

Комбинированный алгоритм вычисления расстояния со знаком для задач численного моделирования физических процессов и визуализации движения твердых тел

С.А. Суков¹

Институт прикладной математики им. М.В. Келдыша Российской академии наук

¹ ORCID: 0000-0002-0667-6955, ssoukov@gmail.com

Аннотация

В статье рассматривается проблема инициализации поля функции расстояния со знаком до поверхности движущегося твердого тела произвольной формы. Предложен комбинированный алгоритм быстрого вычисления приближенных значений функции с контролируемой потерей точности. Его идея состоит в том, чтобы интерполировать функцию по ячейкам адаптивной решетки с локальным переключением на поиск расстояния до поверхностной триангуляции. Данный алгоритм может применяться как для визуализации движения поверхностей, так и для решения различных геометрических задач, возникающих в процессе численного моделирования физических процессов. Ошибка определения функции не зависит от формы тела и особенностей траектории движения. Работа содержит описание алгоритма генерации интерполяционной решетки с учетом заданной погрешности вычислений и алгоритма вычисления расстояния со знаком до триангуляции с использованием двоичного дерева поиска. На примерах обработки сферической поверхности и модели крылатой ракеты продемонстрирована возможность использования комбинированного подхода для визуализации движения твердых тел и в численных расчетах газодинамических течений.

Ключевые слова: поле функции расстояния со знаком, визуализация данных, метод линий уровня, поверхностная триангуляция.

1. Введение

При решении широкого круга научных проблем, связанных с геометрическим моделированием, компьютерной анимацией, обработкой изображений, моделированием физических процессов и визуализацией результатов численных экспериментов, для описания формы тел используется функция расстояния со знаком [1]. Геометрия тела Ω с границей $\partial\Omega$ (рис. 1) задается функцией $\phi_{\partial\Omega}(\mathbf{x})$, которая принимает положительные значения во внешней подобласти Ω^+ и отрицательные значения в подобласти Ω^- внутри тела. Границе тела соответствует нулевая изоповерхность $\phi_{\partial\Omega}$.

Абсолютная величина функции равна кратчайшему расстоянию до границы:

$$\phi_{\partial\Omega}(\mathbf{x}) = \begin{cases} \inf_{\mathbf{x}_B \in \partial\Omega} \|\mathbf{x} - \mathbf{x}_B\|, & \mathbf{x} \in \Omega^+ \\ 0, & \mathbf{x} \in \partial\Omega \\ -\inf_{\mathbf{x}_B \in \partial\Omega} \|\mathbf{x} - \mathbf{x}_B\|, & \mathbf{x} \in \Omega^- \end{cases}.$$

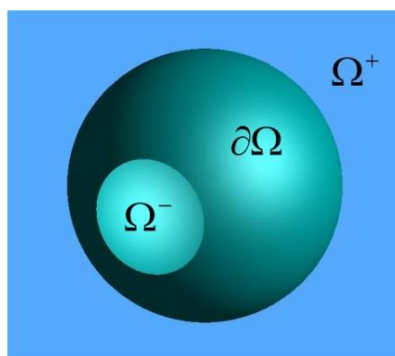


Рис. 1. Определение функции расстояния со знаком относительно поверхности сферы.

В задачах численного моделирования течений жидкости и газа $\phi_{\partial\Omega}$ обычно используется для реконструкции границ движущихся в потоке препятствий на элементах расчетной сетки [2] и адаптации сетки к актуальному положению $\partial\Omega$. В последнем случае дополнительным критерием адаптации становится единичный вектор градиента $\nabla\phi_{\partial\Omega}$. Значения $\phi_{\partial\Omega}$ могут так же быть коэффициентами уравнений математической модели [например, 3]. Визуализация результатов вычислительных экспериментов здесь включает отображение движущейся по рассчитанной траектории поверхности тела.

В плане требований к точности определения $\phi_{\partial\Omega}$ область численного моделирования условно делится на три зоны (рис. 2). Минимальная ошибка должна обеспечиваться вблизи поверхности тела, где она влияет на точность реконструкции $\partial\Omega$ (зона А) и эквивалентна погрешности определения коэффициентов уравнений математической модели (зона В). В зоне С значения $\phi_{\partial\Omega}$ и компоненты вектора $\nabla\phi_{\partial\Omega}$ используются только как критерии сеточной адаптации, что позволяет повысить уровень допустимой погрешности вычислений.

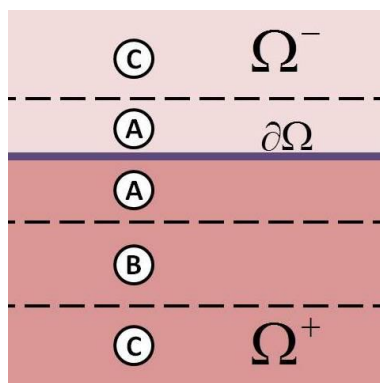


Рис. 2. Выделение зон в области решения задачи.

Для моделирования изменения поля $\phi_{\partial\Omega}$ в процессе перемещения Ω , как правило, применяются методы [4-6], основанные на численном решении уравнения переноса

$$\frac{\partial\phi_{\partial\Omega}}{\partial t} + \mathbf{u}\nabla\phi_{\partial\Omega} = 0. \quad (1)$$

Траектория движения тела задается мгновенной скоростью $\mathbf{u}(\mathbf{x}, t)$. Подход относительно просто обобщается на случай деформации $\partial\Omega$, но имеет ряд существенных недостатков с точки зрения моделирования движения твердых тел. Так, даже при численном интегрировании (1) с использованием высокоточных методов накопление ошибок приводит к постепенному искажению границы. Дополнительным препятствием для практического применения программных реализаций соответствующих алгоритмов может стать их высокая ресурсоемкость. Кроме того, значения искомой функции определяются в узлах или центрах ячеек расчетных сеток. Поэтому поиск $\phi_{\partial\Omega}(\mathbf{x})$ в произвольной точке связан с интерполяцией, которая приводит к потере точности.

Максимальную скорость вычисления $\phi_{\partial\Omega}$ с минимальной погрешностью, теоретически, должны обеспечивать аналитические алгоритмы. Однако для тел, поверхность которых состоит из множества криволинейных граней, вывод явной функциональной зависимости между расстоянием и координатой представляет серьезную проблему. В таком случае рассматривается дискретная модель тела. Поверхность $\partial\Omega$ аппроксимируется множеством многоугольников. Время вычисления расстояния до дискретной модели зависит от ее размерности (общего числа многоугольников) и эффективности применяемого метода упорядочивания данных.

В компьютерной графике функция расстояния со знаком рассматривается в качестве неявного представления твердых и деформируемых поверхностей в алгоритмах построения воксельных и полигональных моделей объектов [7], а также в алгоритмах генерации изображений методом трассировки лучей [8]. При этом поле функции задается на регулярных или адаптивных решетках, где в вершинах хранятся точные значения $\phi_{\partial\Omega}$, а приближенные значения в произвольных точках вычисляются интерполяцией по ячейкам [9-11]. Данный алгоритм характеризуется фиксированной по времени ошибкой и высоким быстродействием, но в чистом виде неприменим в расчетах физических процессов. Максимальная погрешность определения расстояния со знаком в зонах А и В равна нескольким процентам от шага расчетной сетки. Поэтому число ячеек и объем данных описания структуры интерполяционной решетки могут выходить за пределы ограничений на размер представления геометрии.

В настоящей работе изложен комбинированный подход к вычислению $\phi_{\partial\Omega}$, в котором сочетаются интерполяция и поиск расстояния до дискретной модели поверхности. Принципиальная возможность его использования в численных расчетах показана в [12] на примере моделирования двумерных течений. Далее обсуждаются детали реализации комбинированного подхода в трехмерном случае. Приводится описание алгоритма генерации адаптивных интерполяционных решеток и алгоритма вычисления расстояния со знаком до триангуляции. Представлены примеры и параметры решеток для инициализации поля функции расстояния со знаком в задачах визуализации поверхностей и моделирования газодинамического обтекания твердых тел.

