

УДК 519.6

АЛГОРИТМ КОНТУР ДЛЯ НАХОЖДЕНИЯ И ВИЗУАЛИЗАЦИИ ПЛОСКИХ СЕЧЕНИЙ 3D-ОБЪЕКТОВ

С. В. Коломейко, С. Н. Мельникова, Д. В. Могиленских
(РФЯЦ-ВНИИТФ)

На основе формирования вспомогательной адаптивной сетки и вспомогательных скалярных полей предлагается алгоритм КОНТУР для генерации ориентированного контура плоского сечения трехмерного объекта, заданного комбинаторно.

Введение

Известны различные способы задания трехмерной геометрии, которые определяются классом решаемых задач [1]. В задачах, решаемых методом Монте-Карло (ММК), для описания форм материалов и сред применяется комбинаторная геометрия. Обычно определен класс поверхностей, например поверхности 1-го, 2-го, 3-го и 4-го порядка, из которых с помощью булевых операций строятся поверхности ограниченных объемов. Комбинаторный способ позволяет обеспечить быструю логическую локализацию частиц в ММК. Такой способ задания геометрии имеет очевидное преимущество — относительную краткость описания сложной геометрии. Однако существуют некоторые издержки такого задания с точки зрения визуального контроля геометрии, так как задача визуализации комбинаторной геометрии сложнее, чем, например, сеточных объектов.

В данной работе предлагается одно решение (алгоритм КОНТУР) для нахождения и визуализации контуров и внутренностей произвольных сечений объемов, заданных аналитически, на базе процедур, которые существуют в любой программе ММК.

Наряду с комбинаторным заданием трехмерной геометрии применяется и упрощенный способ, например параметрическое задание примитивов или поверхностей вращения. Визуализация таких упрощенных объектов является более легкой. Однако, когда из объектов создаются сложные комбинации с помощью булевых операций, задача визуализации значительно усложняется.

Приведем несколько вариантов комбинаторного задания геометрии. В программе ММК "Призма" [2, 3] используется класс поверхностей 1-го и 2-го порядка и поверхности вращения. В программе ММК MCNP [4] применяется класс поверхностей 1-го и 2-го порядка, тор, небольшой набор примитивов. В программе ММК Tart [5] используется класс поверхностей 1-го, 2-го и 3-го порядка, тор, поверхности вращения, примитивов нет. В программе ММК COG [6] используется класс поверхностей 1-го и 2-го порядка, поверхности вращения, большой набор примитивов.

Один способ задания произвольной геометрии трехмерного объекта

Рассмотрим один вариант комбинаторного описания трехмерной формы объемов, который изложен в работе [2]. Способ основывается на теоретико-множественном обосновании [7].

Рассмотрим трехмерный объем, ограниченный поверхностью S . Поверхность S описывается множеством поверхностей до 2-го порядка (П2П): $\{F_1, F_2, \dots, F_n\}$. Общее уравнение П2П: $A_{xx}x^2 + A_{yy}y^2 + A_{zz}z^2 + A_{xy}xy + A_{xz}xz + A_{yz}yz + A_x x + A_y y + A_z z + A = 0$.

Пусть A — произвольная точка пространства R^3 . Геометрия поверхности S задана однозначно, если: $\forall A \in R^3 \Rightarrow A \in S \vee A \notin S$. Однозначность описания поверхности S достигается применением булевых операций над множеством $\{F_1, F_2, \dots, F_n\}$. Булеву операцию обозначим, как \otimes . Можно применять разный набор булевых операций.

Рассмотрим широкий набор операций, т. е. значок \otimes будет означать одну из булевых операций:

- 1) \cup — объединение;
- 2) \cap — пересечение;
- 3) \setminus — дополнение;
- 4) $-$ — унарный минус.

В результате применения операций поверхность S будет задана однозначно: $S = F_1 \otimes F_2 \otimes \dots \otimes F_n$.

Формализуем факт, что точка $A(x_a, y_a, z_a) \in R^3$ принадлежит поверхности S . Важным является свойство знакоопределенности поверхностей:

Поверхность $F(X, Y, Z) = 0$ называется *знакоопределенной*, если $\forall A(x_a, y_a, z_a) \in R^3 \Rightarrow F(x_a, y_a, z_a) = 0 \vee F(x_a, y_a, z_a) < 0 \vee F(x_a, y_a, z_a) > 0$, то $F(X, Y, Z)$.

Введем понятие логического индекса $IND(F_i)$, который характеризует положение точки $A \in R^3$ относительно каждой поверхности множества $\{F_1, F_2, \dots, F_n\}$. Согласно теории множеств индекс может принимать значения 1 — истина или 0 — ложь. Согласно свойству знакоопределенности будем считать, что если $F_i(x_a, y_a, z_a) = 0 \vee F_i(x_a, y_a, z_a) < 0$, то $IND(F_i) = 0$, иначе $IND(F_i) = 1$. Тогда $A(x_a, y_a, z_a) \in R^3$: $A \in S$, если $IND(S) = IND(F_1) \otimes IND(F_2) \otimes \dots \otimes IND(F_n) = 1$, иначе $A \notin S$.

Постановка задачи

Пусть задана конечная топология объемов $S_m = \{S_i\}^*$, где $i \in [1, m]$, m — количество объемов. Пусть задана произвольная плоскость сечения P в R^3 : $A_P x + B_P y + C_P z + D_P = 0$.

Требуется найти ориентированные контуры (контур может быть не один, так как объемы не всегда выпуклые) сечения объемов из S_m плоскостью P и визуализировать сечения, а именно:

- прорисовать границы сечения;
- закрасить "внутренность" сечения.

Не ограничивая общности, будем рассматривать один объем S из топологии S_m .

Системы координат

Все объекты определены в декартовой базовой системе координат (БСК), т. е. все коэффициен-

*Здесь и всюду дальше термин *объем* S используется для краткого обозначения объема, ограниченного поверхностью S .

ты уравнений согласованы с БСК. Виртуальный экран, на котором будет формироваться изображение, согласуется с плоскостью P . С плоскостью P согласуется экранная система координат (ЭСК). Уравнение плоскости не определяет однозначно ЭСК, поэтому необходимо учитывать дополнительные условия. Также необходимо вычислить матрицу T преобразования координат из БСК в ЭСК и соответственно матрицу T^{-1} обратного преобразования.

Важные процедуры

При реализации различных алгоритмов локализации частиц в ММК, в том числе при визуализации, важное значение имеют две процедуры, которые обязательно присутствуют в любой программе ММК и которые применяются в алгоритме КОНТУР:

Процедура 1. Локализация произвольной точки из трехмерного пространства относительно топологии объемов S_m .

