

自适应网格在水平集方法中的优化与应用

任宇佳¹, 马朝青^{1,2}

(1. 烟台大学 计算机与控制工程学院 山东 烟台 264010;

2. 高端海洋工程装备智能技术烟台市重点实验室 山东 烟台 264010)

摘要: 在水平集方法对运动界面进行追踪的过程中,利用自适应笛卡尔网格可以在水平集演化的关键区域对网格进行自动细化。水平集方程每个时间步求解后,细化区域和粗化区域的选择是决定迭代求解效率的一个重要因素,由于不同细化级别边界处的节点部分缺少同级别的邻接点,加大了求解水平集演化方程中偏微分近似解的复杂度。提出一种优化后的笛卡尔自适应网格方法,引入了窄带法对细化区域进行自动选择,使用了 red-green 混合细化策略,在窄带的边界处对 T 节点进行消除。加入了优化笛卡尔自适应网格的水平集方法被应用于模拟血管内血栓生长的计算模型中,提高了血栓仿真实验的计算效率。

关键词: 自适应网格法; 水平集; 窄带法; red-green 细化

中图分类号: TP399

文献标志码: A

文章编号: 1671-6841(2022)02-0016-08

DOI: 10.13705/j.issn.1671-6841.2021223

Optimization and Application of Cartesian Adaptive Grid in Level Set Method

REN Yujia¹, MA Chaoqing^{1,2}

(1. School of Computer and Control Engineering, Yantai University, Yantai 264005, China; 2. Yantai Key Laboratory of High-end Ocean Engineering Equipment and Intelligent Technology, Yantai 264005, China)

Abstract: When using level set method to track a free interface, the adaptive Cartesian grid method could automatically refine the grid of key area in the evolution. In the process of solving level set equation, the refined region and coarse region should be redecided after each time step. Due to hierarchical refinement, the nodes between different refinement levels have an incomplete neighbor node set, it becomes complex to obtain the approximate solution of the partial differential equation in Level set equation. An optimized adaptive Cartesian grid method was proposed. Firstly, the narrow-band method was introduced to select the refinement area automatically. Then, the red-green hybrid refinement strategy was used to eliminate the T node at the boundary of the narrowband. The Level set method with optimized adaptive Cartesian grid was applied to the computational model of thrombus growth inside vessel to improve the computation efficiency of the thrombosis simulation.

Key words: adaptive mesh refinement; level set method; narrow-band method; red-green refinement

0 引言

水平集方法(level set method, LSM)是一种用于

界面追踪和形状建模的数值技术。水平集方法的优点是可以在笛卡尔网格(Cartesian grid)上基于欧拉法(Eulerian approach)直接对演化中的曲线、曲面进行描述和数值计算,不必对曲线、曲面进行参数

收稿日期:2021-04-30

基金项目:国家自然科学基金项目(61802331,61802330);山东省自然科学基金项目(ZR2018BF008)。

第一作者:任宇佳(1994—),女,硕士研究生,主要从事医学仿真研究,E-mail:313011054@qq.com。

通信作者:马朝青(1987—),女,讲师,主要从事医学图像处理与虚拟现实研究,E-mail:chaoqingqing@ytu.edu.com。

化^[1-2]。水平集方法的另一个优点是可以方便地追踪物体的拓扑结构的改变。在水平集方法应用中,为了在控制算法时间复杂度的同时保证轮廓的分辨率和水平集方程的求解精度,通常在算法中引入自适应网格^[3-4]。

自适应网格方法(adaptive mesh refinement, AMR)是求解偏微分方程的工具。AMR的核心思想是:在迭代过程中,当某些区域变化剧烈(例如大变形、激波面、触间断面和滑移面等)时,该区域网格自动实现分级细化。因为AMR网格生成过程简单快速、自动化程度高、易于实现,在众多领域被广泛应用^[5-6]。

但是采用AMR的水平集方法在实现过程中仍存在难点,由于AMR是一种分级细化的不均匀网格,在不同细化级别之间的节点(T 节点)处会出现邻居节点集合不完整的情况,影响偏微分方程的近似求解。为了解决这一问题,近几年AMR相关研究中提出了最长边二分法^[7-8]、最新顶点二分法^[9-10]和red-green细化^[11-13]等方法。使用最长二分法细化后不能保证子网格的形状规律性。尽管在文献[14-15]中证明了二维三角网格经过最长边二分法后得到的子网格的形状规律性,但在三维中仍是一个不确定问题。后两种则可以保持初始网格中某一元素的所有子代的形状规律性,相较于red-green细化,最新顶点二分法属于复杂度较高的网格嵌套细化。事实上,red-green细化是对带有 T 节点的1-irregular网格常规细化的改进,其目的是实现细化网格的一致性。该细化规则首先在文献[11-12]提出,并在文献[16-18]中成功地扩展到高维的单纯网格。