2. Расстояние со знаком до триангуляции

Пространственная триангуляция является одним из основных подходов к дискретизации границ тел. Поверхность $\partial\Omega$ аппроксимируется триангуляцией T , которая состоит из N_T ориентированных треугольников T_i :

$$\partial\Omega = T = \bigcup_{i=1}^{N_T} T_i.$$

Условие ориентации треугольника означает соблюдение известного порядка перечисления его вершин (например, против часовой стрелки со стороны Ω^+), что позволяет однозначно определить направление единичной внешней нормали \mathbf{n}_{T_i} к плоскости треугольника. В свою очередь в любой точке $\mathbf{c} \in T$ может быть найден вектор взвешенной по углу единичной псевдонормали

$$\mathbf{N}_{\mathbf{c}} = \frac{\sum_{\mathbf{c} \in T_i} \alpha_i \mathbf{n}_{T_i}}{\left\| \sum_{\mathbf{c} \in T_i} \alpha_i \mathbf{n}_{T_i} \right\|}.$$

Направление псевдонормали вычисляется как сумма нормалей инцидентных по отношению к \mathbf{c} треугольников, взятых с угловыми коэффициентами [13]. Для \mathbf{c} , расположенной строго внутри треугольника (рис. 4а), коэффициент принимает значение 2π , а псевдонормаль совпадает с нормалью к плоскости треугольника. Угловые коэффициенты в точках, находящихся на ребрах триангуляции, равны π (рис. 4б). В таком случае $\mathbf{N}_{\mathbf{c}}$ сонаправлен с вектором суммы нормалей двух граничащих по ребру треугольников. Если \mathbf{c} совпадает с вершиной триангуляции (рис. 4в), то коэффициенты приравниваются углам при вершине.

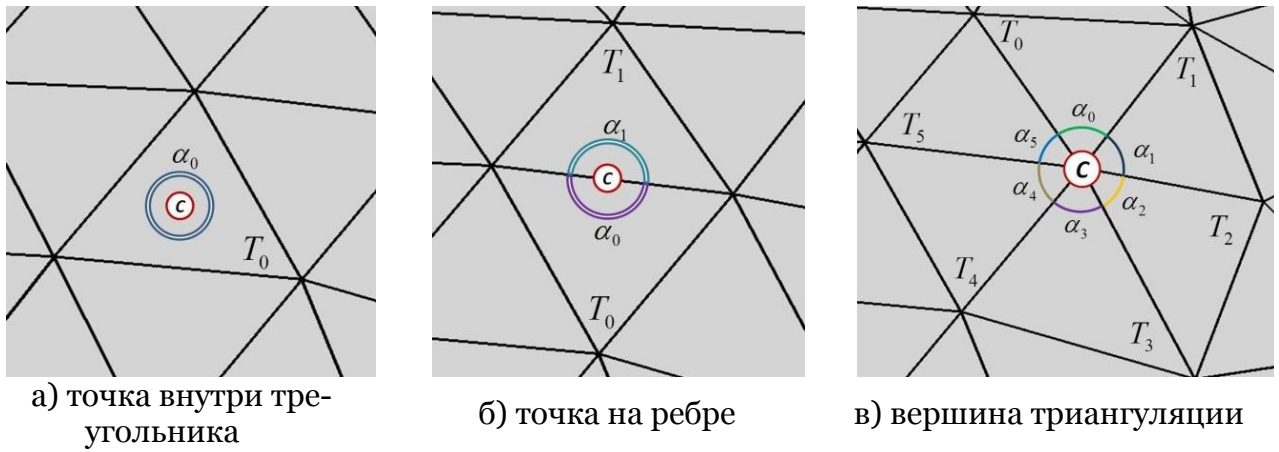


Рис. 3. Определение угловых коэффициентов.

Расстояние со знаком от точки x до триангуляции T вычисляется по формуле

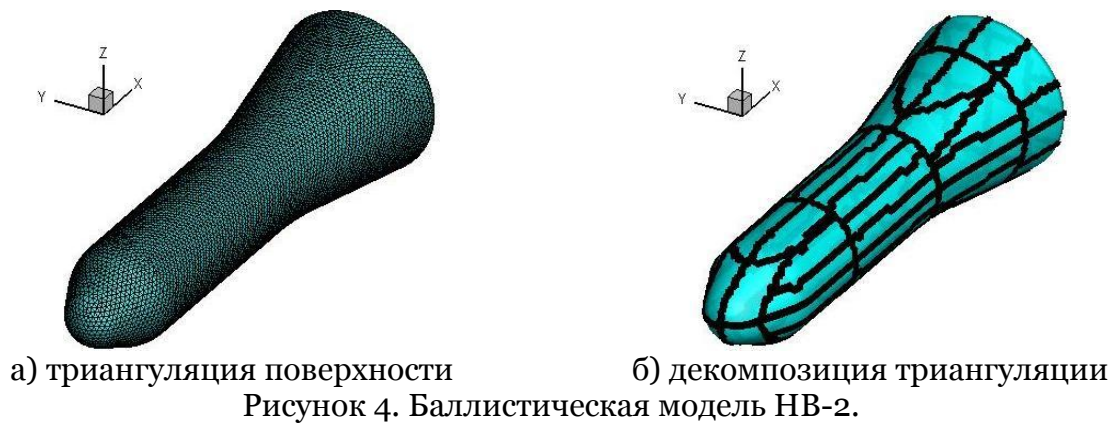
$$\phi_T(x) = \|x - c\| \text{sign}((x - c) \cdot N_c),$$

где c - ближайшая к x точка на поверхности триангуляции. Направление вектора градиента задается как

$$\nabla \phi_T(x) = \begin{cases} N_c, & \phi_T(x) = 0 \\ \frac{x-c}{\|x-c\|}, & \phi_T(x) > 0 \\ \frac{c-x}{\|c-x\|}, & \phi_T(x) < 0 \end{cases}.$$

Базовый алгоритм определения $\phi_T(x)$ предполагает поиск минимальной по модулю величины в процессе последовательного вычисления расстояний $\phi_{T_i}(x)$ до каждого из треугольников. В оптимизированных алгоритмах полный перебор треугольников исключается путем предварительной сортировки данных по геометрическому принципу. Упорядочивание позволяет ограничить обход треугольниками, находящимися в ближайшей окрестности x .

Для задач поиска минимальных расстояний в пространстве эффективна сортировка объектов на основе k-d деревьев [14], которые представляют собой разновидность двоичного дерева поиска. Принцип построения k-d дерева состоит в рекурсивной бисекции пространства решения задачи плоскостями, перпендикулярными осям координат. Триангуляция T погружается в параллелепипед, ребра которого параллельны координатным осям. Охватывающий параллелепипед принимается за область построения k-d дерева и становится его корнем. Ориентация и положением плоскостей разбиения узлов дерева выбираются таким образом, чтобы появляющиеся в результате декомпозиции потомки содержали равное число вершин триангуляции. Рекурсивная бисекция узлов продолжается до достижения заданного числа вершин, либо прерывается по ограничению на максимальную глубину дерева. В финале процедуры для листьев составляются списки принадлежащих им треугольников. Треугольник относится к узлу дерева, когда частично или полностью находится внутри ассоциированного с этим узлом параллелепипеда. Элементы триангуляции, пересекающиеся с гранями параллелепипедов нескольких узлов, включаются в списки каждого из них. На рис. 4 показан пример равномерной триангуляции поверхности баллистической модели НВ-2 (рис. 4а) и разбиение ее треугольников на уровне узлов-листьев k-d дерева (рис. 4б). Геометрические параметры тела взяты из работы [15]. Треугольники, оказавшиеся на пересечении с границами листьев дерева, закрашены на рис. 4б черным цветом.