Процедура 2. Определение кратчайшего расстояния от точки до поверхности объема S из топологии S_m в заданном направлении.

Надо отметить, что эти процедуры различаются спецификой реализации, так как применяются для конкретной геометрии и ее структуры описания для конкретной программы ММК. Трудно определить универсальную реализацию процедур. Авторы применяют процедуры, которые разработаны математиками РФЯЦ-ВНИИТФ [8]. Реализация процедур основана на теоретико-множественном подходе с учетом топологии объемов S и с учетом свойства знакоопределенности.

Габариты сечения объекта

Необходимо кратко упомянуть одну из важных подзадач — определение описывающих габаритов потенциального сечения. Для успешной локализации контуров сечения в плоскости сечения нужно знать примерные габариты сечения. Трудность задачи заключается в сложности точного ее решения в автоматическом режиме. Для поиска габаритов можно воспользоваться процедурами 1 и 2. Допустим, габариты сечения объекта определены: X_{\max} , X_{\min} — по оси абсцисс, Y_{\max} , Y_{\min} — по оси ординат. Они определяют прямоугольную область на плоскости P , с которой сопоставляется виртуальный экран.

Алгоритм локализации и визуализации контура сечения (КОНТУР)

Нужно отметить, что наиболее очевидный способ визуализации — плотная трассировка (рендеринг) плоскости сечения, которая осуществляется с помощью вышеупомянутых процедур. Данный способ позволяет получить контуры сечений в виде набора пар точек пересечения лучей с поверхностью объема. Отметим недостатки этого способа:

1. Внутренность контура сечения не описывается полигонами, поэтому рисование границ контура происходит по точкам, а закрашка внутренности — с помощью прорисовки внутренних отрезков.
2. Возможен способ генерации ориентированного контура сечения после получения точек пересечения, однако при сложной геометрии (объекты с отверстиями, тонкие объекты, невыпуклые объекты) этот способ становится практически трудно реализуем.
3. Для получения плотной закрашки внутренностей сечений необходимо количество лучей сканирования, которое точно согласуется с разрешением экрана, а при увеличении изображения необходимо заново трассировать плоскость сечения, так как получаются пробелы между лучами сканирования.
4. При визуализации границы контура по точкам на экране или на бумаге качество изображения плохое — очевиден лестничный эффект, разная видимая толщина линий. При этом подходе невозможно применить векторную графику.
5. Практика показала еще один значимый недостаток, который выявился в трехмерной визуализации сечения функциями OpenGL — закрашка плоских фигур отрезками дает плохое качество. Особенно это заметно при интерактивном трехмерном повороте объекта вместе с сечением.

В алгоритме КОНТУР остается неизменным принцип трассировки, но меняются интерпретация и обработка результатов трассировки.

Основная идея. Плоскость сечения рассматривается как дискретная сеточная область, а объемы остаются заданными комбинаторно. В плоскости сечения в габаритной области строится регулярная вспомогательная сетка (ВС) размером $M_Grid \times N_Grid$, чтобы на ней получить специальное дискретное скалярное поле

при помощи трассировки, а затем с помощью алгоритма генерации изолиний восстановить ориентированный контур сечения границ объема. Пусть ВС является равномерной и прямоугольной. Пусть направляющие ВС параллельны соответственно координатным осям абсцисс и ординат ЭСК.

Вводится вспомогательная скалярная величина DAT , по которой строится дискретное скалярное поле на ВС. Матрица $DAT[M_Grid, N_Grid]$ служит для записи значений DAT , которые вычисляются в узлах ВС. Смысл величины DAT для каждого узла ВС — это определение факта его принадлежности внутренности сечения объема S : если $DAT[i, j] \geq 0$, то узел (i, j) принадлежит сечению объема S , если $DAT[i, j] < 0$, то узел (i, j) не принадлежит сечению объема S .

Формирование скалярного поля. Вычисление DAT осуществляется с помощью трассировки по направляющим ВС. Допустим, трассировка идет по горизонтальным направляющим, т. е. выпускаются лучи вдоль направляющих из узлов на границе сетки. Схема вычисления DAT и фрагмент ВС представлены на рис. 1.

Шаг 1. Рассматривается направляющая j . Анализируется точка начала луча сканирования $B = (X[0, j], Y[0, j])$. Применяется процедура 1. Если $B \in S$, то $DAT[0, j] = 1$, иначе $B \notin S$ и $DAT[0, j] = -1$.

Пусть $B \notin S$ и $DAT[0, j] = -1$.

Шаг 2. Выпускается луч из точки B в горизонтальном направлении, т. е. применяется процедура 2 с направляющим вектором V вдоль направляющей сетки. Если нет точек пересечения с границами объема S , то $DAT[i, j] = DAT[0, j]$ для $1 \leq i \leq M_Grid$ и осуществляется переход к следующей направляющей $j + 1$, т. е. на шаг 1. Если найдена точка пересечения $A(X_a, Y_a)$ с границей объема S , то переход на шаг 3.

Шаг 3. Допустим, что точка A лежит между узлами $(i - 1, j)$ и (i, j) . Основной момент: необходимо, чтобы при линейной интерполяции величины DAT в точку A из узлов $(i - 1, j)$ и (i, j) было $DAT(A) = 0$. Для этого нужно экстраполировать значение DAT в узел (i, j) вдоль горизонтального направления. Составим пропорцию: $DAT[i, j] = DAT[i - 1, j] * ((X[i, j] - X_a) / (X_a - X[i - 1, j]))$.

