

## СПИСОК ЛИТЕРАТУРЫ

1. Корнеев В.Д. Параллельное программирование в MPI. – Новосибирск: ИВМиМГ, 2002. – 215 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
3. Вальковский В.А., Малышкин В.Э. Синтез параллельных программ и систем на вычислительных моделях. – Новосибирск: Наука, 1988. – 128 с.
4. Андрианов А.Н., Бугеря А.Б., Ефимкин К.Н., Задыхайло И.Б. Норма. Описание языка. Рабочий стандарт. – Препринт ИПМ им. М.В. Келдыша РАН, № 120, 1995. – 50 с.
5. Востокин С.В. Графическая объектная модель параллельных процессов и ее применение в задачах численного моделирова-

6. ния. – Самара: Изд-во Самарского научного центра РАН, 2007. – 286 с.
6. Новосельцев В.Б. Теория структурных функциональных моделей // Сибирский математический журнал. – 2006. – Т. 47. – № 6. – С. 1342–1354.

Поступила 16.04.2008 г.

### Ключевые слова:

Многопроцессорная вычислительная система, суперЭВМ, предложение вычислимости, параллелизм, линейный участок, синтез программ, граф алгоритма, ярусно-параллельная форма, параллельная программа, C-модель.

УДК 519.67(004.02)

## АЛГОРИТМЫ ЛОКАЛИЗАЦИИ ТОЧКИ В ТРЕХМЕРНОМ ПРОСТРАНСТВЕ ДЛЯ ГЕНЕРАЦИИ ОБЪЕКТА ПРИ МОДЕЛИРОВАНИИ МЕТОДОМ ЧАСТИЦ

В.В. Сергеев, С.Ю. Коростелев, С.Г. Псахье

Институт физики прочности и материаловедения СО РАН, г. Томск

E-mail: svalera@ispms.tsc.ru

*Для решения задачи построения трехмерных объектов сложной конфигурации и заполнения расчетной области моделируемыми частицами предложено два алгоритма. Представлены основные положения, достоинства и недостатки каждого из алгоритмов. На основе тестовых расчетов предложены способы увеличения эффективности решения данной задачи с помощью современных компьютерных технологий.*

### 1. Введение

При компьютерном моделировании в рамках метода частиц [1] одной из топологических проблем является построение трехмерных объектов сложной конфигурации. Процесс генерации объекта для моделирования состоит из двух этапов. На первом этапе необходимо создать геометрическую модель объекта представленную одним или несколькими многогранниками. Для этого можно использовать любую коммерческую или свободно распространяемую программу, например, AutoCAD®. На втором этапе полученные многогранники заполняются моделируемыми частицами. Однако для этого не существует реализованных алгоритмов, поэтому приходится разрабатывать новые.

Предлагаемые в работе алгоритмы тестировались в программах для моделирования методом молекулярной динамики и методом подвижных клеточных автоматов [2, 3]. Основное применение рассматриваемые подходы могут получить в методе подвижных клеточных автоматов, поскольку этот метод позволяет моделировать сложные системы и конструкции.

В рамках этого метода моделируемая система представляет собой ансамбль взаимодействующих автоматов, имеющих конечный размер. Образец представляет собой трехмерное тело произвольной формы. В настоящее время самым удобным явля-

ется способ описания 3D-объекта с помощью полигональной сетки. Это позволяет использовать все доступные на сегодняшний день трехмерные редакторы для создания геометрии образца любой сложности. Имея описанный сеткой образец произвольной геометрической формы, необходимо представить его в виде набора автоматов. Автомат может иметь различную форму в зависимости от упаковки (подробнее в разделе 2). Основной задачей является определение принадлежности автомата (представленного точкой) к образцу (представленному триангуляционной сеткой), т. е. локализация точки в 3D-пространстве.

При заполнении внутренней области многоугольника необходимо определить те точки, которые находятся внутри многоугольника, и те, которые находятся снаружи. Для этого проводится луч, выходящий из проверяемой точки и уходящий в бесконечность в любом направлении и не проходящий через вершины. Если луч пересекает ребра многоугольника нечетное число раз, то точка лежит во внутренней области. Такой подход получил название правило «нечетного равенства». Первый алгоритм, использовавший данную схему, описан в работах [4, 5]. Д.Т. Ли и Ф.П. Препарата предложили алгоритм локализации точки в трехмерном пространстве относительно выпуклого многогранника [6].