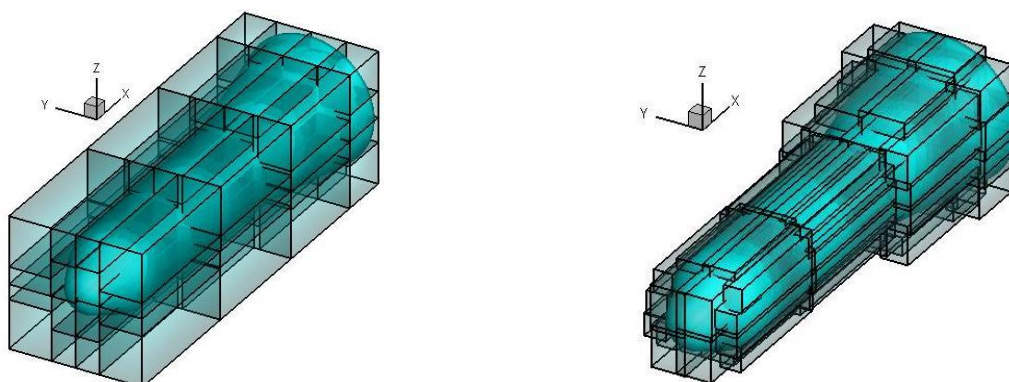


а) триангуляция поверхности

б) декомпозиция триангуляции

Рисунок 4. Баллистическая модель НВ-2.

Процедура поиска $\phi_T(x)$ реализует спуск по k-d дереву посредством рекурсивного вызова функции обработки узлов [16]. Порядок и необходимость обхода дочерних узлов родителя устанавливаются в зависимости с текущим статусом решения задачи. Обработка листового узла состоит в последовательном вычислении расстояний $\phi_{T_i}(x)$ до связанных с ним треугольников. Стартовой целью поиска ставится определение ближайшего к x листа дерева. Минимальное по модулю из расстояний со знаком до его треугольников становится радиусом локальной области поиска с центром в точке x . В дальнейшем при спуске по дереву рассматриваются только те ветви и листья, которые пересекаются с локальной областью поиска. Если расстояние между точкой и очередным обработанным треугольником оказывается меньше радиуса области, то ее размер соответствующим образом уменьшается. Необходимо отметить, что в общем случае связанный с узлом k-d дерева параллелепипед совпадает с минимальным охватывающим параллелепипедом множества принадлежащих узлу треугольников только у корня дерева. Охватывающие параллелепипеды треугольников пересекаются за счет наличия общих элементов, но, как правило, имеют меньший объем (рис. 5). Поэтому с точки зрения сокращения вычислений более эффективной оказывается проверка пересечений узла и локальной области поиска с использованием охватывающего треугольники параллелепипеда.



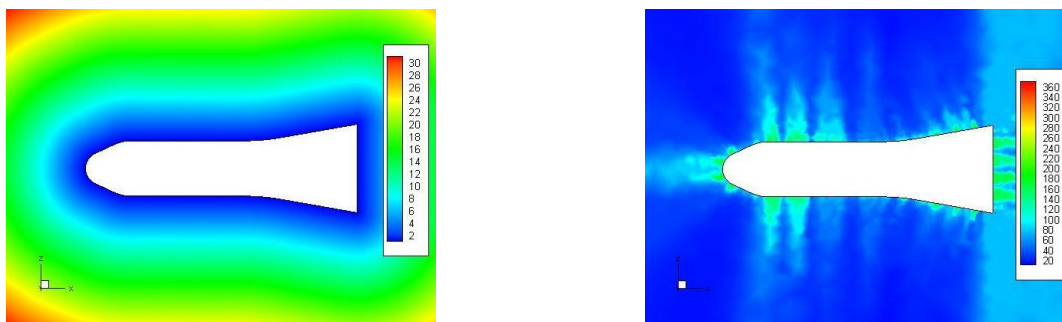
а) параллелепипеды, полученные в результате рекурсивной бисекции корня дерева

б) охватывающие параллелепипеды принадлежащих узлам треугольников

Рис. 5. Варианты геометрии ассоциированных с листьями k-d дерева параллелепипедов.

Если считать, что ресурсоемкость процедуры определения $\phi_T(x)$ пропорциональна числу обработанных треугольников N_{check} , то ускорение от использования двоичного дерева поиска можно оценить по значению коэффициента $S_{kd} = N_T/N_{check}$. Число треугольников N_{check} зависит от нескольких факторов: форма триангулированной поверхности, метод декомпозиции узлов k-d дерева и конкретное расположение точки x по

отношению к T . Минимальному объему вычислений в процессе поиска $\phi_T(x)$ соответствует вариант спуска по дереву с тестированием треугольников только содержащего точку x листового узла. Вероятность расположения точки и ближайшего к ней треугольника внутри одного листа возрастает для x , находящихся вблизи T . Поэтому можно предположить, что в окрестности триангуляции, то есть в зонах, где важна высокая точность вычисления $\phi_{\partial\Omega}$, будет возрастать и коэффициент ускорения S_{kd} .

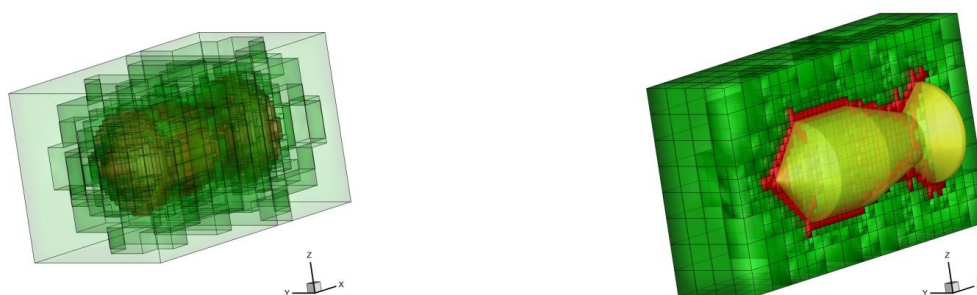


а) поле функции ϕ_T
 б) распределение S_{kd}
 Рис. 6. Расстояние до поверхности баллистической модели НВ-2.

На рис. 6 показано распределение значений функции ϕ_T и коэффициента ускорения S_{kd} с наружной стороны от поверхности баллистической модели НВ-2. В данном случае не прослеживается какой-либо однозначной зависимости между значением коэффициента и расстоянием до поверхности. Тем не менее, на иллюстрациях видно, что значение S_{kd} падает при движении от границы вдоль лучей, начало которых находится в точках $c \in T$, а направление совпадает с вектором N_c .

3. Комбинированный алгоритм вычисления расстояния со знаком

Комбинированный алгоритм реализует ускоренную процедуру определения приближенных значений функции расстояния со знаком с контролируемой потерей точности. Область решения задачи заполняется адаптивной решеткой G , состоящей из кубических ячеек. В узлах решетки хранятся значения функции расстояния со знаком до триангуляции ϕ_T и компоненты вектора градиента $\nabla\phi_T$. Ячейки решетки делятся на два типа: интерполяционные (подобласть G_I) и модельные (подобласть G_T).



а) общий вид
 б) структура решетки в срединном сечении

Рис. 7. Пример построения решетки для пули БМК.

На рис. 7 представлен пример адаптивной решетки для вычисления ϕ_T относительно осесимметричной поверхности пули БМК. Граница тела состоит из цилиндрических и конических поверхностей с диаметром от 8 до 15.5 мм. Длина пули - 34 мм. Красным цветом на рис. 7б отмечена подобласть модельных ячеек.