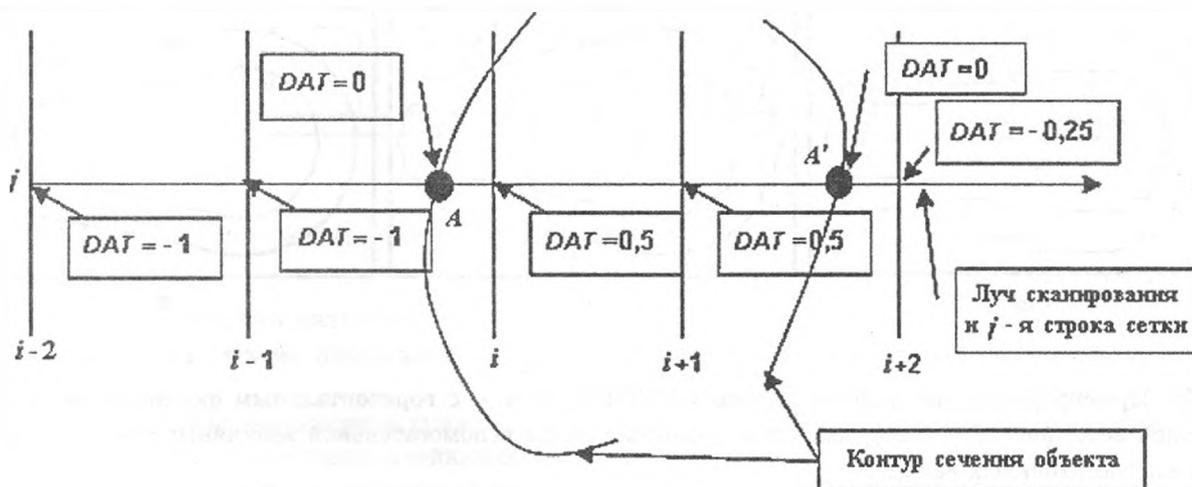


Рис. 1. Схема вычисления вспомогательной величины DAT

На рис. 1 представлено изменение величины DAT вдоль луча сканирования.

Шаг 4. Замена текущей точки: $B = A$. Применением процедуры 2 из точки B выпускается луч в том же направлении. Если нет точек пересечения с границей объема S , то $DAT[k, j] = DAT[i, j]$ для $i + 1 \leq k \leq M_Grid$ и переходим к следующему лучу $j + 1$, т. е. на шаг 1. Если найдена точка пересечения $A^*(X_a, Y_a)$, то допустим, что точка A^* лежит между узлами $(k - 1, j)$ и (k, j) , $DAT[n, j] = DAT[i, j]$ для $i + 1 \leq n \leq k - 1$ и переходим на шаг 4.

И так далее для каждой направляющей ВС вычисляются значения вспомогательной величины DAT в узлах ВС. В результате вычисляется специализированное дискретное скалярное поле. Для такого распределения величины DAT применяется любой корректный алгоритм последовательного построения изолинии на двумерной регулярной сетке для значения $DAT = 0$. Результатом работы алгоритма построения изолинии является восстановленный ориентированный контур сечения. Авторы модифицировали известный алгоритм прослеживания изолинии [9].

На рис. 2,а представлен пример результата работы алгоритма КОНТУР на ВС размером 50×25 с горизонтальной генерацией величины DAT , т. е. построенные изолинии — это и есть контур сечения. Из рисунка видно, что:

- 1) слева и справа гладкость контура адекватная, сверху и внизу — неадекватная;

- 2) необходимо дополнительно при генерации эмпирической величины DAT применить сканирование по вертикальным направляющим сетки.

Формирование скалярного поля на ВС по вертикальным направляющим. Вводится вторая эмпирическая величина $DAT1$, аналогичная величине DAT , но она вычисляется вдоль вертикальных направляющих ВС, т. е. повторяет DAT с точностью до перемены индексов узлов сетки. Пусть значения величины вычислены и записаны в массив $DAT1[M_Grid, N_Grid]$. Далее по данному скалярному полю строится изолиния для $DAT1 = 0$.

На рис. 2,б представлен пример результата работы алгоритма КОНТУР с вертикальным формированием величины $DAT1$, т. е. построенные изолинии — это и есть контур сечения. Из рисунка следует, что:

- 1) сверху и внизу гладкость контура адекватная, справа и слева — неадекватная;
- 2) при построении изолинии необходимо учитывать обе эмпирические величины.

Утверждение. Корректное объединение накопленной информации в массивах DAT и $DAT1$ в одном массиве невозможно из-за корреляции величин.

Интегральное формирование контура сечения. Предлагается в процессе вычисления изолинии по определенному правилу учитывать

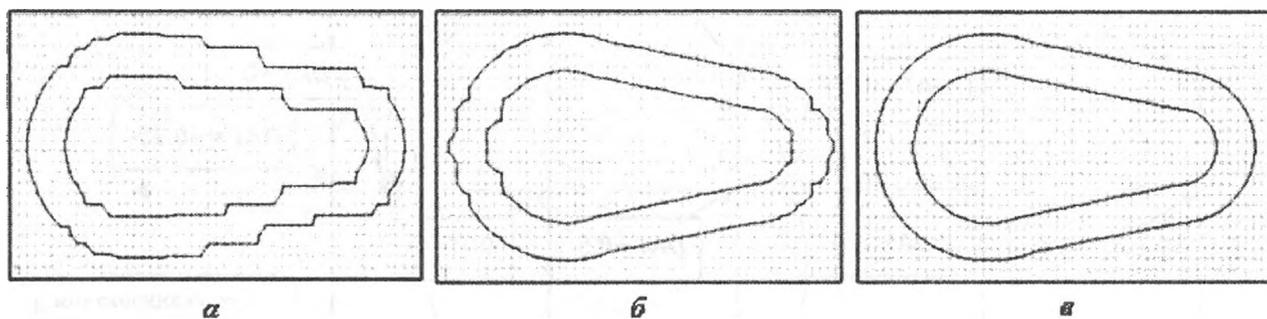


Рис. 2. Пример результата работы алгоритма КОНТУР: *a* — с горизонтальным формированием вспомогательной величины; *b* — с вертикальным формированием вспомогательной величины; *v* — с применением обеих вспомогательных величин

значения обоих скалярных полей. Рассмотрим это решение при реализации алгоритма вычисления изолинии. Построение изолинии ведется последовательно, от одной ячейки сетки к другой, соседней, ячейке. Анализируются точки пересечения изолинии с ребрами ячейки и направление выхода изолинии из ячейки. Идея алгоритма генерации изолинии остается без изменений.

Алгоритм модифицируется проверкой двух условий:

1. Если изолиния пересекает горизонтальное ребро ячейки, то координаты точки пересечения с ребром находятся линейной интерполяцией с использованием значений величины DAT в узлах соответствующей ячейки.
2. Если изолиния пересекает вертикальное ребро ячейки, то координаты точки пересечения с ребром находятся линейной интерполяцией с использованием значений величины $DAT1$ в узлах соответствующей ячейки.

На рис. 2, *v* представлен финальный результат работы алгоритма КОНТУР с горизонтальным и вертикальным вычислением значений величин DAT и $DAT1$ соответственно.

Таким образом, предложен алгоритм кусочно-линейной аппроксимации ориентированного контура сечения объема S , описанного комбинаторным способом. Алгоритм КОНТУР является алгоритмом второго порядка точности.

