

**ПРОЕКТИРОВАНИЕ, ИЗГОТОВЛЕНИЕ И ВВОД
В ЭКСПЛУАТАЦИЮ ОБОРУДОВАНИЯ
ОБЪЕКТОВ АТОМНОЙ ОТРАСЛИ**
DESIGN, MANUFACTURE AND COMMISSIONING
COMMISSIONING OF EQUIPMENT
NUCLEAR INDUSTRY FACILITIES

<https://doi.org/10.26583/gns-2026-01-03>

УДК 004.94

EDN LNQMDP

Оригинальная статья / Original paper



**Реализация инструментов интерактивного скрывания полупространства
моделируемой области**

А.Э. Копейкин¹ ✉, М.Г. Кузнецов²

¹Саровский физико-технический институт – филиал НИЯУ МИФИ, г. Саров,
Нижегородская обл., Российская Федерация

²Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт
экспериментальной физики, г. Саров, Нижегородская обл., Российская Федерация

✉ copeikin.artem92@gmail.com

Аннотация. В данной статье рассматривается разработка и реализация инструментов интерактивного скрывания полупространства моделируемой области для систем проектирования. Актуальность исследования обусловлена необходимостью эффективного анализа внутренних структур сложных 3D-моделей на препроцессорном этапе. В качестве инструментов для реализации алгоритма интерактивного скрывания было выбрано использование языка программирования C++ в интеграции с библиотеками визуализации VTK и OpenGL, а также применение языка шейдеров GLSL. В работе подробно описана математическая модель и реализация интерактивного 3D-виджета, позволяющего пользователю в реальном времени управлять положением и ориентацией секущей плоскости посредством аффинных преобразований (переноса, вращения, масштабирования). Особое внимание в исследовании уделено реализации алгоритмов для двух типов данных. Для геометрических моделей разработан подход на базе вершинных и фрагментных шейдеров: вычисляется положение вершин относительно уравнения плоскости, выполняется отсечение элементов и отбрасываются только те, которые находятся над плоскостью, затем выполняется интерполяция по примитиву, а использование буфера трафарета (stencil buffer) обеспечивает корректное «закрашивание» сечения для сохранения иллюзии сплошного объекта. Для сеточных моделей реализован алгоритм топологической фильтрации, который физически отсекает ячейки и выполняет перестроение элементов, пересекаемых плоскостью. Разработанный функционал имеет высокую практическую значимость для ускорения процесса подготовки моделей. Результаты тестирования на ряде моделях демонстрируют высокую скорость работы алгоритма, корректность отображения срезов, эффективное использование ресурсов для своей задачи, а также высокую масштабируемость. Практическая значимость работы заключается в интеграции разработанных инструментов интерактивного скрывания в специализированное программное обеспечение, что позволяет инженерам-расчетчикам существенно ускорить процесс подготовки моделей, эффективно выявлять геометрические коллизии и задавать граничные условия как на поверхностных, так и на внутренних элементах.

Ключевые слова: интерактивное скрывание, плоскость отсечения, визуализация, VTK, сеточная модель, геометрическая модель, закрашивание сечения, виджет, GLSL, OpenGL

Для цитирования: Копейкин А.Э., Кузнецов М.Г. Реализация средств интерактивного скрывания полупространства моделируемой области. *Глобальная ядерная безопасность*. 2026;16(1):23–33. <https://doi.org/10.26583/gns-2026-01-03>

For citation: Kopeikin A.E., Kuznetsov M.G. Implementation of interactive means of concealing the half-space of the modeled area. *Nuclear Safety*. 2026;16(1):23–33. (In Russ.). <https://doi.org/10.26583/gns-2026-01-03>

Implementation of interactive tools of concealing the half-space of the modeled area

Artyom E. Kopeikin¹ ✉, Mikhail G. Kuznetsov²

¹ Sarov Institute of Physics and Technology – Branch of NRNU MEPhI, Sarov, Nizhny Novgorod region, Russian Federation

² Institute of Theoretical and Mathematical Physics, Russian Federal Nuclear Center - All-Russian Scientific Research Institute of Experimental Physics, Sarov, Nizhny Novgorod region, Russian Federation