Способ вычисления значений искомой функции в точке \mathbf{x} зависит от типа ячейки:

$$\phi_G(\mathbf{x}) = \begin{cases} P(\mathbf{x}), & \mathbf{x} \in G_I \\ \phi_T(\mathbf{x}), & \mathbf{x} \in G_T \end{cases}$$

В пределах интерполяционной ячейки функция расстояния со знаком заменяется многочленом

$$P(\mathbf{x}) = C_0xyz + C_1xy + C_2xz + C_3yz + C_4x + C_5y + C_6z + C_7.$$

Коэффициенты $C_{k=0,1,\dots,7}$ определяются по значениям $\phi_T(\mathbf{x})$ в узлах ячейки (трилинейная интерполяция). Аналогичный интерполяционный подход используется для реконструкции компонент вектора градиента. Внутри модельных ячеек явным образом вычисляется расстояние со знаком до поверхностной триангуляции.

Механизм генерации G заключается в иерархическом изотропном измельчении ячеек кубической сетки (шаг h_0). То есть решетка имеет топологию восьмеричного дерева с выделением двух типов листовых ячеек. На структуру G накладывается ограничение максимальной глубины измельчения на уровне l_d (ребро ячейки $h_{l_d} = h_0/2^{l_d}$). Декомпозиция ячеек в процессе построения решетки продолжается до соблюдения условия заданной погрешности интерполяции

$$|\phi_G(\mathbf{x}) - \phi_T(\mathbf{x})| \leq R_\phi(\phi_T(\mathbf{x})). \quad (2)$$

Функция $R_\phi(\phi_T(\mathbf{x})) > 0$ описывает зависимость между допустимой погрешностью вычислений и расстоянием со знаком до триангуляции. Ячейка уровня l_d , которая не отвечает условию (2), относится к подобласти G_T . Кроме того, модельными ячейками становятся ветви дерева, если после их рекурсивного разбиения вплоть до уровня l_d не появляется ни одного потомка интерполяционного типа.

Оценка точности интерполяции реализуется через сравнение значений ϕ_G и ϕ_T в узлах контрольной сетки (шаг $h_t = h_0/2^{l_t}$, $l_t > l_d$). Таким образом, проверка выполнения условия (2) в ячейке уровня l_m включает сравнение точного и приближенного значений функции в $(2^{l_t-l_m} + 1)^3 - 8$ точках. Множество точек контрольной сетки может дополняться списком координат вершин, расположенных на поверхности и в ближайшей окрестности триангуляции.

Интерполяционная решетка генерируется в связанной с телом системе координат. Выбор ее начала и направлений осей учитывает форму тела. Например, для воздушных судов начало связанной системы координат помещается в центр масс, а продольная, вертикальная и поперечная оси определяются конструкцией самолета. Текущее положение тела задается координатами центра и направлениями осей связанной системы координат в системе координат расчетной области (рис. 8).

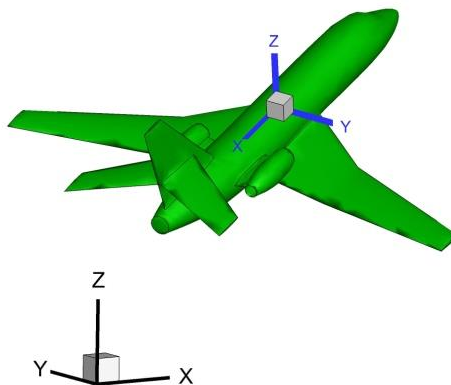


Рис. 8. Позиционирование тела в расчетной области.

Синим цветом отображаются оси связанной системы координат, черным цветом - оси системы координат расчетной области.

Поиск расстояния со знаком до поверхности движущегося объекта включает преобразование координат, вычисление расстояния и обратный поворот вектора градиента.

Геометрически ситуацию можно трактовать так, что Ω перемещается по расчетной области вместе с G .

С точки зрения быстродействия узким местом комбинированного алгоритма является локальный переход от интерполяции к явному поиску ϕ_T внутри подобласти модельных ячеек. Косвенным критерием для оценки размера G_T и эффективности использования комбинированной решетки может быть отношение объема подобласти к площади триангуляции

$$h_{G_T} = V_{G_T}/S_T.$$

В предположении о том, что ячейки модельного типа группируются у поверхности тела, коэффициент h_{G_T} должен соответствовать толщине зоны перехода к работе с триангуляцией. Фактический объем V_{G_T} и картина распределения модельных ячеек зависят от формы рассматриваемой поверхности, шага триангуляции, ориентации $\partial\Omega$ относительно ребер G , базового шага h_0 , глубины измельчения l_d и вида функции R_ϕ . С ростом глубины измельчения из G_T постепенно исключаются объемы призм, полученных протягиванием поверхностных треугольников вдоль перпендикуляров к их плоскостям. Модельные ячейки группируются в основном при вершинах и ребрах T , а также вблизи биссекторных плоскостей острых двугранных углов (рис. 9).

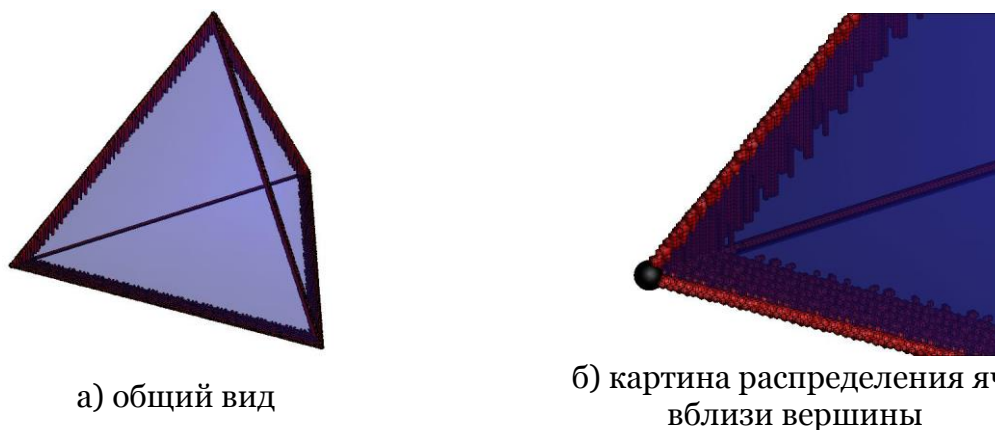


Рис. 9. Пример распределения модельных ячеек решетки для вычисления ϕ_T относительно граней тетраэдра.

Минимизация V_{G_T} за счет измельчения ячеек приводит к нежелательному росту размерности G , то есть общего числа ячеек решетки. Данная проблема частично решается использованием нескольких решеток для вычисления приближенных значений ϕ_G . Генерация отдельной решетки G_i в окрестности каждой из особенностей поверхности позволяет локально максимизировать шаг h_{l_d} . Для набора вложенных решеток задается порядок их инициализации и обхода. Из области генерации каждой последующей решетки исключаются подобласти пересечения с построенными ранее решетками. Находящимся в них ячейкам без тестирования точности интерполяции присваивается модельный тип. Аналогичная очередность обхода соблюдается и во время поиска расстояния до поверхности.

На рис. 10 проиллюстрирован пример схемы расположения четверки вложенных решеток для определения расстояния относительно поверхности самолета. Решетки G_1 и G_2 с повышенной плотностью ячеек окружают двигатели. Решетка G_3 содержит объект и его ближайшую окрестность. Последняя по очередности построения решетка G_4 с максимальным размером ячеек используется для интерполяции расстояния в области дальнего поля.

