УДК 519.65

КОМБИНИРОВАННОЕ ПРЕДСТАВЛЕНИЕ ГЕОМЕТРИИ ДЛЯ ВЫЧИСЛЕНИЯ РАССТОЯНИЯ В ГИДРОДИНАМИЧЕСКИХ РАСЧЁТАХ С МЕТОДОМ ВМОРОЖЕННЫХ ГРАНИЦ

© 2024 М. Ю. Хребтов[1,2a], Р. И. Мулляджанов[1,2b]

[1] *Институт теплофизики СО РАН,*
просп. Лаврентьева, 1, г. Новосибирск 630090, Россия,
[2] *Новосибирский государственный университет,*
ул. Пирогова 1, г. Новосибирск 630090, Россия

E-mails: [a]weexov@yandex.ru, [b]rustammul@gmail.com

Поступила в редакцию 27.11.2024 г.; после доработки 27.11.2024 г.;
принята к публикации 11.12.2024 г.

Предложен метод вычисления расстояния до трёхмерных геометрических моделей, путём представления их в виде результата булевых операций между элементарными объектами, для каждого из которых известно знаковое расстояние. Предложено две версии алгоритма, упрощённый, позволяющий быстрее рассчитать аппроксимацию расстояния (с точной нулевой изоповерхностью расстояния и разделением областей внутри и снаружи модели), и с дополнительным расчётом расстояния до контуров пересечения между элементами, позволяющий восстановить расстояние с большей точностью без существенных дополнительных затрат. Оба метода позволяют существенно сократить время вычисления по сравнению с расчётом расстояния до поверхностей путём представления их в виде связного набора треугольников. Также подход позволяет интерактивно изменять параметры и относительное положение частей геометрии, что даёт возможность проводить расчёты с подвижными границами. Подход протестирован в гидродинамических расчётах с межфазной границей и адаптивным многоуровневым сгущением сетки в открытом коде для моделирования сплошных сред — `Basilisk`.

Ключевые слова: расстояние до объекта, вычислительная геометрия, численное моделирование, сплошные среды, динамические сетки.

DOI: 10.33048/SIBJIM.2024.27.411

ВВЕДЕНИЕ

В численном моделировании многофазных сред набирает популярность подход, связанный с адаптивным сгущением расчётной сетки [1–3]. Данный подход позволяет отслеживать положение межфазной поверхности и поддерживать необходимую степень разрешения при её эволюции. При этом, чтобы обеспечить возможность динамического сгущения, структура сетки должна быть достаточно простой. Наиболее удобным здесь представляется использование структурированной декартовой сетки с многоуровневым её представлением в виде дерева. Одна ячейка сетки может быть разбита на целое число равных ячеек (как правило, с соотношением размеров новых ячеек к исходным, равным $1/2$) [2, 4, 5]. Таким образом, можно построить иерархическое представление сетки, при этом глубина разбиения может варьироваться динамически в широких пределах в зависимости от выполнения заданного локального критерия (погрешности аппроксимации градиентов, расстояния до некой поверхности, величины скалярного поля и т. д. [3, 6]).

В данном подходе сложностью является учёт произвольной геометрии твёрдых тел, ограничивающих данное течение. Как правило, используется метод вмороженных или встроенных границ, когда поверхность границы может проходить внутри расчётной ячейки. Если известно положение этой границы, то далее можно построить локальные численные схемы для задания граничных условий с необходимым порядком аппроксимации с помощью построения шаблона с использованием координат центров и узлов ячеек, расположенных вблизи данной границы [7].

При использовании динамического сгущения эти схемы необходимо перестраивать и увеличивать либо уменьшать точность разрешения твёрдых границ в зависимости от локальных условий расчёта (наличия межфазных границ, зон отрыва и т. д.). Также перестройка сетки в процессе расчёта необходима при использовании подвижных границ от движущихся элементов геометрии.

Помимо задачи перестроения сетки, знать расстояние до поверхности необходимо для замыкания различных моделей турбулентности, которые явно учитывают пристенную область течения для переключения параметризации турбулентной диффузии или используют различные пристенные демпфирующие факторы. Такое разделение используется в наиболее распространённой RANS-модели $k - \omega$ SST [8], применяющейся для инженерных расчётов.

Таким образом, существует необходимость разработки алгоритма для быстрого расчёта расстояния до твёрдых границ, который бы можно было интерактивно использовать для проведения моделирования потоков с использованием динамического сгущения сетки.

Знание расстояния до границы позволяет находить нормаль к поверхности вблизи границы по градиенту расстояния и восстанавливать форму поверхности с учётом этих нормалей (и знака расстояния, определяющего с какой стороны от поверхности находится точка) в ближайших узлах сетки [6, 9, 10].

Вычисление расстояния во всех точках расчётной сетки для нетривиальной геометрии требует значительных вычислительных затрат. В случае, если перестройка сетки происходит часто (раз в несколько итераций), это требует перерасчёта расстояний каждый раз при сгущении или огрублении сетки.

Существует ряд подходов [6, 11] к быстрому восстановлению значений расстояния во время расчёта, однако они либо используют алгоритмы представления геометрии на фиксированной сетке (с высоким разрешением), что требует дополнительных затрат памяти, либо имеют низкую точность восстановления, отражающуюся в ошибках положения границы твёрдого тела при динамическом сгущении сетки. Кроме того, эти подходы не будут работать при перемещении или деформировании твёрдых границ, если части геометрии будут смещаться относительно друг друга.

С другой стороны, есть подходы [12], которые связаны с геометрическими методами расчёта расстояний на основе исходной 3D-модели, когда модель представляется в виде аппроксимации поверхности (например, связанным набором треугольников, многогранником), а расстояние до поверхности вычисляется как расстояние до ближайшего элемента, аппроксимирующего поверхность многогранника (вершины, рёбра или грани). Такое вычисление требует прохождения по всем треугольникам поверхности (в общем случае) с количеством операций $O(n \cdot m)$, где n — это число элементов многогранника (вершина, ребро, грань), и m — это число ячеек, в которых рассчитываются значения расстояния. При детальном представлении модели и n, и m могут быть большими, что создаёт большую вычислительную нагрузку на перерасчёт расстояний и делает его непрактичным. С этой точки зрения, требуется разработать методы расчёта расстояний, которые позволяют проводить его с большей скоростью.

В данной статье предлагается представлять целевую 3D-модель в виде булевой суперпозиции элементарных объектов, расстояния до которых могут быть рассчитаны с небольшими вычислительными затратами. В качестве примеров таких элементарных объектов рассматриваются поверхности вращения и цилиндрические "капсулы". Многие 3D-модели в инженерных приложениях можно аппроксимировать комбинацией таких объектов. При этом, предлагается рассчитывать итоговое расстояние до объекта, представленного в виде набора булевых операций над такими элементарными объектами, через известные значения расстояний от каждого из элементов данных операций. В качестве практического примера приводится расчёт истечения жидкой струи из центробежной форсунки, сделанный с помощью открытого программного пакета *Basilisk* [1].