水平集方程的求解是一个多次迭代求解的过程。引入AMR的水平集方法在结束每个时间步的计算后,需要重新确定细化和非细化网格,如果每次都进行网格的全局遍历会增加复杂度。针对这一问题,文献[19]提出了一种快速判断方法,即将遍历区域缩小到每个时间步计算所得曲面的临近网格,有效地减少了界面演化时的计算工作量。文献[20]开发了一种定位水平集,不需要在空间域中显式地找到界面位置的快速方法。该方法解决了如何将只在界面上给出的量扩展到界面邻域的重要问题。文献[21]将窄带法和快速行进近似法(the fast marching approximations)相结合,提出了一种用于研究生长过程中界面演化的算法,在重要的物理边界上建立了一个最高分辨率的可控有限宽度区域。

本文为了提高AMR水平集方法的迭代效率,在采用了局部水平集方法的基础上结合了窄带法^[21]对需要标记的区域进行选择,优化了细化区域的选择机制。同时,采用基于red-green细化的改进算法优化了 T 节点的近似计算。最后,为验证所提出算法的效果,将该算法应用于基于水平集的血栓生长模型中,优化了模型中水平集求解部分,并利用该模型进行了血栓形成过程的仿真实验。

1 局部水平集方法基本原理

水平集方法^[22]在解决轮廓演化问题时,把低维的计算问题上升到高一维,即把 N 维的轮廓利用 $N+1$ 维的水平集函数进行描述,获得一个闭合的超曲面,水平集函数值为0的零水平集即为被描述的轮廓。这样,轮廓的运动就可以通过闭合超曲面的演化方程得到,仅需要确定演化过程中零水平集的位置就可对给定轮廓进行追踪。

局部水平集的基本思想是在零水平集周围设定一条覆盖零水平集的带状区域,区域内的水平集符合 $|\phi| < K, 2K$ 即为窄带宽度。因为曲线在演化过程中每个时间步中的移动距离较小,所以不需要对水平集方程在整个定义域内进行计算,仅需要在窄带区域进行计算,从而降低计算的时间复杂度。假设一副二维图像的网格大小为 $N \times N$,节点数为 N^2 。用传统水平集方法对图像的轮廓进行追踪,那么将二维图像处理上升为三维函数的演化问题,每个时间步中水平集函数计算的时间复杂度为 $O(N^2)$ 。而局部水平集可以将单位时间内的计算量下降到 $O(kN)$, k 则表示在窄带内部像素点的数量。窄带的引入大大减少了曲线演化的时间,使计算效率得到有效提高。

在局部水平集方法^[23]中,局部水平集方程被定义为

$$\frac{\partial \phi}{\partial t} + C(\phi) \vec{F} \cdot \nabla \phi = 0, \quad (1)$$

其中: $\phi(x, t)$ 是局部水平集函数;截断函数 $C(\phi)$ 为

$$C(\phi) = \begin{cases} 1, & |\phi| > \Delta, \\ \frac{(|\phi| - \delta)^2 (2|\phi| + \delta - 3\Delta)}{(\delta - \Delta)^3}, & \Delta < |\phi| \leq \delta, \\ 0, & |\phi| < \Delta, \end{cases} \quad (2)$$

水平集的初始化仅在 $T_0 = \{x: |\phi(x)| < \delta\}$ 处进行。

2 Red-green 混合细化方法

本节将对现有的 red-green 及改进后的 red-green 分别进行细化描述。

2.1 传统的 red-green 细化

Red-green 细化最初由文献[11]提出,是一种基于三角网格的局部细化方法。它的实质是用两种不同的策略对单元格进行细化,从而减少 T 节点。随着三维网格的发展,这种 red-green 技术也适用于三维四面体网格^[16-18]。二维三角网格分割的 red 细化策略和 green 细化策略如图 1 所示。从图 1(a)可知二维的 red 细化(也称为正则细化)是通过连接边的三个中点将一个三角形分成四个子三角形(称为红色元素)。从图 1(b)可知 green 细化(也称为不规则求精)是连接顶点和与之相对的边的中点,通过 green 细化分割获得的两个子三角形称为绿色元素,两条分割边 P_3P_1 和 P_1P_4 称为绿色边(或不规则边)。元素 $T = T_1 \cup T_2 \cup T_3 \cup T_4$ 的 red 细化为四个全等子域 T_1, T_2, \dots, T_4 ; 元素 $T = T_1 \cup T_2$ 的 green 细化为两个子三角形 T_1 和 T_2 。