Можно добиваться практически любой гладкости и детализации линии контура путем увеличения размеров BC локально или глобально.

Однако алгоритм КОНТУР работает корректно, если на каждом ребре ячейки не более од-

ной точки пересечения с поверхностью объема. Ниже рассматривается модификация алгоритма КОНТУР (КОНТУР_М), которая позволяет снять данное ограничение.

Классификация ячеек BC

Для описания модификации алгоритма КОНТУР необходимо ввести некоторые понятия.

Определение 1. Если поверхность объема пересекает любое ребро ячейки BC более одного раза, то такую ячейку назовем *особой* ячейкой (ОЯ).

Замечание. Количество пересечений поверхности объема с ребрами ячейки четное, так как для каждой точки относительного входа контура сечения в ячейку должна быть одна точка относительного выхода из ячейки (предполагается, что объект всегда имеет толщину).

Определение 2. Ячейка называется *пустой*, если поверхность объема не пересекает ее ребер.

Определение 3. Ячейка называется *простой*, если поверхность объема пересекает ее ребра два раза и оба пересечения не принадлежат одному ребру.

На рис. 3, *a* приведены два типа простых ячеек, все другие возможные варианты простых ячеек приводятся к этим типам.

Определение 4. Ячейка называется *условно простой*, если поверхность объема пересекает ее ребра четыре раза и на одном ребре не больше двух пересечений.

На рис. 3, *b* приведены пять типов такого вида ячеек, все другие возможные варианты условно простых ячеек приводятся к этим типам.

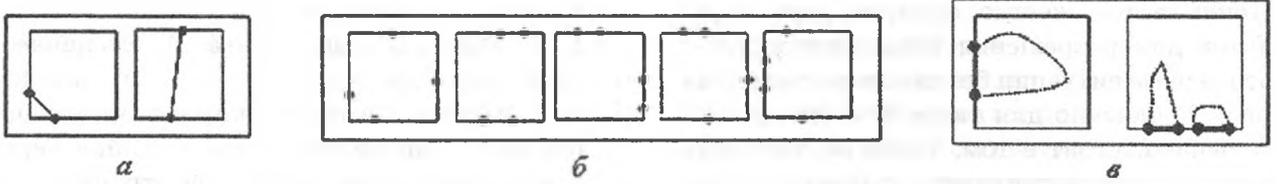


Рис. 3. Пример простых (а), условно простых (б) и граничных (в) ячеек

Определение 5. Ячейка называется *граничной*, если поверхность объема пересекает только одно ее ребро.

На рис. 3, в приведены два примера граничных ячеек. Можно считать, что в таких ячейках объем ограничивается. Пунктиром показаны подобные ситуации.

Определение 6. Ячейка называется *сложной*, если она не является пустой, простой, условно простой или граничной.

Определение 1*. Ячейка называется *особой*, если она не является пустой и простой.

Данная классификация позволяет упорядочить дальнейшие рассуждения.

Алгоритм КОНТУР_M

Модификации алгоритма КОНТУР снимают существующее ограничение (см. выше) и решают задачу нахождения ориентированного контура сечения в целом. Размеры ВС выбираются эмпирически, т. е. можно примерно оценить средние толщины объемов. При появлении ОЯ увеличение размеров ВС не является оптимальным способом разрешения ограничения, так как, во-первых, количество ОЯ обычно небольшое по отношению к неособым ячейкам, а во-вторых, ОЯ будут появляться в других местах и их количество будет возрастать.

Идея первой модификации заключается в выделении и обработке ОЯ, т. е. контур сечения уточняется внутри ОЯ. Множество ОЯ можно найти в процессе формирования скалярных полей *DAT* и *DAT1*. Допустим, идет процесс построения контура сечения от ячейки к ячейке и на каком-то шаге контур сечения входит в ОЯ.

Рассмотрим возможные способы обработки такой ситуации в зависимости от вида особенности.

1. Простые ячейки обрабатываются тривиально (алгоритм КОНТУР).
2. Условно простые ячейки однозначно не обрабатываются, но можно выработать прави-

ла прослеживания контура внутри ячейки или получить дополнительную информацию внутри ячейки и воспользоваться ею. В процессе прослеживания контура сечения в такие ячейки контур "приходит" два раза.

Правило. Контур сечения не должен взаимно пересекаться внутри ячейки.

На рис. 4 приведены пять возможных типов ситуаций, по три варианта конфигурации контура сечения для каждого типа условно простых ячеек.

Варианты с самопересечением аналитически обрабатываются и отбрасываются, но ситуация остается двужаночной. Для разрешения двужаночности предлагаются следующие варианты решения:

- A. Данной ячейке присваивается статус сложной, и она пересылается на обработку сложной ячейки.

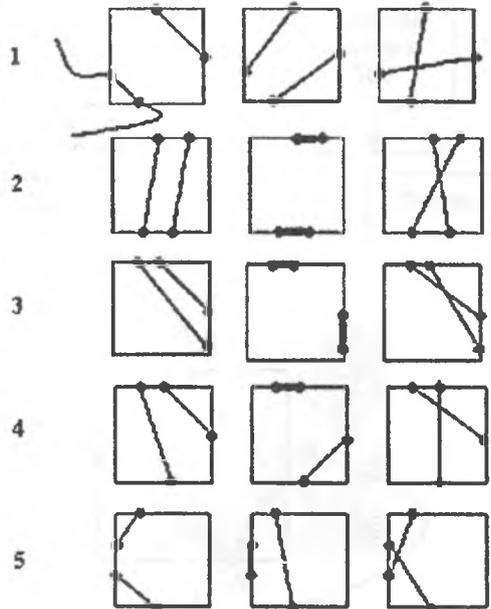


Рис. 4. Примеры условно простых ячеек и варианты конфигурации контура в них

- Б. Принимается жестко один из двух вариантов для разрешения неоднозначности — это первые ситуации без самопересечений на рис. 4. Обычно для типов 2—4 (см. рис. 4) решение состоит в том, чтобы не соединять точки, которые принадлежат одному ребру.
- В. Необходимо найти дополнительную информацию внутри ячейки с помощью поиска дополнительных точек контура сечения внутри ячейки.

Рассмотрим два способа нахождения дополнительной информации, например, для 1-го типа условно простых ячеек.