1. АЛГОРИТМ ЗАДАНИЯ ПОВЕРХНОСТЕЙ И ОПРЕДЕЛЕНИЯ ЗНАКОВОГО РАССТОЯНИЯ

Определим знаковое расстояние $D(S,\vec{r})$ в точке $\vec{r} = \{x,y,z\}$ до замкнутой несамопересекающейся поверхности S как расстояние (заданное для двух точек $\vec{p_1}$ и $\vec{p_2}$ через $\sqrt{(\vec{p_1}-\vec{p_2})\cdot(\vec{p_1}-\vec{p_2})}$) от $\vec{r}$ до ближайшей точки поверхности S, умноженное на -1, если точка находится внутри объёма, ограниченного S.

Предлагается рассмотреть в качестве примера базовых геометрических элементов два типа объектов. Первый тип объектов представляет собой круглый цилиндр с высотой L и радиусом R, заканчивающийся полусферическими закруглениями такого же радиуса, гладко сопряжёнными с поверхностью цилиндра (капсула) (рис. 1(a)).

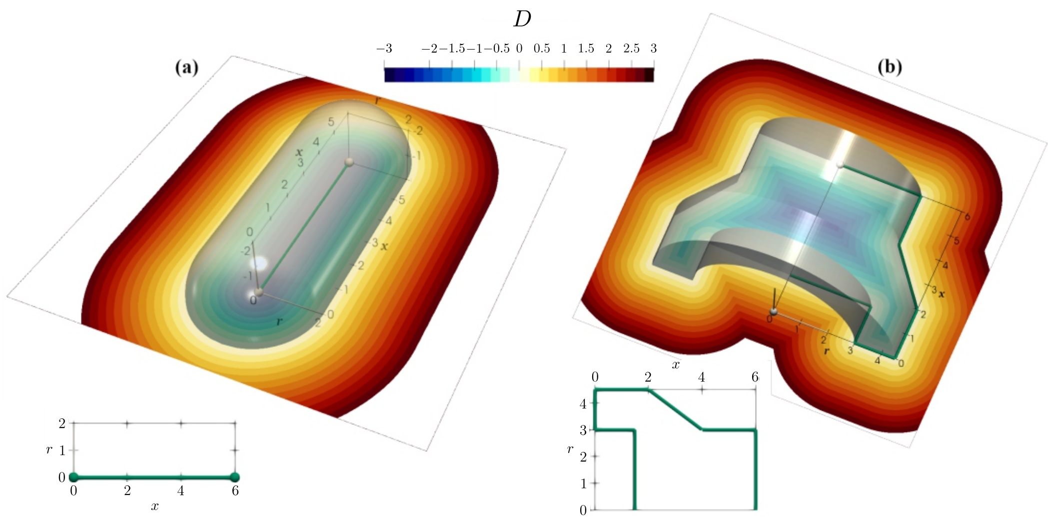

Рис. 1. Базовые геометрические элементы и расстояния до них; (a) — капсула; (b) — тело вращения

Для каждой капсулы, в памяти хранится её локальный базис из трёх ортонормальных векторов $\vec{a}$, $\vec{b}$, $\vec{c}$, где вектор $\vec{a}$ направлен вдоль оси цилиндра, а два других произвольно перпендикулярны этой оси и друг другу. Также в памяти хранятся координаты основания цилиндра $\vec{a_0}$ (центр основания цилиндрической части капсулы), его радиус R и высота L цилиндрического участка. Такое представление позволяет быстро находить знаковое расстояние $D(S,\vec{r_0})$ до поверхности капсулы S от произвольной точки $\vec{r_0}$ по следующему алгоритму.

Алгоритм 1. Вычисление расстояния до капсулы.

if $(0 \geqslant (\vec{r_0}-\vec{a_0})\cdot\vec{a} \geqslant L)$ **then**
  $D(S,\vec{r_0}) = \sqrt{((\vec{r_0}-\vec{a_0})\cdot\vec{b})^2 + ((\vec{r_0}-\vec{a_0})\cdot\vec{c})^2} - R$;
else if $((\vec{r_0}-\vec{a_0})\cdot\vec{a} < 0)$ **then**
  $D(S,\vec{r_0}) = \sqrt{(\vec{r_0}-\vec{a_0})^2} - R$;
else
  $D(S,\vec{r_0}) = \sqrt{(\vec{r_0}-(\vec{a_0}+L\cdot\vec{a}))\cdot(\vec{r_0}-(\vec{a_0}+L\cdot\vec{a}))} - R$;
endif

где оператор «·» для векторов означает скалярное произведение. Данный геометрический элемент (капсулу) можно использовать для создания связанных наборов труб и отверстий путём сопряжения капсул через сферические участки, что является распространённым элементом в инженерном проектировании.

Второй геометрический элемент — это тело вращения с образующей в виде односвязного набора отрезков прямых без самопересечений (рис. 1(b)). Для данного элемента также хранится локальный ортонормированный базис из векторов $(\vec{a}, \vec{b}, \vec{c})$ и координаты точки $\vec{a_0}$, находящейся на оси симметрии, относительно которой записываются координаты отрезков. Точки контура хранятся в виде локальных $2D$-координат $\{x_i, r_i\}$ относительно локального базиса. Для вычисления расстояния от поверхности вращения до произвольной точки $\vec{r_0}$, сначала её координаты переводятся в систему координат контура (с учётом осевой симметрии). Затем, в двумерном пространстве ищется расстояние до ближайшего элемента контура (рёбра либо вершины многоугольника). Данный процесс может быть ускорен, если для каждой стороны многоугольника (отрезка $\{x_i - x_{i-1}, r_i - r_{i-1}\}$) хранить единичную нормаль (в 2D-пространстве) $\vec{n_i}$, направленную наружу контура. Тогда алгоритм нахождения расстояния сводится к следующему (для ускорения будем минимизировать квадрат расстояния, обозначенный как $D2$).

Алгоритм 2. Вычисление расстояния до осесимметричного контура.

$x = (\vec{r_0} - \vec{a_0}) \cdot \vec{a};$
$r = \sqrt{((\vec{r_0} - \vec{a_0}) \cdot \vec{b})^2 + ((\vec{r_0} - \vec{a_0}) \cdot \vec{c})^2}$
$D2 = (x - x_0)^2 + (r - r_0)^2\,;$
for (i=1 **to** n)
  if $(0 < (x - x_{i-1}) \cdot (x_i - x_{i-1}) + (r - r_{i-1}) \cdot (r_i - r_{i-1}) < 1)$ **then**
    $D2 = \min(D2, ((x - x_{i-1}) \cdot (n_i)_x + (r - r_{i-1}) \cdot (n_i)_r)^2)\,;$
  else
  $D2 = \min(D2, (x - x_i)^2 + (r - r_i)^2)\,;$
  endif
endfor
$D(S, \vec{r_0}) = \sqrt{D2}\,;$

Для вычисления знака расстояния (определения, внутри или снаружи многоугольника находится точка) можно использовать различные подходы [13]. В нашем случае используется метод вычисления знака расстояния через обобщённую нормаль [14]. Знак сводится к вычислению знака скалярного произведения до нормали к ближайшему элементу контура, направленной наружу. В случае, если ближайшим элементом является отрезок, то нормаль будет определяться однозначно (можно использовать ориентацию контура по часовой стрелке, умножая векторно каждый отрезок на вектор $\vec{c}$ из локального базиса). Если ближайшим элементом является вершина, то нормалью, определяющей знак расстояния, будет являться сумма нормалей от прилежащих к вершине отрезков.