✉ copeikin.artem92@gmail.com

Abstract. This paper discusses the development and implementation of tools for interactive hiding of the half-space of a modeled area for design systems. The relevance of this research comes from the need for effective analysis of the internal structures of complex 3D models at the pre-processing stage. The C++ programming language is chosen as the tool to implement the interactive occlusion algorithm, integrated with the VTK and OpenGL visualization libraries, as well as the GLSL shader language. The paper describes in detail the mathematical model and implementation of an interactive 3D widget that allows the user to control the position and orientation of the cutting plane in real time using affine transformations (translation, rotation, scaling). The study pays particular attention to the implementation of algorithms for two types of data. An approach based on vertex and fragment shaders has been developed for geometric models: the position of vertices relative to the plane equation is calculated, elements are clipped, and only those above the plane are discarded, then interpolation is performed on the primitive, and the use of a stencil buffer ensures correct “coloring” of the cross-section to preserve the illusion of a solid object. A topological filtering algorithm is implemented for mesh models that physically cuts off cells and rearranges elements intersected by the plane. The developed functionality is of high practical importance, as it is designed to speed up the model preparation process. Testing results on a number of models demonstrate the high speed of the algorithm, the correctness of the display of cross-sections, the efficient use of resources for its task, and high scalability. The practical significance of the work lies in the integration of the developed interactive occlusion tools into specialized software, which allows engineers to significantly speed up the model preparation process, effectively identify geometric collisions, and set boundary conditions on both surface and internal elements.

Keywords: interactive hiding, clipping plane, visualisation, VTK, mesh model, geometric model, section colouring, widget, GLSL, OpenGL

Введение

Современные научные, инженерные и производственные процессы невозможно представить без применения компьютерных технологий. С развитием вычислительных мощностей появилась возможность создания численных моделей различных объектов. Эти модели активно используются в системах, занимающихся проектированием (Computer Aided Design), расчетами (Computer Aided Engineering) и производством (Computer Aided Manufacturing) моделируемых объектов. Во всех этих системах необходимо точно или приближенно описать геометрическую форму объектов.

Особую значимость данные подходы приобретают в задачах атомной отрасли. Проектирование реакторных установок и их элементов (например, тепловыделяющих сборок, парогенераторов, насосного оборудования) сопряжено с созданием сложнейших геометрических моделей, содержащих тысячи скрытых деталей. Инструменты интерактивного скрытия позволяют инженеру-

конструктору визуально верифицировать внутреннюю компоновку оборудования и выявлять коллизии на ранних стадиях проектирования (при работе с CAD-геометрией).

В свою очередь, на этапе подготовки расчетных моделей важно корректно задавать начальные и граничные условия как на поверхностных элементах, так и на внутренних. Инструменты интерактивного скрытия позволяют инженеру-расчетчику быстро выделять необходимые внутренние зоны для задания различных нагрузок, креплений и т.д., что делает процесс подготовки расчетных моделей более эффективным.

Процесс решения задач обычно включает несколько этапов: препроцессорный (подготовительный), этап вычислений и постпроцессорный (обработка результатов)¹.

¹ Денисов М.А. Математическое моделирование тепловых процессов. ANSYS и CAE-проектирование: учебное пособие. – Екатеринбург: УрФУ, 2011. – 149 с. – Режим доступа: <https://elar.urfu.ru/handle/10995/41264> (дата обращения: 02.12.2025).

На препроцессорном этапе определяется тип расчета, строится или импортируется из других программ геометрическая модель объекта, выполняется конечно-элементное разбиение – строится сетка [1], задаются свойства материалов и прикладываются нагрузки, включая граничные условия, усилия и перемещения, а также вводятся исходные данные.

На этапе вычислений запускается расчетная программа, а результаты (например, перемещения, температуры, напряжения, деформации, скорости и тепловые потоки) записываются в специальные файлы.

На постпроцессорном этапе результаты визуализируются с помощью программного обеспечения, которое позволяет отображать данные одного шага решения, а также позволяет отслеживать изменения результатов во времени. Эти данные могут быть представлены графически или в виде текстового листинга, при этом возможно проведение дополнительных вычислений.

Одним из важных аспектов работы на препроцессорном этапе является использование интерактивных виджетов. Это небольшие приложения, которые позволяют пользователю работать с параметрами и настройками модели, улучшая процесс ее создания и повышая его эффективность.

В рамках представленной статьи будет рассмотрена разработка интерактивного виджета для скрытия полупространства моделируемой области с использованием технологий VTK (Visualization Toolkit) [2,3], OpenGL^{2,3}, GLSL на языке программирования C++.

Плоскости отсечения широко используются в технической визуализации для исследования сложных 3D-форм⁴. Устраняя пространственную преграду (occlusion), они

позволяют пользователю более эффективно просматривать внутренние или скрытые детали объектов. Этот механизм отсечения является быстрой и легко понятной техникой с простым взаимодействием. Он часто используется в системах рендеринга и API для реализации процесса растеризации, где важно обрезать объекты относительно плоскости отсечения. Однако для ряда приложений, таких как автоматизированное проектирование, необходима возможность дополнительной настройки плоскостей отсечения, управляемых пользователем. Кроме того, требуется высокая частота кадров для интерактивного изменения параметров плоскости в реальном времени, обеспечивая при этом высокое качество визуализации.

