based interface (QUASI) reconstruction algorithm for accurate tracking of two-phase flows [J]. *Journal of Computational Physics*, 2009, 228(24): 9107-9130.
- [26] MAITY R K, SUNDARARAJAN T, VELUSAMY K. An accurate interface reconstruction method using piecewise circular arcs [J]. *International Journal for Numerical Methods in Fluids*, 2021, 93(1): 93-126.
- [27] CHEN Yujie, GONG Junhua, LU Wei, et al. A circle-based interface reconstruction algorithm based on the coupled volume-of-fluid and level set method [J]. *Physics of Fluids*, 2024, 36(3): 032128.
- [28] PILLIOD J E J, PUCKETT E G. Second-order accurate volume-of-fluid algorithms for tracking material interfaces [J]. *Journal of Computational Physics*, 2004, 199(2): 465-502.

(编辑 亢列梅)