1.1. Булевы операции над введёнными геометрическими элементами

Использование двух описанных типов объектов позволяет с помощью их булевых комбинаций описать достаточно обширный набор геометрий (форсунки, элементы фронтовых устройств и камер сгорания), используемых в практических приложениях CFD-моделирования. Рассмотрим реализацию вычисления расстояния от булевых комбинаций (объединения и вычитания) рассмотренных геометрических элементов. Известны аппроксимации расстояния до булевой комбинации объектов с известными знаковыми расстояниями [12]. Так, расстояния до булева объединения объектов можно аппроксимировать через минимум от расстояний до двух объектов A и B:

$$D(A \cup B, \vec{r}) \approx \min(D(A, \vec{r}), D(B, \vec{r})), \tag{1}$$

где $D(f, \vec{r})$ — знаковое расстояние от объекта f в точке $\vec{r}$.

Снаружи обоих объектов, входящих в объединение, эта аппроксимация точно соответствует расстоянию до их объединения. Действительно, снаружи обоих объектов знаки расстояний до обоих объектов положительны, при этом элементы поверхности, находящиеся внутри объединения двух объектов (там, где знак расстояния до хотя бы одного из объектов отрицателен) находятся на ненулевом расстоянии от поверхности объединения, и в итоге расстояние до них будет большим, чем расстояние до ближайшей поверхности булева объединения объектов.

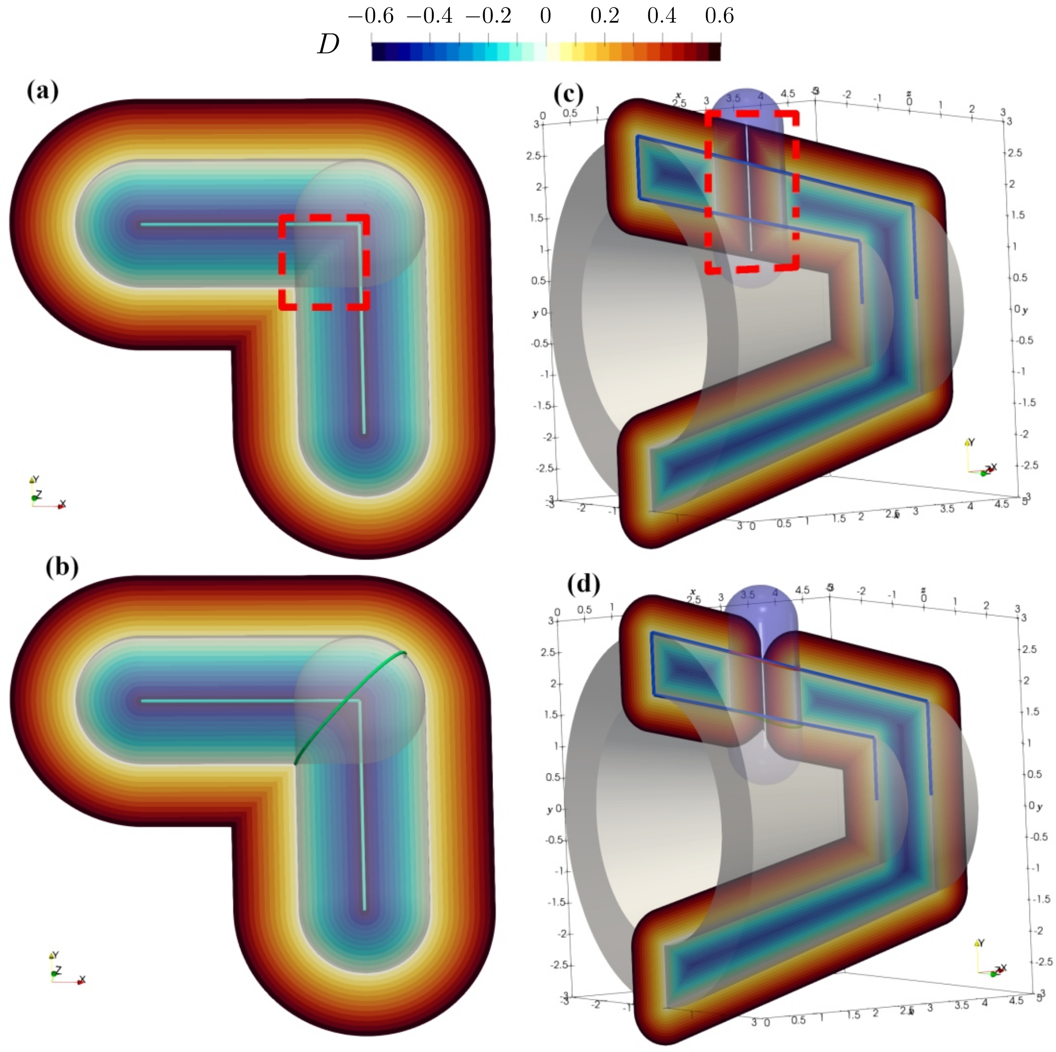

Рис. 2. Распределения расстояний для: булева объединения двух капсул (a), (b); булева вычитания капсулы из осесимметричного контура (c), (d). Области с ошибкой вычисления по формулам (1), (2) показаны красным (a), (c). Уточнённые результаты с помощью учёта расстояний до контуров пересечения поверхностей (b), (d)

Во внутренней области булева объединения (там, где знак хотя бы одного из расстояний до элементов отрицателен) выражение (1) не будет точно представлять расстояние. Вблизи пересечения поверхностей A и B могут быть области, где минимальным будет расстояние до контура пересечения поверхностей A и B, не совпадающее с расстояниями ни до первого, ни до второго объектов. Ошибка в вычислении расстояния с помощью (1) продемонстрирована на рис. 2(a), 2(c) в сравнении с правильными расстояниями, приведёнными на рис. 2(b), 2(d).

Прежде, чем перейти к уточнённому алгоритму определения расстояния, заметим, что выражение (1) правильно восстанавливает форму изоповерхности нулевого расстояния до объединения объектов, а также позволяет верно определить знак расстояния (внутри или снаружи находится точка). Функция ведёт себя монотонно при переходе через поверхность нулевого расстояния и поэтому может быть использована для задания твёрдых границ в гидродинамических расчётах, но не подходит для случаев, когда нужно восстановить значение величины расстояния до поверхности точно.