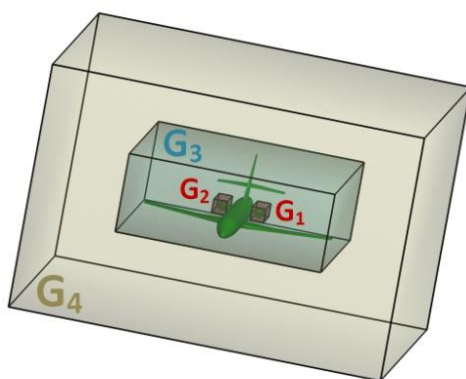


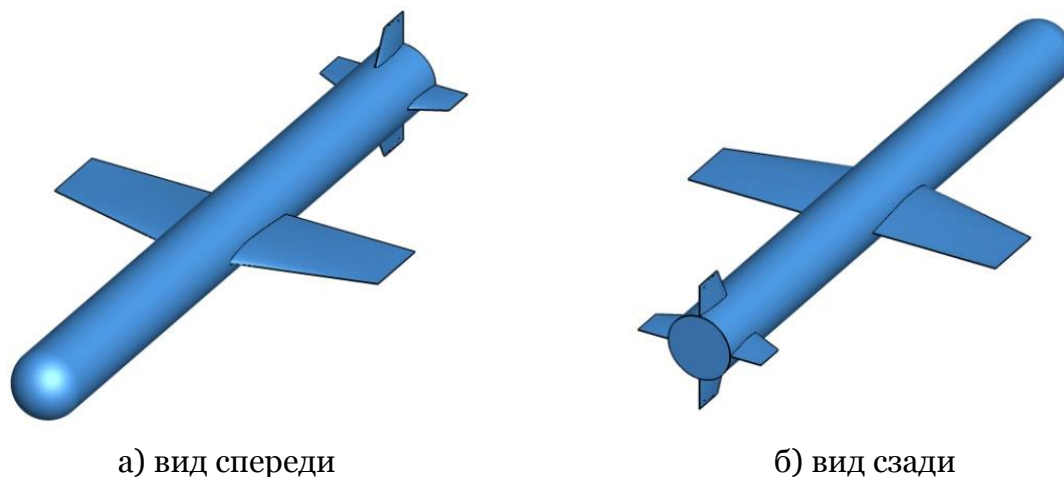
Рис. 10. Пример схемы расположения набора вложенных решеток.

4. Вычислительный эксперимент

Комбинированный подход к вычислению поля функции расстояния со знаком реализован в виде библиотеки подпрограмм для систем с многоядерными процессорами общего назначения (CPU). В состав библиотеки входят модули построения k-d дерева, вычисления расстояния со знаком до триангуляции, модуль генерации и визуализации интерполяционных решеток, а также непосредственно реализация комбинированного алгоритма. Процедура определения приближенных значений функции расстояния со знаком работает с собственными структурами данных, инициализация которых выносятся на предварительный этап. Таким образом, программная реализация комбинированного алгоритма без каких-либо модификаций добавляется в коды как последовательных, так и параллельных программ численного моделирования физических процессов и визуализации данных. В распределенных MPI-приложениях все процессы группы считывают топологию интерполяционной решетки и описание триангуляции, после чего задача решается в последовательном режиме. Для распараллеливания на многоядерную архитектуру цикла с поиском расстояний до поверхности в заданном множестве точек достаточно добавления в код программы соответствующей директивы OpenMP.

Ниже приводится описание двух примеров генерации интерполяционных решеток. Первая решетка строится вокруг упрощенной модели крылатой ракеты для определения поля функции расстояния со знаком в численных расчетах. Вторая решетка содержит сферу единичного диаметра и используется для визуализации поверхности на элементах смешанной сетки.

Внешний вид модели крылатой ракеты показан на рис. 11.



а) вид спереди

б) вид сзади

Рис. 11. Модель поверхности.

Геометрические параметры объекта и поверхностной триангуляции даны в таб. 1.

Таблица 1.

Параметр	Значение
Число вершин	160322
Число треугольников	322154
Охватывающий параллелепипед	10 x 5 x 2
Шаг триангуляции	0.00553
Площадь поверхности	44.3162

Область построения интерполяционной решетки представляет собой охватывающий параллелепипед модели с увеличенными на диаметр корпуса ракеты ($D \approx 1$) ребрами. Кусочная функция допустимой ошибки интерполяции имеет вид:

$$R_\phi(\phi_T(x)) = \begin{cases} 0.075 \cdot |\phi_T(x)|, \phi_T(x) < -2.5\delta \\ 10^{-12} + 0.005 \cdot |\phi_T(x)|, -2.5\delta \leq \phi_T(x) \leq 2.5\delta \\ 0.05 \cdot \phi_T(x), 2.5\delta < \phi_T(x) \leq 5\delta \\ 0.1 \cdot \phi_T(x), 5\delta < \phi_T(x) \end{cases}$$

Коэффициент $\delta = 8.2207116E - 02$ соответствует толщине турбулентного пограничного слоя на плоской пластине при скорости набегающего потока $2.5M$ и параметрах среды на высоте 10 км. Прочие параметры генерации решетки приводятся в таб. 2.

Таблица 2

Параметр	Значение
Область генерации решетки	11 x 6 x 3.2
Базовый шаг h_0	0.2
Минимальная длина ребра ячейки h_{l_d}	0.0125 ($l_d = 4$)
Шаг тестирования точности интерполяции h_t	0.00625 ($l_t = 5$)

Общий вид решетки, построенной для определения значений функции расстояния со знаком, показан на рис. 12. Как и модель поверхности решетка имеет симметричную относительно плоскостей $Y = 0$ и $Z = 0$ структуру. Поэтому для наглядности на иллюстрациях отображается ее четверть на фоне поверхности ракеты. Зеленым цветом выделены внешние границы и структурные особенности подобласти интерполяционных ячеек, красным цветом - подобласть ячеек модельного типа.

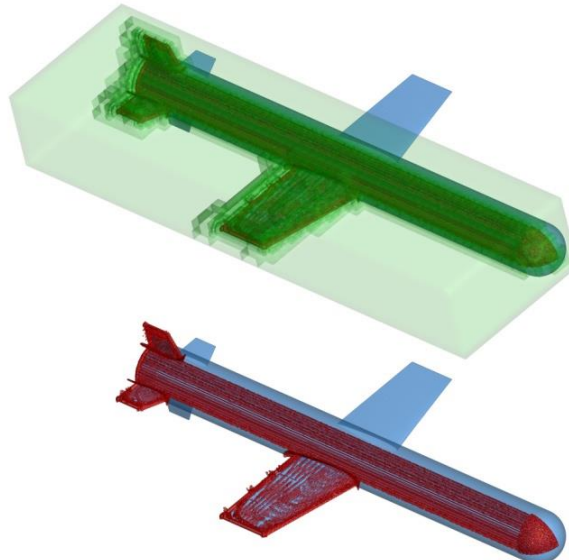


Рис. 12. Общий вид решетки.

В общей сложности решетка содержит 5541528 листовых ячейки, из которых 18.5% относятся к подобласти явного вычисления расстояния до триангуляции. Визуализация распределения модельных ячеек вблизи различных частей конструкции приводится на рис. 13.

Толщина зоны модельных ячеек составляет $h_{G_T} = 0.028819 \approx 0.35\delta$. При этом максимальное расстояние от поверхности до ячейки модельного типа равно 0.206421. На иллюстрациях видно, что наиболее удаленные ячейки располагаются вдоль биссекторных плоскостей двугранных углов между поверхностями корпуса и крыльев. Осредненный коэффициент ускорения поиска расстояния до триангуляции в центрах масс модельных ячеек принимает значение $S_{kd} \approx 414$, что примерно в 3 раза больше среднего значения коэффициента по всей области генерации решетки.