Способ 1. Рассмотрим рис. 5,а. Допустим, A — точка входа контура в ячейку. Эта точка потенциально может быть соединена с точками B и C . Введем дополнительный луч трассировки таким образом, чтобы он пересекал потенциальные отрезки. Применим процедуру 2. Допустим, найдена точка M пересечения луча с поверхностью объема. Тогда можно ввести эмпирический критерий, например: если точка M лежит левее от точки A по вертикали, то результирующие отрезки $|AB|$ и $|CD|$, иначе $|AC|$ и $|BD|$. Все методы дополнительной обработки внутри ячейки носят приближенный характер, но дают возможность более точно автоматически проследить контур сечения внутри ячейки.

Способ 2. Точка первого входа контура в ячейку, например точка A , соединяется с одной из точек на смежных ребрах, т. е. строится первый аппроксимирующий отрезок. При втором входе в ячейку через одну из двух оставшихся точек строится второй отрезок, отличный от первого. В случае неправильного соединения точек пересечения возникают ошибки двух типов, которые представлены на рис. 5,б и 5,в. Сплошной линией показано правильное соединение, пунктирной — ошибочное. Чтобы свести к минимуму подобные ошибки, применяются различные методы дополнительного анализа, такие как способ 1.

3. Граничную ячейку можно обрабатывать тремя способами:

- Предполагается, что в этой ячейке контур сечения закругляется, т. е. точки соединяются попарно (см. рис. 3,е).
- При необходимости более точной детализации контура сечения ячейке присваивается статус сложной.
- Обработка сложной ячейки.

Определение 7. Ячейки ВС назовем *базовыми* ячейками.

Идея второй модификации — это обработка сложной ячейки, которая заключается в ее би-

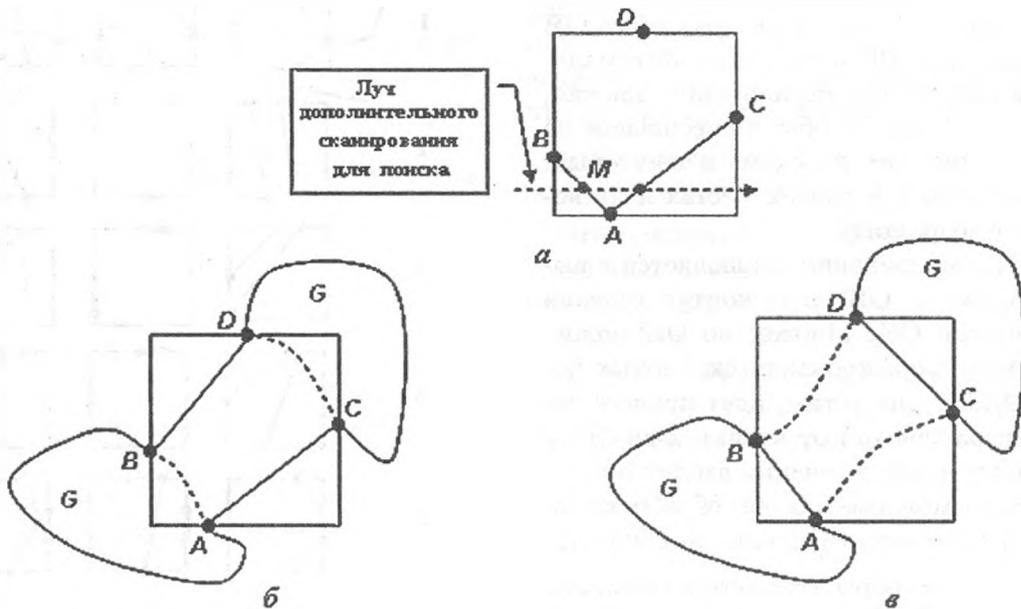


Рис. 5. Примеры неоднозначностей в условно простых ячейках: а — способ 1 разрешения неоднозначности; б, в — два типа ошибок при соединении точек пересечения в процессе применения способа 2

нарном делении на более мелкие дополнительные ячейки (ДЯ), т. е. внутри ячейки строится дополнительная адаптивная сетка, что позволяет детализировать контур сечения и разрешить неоднозначности, разделяя сложную ячейку неоднозначности на множество условно простых, простых и пустых ячеек. Деление ячейки осуществляется до тех пор, пока все ДЯ не перестанут быть сложными. ДЯ обрабатываются аналогично базовым. После образования новой ДЯ анализируется ее статус.

Можно применять бинарное деление ячеек. Для сложной ячейки такой алгоритм является единственным, обязательным при данном подходе, а для ячеек другого вида он применяется в случае необходимости. Авторы представляют бинарное деление ячейки.

Допустим, ячейка особая. Тогда выбирается одно из ее ребер, которое имеет несколько точек пересечения с поверхностью объема S , и между одной парой точек ставится дополнительная точка деления на ребре. На противоположном ребре вычисляется адекватная точка деления.

На рис. 6,а,б приведена схема бинарного деления сложной ячейки с шестью точками пересечения на ребрах. Пунктирными линиями на рис. 6,а выделен результирующий контур сечения внутри ячейки. На рис. 6,б представлена та же ячейка в составе фрагмента ВС и сечение объема S в целом, где выделена рассматриваемая ячейка.

Рассмотрим подробнее данный пример дополнительной обработки сложной ячейки. Новые ячейки, полученные бинарным делением, проу-