Для того, чтобы уточнить значение расстояния от произвольной точки $\vec{r}$ до поверхности булева объединения объектов A и B, необходимо в явном виде учесть расстояние до контуров пересечения поверхностей этих объектов и исключить влияние участков поверхностей этих объектов, которые находятся внутри контуров пересечения. Запишем алгоритм расчёта расстояния до булева объединения объектов с учётом контура пересечения их поверхностей. Для простоты рассмотрим случай объединения двух элементарных объектов, имеющих один односвязный контур пересечения геометрий.

Алгоритм 3. Схема определения расстояния до объединения элементарных объектов $D(A \cup B, \vec{r})$:

---

1. Определить контур пересечения поверхностей.
2. Для каждого из объектов определить область, где минимальным будет расстояние до контура.
3. Вне этой области расстояние вычисляется по формуле (1).
4. Внутри этой области расстояние вычисляется как расстояние до ближайшей точки контура пересечений поверхностей A и B.
5. Для определения знака расстояния использовать знак выражения (1).

То есть, необходимо исключить из рассмотрения части поверхностей объектов, находящиеся внутри контуров пересечения поверхностей объектов.

Рассмотрим, как это сделать для объекта типа «капсула». При наличии контура пересечения, опоясывающего капсулу, необходимо определить, с какой стороны от контура находится точка на капсуле, ближайшая к точке, в которой вычисляется расстояние. Определим точку $\vec{r}_{nearest}$, как лежащую на поверхности капсулы A и ближайшую к точке $\vec{r}$. Точка $\vec{r}_{nearest}$ может быть легко найдена следующим образом. Если ближайшей является цилиндрическая часть капсулы, то нужно провести прямую перпендикулярно оси капсулы до пересечения с поверхностью цилиндра. Если ближайшей является одна из полусфер, то нужно провести прямую к центру основания полусферы, до пересечения с её поверхностью. Данную процедуру можно описать с помощью следующего алгоритма.

Алгоритм 4. Определение ближайшей точки на поверхности капсулы.

$r_a = (\vec{r} - \vec{a_0}) \cdot \vec{a}$
$r_b = (\vec{r} - \vec{a_0}) \cdot \vec{b}$
$r_c = (\vec{r} - \vec{a_0}) \cdot \vec{c}$
if $(0 > r_a > L)$ **then**

$k = \sqrt{R^2/(r_b^2 + r_c^2))}$;
$\vec{r}_{nearest} = \vec{r} - (1-k)\cdot(\vec{b}\cdot r_b + \vec{c}\cdot r_c)$;
else if $(r_a < 0)$ **then**
$k = \sqrt{R^2/((r_a)^2 + (r_b)^2 + (r_c)^2)}$;
$\vec{r}_{nearest} = \vec{r} - (1-k)\cdot(\vec{a}\cdot r_a + \vec{b}\cdot r_b + \vec{c}\cdot r_c)$;
else
$k = \sqrt{R^2/((r_a - L)^2 + (r_b)^2 + (r_c)^2)}$;
$\vec{r}_{nearest} = \vec{r} - (1-k)\cdot(\vec{a}\cdot(r_a - L) + \vec{b}\cdot r_b + \vec{c}\cdot r_c)$;
endif

Аналогичную операцию можно провести и для второго типа рассматриваемых базовых объектов — тел вращения, с образующей, состоящей из связанных отрезков прямых. Для данного типа объектов также легко можно найти точку $\vec{r}_{nearest}$, ближайшую к произвольной точке $\vec{r}$. Для этого нужно найти, какой из отрезков контура является ближайшим к данной точке в 2D-пространстве локального базиса объекта (в силу осевой симметрии поверхности объекта). Нахождение ближайшего отрезка происходит при вычислении расстояния $D(A, \vec{r}))$ и известно заранее при вычислении значения выражения (1).

После нахождения ближайшего отрезка нужно спроектировать исходную точку $\vec{r}$ в направлении, перпендикулярном к данному отрезку в цилиндрических координатах, до пересечения с изоповерхностью $D(A, \vec{r}_{nearest}) = 0$. Если ближайшим является не отрезок, а вершина многоугольника, то её координаты и будут координатой ближайшей точки в 2D-пространстве.

После того, как определены координаты ближайшей точки в локальном 2D-пространстве, её нужно спроектировать обратно в исходное 3D-пространство, что легко сделать, используя представление о цилиндрической симметрии распределения расстояния $D(A, \vec{r})$ для тела вращения. Алгоритм вычисления $\vec{r}_{nearest}$ можно формально записать следующим образом (считаем, что индекс $imin$ — это индекс отрезка $\{x_{imin} - x_{imin-1}, r_{imin} - r_{imin-1}\}$, ближайшего к точке $\vec{r}$, при этом $\vec{n}_{imin}$ — это вектор нормали к данному отрезку в 2D-пространстве):

Алгоритм 5. Определение ближайшей точки на поверхности тела вращения.

$p_r = (r - r_{imin-1})\cdot(n_{imin})_r$;
$p_x = (x - x_{imin-1})\cdot(n_{imin})_x$;
$\vec{r}_{loc} = (\vec{r} - \vec{a}_0)$;
$p_a = \vec{r}_{loc}\cdot\vec{a}$;
$\vec{r}_{nearest} = \vec{a}_0 - p_r\cdot(\vec{r}_{loc} - \vec{a}\cdot p_a) - p_x\cdot(\vec{a}\cdot p_a)$;

Разделение областей пространства, ближайшими к которым являются части поверхности по одну и другую сторону контура пересечения для капсулы, приведены на рис. 3(a). Как видно, пространство разделяется на две части, разделённые поверхностью, образованной прямыми, проведёнными в сторону нормали к поверхности капсулы из всех точек контура пересечения капсулы с другим объектом. Аналогичное разделение пространства для тела вращения с вырезанным на его поверхности контуром пересечения с другим объектом показано на рис. 3(b). Видно, что в этом случае поверхность разделения напоминает конус, сходящийся к оси симметрии тела вращения. Точки, находящиеся внутри конуса, будут ближайшими к области поверхности вращения, ограниченной контуром пересечения.

После нахождения ближайшей к $\vec{r}$ точки на поверхности A, необходимо определить, находится ли эта точка $\vec{r}_{nearest}$ внутри или снаружи другого тела B из булевой операции. Это можно сделать, используя известные значения расстояния до поверхности второго тела $D(B, \vec{r}_{nearest})$. Если $D(B, \vec{r}_{nearest}) < 0$, то надо исключить вклад тела A в величину расстояния до точки $\vec{r}$ и заменить его вкладом контура пересечения поверхностей A и B. Таким образом, удаётся исключить из рассмотрения участок поверхности тела A находящийся внутри объединения $A \cup B$. Аналогичным образом необходимо исключить и участок поверхности тела B, находящийся внутри $A \cup B$.