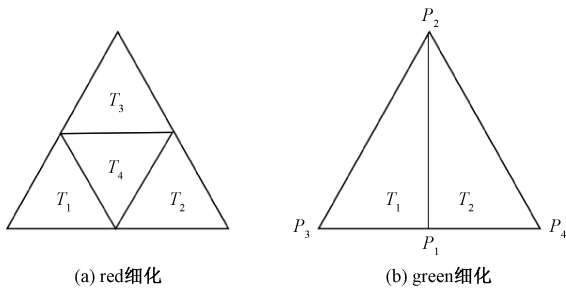


图 1 传统的 red-green 细化示意图

Figure 1 The diagram of traditional red-green refinement

Red 细化策略的优势在于,可以保证细化得到的子网格的形状与原始网格比较接近,但是会产生较多的 T 节点,在计算时增加难度;Green 细化策略可以减少 T 节点的产生,但是不能保证子网格的形状。因此,需要将两种策略结合使用。

2.2 无 T 节点四边形网格的局部细化

原始的四边形网格细化一直使用四分法作为 red 细化策略。在实际应用中,如果每个边上最多允许一个 T 节点存在,red 细化的对分完全可以满足四边形网格上的局部细化。但是在实际的计算中, T 节点数量的增加会导致计算难度的升高。在数值算法中,为了保持离散解的连续性,必须对 T 节点施加约束。为了尽量减少求解过程中对 T 节点特殊处理,Bank 等^[11]给出了 green 细化策略,它们直

接将 T 节点与四边形顶点相连。由于 T 节点出现在边的中间,在将 T 节点与不相邻顶点相连的情况下,即使初始网格为四边形,三角形仍将出现在细化网格中,生成的三角形和四边形的混合网格需要较为复杂的数据结构进行描述。因此,将 red 细化和 green 细化结合后可以形成新的混合细化,首先构造合适的 red 细化策略,保证细化网格与原始网格的一致性,然后辅助 green 细化策略进一步减少 T 节点的数量。同时,将混合细化方法扩展到四边形以消除 T 节点。

如图 2 所示,在二维混合细化算法中,原始网格首先通过三等分单元格 K 的边,将其分割为 9 个子单元格 K_1, K_2, \dots, K_9 进行 red 细化。内部节点 P_5, P_6, P_7, P_8 相对原始单元格顶点 P_1, P_2, P_3, P_4 的位置如下:

$$P_5 = (4P_1 + 2P_2 + P_3 + 2P_4)/9; \quad (3)$$

$$P_6 = (2P_1 + P_2 + 2P_3 + 4P_4)/9; \quad (4)$$

$$P_7 = (2P_1 + 4P_2 + 2P_3 + P_4)/9; \quad (5)$$

$$P_8 = (P_1 + 2P_2 + 4P_3 + 2P_4)/9, \quad (6)$$

其中 $P_i = (x_i, y_i), i = 1, 2, \dots, 8$ 为坐标。

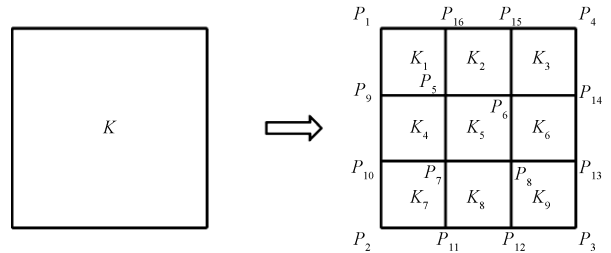


图 2 Red 细化

Figure 2 Red refinement

如图 3 所示,处于细化边缘的非细化单元格根据邻接网格的细化情况可以分为四种模式:三侧细化;相对两侧细化;相邻两侧细化和单侧细化,分别对应产生 6 个、4 个、4 个、2 个 T 节点(实心圆)。消除这些 T 节点的方案之一是该非细化网格进行同样的 red 细化(虚线),但会产生新的 T 节点,数量分别是 2 个、4 个、4 个、6 个(空心圆)。该方案仅对三侧细化模式具有减少 T 节点的效果。根据文献[11]提出的“ $K_c - 1$ Neighbor 规则”,后三种情况需要采用 green 细化(实线)来消除 T 节点。

2.3 水平集方法中的笛卡尔细化网格

当使用局部水平集方法对水平集函数进行演化时,只需要更新窄带内节点的水平集。水平集方程可采用 WENO 方法在空间上进行离散求近似解。WENO^[24]是由 ENO 格式发展而来的重构或插值过程。WENO 中的单侧后导数 $D_x^- \phi$ 和前导数 $D_x^+ \phi$ 的

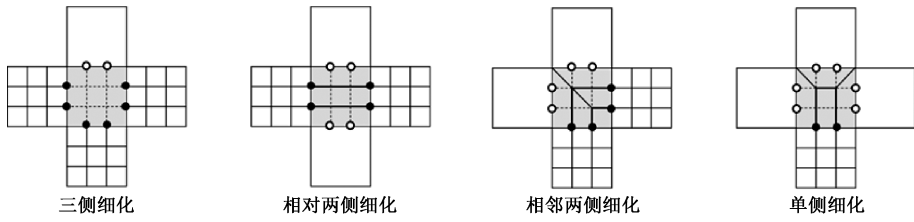


图 3 边缘单元格的不同细化情况及相应的 green 细化模式

Figure 3 The different adjacent refinement of edge cells and the corresponding green refinement mode