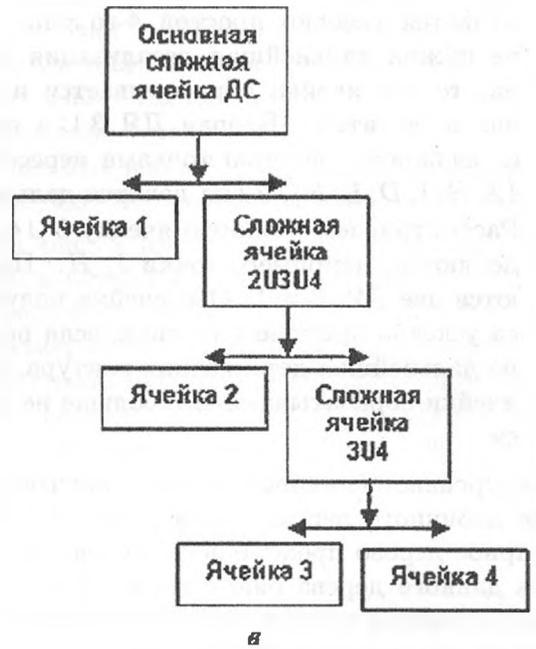
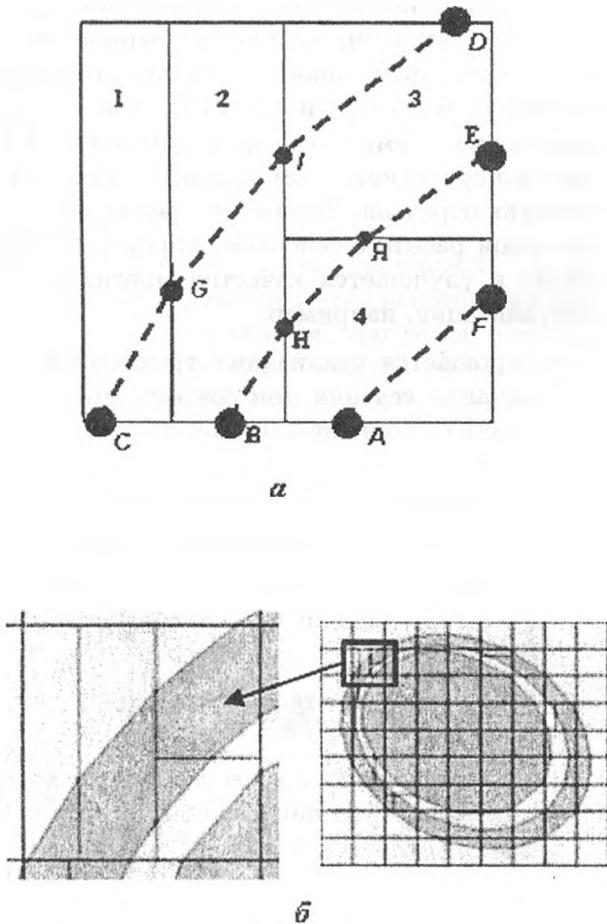


Рис. 6. Бинарное деление сложной ячейки: а — схема деления ячейки; б — визуализация ячейки в составе фрагмента ВС и сечения объема S в целом; в — описание рекурсивного бинарного деления в виде двоичного дерева; г — другая стратегия деления ячейки

мерованы на рис. 6,а. Схема деления ячейки, т. е. порядок получения пар ячеек, может быть разным. От выбора стратегии деления зависит скорость решения, количество дополнительных ячеек, форма контура. Приведем пример одной стратегии (см. рис. 6,а):

1. Сначала разделяются точки B, C . Получаются две ячейки: 1 и объединение $2 \cup 3 \cup 4$. На этом шаге еще нет объединения, но известен результат, и поэтому для наглядности вторую ячейку представим именно так. В результате ДЯ 1 получается простой; если не нужна дальнейшая детализация контура, то эта ячейка тривиально обрабатывается и больше не делится. Вторая ячейка остается сложной с шестью точками пересечения $\{A, B, G, D, E, F\}$, и она делится дальше.
2. Рассматриваем сложную ячейку $2 \cup 3 \cup 4$. Разделяются точки A, B . Получаются две ячейки: 2 и объединение $3 \cup 4$. ДЯ 2 получается условно простой 4-го типа; если не нужна дальнейшая детализация контура, то эта ячейка обрабатывается и больше не делится. Вторая ДЯ $3 \cup 4$ остается сложной с шестью точками пересечения $\{A, H, I, D, E, F\}$, и она делится дальше.
3. Рассматриваем сложную ячейку $3 \cup 4$. Разделяются, например, точки I, H . Получаются две ДЯ: 3 и 4. Обе ячейки получаются условно простые 1-го типа; если не нужна дальнейшая детализация контура, то эти ячейки обрабатываются и больше не делятся.

Рекурсивное бинарное деление описывается в виде двоичного дерева. Для данного примера бинарное дерево представлено на рис. 6,в. Глубина данного дерева равна трем. В результате контур сечения внутри данной сложной ячейки аппроксимируется семью отрезками.

На рис. 6,г представлена другая стратегия деления той же ячейки. В результате базовая ячейка делится на семь ДЯ и контур аппроксимируется одиннадцатью отрезками, что, естественно, увеличивает время работы алгоритма, но, с другой стороны, контур сечения получается более детальным.

На этом обработка сложных ячеек завершена, отрезки контура сечения внутри сложной ячейки найдены и могут быть включены в общий контур сечения.

Построение ориентированного контура сечения в целом. Заметим, что для прорисов-

ки контура сечения достаточно найти неупорядоченное множество отрезков и нарисовать их, но для записи контура в виде набора ориентированных полигонов нужна дополнительная обработка, которая связана с правильным соединением отрезков. Этот шаг лучше делать одновременно с алгоритмом прослеживания линии от ячейки к ячейке. В особой ячейке находятся отрезки, которые внутри особой ячейки аппроксимируют контур, и далее они соединяются с формирующимся контуром сечения.

Описание алгоритма КОНТУР_М завершено.

Закраска внутренности сечения

В результате работы алгоритма КОНТУР удастся сформировать кусочно-линейную аппроксимацию контуров сечений в виде множества замкнутых ориентированных многоугольников. Это позволяет закрасить внутренность сечения с помощью стандартных графических функций для многоугольников или же выполнить триангуляцию внутренности сечения, например с помощью библиотеки OpenGL, что позволяет уменьшить время закраски и улучшить ее качество по сравнению с закраской с помощью внутренних отрезков. Также становится более естественной работа с сечением, упрощается реализация и улучшается качество многих функций визуализации, например:

- упрощается реализация трехмерной визуализации сечения при совместном представлении с отсеченной частью трехмерного объема;
- увеличение части изображения становится более простой и быстрой операцией;
- при повороте объекта в плоскости сечения рисование границ и закраска внутренности осуществляется корректно;
- быстрее осуществляется экранное отсечение;
- есть возможность качественного и корректного рисования линий на векторных печатающих устройствах.

Кратко остановимся на вариантах решения задачи выбора цвета закраски, которая является отдельной и важной задачей научной визуализации. Авторы применяют следующие способы закраски объемов и их сечений:

1. *По номерам объемов.* Обычно применяется палитра цветов с резко отличающимися цветами, которая сопоставляется с номерами объемов. Если применять палитры с плавным изменением цветов, то будут плохо различимы соседние граничащие объемы, так как обычно их номера близки. Выделять объемы можно разными способами:

- цвета палитры формировать случайно, тогда она получается пестрой;
- применять палитру с плавным изменением цветов, тогда для каждого номера объема случайным образом выбирается цвет из палитры;
- "вручную" назначить цвет для объема, но обычно количество объемов большое и такая операция будет очень трудоемкой.