Актуальность работы обусловлена тем, что обрезка модели с помощью плоскости отсечения дает возможность эффективно исследовать внутренние структуры объекта. Для моделей, состоящих из нескольких деталей, желательно сохранить иллюзию сплошного объекта, что требует закрытия сечения модели после выполнения операции обрезки. Это особенно важно для систем проектирования, где требуется создание поперечных сечений массивных объектов, что позволяет исключить пространственные помехи и улучшить анализ внутренней структуры.

Цель – разработать и реализовать инструменты интерактивного скрытия полупространства моделируемой области, которые могут быть использованы на этапах подготовки геометрических и сеточных моделей, а также при задании начальных и граничных условий.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) реализовать интерактивный виджет для плоскости отсечения;
- 2) реализовать алгоритм скрытия полупространства для геометрической модели;
- 3) реализовать алгоритм скрытия полупространства для сеточной модели;
- 4) реализовать алгоритм закрашивания сечения.

Научная новизна заключается в разработке и оптимизации алгоритмов скрытия полупространства для геометрических и сеточ-

² Joey de Vries. Learn OpenGL. An offline transcript of learnopengl.com. – Kendall & Welling, 2020. – 523 p. – Режим доступа: <https://ia600102.us.archive.org/5/items/learnopengl/learnopengl.pdf> (дата обращения: 02.12.2025).

³ OpenGL red book (Russian version). – 2013. – 476 p. – Режим доступа: <http://www.cosmic-rays.ru/books61/RedBook.pdf> (дата обращения: 02.12.2025).

⁴ Голованов Н.Н. Геометрическое моделирование : учебник – Москва: КУРС : ИНФРА-М, 2016. – 400 с. Режим доступа: https://lib.yvsu.am/open_books/415823.pdf (дата обращения: 02.12.2025).

ных моделей, а также в интеграции этих алгоритмов с современными технологиями визуализации. Это позволяет повысить эффективность и гибкость взаимодействия с моделями в процессе подготовки и анализа в инженерных и научных приложениях.

Практическая значимость: разработанные средства позволят ускорить процесс подготовки сложных геометрических и сеточных моделей, а также процесс задания начальных и граничных условий.

В данной работе описывается процесс разработки и реализации инструментов скрытия полупространства моделируемой области. Приведены примеры использования реализованных средств.

Полупространство и скрытие

Полупространство, ограниченное гиперплоскостью α , – это геометрическая фигура в пространстве, для которой выполняется следующее:

1) фигура включает в себя плоскость α , но не сводится к ней (рис. 1а);

2) любой отрезок, ограниченный произвольными точками этой фигуры A и B , не принадлежащими α , не имеет пересечений с плоскостью α (рис. 1б);

3) любой отрезок, ограниченный произвольными точками этой фигуры A и B , где A принадлежит α , а B – нет, имеет пересечение с плоскостью α (рис. 1в).

Интерактивное скрытие полупространства основывается на применении геометрических операций и алгоритмов для определения видимых и скрытых частей трехмерных моделей.

Разработка интерактивного виджета для задания параметров плоскости

Для выполнения поставленных задач необходим интерактивный элемент, с помощью которого будут выполняться вышеописанные действия. Наилучшим образом подходит плоскость, которая будет изменять свои размеры, перемещаться и вращаться в пространстве при использовании ее пользователем.