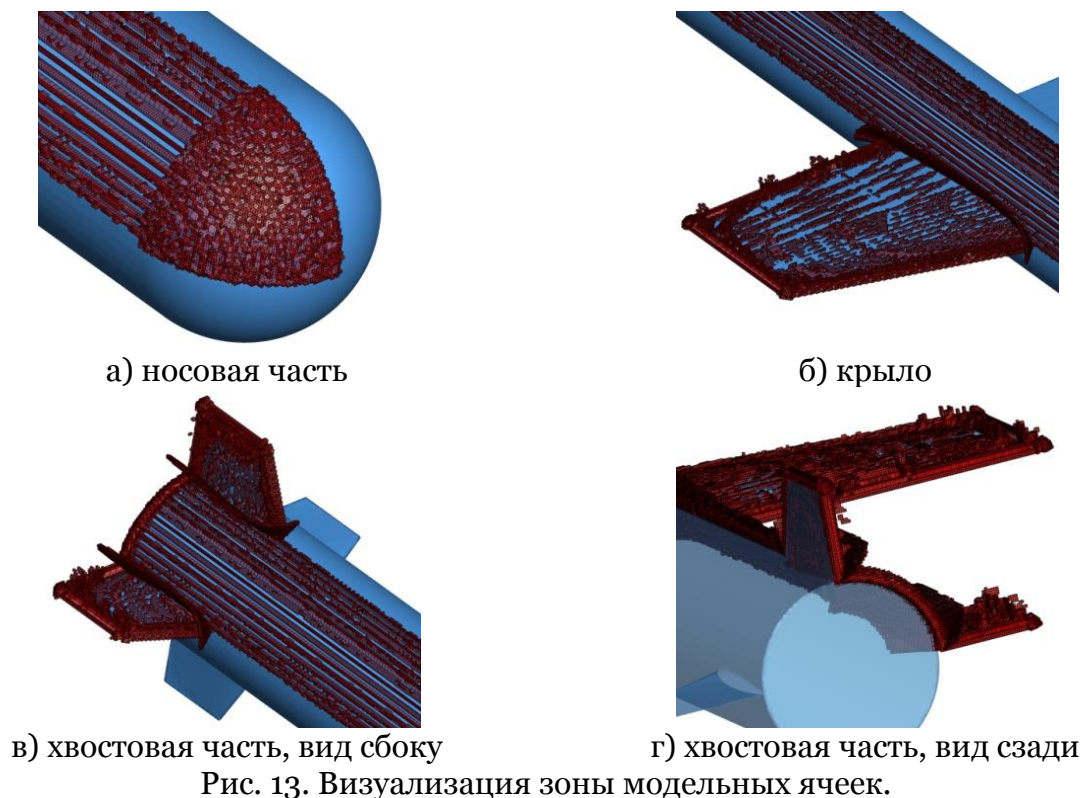


Рис. 13. Визуализация зоны модельных ячеек.

Быстродействие программных модулей и фактическая точность интерполяции оценивались по результатам определения ϕ_G в узлах равномерной ортогональной сетки. Сетка строится внутри области генерации решетки и содержит более $20 \cdot 10^6$ вершин, которые не совпадают с точками проверки точности интерполяции на этапе генерации решетки. Среднее отношение между фактической и заданной погрешностью вычислений

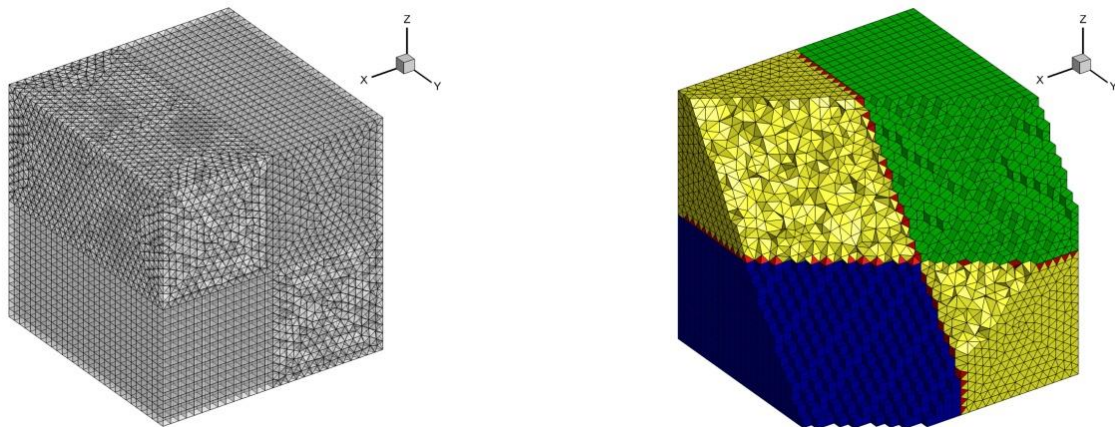
$$d_R(x) = \frac{|\phi_G(x) - \phi_T(x)|}{R_\phi(\phi_T(x))}$$

по вершинам сетки, оказавшимся внутри интерполяционных ячеек, равно $d_R \approx 0.0578$. То есть ошибка интерполяции в среднем не превосходит 6% от пороговой величины. Однако генерация решетки с дискретным тестированием точности в общем случае не гарантирует строгое соблюдение заданной погрешности вычислений. В 197 вершинах, принадлежащих подобласти $|\phi_T(x)| \leq 0.09$, фактическая точность интерполяции оказывается ниже заданной. Максимальное значение $d_R(x) = 2.184$ фиксируется для $\phi_T(x) \approx 0.00042$. Ошибка интерполяции здесь равна 1.1% от абсолютного значения ϕ_T вместо заданных 0.5%. В свою очередь максимальная по модулю ошибка $|\phi_G(x) - \phi_T(x)| \approx 5.9 \cdot 10^{-4}$ соответствует $d_R(x) = 1.354$ или отклонению 0.68% от абсо-

лутного значения ϕ_T . Данные результаты сопоставимы по точности с альтернативными подходами (например, [6]) к вычислению функции расстояния со знаком до границы подвижного твердого тела сложной формы. Стоит также отметить, что речь идет о поиске значений функции в произвольных точках пространства, а не о моделировании изменения поля сеточной функции на некоторой сетке.

Применение комбинированного алгоритма сокращает время вычислений более чем в 300 раз по сравнению с временем поиска точного расстояния до поверхностной триангуляции. Описание геометрии тела (структура k-d дерева поиска, триангуляция поверхности и топология интерполяционной решетки) занимает 250 МБайт дискового пространства.

Второй пример демонстрирует возможность использования комбинированного алгоритма в компьютерной графике. Рассматривается задача реконструкции и отображения поверхности единичной сферы на элементах смешанной сетки (рис. 14). Квазиравномерная сетка с шагом $h_{mesh} \approx 0.0625$ заполняет объем куба с длиной ребра 1.8 (рис. 14а). Она состоит из 24881 вершин и 77442 элементов: 5684 шестигранников (на рис. 14б выделены синим цветом), 57316 тетраэдров (выделены желтым цветом), 12818 треугольных призм (выделены зеленым цветом) и 1625 четырехугольных пирамид (выделены красным цветом).



а) область построения сетки и дискретизация границ

б) структура смешанной сетки в срезе

Рис. 14. Квазиравномерная смешанная сетка.

Параметры интерполяционной решетки для поиска расстояния со знаком относительно сферической поверхности даны в таб. 3.

Таблица 3

Параметр	Значение
Область генерации решетки	2 x 2 x 2
Базовый шаг h_0	0.1
Минимальная длина ребра ячейки h_{l_d}	0.00625 ($l_d = 4$)
Шаг тестирования точности интерполяции h_t	0.0015625 ($l_t = 6$)

Ошибка определения функции ограничивается на уровне 5% от ее абсолютной величины:

$$R_\phi(\phi_{\partial\Omega}(\mathbf{x})) = 10^{-6} + 0.05 \cdot |\phi_{\partial\Omega}(\mathbf{x})|.$$

Граница объекта в данном случае описывается аналитически

$$\phi_{\partial\Omega}(\mathbf{x}) = \sqrt{x^2 + y^2 + z^2} - 0.5.$$

То есть погрешность интерполяции тестируется относительно точных значений $\phi_{\partial\Omega}$.

Поскольку эксперимент предполагает использование решетки исключительно в целях визуализации поверхности, то всем листовым ячейкам уровня l_d вне зависимости от конечной погрешности вычислений присваивается интерполяционный тип ($h_{GT} = 0$). Структура решетки в срединном сечении показана на рис. 15.