В настоящей работе рассматриваются два различных подхода к решению проблемы заполнения 3D-модели, представленной произвольным замкнутым многогранником.

## 2. Алгоритмы заполнения и типы упаковок

Для проведения моделирования методом подвижных клеточных автоматов в 3D пространстве образец должен представлять собой ансамбль взаимодействующих или контактирующих автоматов.

Предложенный метод заполнения моделируемого объекта клеточными автоматами состоит из двух этапов. На первом этапе генерируется проверяемая точка, представляющая центр клеточного автомата. Координаты точки определяются в зависимости от выбранного типа упаковки и размера автомата.

Рассматриваются три варианта упаковки: простая кубическая упаковка, гранецентрированная кубическая упаковка и гексагональная плотная упаковка [7]. При этом в случае простой кубической упаковки автомат имеет 6 соседей, а в случае плотных упаковок – 12 соседей. На втором этапе точка проверяется на принадлежность к образцу. Образец, описанный триангуляционной сеткой, может быть выпуклой и невыпуклой формы. Это накладывает определенные трудности на алгоритм проверки локализации точки в объекте при разработке алгоритмов решения данной задачи.

### 2.1. Алгоритм полного перебора граней

Рассмотрим случай выпуклого трехмерного тела (рис. 1). Для определения положения точки необходимо проверить, где она находится относительно всех граней сетки, описывающей образец. Для этого для каждой грани необходимо вычислить:

$$\begin{aligned} C_1 &= N_x(P_x - F_x); \\ C_2 &= N_y(P_y - F_y); \\ C_3 &= N_z(P_z - F_z); \\ C &= C_1 + C_2 + C_3; \end{aligned} \quad (*)$$

где  $N$  – нормаль грани,  $P$  – проверяемая точка,  $F$  – любая точка, принадлежащая грани (например, одна из вершин). Тогда, если  $C < 0$ , точка находится внутри образца, относительно данной грани, в противном случае – вне. Таким образом, если проверяемая точка будет находиться внутри объекта относительно всех граней, то она будет принадлежать образцу. Если же найдется хоть одна грань, относительно которой точка будет находиться снаружи, то она будет вне образца.

В случае невыпуклого многогранника данный алгоритм не корректен, т. к. любая точка будет находиться снаружи относительно какой-либо грани. Так, например, на рис. 2 точка  $P$  лежит внутри объекта относительно нормалей  $n_2, n_4, n_5, n_6$  и снаружи относительно нормалей  $n_1, n_3$ . Однако предложенный алгоритм можно модифицировать для работы с невыпуклыми телами. Для этого необходимо отобрать грани, относительно которых нужно проверить положение точки, ур. (\*). Алгоритм проверки

грани будет следующий. Необходимо провести отрезок от проверяемой точки до внутренней точки тестируемой грани, затем определить, пересекает ли данный отрезок другие грани, и если он пересекает, то проверять положение точки относительно данной грани не нужно. На рис. 3 отрезок  $PY$  от проверяемой точки  $P$  до внутренней точки  $Y$  грани  $HOMG$  с нормалью  $n_2$  не пересекает других граней, а отрезок  $PX$  от  $P$  до грани  $RSTU$  с нормалью  $n_1$  пересекает другую грань  $EUTF$  в точке  $I$ . Следовательно, проверять положение точки  $P$  относительно грани  $RSTU$  не нужно, а относительно грани  $HOMG$  нужно.