Пример распределения расстояний от капсулы, пересечённой поверхностью тела вращения, приведён на рис. 3(c). Показано, как выглядит распределение расстояния, с исключением части поверхности, ограниченной контуром пересечения с другим объектом. Аналогичное распределение для тела вращения, с вырезанным участком поверхности внутри контура пересечения, показано на рис. 3(d).

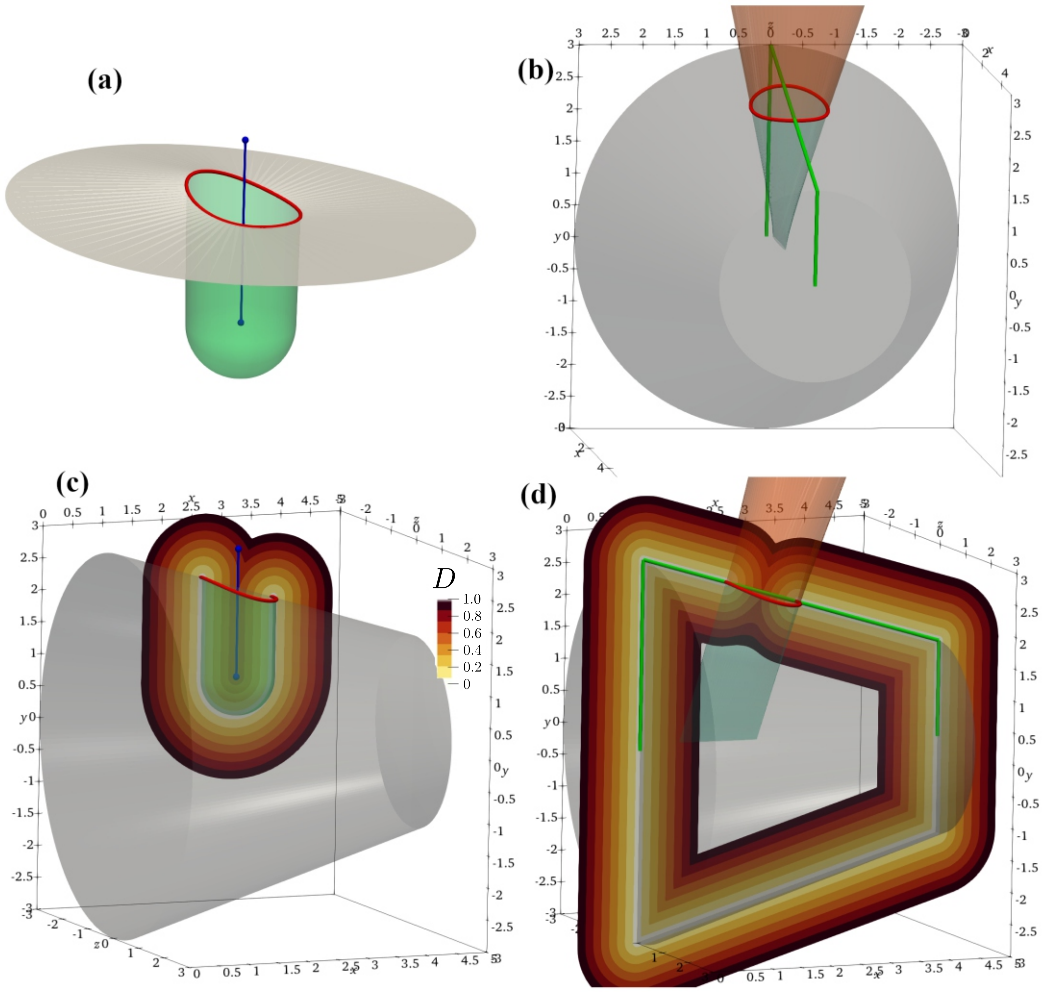

Рис. 3. Разделение областей пространства, ближайшими к которым являются части поверхности по одну и другую сторону контура пересечения: (a) — для объекта типа капсула, (b) — для объекта типа тело вращения. Распределения расстояний от объектов с исключением части поверхности внутри контура пересечения: (c) — для капсулы, (d) — для тела вращения

Для булева вычитания объектов в качестве аппроксимации расстояния можно брать следующее выражение:

$$D(A - B, \vec{r}) \approx \max(D(A, \vec{r}), -D(B, \vec{r})). \tag{2}$$

В данном случае расстояние определяется точно в области разницы A и B (внутри объекта A и снаружи объекта B). Расстояние, от булевой разности $A - B$, рассчитанное по формуле (2), будет иметь погрешность в области снаружи от A и внутри B. Но, как и формула (1), формула (2) позволяет правильно определять нулевую изоповерхность расстояния, а также отделять внутреннюю часть булевой комбинации объектов от внешней.

Для уточнения значений расстояния до булевой разности $D(A - B, \vec{r})$, полученных по формуле (2) необходимо, как и ранее, вычислить контуры пересечения поверхностей A и B, после чего, исключить из рассмотрения области поверхностей, находящиеся внутри контуров пересечения. Это можно сделать путём нахождения ближайших точек к $\vec{r}$ на поверхностях A и B, аналогично тому, как это было описано для случая булева объединения выше. После нахождения ближайших точек $\vec{r}_{nearest}$ нужно определить, лежат ли они вне или внутри второго объекта из булевой разности, и в случае, если они лежат внутри второго объекта, то расстояние, найденное по формуле (2) нужно заменить на расстояние до ближайшей точки контура пересечения поверхностей.

Таким образом, был описан алгоритм вычисления расстояния до двух базовых булевых операций (объединение и разность) от базовых типов геометрических объектов. Набор базовых типов геометрических объектов может быть расширен при необходимости, при этом принцип вычисления расстояния останется тем же.

Остаётся сказать о вычислении расстояния до контура пересечения поверхностей элементарных объектов. Можно описывать контур пересечения различными способами с применением аналитических или численных методов. Поскольку не существует прямого аналитического способа вычисления минимума расстояния до контура пересечения в общем виде, то в данной работе предложено использовать простую аппроксимацию контура через набор связанных отрезков. Отрезки предполагаются имеющими равную длину, при уменьшении которой аппроксимация контура стремится к истинной форме контура. При этом, определение расстояния до контура сводится к определению расстояния до ближайшего элемента контура (отрезка или вершины) путём перебора по всем элементам контура.

Ошибка в определении расстояния до булевой комбинации объектов по предложенному алгоритму будет зависеть от дискретизации контуров пересечения поверхностей. Чем точнее будет дискретизация (больше элементов контура), тем точнее будет восстанавливаться расстояние. При этом, относительная погрешность вычисления расстояния будет уменьшаться с удалением от контура. Однако увеличение количества отрезков в дискретизации контура будет приводить к замедлению вычисления расстояний, поэтому в данном вопросе необходим компромисс. Рациональным является использование отрезков для дискретизации контура пересечения близких к размеру расчётных ячеек гидродинамического расчёта, находящихся вблизи контура пересечения.