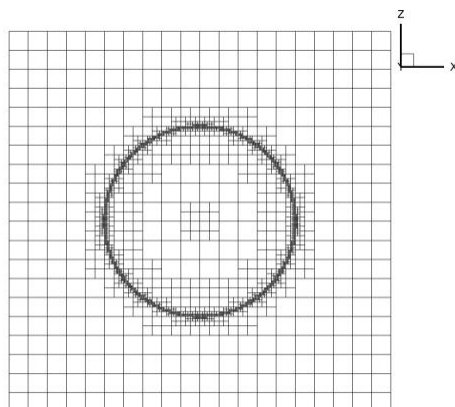
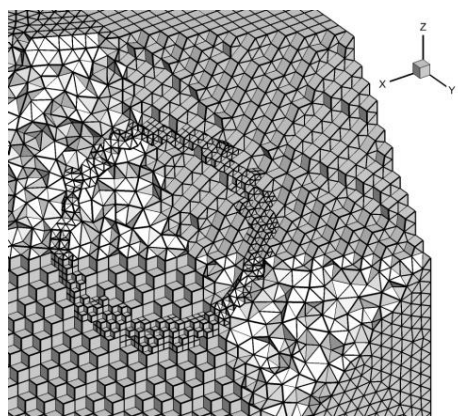


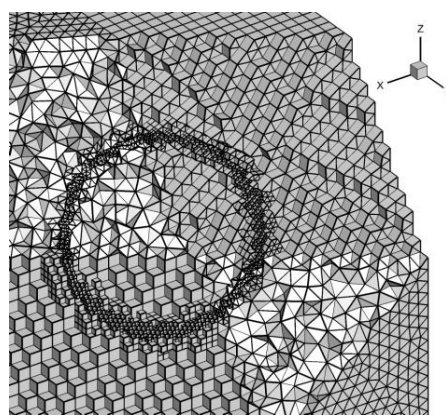
Рис. 15. Структура интерполяционной решетки в срединном сечении.

Визуализация границ сферы на элементах смешанной сетки выполняется двумя способами. Первый способ представляет собой аналог вокселизации. Поверхность аппроксимируется внешней границей множества сеточных элементов, целиком оказавшихся внутри сферы. Второй способ основан на использовании метода линий уровня. Граница реконструируется полигональной сеткой, соответствующей изоповерхности $\phi_{\partial\Omega} = 0$.

С целью повышения степени детализации изображения исходная конформная сетка ($MESH_B$) дважды адаптируется (сетки $MESH_{A1}$ и $MESH_{A2}$) к сферической поверхности (рис. 16). Механизм адаптации состоит в иерархическом изотропном измельчении элементов с добавлением новых вершин на середины ребер, в центры четырехугольных граней и объемов гексаэдров. Область адаптации формируется из ячеек, вершины которых находятся по разные стороны от поверхности сферы.



а) $MESH_{A1}$ - однократное разбиение

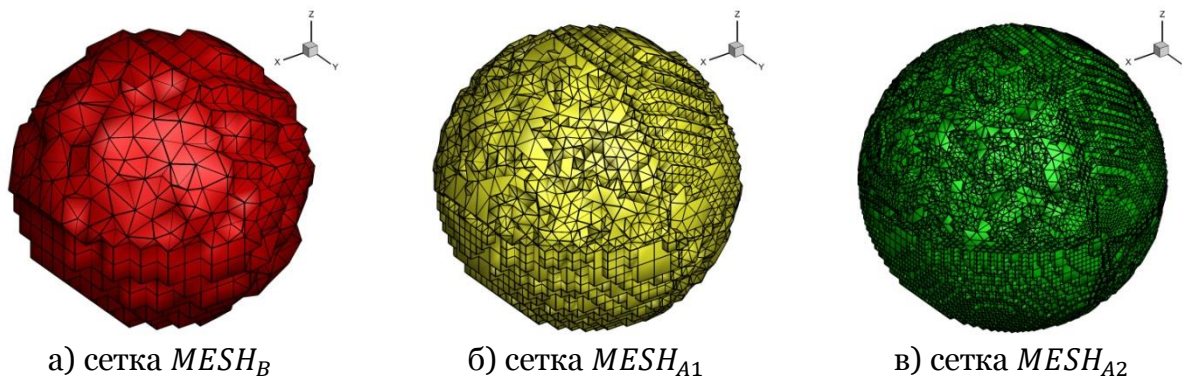


б) $MESH_{A2}$ - два шага измельчения

Рис. 16. Структура сетки после адаптации.

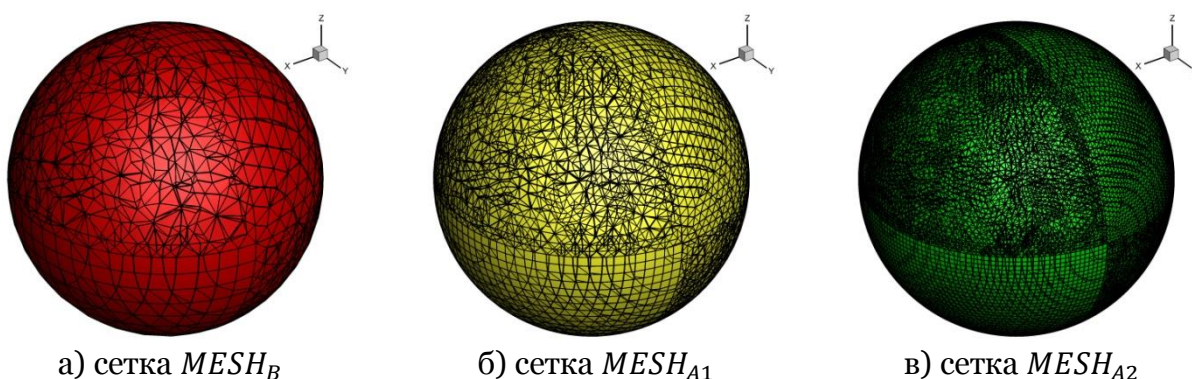
Заданное в вершинах сетки поле функции расстояния со знаком инициализируется интерполяцией по ячейкам сгенерированной решетки. Для построения полигональной сетки и визуализации поверхности использован инструментальный графического пакета TecPlot [17].

Контуры сферы, полученные как отображение внешней границы множества сеточных ячеек, находящихся у нее внутри, показаны на рис. 17. Адаптация сетки повышает точность реконструкции поверхности, но даже на общем плане все изображения заметно отличаются от исходного объекта.



а) сетка $MESH_B$ б) сетка $MESH_{A1}$ в) сетка $MESH_{A2}$
 Рис. 17. Визуализация сферической поверхности границей множества внутренних сеточных ячеек.

Второй способ визуализации с аппроксимацией границ сферы полигональной сеткой дает более качественный результат (рис. 18).



а) сетка $MESH_B$ б) сетка $MESH_{A1}$ в) сетка $MESH_{A2}$
 Рис. 18. Визуализация поверхности на основе полигональной сетки.

При этом разница между реконструкцией поверхности на разных сетках зрительно практически неразличима. Численная оценка (таб. 4) показывает, что максимальное расстояние от вершин реконструированной поверхности до фактической поверхности сферы сокращается после каждого шага адаптации объемной сетки. Одновременно снижается и отклонение площади полигональной сетки от площади поверхности отображаемого объекта.

Таблица 4

	$MESH_B$	$MESH_{A1}$	$MESH_{A2}$
Максимальное расстояние	0.003267	0.000978	0.000385
Отклонение площади	0.518 %	0.136 %	0.039 %

Данный пример подтверждает возможность использования программной реализации интерполяционного подхода в задачах визуализации поверхностей твердых тел.

5. Заключение

В настоящей работе описан комбинированный алгоритм определения расстояния со знаком до границ движущихся твердых тел. Точность представленного алгоритма не зависит от формы тела и параметров траектории его движения. Комбинированный подход может применяться как в численных расчетах физических процессов, так и для визуализации объектов, геометрия которых задается полем функции расстояния со знаком. Программная реализация алгоритма характеризуется повышенным быстродействием и без дополнительных модификаций может добавляться в коды последовательных и параллельных приложений. Дальнейшим направлением работ предполага-

ется реализация процедуры интерполяции с использованием полиномов высокого порядка.