近似值为

$$D_x^- \phi = D_x^+ \phi = \omega_1 \phi_x^1 + \omega_2 \phi_x^2 + \omega_3 \phi_x^3, \quad (7)$$

该方法求解水平集方程 ϕ_i 时需要使用 $\{\phi_{i-3}, \phi_{i-2}, \phi_{i-1}, \phi_i, \phi_{i+1}, \phi_{i+2}, \phi_{i+3}\}$ 。根据求解方法的精度不同,求解过程中使用的相邻节点水平集数量也不同。对于非均匀细化笛卡尔网格,两级细化分界处的单元格的 T 节点处,由于相邻数据不连续,需要通过与其相连节点的 ϕ 值求得近似相邻数据,用于水平集方程求解。

为了消除 T 节点带来的附加计算,我们将改进后的 red-green 细化与窄带法相结合。首先对窄带区域内的节点进行 red 细化,并对靠近窄带边界的节点进行 green 细化,以消除 T 节点。

如图 4 所示,深色区域为 red 细化网格,浅色为 green 细化网格,判断原始网格的“分类”:

- 1) 与 red 细化相邻的原始单元格称为红色元素(图 4(a));
- 2) 与 green 细化相邻的原始单元格称为绿色元素(图 4(b));
- 3) 同时与 red 细化网格和 green 细化网格相邻的原始网格,也将其看作红元素(图 4(c))。

图 4 中 red 细化网格和红色元素为深色,green 细化网格和绿色元素为浅色。然后,忽略绿色元素,仅针对每一个红色元素进行如下操作:

- 1) 统计红色元素边上的 T 节点数 n ;
- 2) 将红色元素进行“预 red 细化”,统计细化后新增 T 节点数 m ;
- 3) 若 $m/n < 1$, 对该红色元素实施 red 细化, 否则实施相应的 green 细化。

3 基于水平集的血栓形成计算模型

血栓^[25]是血液循环中血管表面或心脏内壁上形成的凝块或沉积物。当血管内膜受损时,内皮下基质暴露,损伤附近的血小板被激活。随着血小板的粘附和聚集,形成以血小板为主要成分的初期血栓。根据血栓形成的原理,我们设计了初期血栓的

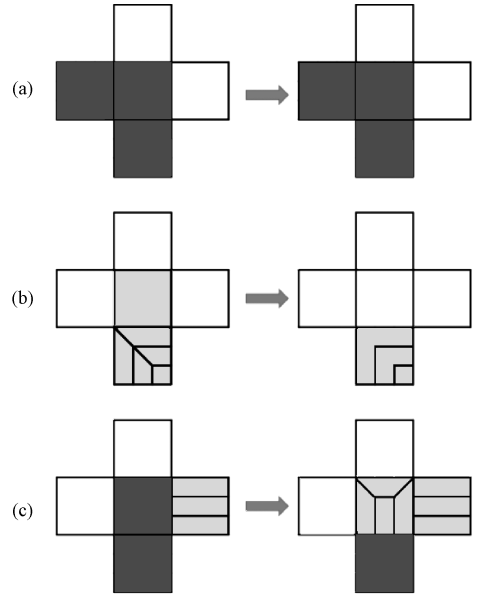


图 4 网格细化过程示例图

Figure 4 Example diagram of mesh refinement process

混合模型,如图 5 所示。血栓生长模型包括血管、破损的血管内膜、血小板、血浆和初期血栓,在设计成管状的血管中,血浆自左向右流动,带着血小板,经过破损的血管内膜这一区域的血小板被凝血因子激活,然后成为初期血栓的一部分。

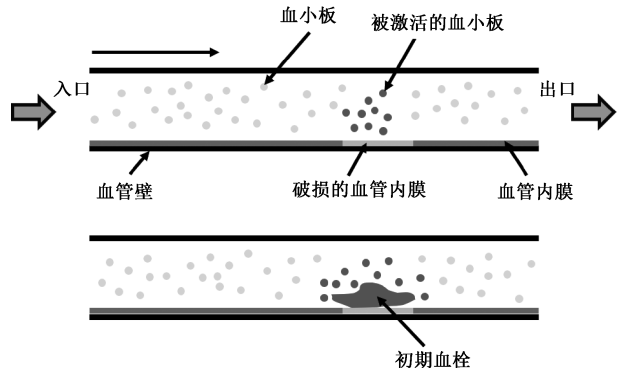


图 5 血栓生长模型示意图

Figure 5 Schematic diagram of thrombus growth model

血栓生长计算模型包括血浆、血小板和血栓三部分。如图 6 所示三个子模型的结构和关系:血浆子模型提供血流速度场并输送血小板;血小板模型是用来模拟血小板粘附和聚集的过程;血小板通过