2. ТЕСТИРОВАНИЕ АЛГОРИТМА В РАСЧЁТЕ ИСТЕЧЕНИЯ СТРУИ ЖИДКОСТИ ИЗ ЦЕНТРОБЕЖНОЙ ФОРСУНКИ

Для тестирования алгоритма вычисления расстояния до геометрии был проведён расчёт течения внутри центробежной форсунки для керосина с расходом 0.9 л/мин (рис. 4).

Форсунка представляет собой цилиндр диаметром 7 мм и высотой 6.2 мм с вырезанной внутренней полостью диаметром 3.2 мм, в которую ведут два тангенциальных входных канала диаметрами по 1.2 мм. Внутренняя полость заканчивается коническим сужением и одним выходным каналом с диаметром 0.8 мм (рис. 4(a)–4(c)). Расход керосина 0.9 л/мин соответствует выходной скорости струи ≈ 30 м/с. Струя керосина с плотностью 810 кг/м^3, кинематической вязкостью $2.71 \cdot 10^{-6}$ м^2/с и коэффициентом поверхностного натяжения 0.023 Н/м истекала в воздушную среду с плотностью 1.2 кг/м^3 и кинематической вязкостью $1.6 \cdot 10^{-5}$ м^2/с.

Задача распыла струи керосина из форсунки рассчитывалась в приближении несжимаемой жидкости с помощью открытого расчётного кода *Basilisk* [1]. Расчёт проводился методом конечных объёмов, детали расчётной схемы и решаемые уравнения описаны в [15].

Расчётная область представляла собой куб со стороной 15 мм. Расчёт проводился с динамическим сгущением ячеек сетки (каждая ячейка могла быть разбита на 8 равных ячеек

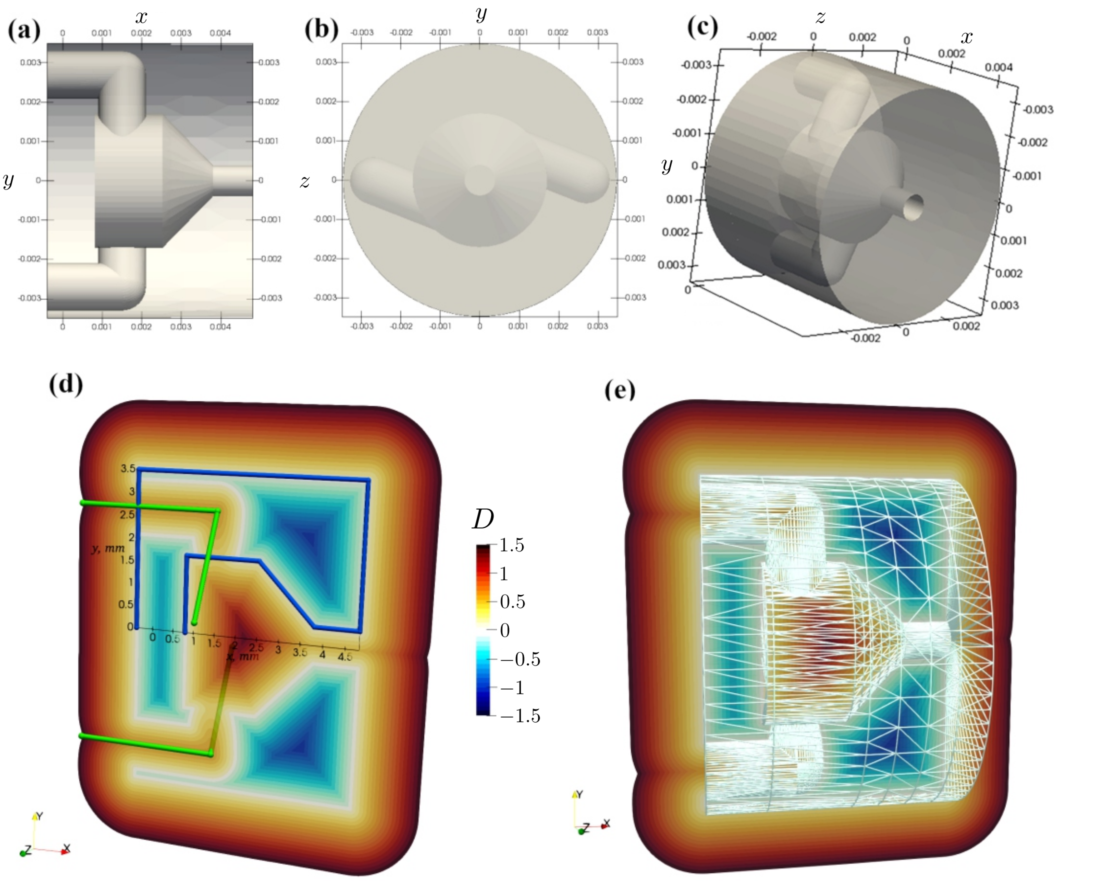

Рис. 4. (a)–(c) — геометрия форсунки; (d) — представление форсунки в виде комбинации базовых элементов между которыми произведено булево вычитание (зелёным показаны центральные оси капсул, синим — осесимметричный контур), а также распределение расстояния в продольном сечении; (e) — представление поверхности форсунки в виде треугольников и расстояние, полученное путём перебора по треугольникам. D — расстояние, мм; x, y, z измеряются в м

с вдвое меньшим линейным масштабом). Разбиение могло проходить вплоть до минимального размера ячейки в 7 мкм. Максимальное число узлов во время расчёта составляло 30 млн. ячеек. В качестве критериев для сгущения использовались помимо значений расстояния до поверхности форсунки также модуль градиента поля объёмной доли жидкой фазы и модуль градиента скорости. Расчёт проводился на 128 процессорных ядрах кластера «Каскад» ИТ СО РАН. На каждое ядро приходилось ≈ 234 тыс. ячеек.

Геометрия форсунки была представлена тремя способами: (i) с помощью булевой комбинации, т. е. вычитания четырёх «капсул» из одного осесимметричного контура (рис. 4(d)) со значениями расстояния, вычисленными по формуле (1) без уточнений; (ii) с уточнением по описанному выше алгоритму, т. е. с явным учётом контуров пересечения поверхностей элементарных объектов (для аппроксимации контуров использовалось кусочно-линейное представление контуров с разрешением в 20 отрезков на контур); (iii) и с помощью связанного набора треугольников, описывающих форму поверхности с минимальным достаточным для расчёта разрешением (рис. 4(e)).

Модель поверхности форсунки в виде треугольников с минимально необходимым для расчёта разрешением содержит около 5000 треугольников (рис. 4(e)). Вычисление расстояния до триангулированной поверхности форсунки сводилось к нахождению расстояния до ближайшего элемента треугольника (вершины, рёбра, грани) из всего набора треугольников. Для вычисления расстояния до треугольника использовался алгоритм, описанный в [16].