Список литературы

1. Jones M., Bærentzen A., Sramek M. 3D distance fields: A survey of techniques and applications // IEEE transactions on visualization and computer graphics., 2006, 12, 581-99. DOI: 10.1109/TVCG.2006.56.
2. Mittal R., Iaccarino G. Immersed boundary methods // Annu. Rev. Fluid Mech., 2005, v.37, p. 239–261.
3. Spalart P.R., Allmaras S.R. A one-equation turbulence model for aerodynamic flows // AIAA Paper 92-0439, 30th Aerospace Science Meeting, Reno, Nevada, 1992.
4. Osher S., Fedkiw R. Level set methods and dynamic implicit surfaces // New York: Springer-Verlag, 2003, 273p.
5. Shervani-Tabar N., Vasilyev O.V., Stabilized conservative level set method // Journal of Computational Physics, 2018, 375, pp. 1033-1044.
6. Morgan N., Waltz J. 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement // Journal of Computational Physics. 2017, 336, 492-512. DOI: 10.1016/j.jcp.2017.02.030.
7. Jones M. The production of volume data from triangular meshes using voxelization // Computer Graphics Forum, 1996, 15:311-318.
8. Hart J. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. // The Visual Computer, 1996, 12, 527–545. DOI: <http://doi.org/10.1007/s003710050084>.
9. Museth K. VDB: High-resolution sparse volumes with dynamic topology // ACM Trans. Graph. 32, 3, Article 27 (June 2013) 22 pages. DOI: <http://dx.doi.org/10.1145/2487228.2487235>.
10. Koschier D., Deul C., Bender J. Hierarchical hp-adaptive signed distance fields // In: Proceedings of ACM SIGGRAPH EUROGRAPHICS Symposium on Computer Animation (SCA), 2016.
11. Frisken S.F., Perry R.N., Rockwood A.P., Jones T.R. Adaptively sampled distance fields: a general representation of shape for computer graphics // ACM Transactions on Graphics, 2000, 249–254.
12. Abalakin I.V., Kozubskaya T.K., Soukov S.A., Zhdanova N.S. Numerical Simulation of Flows over Moving Bodies of Complex Shapes Using Immersed Boundary Method on Unstructured Meshes // In: Garanzha V., Kamenski L., Si H. (eds) Numerical Geometry, Grid Generation and Scientific Computing. Lecture Notes in Computational Science and Engineering, vol 131. Springer, Cham. 2019. DOI: https://doi.org/10.1007/978-3-030-23436-2_13
13. Baerentzen J.A., Aanaes H. Signed distance computation using the angle weighted pseudonormal // IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 3, pp. 243-253, May-June 2005. DOI: 10.1109/TVCG.2005.49.
14. Bentley J.L. Multidimensional binary search trees used for associative searching // Communications of the ACM. 1975, 18 (9): 509–517. DOI:10.1145/361002.361007.
15. Kryuchkova A.S. Numerical simulation of a hypersonic flow over HB-2 model using UST3D programming code // 2019 J. Phys.: Conf. Ser. 1250 012010. DOI: 10.1088/1742-6596/1250/1/012010.
16. Yianilos P. Data structures and algorithms for nearest neighbor search in general metric spaces // Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. 1993. DOI: 10.1145/313559.313789.
17. TecPlot – CFD Visualization & Analysis Software URL: <https://www.tecplot.com/>

Combined signed distance calculation algorithm for numerical simulation of physical processes and visualization of solid bodies movement

S.A. Soukov¹

Keldysh Institute of Applied Mathematics Russian Academy of Sciences

¹ ORCID: 0000-0002-0667-6955, ssoukov@gmail.com

Abstract

The article deals with the problem of initializing the field of the signed distance function to the surface of a moving solid body of arbitrary shape. A combined algorithm is proposed for fast calculation of approximate values of a function with a controlled loss of accuracy. The idea is to interpolate the function over the cells of an adaptive grid with local switching to find the distance to surface triangulation. This algorithm can be used both for visualizing the motion of surfaces and for solving various geometric problems arising in the process of numerical modeling of physical processes. The error in determining the function does not depend on the shape of the body and the features of the movement trajectory. The paper contains a description of an algorithm for generating an interpolation grid taking into account a given computational error and an algorithm for calculating the signed distance to triangulation using a binary search tree. Using the examples of processing a spherical surface and a cruise missile model, the possibility of using a combined approach for visualizing the motion of solid bodies and in numerical calculations of gas-dynamic flows is demonstrated.

Keywords: signed distance field, data visualization, level-set method, surface triangulation.

References

1. Jones M., Bærentzen A., Sramek M. 3D distance fields: A survey of techniques and applications // IEEE transactions on visualization and computer graphics., 2006, 12, 581-99. DOI: 10.1109/TVCG.2006.56.
2. Mittal R., Iaccarino G. Immersed boundary methods // Annu. Rev. Fluid Mech., 2005, v.37, p. 239–261.
3. Spalart P.R., Allmaras S.R. A one-equation turbulence model for aerodynamic flows // AIAA Paper 92-0439, 30th Aerospace Science Meeting, Reno, Nevada, 1992.
4. Osher S., Fedkiw R. Level set methods and dynamic implicit surfaces // New York: Springer-Verlag, 2003, 273p.
5. Shervani-Tabar N., Vasilyev O.V., Stabilized conservative level set method // Journal of Computational Physics, 2018, 375, pp. 1033-1044.
6. Morgan N., Waltz J. 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement // Journal of Computational Physics. 2017, 336, 492-512. DOI: 10.1016/j.jcp.2017.02.030.
7. Jones M. The production of volume data from triangular meshes using voxelization // Computer Graphics Forum, 1996, 15:311-318.
8. Hart J. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. // The Visual Computer, 1996, 12, 527–545. DOI: <http://doi.org/10.1007/s003710050084>.

9. Museth K. VDB: High-resolution sparse volumes with dynamic topology // ACM Trans. Graph. 32, 3, Article 27 (June 2013) 22 pages. DOI: <http://dx.doi.org/10.1145/2487228.2487235>.
10. Koschier D., Deul C., Bender J. Hierarchical hp-adaptive signed distance fields // In: Proceedings of ACM SIGGRAPH EUROGRAPHICS Symposium on Computer Animation (SCA), 2016.
11. Frisken S.F., Perry R.N., Rockwood A.P., Jones T.R. Adaptively sampled distance fields: a general representation of shape for computer graphics // ACM Transactions on Graphics, 2000, 249–254.
12. Abalakin I.V., Kozubskaya T.K., Soukov S.A., Zhdanova N.S. Numerical Simulation of Flows over Moving Bodies of Complex Shapes Using Immersed Boundary Method on Unstructured Meshes // In: Garanzha V., Kamenski L., Si H. (eds) Numerical Geometry, Grid Generation and Scientific Computing. Lecture Notes in Computational Science and Engineering, vol 131. Springer, Cham. 2019. DOI: https://doi.org/10.1007/978-3-030-23436-2_13
13. Baerentzen J.A., Aanaes H. Signed distance computation using the angle weighted pseudonormal // IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 3, pp. 243-253, May-June 2005. DOI: 10.1109/TVCG.2005.49.
14. Bentley J.L. Multidimensional binary search trees used for associative searching // Communications of the ACM. 1975, 18 (9): 509–517. DOI:10.1145/361002.361007.
15. Kryuchkova A.S. Numerical simulation of a hypersonic flow over HB-2 model using UST3D programming code // 2019 J. Phys.: Conf. Ser. 1250 012010. DOI: 10.1088/1742-6596/1250/1/012010.
16. Yianilos P. Data structures and algorithms for nearest neighbor search in general metric spaces // Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. 1993. DOI: 10.1145/313559.313789.
17. TecPlot – CFD Visualization & Analysis Software URL: <https://www.tecplot.com/>