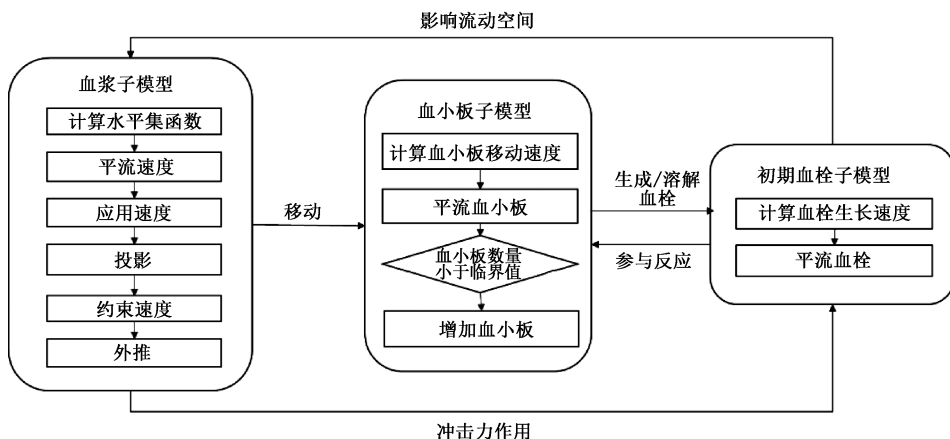


图6 血栓生长计算模型结构图

Figure 6 Structure diagram of thrombus growth calculation model

血栓子模型合并到血栓中。我们将初期血栓的表面定义为零水平集,并通过水平集方法追踪界面位置的变化,窄带的宽度值与血小板的直径成正比,确保新粘附上的血小板可以被包含在窄带内。

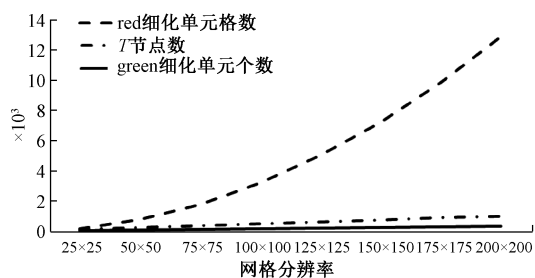
血浆子模型是用来模拟血液流动的子模型,提供了血液流速,并且负责运输血小板。我们假设血浆是不可压缩的,通过求解 Navier-Stokes 方程获得血流速度。血小板是初期血栓的主要组成部分。血小板模型用于模拟血小板的激活、粘附和聚集反应。分别计算血小板在血流冲击力、破损内膜粘附力和初期血栓的聚集力作用下的移动速度,并使血小板在该速度下移动。

模拟血栓生长过程的子模型中,我们将血栓的表面定义为由零水平集描述的自由移动界面,水平集 ϕ 由符号距离公式计算得出:血栓内部 $\phi < 0$; 血栓外部 $\phi > 0$; 血栓表面 $\phi = 0$ 。将破损表面定义为初期血栓的初始表面。在血栓生长过程中,发生粘附和聚集的血小板被定义为组成初期血栓的一部分,因此,当血小板模型中判定某一血小板发生粘附或聚集后,在该血小板所在位置定义界面移动速度,使血栓表面向外扩张产生扩张的表面通过水平集方法追踪界面位置的变化。在追踪过程中,定义窄带的宽度值不小于血小板的直径,以确保新捕获的血小板完全包含在窄带内。

4 实验结果与分析

为了验证改进的 red-green 细化策略在局部水平集中的有效性,我们建立了不同分辨率下的轮廓和窄带,并对细化数据进行统计并加以分析。在边长为 5 mm 的正方形网格中,建立半径为 15 mm 的

圆形轮廓,窄带宽度为 0.2 mm。网格的分辨率从 25×25 开始,以 25 为间隔递增至 200×200 。网格中 red 细化单元格数、red 细化后 T 节点数和追加 green 细化单元格数的变化曲线如图 7 所示。随着分辨率的提高,被细化的网格数增长幅度非常大。由于圆形轮廓较为简单, T 节点数目虽然增长,但幅度相对较小。消除这些 red 细化出现的 T 节点所追加的 green 细化单元格数目的增长幅度小于 T 节点的增长幅度,说明 green 细化在有效消除 T 节点时并不会带来更多的计算量。

图7 不同分辨率下细化单元格数及 T 节点数变化曲线Figure 7 Variation curve of refined cell number and T -node number under different resolutions