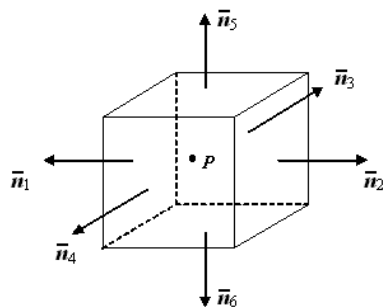


Рис. 1. Выпуклый многогранник

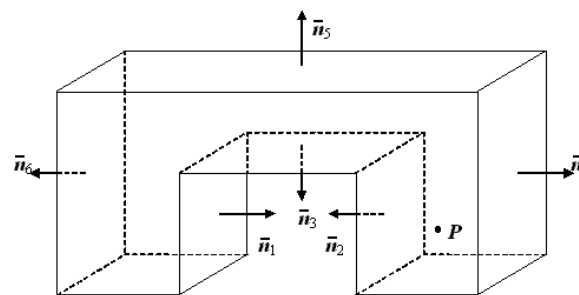


Рис. 2. Невыпуклый многогранник

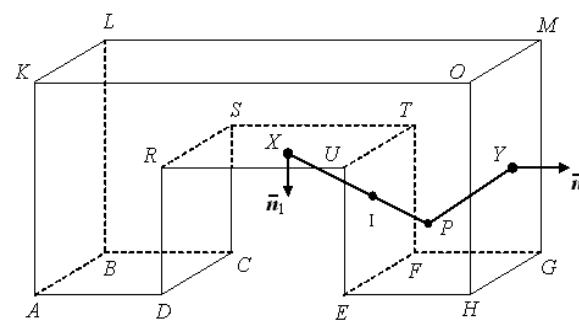


Рис. 3. Проверка граней

Преимуществом данного алгоритма является его верное выполнение даже на сетках с некоторыми погрешностями (например, наличие лишнего ребра), которые могут получиться при её создании. При этом качество его заполнения не зависит от размеров сетки и автомата. Цена данного алгоритма проверки точки  $O(n \cdot n)$ , где  $n$  – количество полигонов в сетке. Соответственно для сложных сеток, содержащих большое количество полигонов, алгоритм будет дорогим. Более дешевым и эффективным является трехмерный алгоритм трассировки луча.

### 2.2. Трехмерный алгоритм трассировки луча

Двухмерный аналог данного алгоритма представлен в работах [8, 9]. Принцип работы алгоритма в двух- и трехмерном пространстве остается одинаковым. Для его применения в трехмерном пространстве рассматривается луч, выходящий из проверяемой точки (рис. 4). Затем рассчитывается количество точек его пересечений с полигонами сетки. Если это количество окажется четным, то точка лежит вне объекта, если нечетным – то внутри. Особенностью данного алгоритма является отсутствие ограничений на луч. В отличие от двухмерного варианта луч может пересекать грани, ребра и проходить через вершины. При этом следует учесть то, что одно и то же ребро содержат две соседних грани. Следовательно, одна точка пересечения луча будет принадлежать обоим граням, и при подсчете общего количества пересечений она будет учитываться дважды, поэтому необходимо проверять попадание точки пересечения на ребро. Отметим, что в этой ситуации количество пересечений следует увеличивать не на 1,0, а на 0,5. Аналогично, при попадании точки пересечения в вершину следует прибавлять величину  $1/g$  к общему количеству пересечений, где  $g$  – количество полигонов, содержащих данную вершину.

На рис. 4 приведен пример проверки двух точек. Точка  $P_2$  находится внутри объекта, т. к. луч, выходящий из нее, имеет нечетное количество точек пересечений с гранями:  $I_3, I_4$  и  $I_5$ . А точка  $P_1$  находится снаружи объекта, т. к. луч имеет четное количество точек пересечения:  $I_1$  и  $I_2$ .

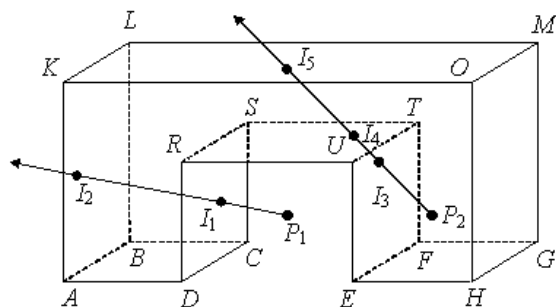


Рис. 4. Метод трассировки луча

Цена алгоритма равна  $O(n)$ , что намного дешевле предыдущего алгоритма. Однако в отличие от него, при наличии ошибки в сетке (например, наличие лишнего ребра) данный алгоритм может привести к некорректному результату работы.