Каждая итерация по времени содержала в себе проход по всем ячейкам с вычислением расстояний до поверхности форсунки, используемых для переразбиения сетки. Временные затраты на вычисление расстояний по трём описанным алгоритмам в сравнении с затратами на расчёт гидродинамики потока приведены в таблице.

Затраты на вычисления для одного шага по времени (30 млн ячеек, 128 вычислительных ядер)

Название функции	Время, с
Расчёт расстояния по треугольникам	35.1
Расчёт расстояния по формулам 1,2	0.151
Расчёт расстояния с уточнением	0.230
Расчёт гидродинамики потока	20.6

Как видно, расчёт расстояния по треугольникам поверхности занимает время, сравнимое со временем, затраченным на расчёт гидродинамики потока, что является крайне неэффективным. Тогда как, время расчёта расстояния с использованием комбинированного представления геометрии (по формуле (1)) занимает лишь 0,7 % от времени расчёта гидродинамики, что слабо влияет на общую длительность расчёта. Использование уточнённого значения расстояния путём явного учёта контуров пересечения поверхностей элементарных объектов (с контурами, состоящими из 20 отрезков) несущественно увеличивает время расчёта расстояния до 1,1 % от времени расчёта гидродинамики, что всё ещё является приемлемым. Разрешение контуров пересечения можно менять в процессе расчёта, что позволит найти баланс между точностью вычисления расстояния и производительностью.

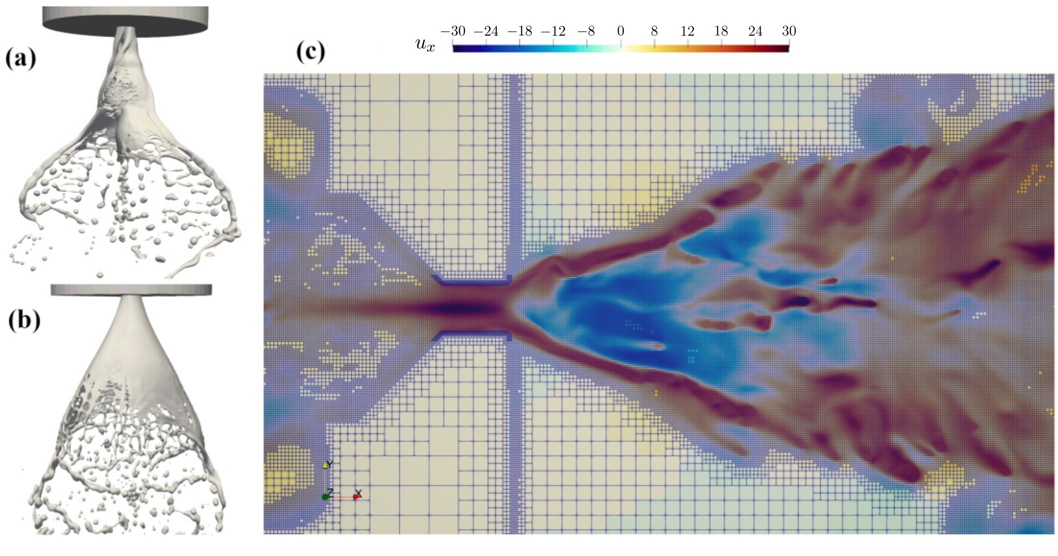

Рис. 5. (a)–(b) — вид струи жидкости в два последовательных момента времени (3 мс и 9 мс от старта расчёта); (c) — мгновенное поле скорости и ячейки динамической сетки в продольном сечении струи (момент времени $t = 4$ мс). u_x измеряется в м/с

Все три способа расчёта расстояния приводят к практически неотличимым гидродинамическим полям и форме струи. Формы струи в переходном режиме (когда из сопла вылетают несколько струй) и в установившемуся режиме (когда струя принимает форму конической плёнки) приведены на рис. 5(a)–(b). Поле скорости в продольном сечении и вид динамической сетки приведены на рис. 5(c).

После переходного процесса струя преобразуется в конусное течение жидкости в виде тонкой плёнки, которая затем распадается на капли из-за утончения под действием центробежной силы. Сравнение установившегося течения из данной форсунки, полученного в расчёте, с наблюдениями показывает качественное согласие.

ЗАКЛЮЧЕНИЕ

В работе предложен алгоритм вычисления расстояния до поверхностей, образованных булевыми комбинациями элементарных объектов двух типов (осесимметричного контура и капсулы). Новизна алгоритма заключается в явном учёте контуров пересечения поверхностей элементарных объектов в процедуре вычисления расстояния. Работа алгоритма протестирована на примере гидродинамического расчёта истечения струи керосина в воздушную среду из центробежной форсунки. Вычисленные значения расстояния используются в расчёте для задания положения границы твёрдого тела (метод вмороженных границ), динамически пересчитываемого во время переразбиения сетки. Расчёт расстояния по предложенному алгоритму показал значительно большую скорость вычисления по сравнению с использованием описания 3D модели треугольниками. Точность восстановления расстояния может быть контролируема во время расчёта, путём выбора разрешения для дискретизации контура пересечений поверхностей элементарных объектов. Вычисление расстояния по предложенному алгоритму требует незначительных вычислительных затрат (по сравнению с затратами на расчёт гидродинамики) и позволяет в перспективе использовать динамически изменяемую форму геометрии путём перемещения или изменения параметров элементарных объектов, входящих в булеву комбинацию, образующую обтекаемое тело.

ФИНАНСИРОВАНИЕ РАБОТЫ

Исследование выполнено при финансовой поддержке Российского научного фонда (проект № 22-79-10246). Вычислительные ресурсы предоставлены в рамках госзадания Института теплофизики СО РАН (проект FWNS-2025-0002). Других источников финансирования проведения или руководства данным конкретным исследованием не было.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы данной работы заявляют, что у них нет конфликта интересов.

ЛИТЕРАТУРА

1. *Popinet S.* A quadtree-adaptive multigrid solver for the Serre—Green—Naghdi equations // J. Comput. Phys. 2015. V. 302, N 2. P. 336–358.
2. *Popinet S.* An accurate adaptive solver for surface-tension-driven interfacial flows // J. Comput. Phys. 2009. V. 228, N 16. P. 5838–5866.
3. *Van Hooft J. A., Popinet S., Van Heerwaarden C. C., Van der Linden S. J., De Roode S. R., Van de Wiel B. J.* Towards adaptive grids for atmospheric boundary-layer simulations // Boundary Layer Meteorol. 2018. V. 167, N 3. P. 421–443.
4. *Almgren A. S., Bell J. B., Colella P., Howell L. H., Welcome M. L.* A conservative adaptive projection method for the variable density incompressible Navier—Stokes equations // J. Comput. Phys. 1998. V. 142, N 1. P. 1–46.
5. *Greaves D.* A quadtree adaptive method for simulating fluid flows with moving interfaces // J. Comput. Phys. 2004. V. 194, N 1. P. 35–56.
6. *Huet D. P., Wachs A.* A Cartesian-octree adaptive front-tracking solver for immersed biological capsules in large complex domains // J. Comput. Phys. 2023. V. 492, N 1. Article 112424.