同时,针对静脉,我们使用第 3 节介绍的计算模型分别在无狭窄的直血管和有部分平滑狭窄的血管中对血栓的形成过程进行了仿真实验,各项血管参数分别为:血流速度 1 m/s;血液黏度 3.5×10^{-4} Pa·s;血管直径 4 mm;血管长度 12 mm;破损区域面积 $1.57 \text{ mm} \times 2 \text{ mm}$;狭窄程度 25%;狭窄长度 2 mm。并利用视觉化工具函数库 (Visualization Toolkit, VTK) 将仿真实验的结果进行可视化,方便观察血栓的形成过程,图 8 为两种血管的形态及破损血管内壁的位置的可视化图像。破损内壁初始为光滑表

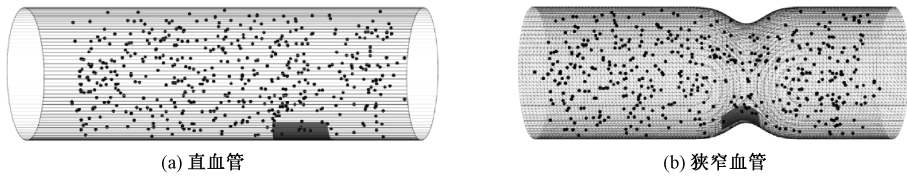


图8 仿真血管

Figure 8 Simulated blood vessels

面,随着血小板被破损血管壁捕获成为血栓,表面不断扩展,形成增多并叠加的颗粒状突起。

在无狭窄血管和有狭窄血管中将血流速度都分别设置为三种情况:2 mm/s,4 m/s 和 6 m/s(对应雷诺数 Re 分别为 0.01、0.02 和 0.03),分别进行 5 次时长为 50 ms 的血栓生长仿真实验,将平均结果绘制生长曲线,如图 9 所示。狭窄血管中的生长速度总体大于无狭窄血管,两种血管中的血栓生长均随

血流速度增加变快。将该结果与 Kamada 等的静脉血栓仿真^[26]结果对比,血栓生长速率以及该速率与雷诺数的关系是一致的。不同之处在于,Kamada 实验中考虑了血栓脱落,生长曲线存在中断,讨论脱落栓子阻塞血管的危险性,本文期望观察血栓在出现位置处的阻塞情况,未考虑血栓脱落,因此血栓在整个仿真时间内持续上升。