### 3. Результаты тестирования и работы алгоритмов

Описанные выше алгоритмы были реализованы в составе системы моделирования методом подвижных клеточных автоматов. На рис. 5 приведены примеры полигональных сеток образцов для моделирования (слева) и результаты работы алгоритма заполнения с гранецентрированной кубиче-

ской упаковкой (справа). Автоматы для наглядности изображены сферами.

Тестирование алгоритмов производилось на ПЭВМ с конфигурацией:

- процессор Intel Core 2 Duo E6550 2,33 ГГц;
- объем оперативной памяти 2 Гб;
- операционная система Linux (Fedora Core 6);
- результат теста производительности «Whetstones»:  $4,9 \cdot 10^9$  WIPS.

По результатам была составлена сравнительная таблица, где «Алг.1» это алгоритм полного перебора граней, а «Алг.2» – алгоритм трассировки луча. Из нее видно, что при увеличении количества полигонов в сети алгоритм трассировки работает гораздо эффективнее алгоритма полного перебора.

Таблица. Результаты тестирования алгоритмов

Кол-во полигонов	Упаковка	Время работы, с		Кол-во автоматов	
		Алг.1	Алг.2	Алг.1	Алг.2
36	Кубическая	16	8	314721	300000
36	Гранецентри- рованная куби- ческая	25	13	428481	428485
2880		9131	438	4022	3979
50640		44913	3626	23314	18492

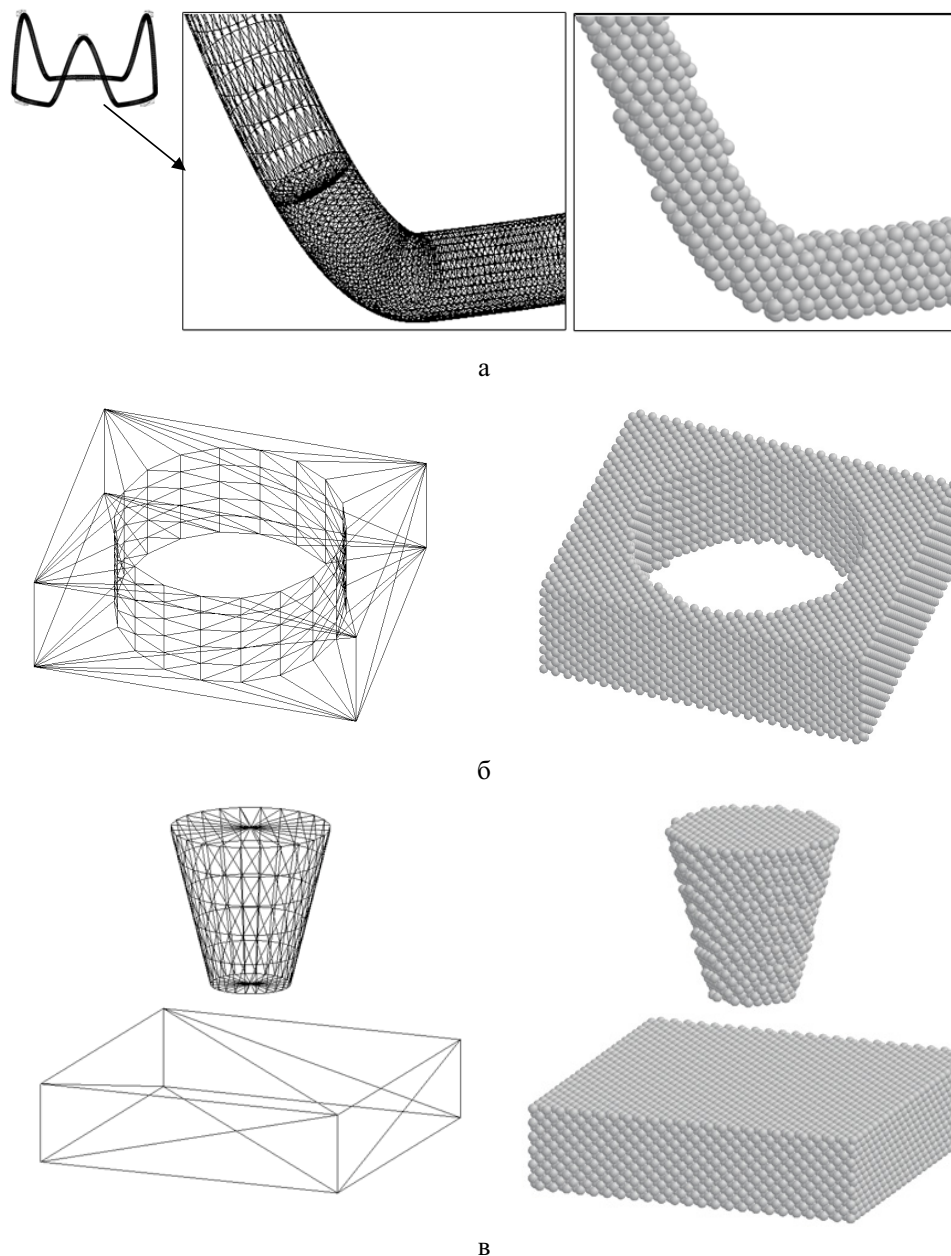
Количество автоматов при работе каждого алгоритма может отличаться, т. к. подход к заполнению в предложенных алгоритмах разный. Какой из алгоритмов является более подходящим, определяется в зависимости от параметров моделирования.