7. *Johansen H., Colella P.* A Cartesian grid embedded boundary method for Poisson's equation on irregular domains // J. Comput. Phys. 1998. V. 147, N 1. P. 60–85.
8. *Menter F. R.* Two-equation eddy-viscosity turbulence models for engineering application // AIAA J. 1994. V. 32, N 8. P. 1598–1605.
9. *Günther C., Meinke M., Schröder W.* A flexible level-set approach for tracking multiple interacting interfaces in embedded boundary methods // Comput. Fluids. 2014. V. 102, N 1. P. 182–202.
10. *Limare A., Popinet S., Josserand C., Xue Z., Ghigo A.* A hybrid level-set/embedded boundary method applied to solidification-melt problems // J. Comput. Phys. 2023. V. 474, N 10. Article 111829.
11. *Cheny Y., Botella O.* The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties // J. Comput. Phys. 2010. V. 229, N 4. P. 1043–1076.
12. *Jones M. W., Baerentzen J. A., Sramek M.* 3D distance fields: A survey of techniques and applications // IEEE Trans. Vis. Comput. Graph. 2006. V. 12, N 4. P. 581–599.
13. *Hormann K., Agathos A.* The point in polygon problem for arbitrary polygons // Comput. Geom. 2001. V. 20, N 3. P. 1043–1076.
14. *Baerentzen J. A., Aanaes H* Signed distance computation using the angle weighted pseudonormal // IEEE Trans. Vis. Comput. Graph. 2005. V. 11, N 3. P. 243–253.
15. *Vozhakov I. S., Hrebtov M. Y., Yavorsky N. I., Mullyadzhanov R. I.* Numerical Simulations of Swirling Water Jet Atomization: A Mesh Convergence Study // Water. 2023. V. 15, N 14. Article 2552.
16. *Jones M. W.* 3D distance from a point to a triangle // Department of Computer Science, University of Wales Swansea Technical Report CSR-5. 1995.

SIBERIAN JOURNAL OF INDUSTRIAL MATHEMATICS

UDC 519.65

COMPUTATION OF A DISTANCE FIELD BY MEANS OF COMBINED GEOMETRY REPRESENTATION IN FLUID DYNAMICS SIMULATIONS WITH EMBEDDED BOUNDARIES

© 2024 **M. Y. Hrebtov**[1,2a], **R. I. Mullyadzhanov**[1,2b]

[1]*Kutateladze Institute of Thermophysics, Siberian Branch, Russian Academy of Sciences, Novosibirsk, 630090 Russia,*
[2]*Novosibirsk State University, Novosibirsk, 630090 Russia*

E-mails: [a]weexov@yandex.ru, [b]rustammul@gmail.com

Received 27.11.2024, revised 27.11.2024, accepted 11.12.2024

Abstract. We present a method for calculating the signed distance field to three-dimensional geometric models by representing them as a result of Boolean operations on elementary objects for each of which the signed distance is known. Two versions of the algorithm are proposed. The first is a simplified version for quick calculation of the rough distance approximation (with an exact zero isosurface and correct separation of domains inside and outside the model). The second version includes calculation of the distance to the intersection contours between elements, allowing the distance to be reconstructed with a greater accuracy without considerable additional computational costs. Both methods are much faster than the computation of distance based on the triangulation of the surfaces. The proposed approach also allows for interactively changing relative positions and orientation of the geometry parts; this makes it possible to perform calculations with moving boundaries. The approach has been tested in fluid dynamics simulation with an interphase boundary and adaptive multilevel grid refinement in `Basilisk` open source code for simulation of multiphase flows.

Keywords: distance to object, computational geometry, numerical modeling, continuous medium, dynamic grid.

DOI: 10.33048/SIBJIM.2024.27.411

REFERENCES

1. S. Popinet, "A quadtree-adaptive multigrid solver for the Serre—Green—Naghdi equations," J. Comput. Phys. **302** (2), 336–358 (2015).
2. S. Popinet, "An accurate adaptive solver for surface—tension—driven interfacial flows," J. Comput. Phys. **228** (16), 5838–5866 (2009).
3. J. A. Van Hooft, S. Popinet, C. C. Van Heerwaarden, S. J. Van der Linden, S. R. De Roode, and B. J. Van de Wiel, "Towards adaptive grids for atmospheric boundary-layer simulations," Boundary Layer Meteorol. **167** (3), 421–443 (2018).
4. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, "A conservative adaptive projection method for the variable density incompressible Navier—Stokes equations," J. Comput. Phys. **142** (1), 1–46 (1998).
5. D. Greaves, "A quadtree adaptive method for simulating fluid flows with moving interfaces," J. Comput. Phys. **194** (1), 35–56 (2004).
6. D. P. Huet and A. Wachs, "A Cartesian-octree adaptive front-tracking solver for immersed biological capsules in large complex domains," J. Comput. Phys. **492** (1), 112424 (2023).

English translation is published in Journal of Applied and Industrial Mathematics, 2024, Vol. 18, No. 4, pp. 697–708.

7. H. Johansen and P. Colella, “A Cartesian grid embedded boundary method for Poisson’s equation on irregular domains,” J. Comput. Phys. **147** (1), 60–85 (1998).
8. F. R. Menter, “Two-equation eddy-viscosity turbulence models for engineering application,” AIAA J. **32** (8), 1598–1605 (1994).
9. C. Günther, M. Meinke, and W. Schröder, “A flexible level-set approach for tracking multiple interacting interfaces in embedded boundary methods,” Comput. Fluids **102** (1), 182–202 (2014).
10. A. Limare, S. Popinet, C. Josserand, Z. Xue, and A. Ghigo, “A hybrid level-set/embedded boundary method applied to solidification-melt problems,” J. Comput. Phys. **474** (10), 111829 (2023).
11. Y. Cheny and O. Botella, “The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties,” J. Comput. Phys. **229** (4), 1043–1076 (2010).
12. M. W. Jones, J. A. Baerentzen, and M. Sramek, “3D distance fields: A survey of techniques and applications,” IEEE Trans. Vis. Comput. Graph. **12** (4), 581–599 (2006).
13. K. Hormann and A. Agathos, “The point in polygon problem for arbitrary polygons,” Comput. Geom. **20** (3), 1043–1076 (2001).
14. J. A. Baerentzen and H. Aanaes, “Signed distance computation using the angle weighted pseudonormal,” IEEE Trans. Vis. Comput. Graph. **11** (3), 243–253 (2005).
15. I. S. Vozhakov, M. Y. Hrebtov, N. I. Yavorsky, and R. I. Mullyadzhanov, “Numerical Simulations of Swirling Water Jet Atomization: A Mesh Convergence Study,” Water **15** (14), 2552 (2023).
16. M. W. Jones, “3D distance from a point to a triangle,” Technical Report CSR-5 (Dep. Comput. Sci., Univ. Wales, Swansea, 1995).