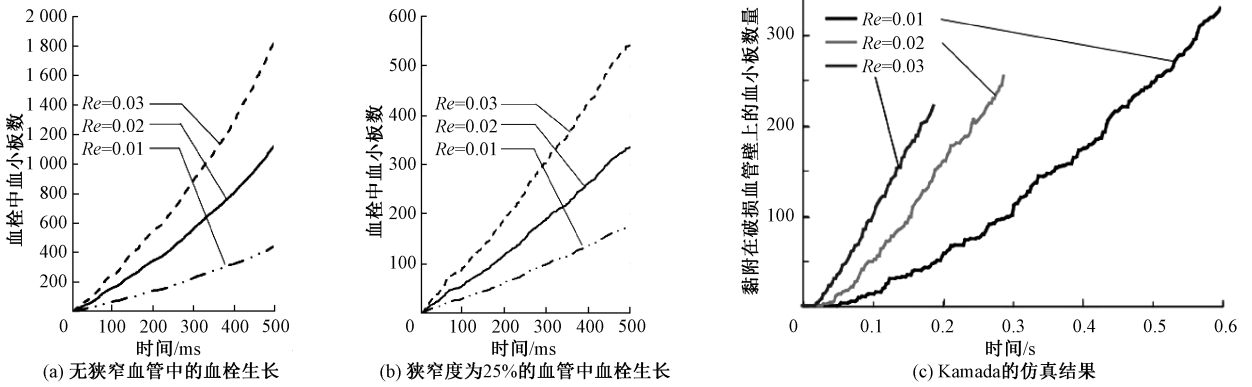


图9 不同血流速度下血栓生长曲线图

Figure 9 Thrombus growth curve under different blood flow velocity

根据仿真过程中网格数据的比较,如图 10 所示,无论是否存在狭窄,如果在目标区域中使用均匀细化,即在覆盖全部血栓的区域建立六面体网格,细

化网格的数量和增长率远高于基于窄带方法的局部细化。使用局部细化后,局部细化网格中仍存在约 10%的 T 节点,求解演化方程时需要分别计算。引

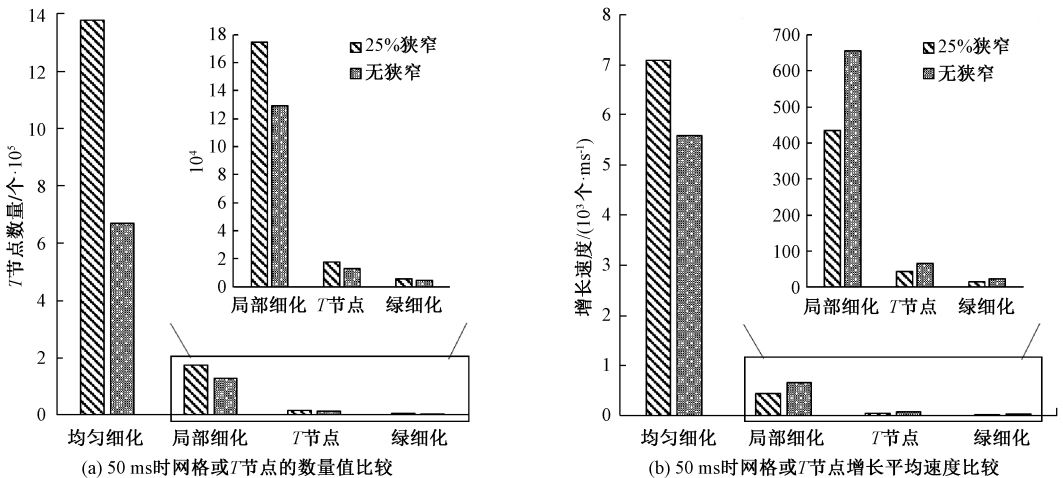


图10 仿真实验中随血栓生长的网格数据比较

Figure 10 Comparison of grid data with thrombus growth in simulation experiment

入 green 细化之后,尽管细化网格的总数增加了约 3%,但 green 细化网格的数量仍小于 T 节点的数量。同时,在统计增长率中,发现尽管在狭窄血管的模拟中细化网格的总数大于非狭窄血管的数目,但其增长率小于非狭窄血管的增长率。这表明由于血管几何结构的复杂性,细化网格的数量仅在狭窄初期才大,但是在血栓生长过程中细化网格的数量却缓慢增加。由于窄带方法的局部细化是基于零水平集界面,该模型中的零水平集是血栓表面,细化网格也是血栓表面的增长。因此,在狭窄的情况下,细化网格的缓慢生长速度表明由于血管狭窄所导致的血小板颗粒的粘附和聚集范围更加集中。

5 结论

本研究中,我们将改进后的 red-green 细化策略融合到局部水平集中,有效地提高了笛卡尔网格的运算效率。一方面,red-green 细化的使用消除了 T 节点,避免了由于找不到邻居节点导致的计算难度的增加,提高了计算效率;另一方面,窄带法的应用,使得每个时间步后,只需要对窄带内的点进行是否需要细化或粗化的判断,定义在零水平集上的速度函数也只需要扩展到窄带内的部分,降低水平集方法的计算量和存储空间。我们先是使用简单的圆形轮廓对其进行实验,又将该方法应用与血栓生长过程的建模和仿真中。实验结果表明改进后的算法在网格分辨率增加时能够更好地减少计算量,并能够有效地提高计算效率。该方法对于血栓生长过程的建模与仿真有利于进一步分析血栓形成的相关影响因素。本文研究中,在选择细化区域时仅考虑了与零水平集的距离,未充分考虑轮廓形态的复杂程度。我们下一阶段的工作将继续针对自适应细化网格进行研究,将能够代表轮廓几何特征的曲率等元素加入判断细化区域的特征参数中。

参考文献:

- [1] OSHER S, SETHIAN J A. Fronts propagating with curvature-dependent speed; Algorithms based on Hamilton-Jacobi formulations[J]. *Journal of computational physics*, 1988, 79(1): 12-49.
- [2] OSHER S, FEDKIWI R. *Level set methods and dynamic implicit surfaces*[M]. New York: Springer, 2003.
- [3] 吕文鹏,许峰. 基于自适应网格方法的免疫多目标进化算法[J]. *软件工程*, 2018, 21(6): 25-28.
LV W P, XU F. An immune multi-objective evolutionary algorithm based on the adaptive grid method[J]. *Software engineering*, 2018, 21(6): 25-28.
- [4] 殷亚军,李阳东,涂志新,等. 基于八叉树自适应网格技术的 Level Set 运动界面追踪方法[J]. *化工学报*, 2016, 67(11): 4732-4741.
YIN Y J, LI Y D, TU Z X, et al. Interface capturing method based on Level Set with octree adaptive mesh technology[J]. *CIESC journal*, 2016, 67(11): 4732-4741.
- [5] 段献葆,党妍,秦玲. Stokes 问题形状优化中自适应水平集方法的应用[J]. *上海大学学报(自然科学版)*, 2020, 26(4): 671-680.
DUAN X B, DANG Y, QIN L. Application of adaptive level set method in shape optimization for Stokes problem[J]. *Journal of Shanghai university (natural science edition)*, 2020, 26(4): 671-680.
- [6] 裴红星,刘金达,葛佳隆,等. 图像拼接技术综述[J]. *郑州大学学报(理学版)*, 2019, 51(4): 1-10, 29.
PEI H X, LIU J D, GE J L, et al. A review on image mosaicing techniques[J]. *Journal of Zhengzhou university (natural science edition)*, 2019, 51(4): 1-10, 29.
- [7] PLAZA A, CAREY G F. Local refinement of simplicial grids based on the skeleton[J]. *Applied numerical mathematics*, 2000, 32(2): 195-218.
- [8] HUAYTA O A, RIVARA M C. Study on the average size of the longest-edge propagation path for triangulations [C]// *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Malta: Science and Technology Publications, 2020: 368-375.
- [9] ARNOLD D N, MUKHERJEE A, POULY L. Locally adapted tetrahedral meshes using bisection [J]. *SIAM journal on scientific computing*, 2000, 22(2): 431-448.
- [10] BORKER R, HUANG D, GRIMBERG S, et al. Mesh adaptation framework for embedded boundary methods for computational fluid dynamics and fluid-structure interaction[J]. *International journal for numerical methods in fluids*, 2019, 90(8): 389-424.
- [11] BANK R E, SHERMAN A H, WEISER A. *Refinement algorithms and data structures for regular local mesh refinement*[M]. Amsterdam: North-Holland Publishing, 1983, 3-17.
- [12] BANK R E. *The efficient implementation of local mesh refinement algorithms*[M]. Babuska: Springer Vienna, 1983, 74-81.
- [13] CERVENKA J. Three-dimensional mesh generation for device and Process simulation[EB/OL]. [2020-11-28]. <http://www.iue.tuwien.ac.at/phd/cervenka/node14.htm>

- ml.
- [14] RIVARA M C, VEMERE M. Cost analysis of the longest-side (triangle bisection) refinement algorithm for triangulations[J]. *Engineering with computers*, 1996, 12(3/4): 224-234.
- [15] ROSENBERG I G, STENGER F. A lower bound on the angles of triangles constructed by bisecting the longest side[J]. *Mathematics of computation*, 1975, 29(130): 390.
- [16] BEY J. Tetrahedral grid refinement [J]. *Computing*, 1995, 55(4): 355-378.
- [17] BEY J. Simplicial grid refinement: on Freudenthal's algorithm and the optimal number of congruence classes [J]. *Numerische mathematik*, 2000, 85(1): 1-29.
- [18] MOLINO N, BRIDSON R, TERAN J, et al. Red green strategy for meshing highly deformable objects with tetrahedra[EB/OL]. (2003-01-01)[2021-01-26]. https://www.researchgate.net/publication/247367408_Red_green_strategy_for_meshing_highly_deformable_objects_with_tetrahedra.
- [19] ADALSTEINSSON D, SETHIAN J A. A fast level set method for propagating interfaces[J]. *Journal of computational physics*, 1995, 118(2): 269-277.
- [20] PENG D P, MERRIMAN B, OSHER S, et al. A PDE-based fast local level set method[J]. *Journal of computational physics*, 1999, 155(2): 410-438.
- [21] SOCHNIKOV V, EFRIMA S. Level set calculations of the evolution of boundaries on a dynamically adaptive grid [J]. *International journal for numerical methods in engineering*, 2003, 56(13): 1913-1929.
- [22] CHOPP D L. Computing minimal surfaces via level set curvature flow [J]. *Journal of computational physics*, 1993, 106(1): 77-91.
- [23] ZHANG W H, ZHANG Y T. Efficient local level set method without reinitialization and its appliance to topology optimization[J]. *Mathematical problems in engineering*, 2016, 1-14.
- [24] 白晓雅, 郑秋亚, 梁益华. 求解欧拉方程的嵌入 WENO 格式[J]. *郑州大学学报(理学版)*, 2020, 52(3): 98-103.
- BAI X Y, ZHENG Q Y, LIANG Y H. The embedded WENO scheme for solving the Euler equation[J]. *Journal of Zhengzhou university (natural science edition)*, 2020, 52(3): 98-103.
- [25] 李丽娟, 梁珊, 何亚州, 等. 血小板自噬在动脉血栓形成中的作用研究进展[J]. *重庆医学*, 2019, 48(3): 470-472.
- LI L J, LIANG S, HE Y Z, et al. The research progress of platelet autophagy in the formation of arterial thrombosis[J]. *Chongqing medicine*, 2019, 48(3): Chongqing Yixue, 2019, 48(3): 70-472.
- [26] KAMADA H, TSUBOTA K I, NAKAMURA M, et al. A three-dimensional particle simulation of the formation and collapse of a primary thrombus [J]. *International journal for numerical methods in biomedical engineering*, 2010, 26(3/4): 488-500.