2. *По группам объемов.* Часто объемы объединяются в группы, и для группы можно назначить один цвет или разные оттенки одного цвета.

3. *По материалам.* Обычно каждый объем представляет определенный материал. Тогда осуществляется соответствие между палитрой и материалами, т. е. материалу назначается свой цвет.

4. *По физическим характеристикам.* Каждый объем характеризуется некоторыми физическими величинами, например плотностью. Тогда осуществляется соответствие между палитрой и значениями физической

величины. Обычно применяются палитры с плавно изменяющимися цветами, чтобы точнее отразить изменение физических величин.

Контур сечения выделяется линиями изменения цветов при закраске внутренности или же выделяется ярким цветом, чтобы более четко выделить объемы друг от друга.

Примеры в иллюстрациях

1. На рис. 7, *а–в* представлены трехмерные виды объекта, заданного контуром вращения. Это деталь реального нейтронного генератора. На рис. 7, *б* представлен объект и плоскость сечения, на рис. 7, *в* представлен объект с закрасленным сечением.
2. На рис. 8 представлена модель реальной установки нейтронного генератора и его окружения, части которого заданы комбинаторно. На рис. 8, *а* — трехмерное изображение установки. На рис. 8, *б* представлена отсеченная плоскостью установка, на рис. 8, *в* — *д* — разные сечения установки.
3. На рис. 9 представлены разные варианты ВС и уровни детализации. На рис. 9, *а* — базовая ВС размером 22×22 , на рис. 9, *в* — ВС размером 42×42 . На рис. 9, *б* и 9, *г* приведено сравнение двух результирующих контуров сечений. На рис. 9, *д* представлено сечение, сформированное с ВС 80×80 .

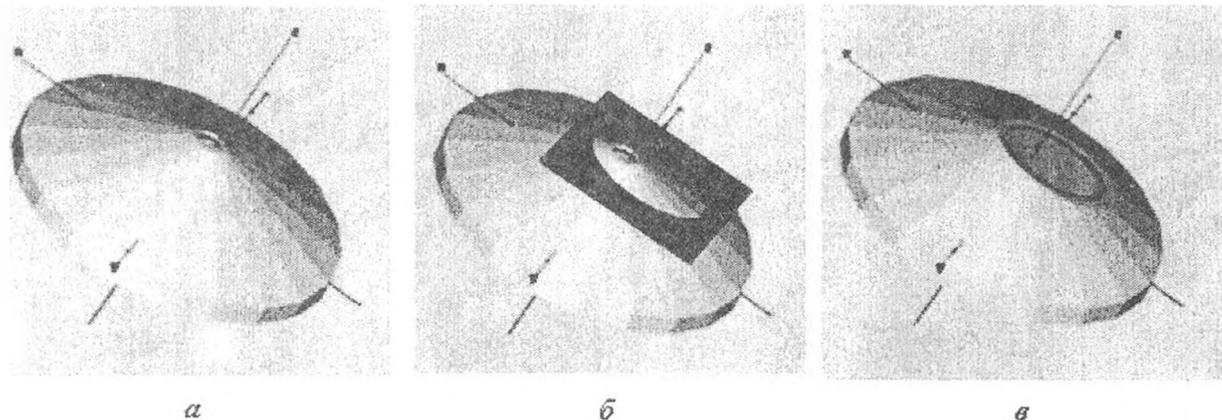


Рис. 7. Трехмерные изображения детали нейтронного генератора: *а* — общий вид детали; *б* — вид детали с плоскостью сечения; *в* — вид отсеченной детали

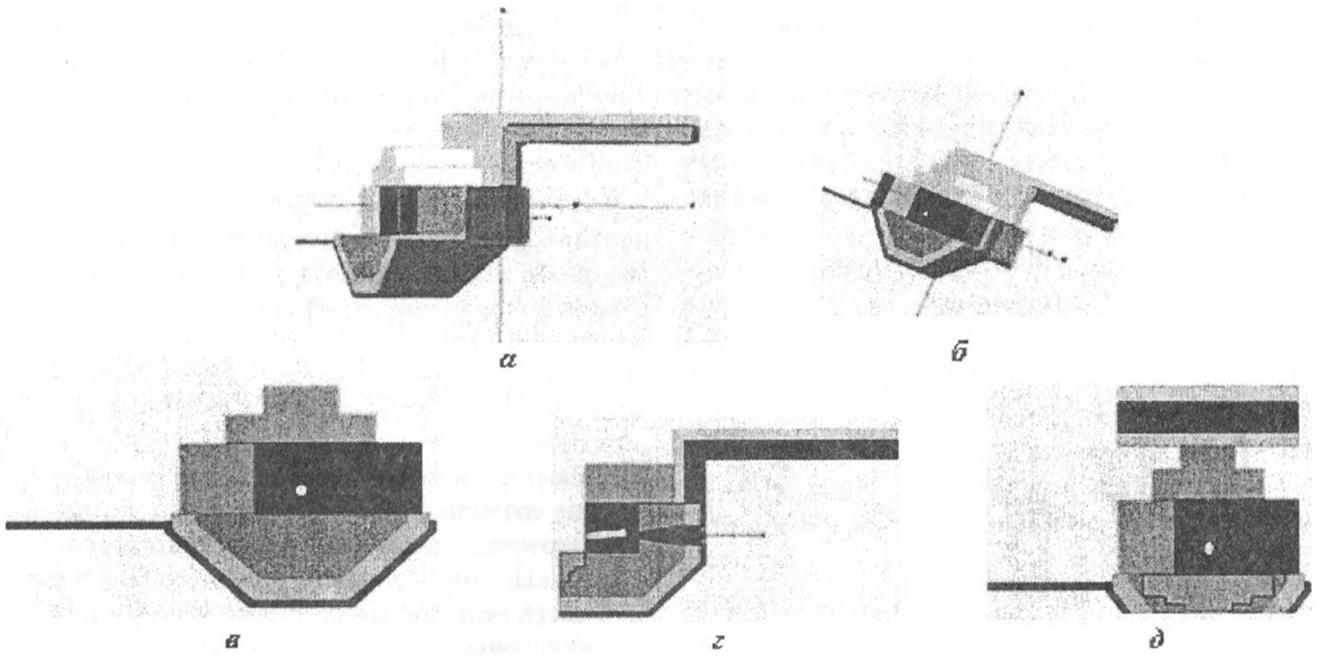


Рис. 8. Изображения модели установки нейтронного генератора и его сечений: *а* — трехмерное изображение установки; *б* — трехмерное изображение отсеченной установки; *в-д* — разные сечения установки