### 4. Заключение

Рассмотрены алгоритмы заполнения частицами произвольного трехмерного объекта представленного замкнутым многогранником. Проведенные тесты показывают, что алгоритм «трассировки луча» будучи весьма эффективным, в некоторых случаях сложных объектов может приводить к неточному заполнению объекта частицами. В тоже время алгоритм полного перебора таких погрешностей не дает. Поэтому в каждом случае необходимо принимать решение об использовании того или иного подхода в зависимости от решаемой физической задачи.

Предложенные алгоритмы можно использовать на многопроцессорных компьютерах. Это даст значительное сокращение времени расчетов. Параллельные вычисления могут быть реализованы за счет независимой обработки каждого автомата, т. е. путем разрешения задачи локализации для точек на различных процессорах.

Область применения описанных выше алгоритмов не ограничивается задачей заполнения. Они могут найти применение при решении любых задач локализации точки в трехмерном пространстве относительно объекта описанного полигональной сеткой, в том числе для задач формирования плотных упаковок из технически сложной геометрической формы.



**Рис. 5.** Результат работы алгоритма заполнения: а) сердечного клапана; б) блока с отверстием; в) мишени и ударника

#### СПИСОК ЛИТЕРАТУРЫ

1. Хокни Р., Иствуд Дж. Численное моделирование методом частиц. – М.: Мир, 1987. – 640 с.
2. Псахье С.Г., Хори Я., Коростелев С.Ю., Смолин А.Ю., Дмитриев А.И., Шилько Е.В., Алексеев С.В. Метод подвижных клеточных автоматов, как инструмент для моделирования в рамках физической мезомеханики // Известия вузов. Физика. – 1995. – № 11. – С. 58–69.
3. Псахье С.Г., Коростелев С.Ю., Смолин А.Ю., Дмитриев А.И. Метод подвижных клеточных автоматов как инструмент физической мезомеханики материалов // Физическая мезомеханика. – 1998. – Т. 1. – № 1. – С. 95–108.
4. Foley J.D., Van Dam A. Fundamentals of Interactive Computer Graphics. – MA: Addison-Wesley, 1984. – 684 p.
5. Foley J.D., Van Dam A. Computer Graphics – Principles and Practice. – MA: Addison-Wesley, 1990. – 1200 p.
6. Lee D.T., Preparata F.P. Location of a point in a planar subdivision and its applications // SIAM J. Comput. – 1977. – V. 6. – № 3. – P. 594–606.
7. Чупрунов Е.В., Хохлов А.Ф., Фаддеев М.А. Основы кристаллографии. – М.: Физматлит, 2004. – 500 с.
8. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение: Пер. с англ. – М.: Мир, 1989. – 478 с.
9. Ласло М. Вычислительная геометрия и компьютерная графика на C++: Пер. с англ. – М.: Изд-во «БИНОМ», 1997. – 304 с.: ил.

Поступила 03.03.2008 г.