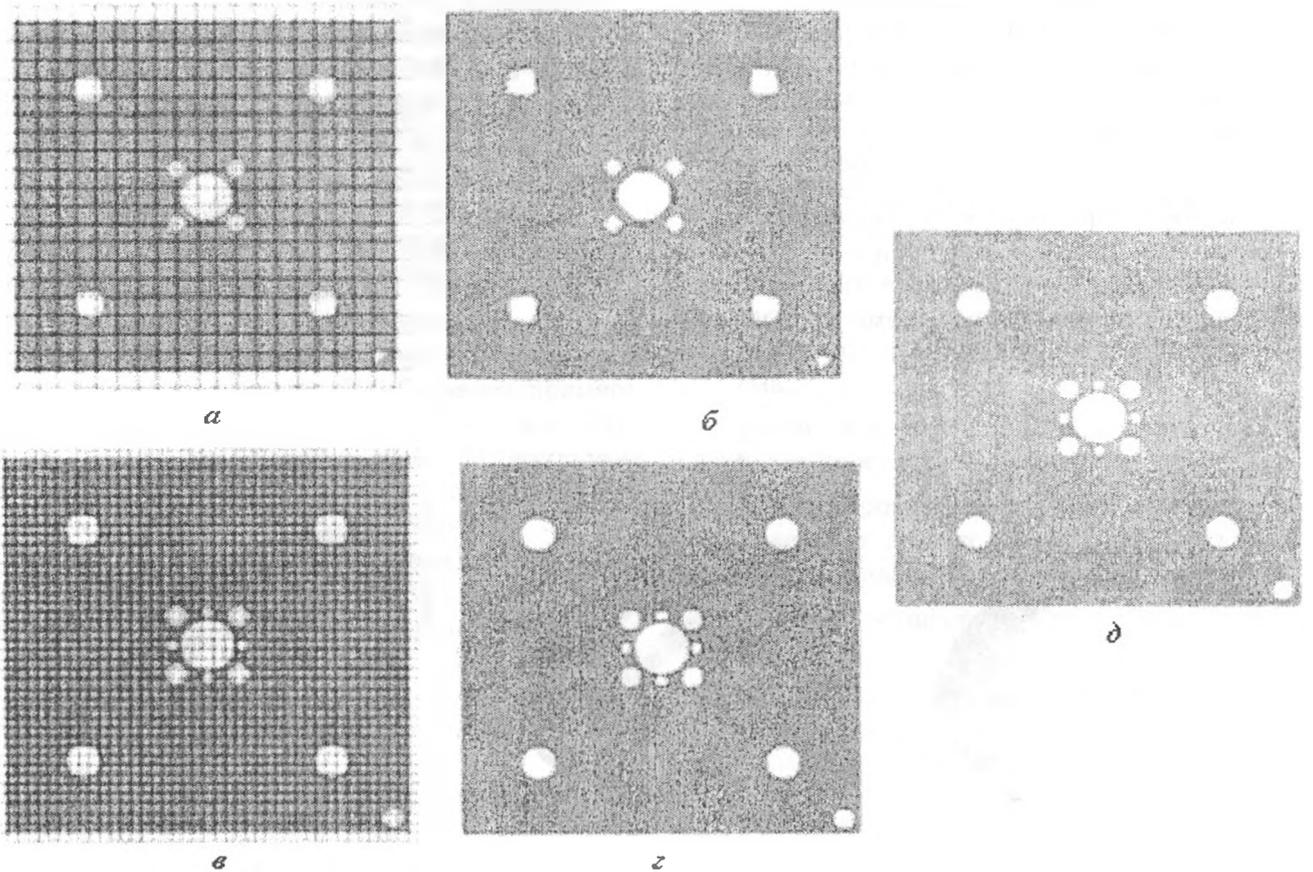


Рис. 9. Разные уровни детализации при формировании сечения с помощью ВС разных размеров: *а, б* — ВС размером 22×22 и результирующее сечение; *в, г* — ВС размером 42×42 и результирующее сечение; *д* — сечение, построенное с помощью ВС размером 80×80

Заключение

Цель данного алгоритма — автоматизировать обработку и анализ трехмерной комбинаторной геометрии. Но человеческий фактор всегда остается, например, для более точного определения габаритов сечения, их уточнения, выбора параметров уровня детализации, выбора размеров и формы ВС. Такое вмешательство является оправданным, так как пользователь является автором заданной геометрии и посвященным специалистом, что позволяет использовать его компетенцию для более качественного и быстрого решения задачи.

Вспомогательная регулярная сетка может быть с криволинейными направляющими, например радиально-концентрическая сетка, что часто улучшает результат и скорость.

В результате на основе формирования вспомогательной адаптивной сетки и вспомогательных скалярных величин предложен алгоритм КОНТУР для формирования кусочно-линейной аппроксимации ориентированных контуров плоского сечения трехмерного объекта, заданного комбинаторно.

Список литературы

1. Иванов В. П., Батраков А. С. и т. д. Трехмерная компьютерная графика. Радио и связь. Москва, 1995.

2. Адеев А. В., Адеева И. В., Павлова Н. А. О геометрическом обеспечении расчетов методом Монте-Карло в РФЯЦ-ВНИИТФ: Препринт № 160. Снежинск: РФЯЦ-ВНИИТФ, 1999.
3. Arnautova M. A., Kandiev Ya. Z., Lukhminsky B. E., Malyshkin G. N. Monte Carlo simulation in nuclear geophysics. Intercomparison of the PRIZMA Monte Carlo program and benchmark experiments // Nucl. Geophys. 1993. Vol. 7, No 3. P. 407–418.
4. Manual. MCNPTM-A General Monte Carlo N-Particle Transport Code. Version 4C. USA, LANL, 2000.
5. User's Manual TART2002. USA, LANL.
6. User's Manual COG second edition 1994. USA, LANL.
7. Куратовский К., Мостовский А. Теория множеств. М.: Мир, 1970.
8. Адеев А. В., Адеева И. В., Павлова Н. А. Теоретико-множественный подход для трехмерной геометрии в пакете программ МК на ЭЛЬБРУС-2 // VII Всесоюзное совещание "Методы Монте-Карло в вычислительной математике и математической физике". Тез. докл. Новосибирск: ВЦ СО АН СССР, 1991.
9. Баяковский Ю. М., Галактионов В. А., Михайлова Т. Н. Графор. Графическое расширение Фортрана. М.: Наука, 1985.