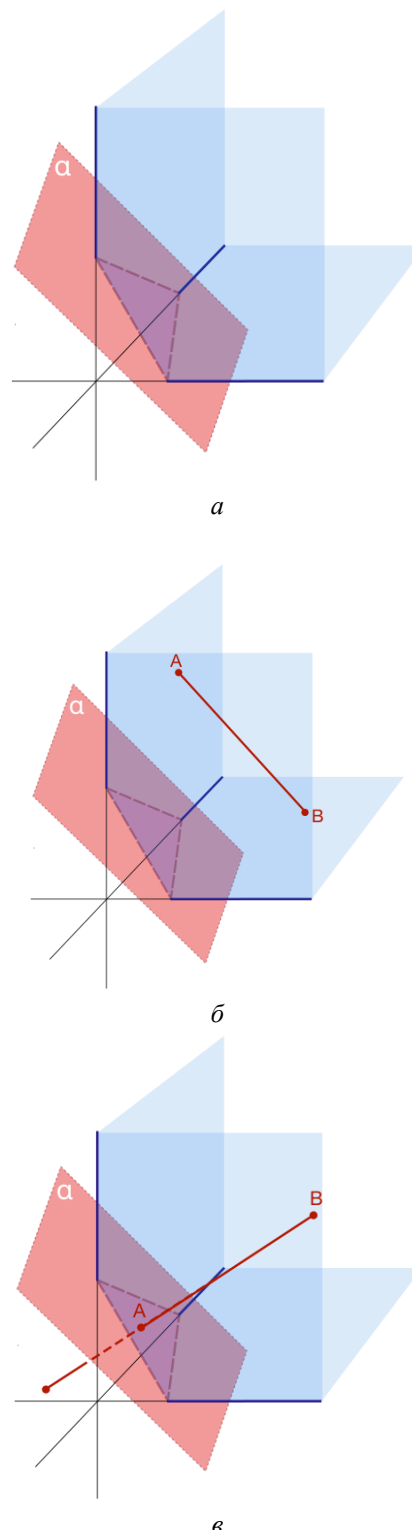


Рисунок 1. Полупространство: а – ограниченное гиперплоскостью; б – ограниченное гиперплоскостью, не включающей отрезок AB ; в – ограниченное гиперплоскостью, включающей точку A

Figure 1. Half-space: а – limited by a hyperplane; б – limited by a hyperplane excluding the segment AB ; в – limited by a hyperplane, including point A

Взаимодействие с виджетом требует применения некоторых математических концепций и формул⁵ [4]. Рассмотрим некоторые из них:

1. Перенос.

Если каждая точка $P(x, y, z)$ отображается на точку $P'(x', y', z')$ в соответствии с уравнениями (1):

$$\begin{cases} x' = x + a_1, \\ y' = y + a_2, \\ z' = z + a_3, \end{cases} \quad (1)$$

где a_1, a_2, a_3 – константы, то этот процесс называется переносом в трехмерном пространстве. Такой перенос может быть записан в матричной форме (2):

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} T, \quad (2)$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_1 & a_2 & a_3 & 1 \end{bmatrix}.$$

2. Масштабирование.

Для изменения размеров элемента можно использовать простые математические формулы.

Растяжение на k_x, k_y, k_z запишем в виде выражений (3):

$$\begin{cases} x' = k_x x, \\ y' = k_y y, \\ z' = k_z z. \end{cases} \quad (3)$$

В матричном виде (4):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (4)$$

Сжатие на k_x, k_y, k_z запишем в виде (5):

$$\begin{cases} x' = x / k_x, \\ y' = y / k_y, \\ z' = z / k_z. \end{cases} \quad (5)$$

В матричном виде (6):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1/k_x & 0 & 0 \\ 0 & 1/k_y & 0 \\ 0 & 0 & 1/k_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (6)$$

3. Поворот.

Для вращения элемента на заданный угол θ можно использовать матрицу поворота.

Любое вращение в трехмерном пространстве может быть представлено как композиция поворотов вокруг трех ортогональных осей (например, вокруг осей декартовых координат, как показано на рисунке 2). Этой композиции соответствует матрица, равная произведению соответствующих трех матриц поворота.

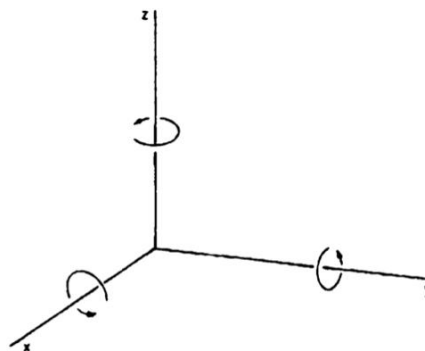


Рисунок 2. Вращение в положительном направлении вокруг координатных осей

Figure 2. Rotation in a positive direction around the coordinate axes

Для поворота вокруг оси Ox формула будет иметь следующий вид (7):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (7)$$

где (x, y, z) – исходные координаты элемента, (x', y', z') – координаты элемента после поворота.

Для поворота вокруг оси Oy будет иметь вид (8):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (8)$$

⁵ Порев В.Н. Компьютерная графика : учебное пособие. – СПб.: БХВ-Петербург, 2002. – 432 с. – Режим доступа: <https://lib.agu.site/upload/iblock/233/Computer%20graphics.pdf> (дата обращения: 02.12.2025).

Для поворота вокруг оси Oz будет иметь вид (9):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (9)$$

Все три поворота делаются независимо друг от друга, т.е. если надо повернуть вокруг осей Ox и Oy , вначале делается поворот вокруг оси Ox , потом применительно к полученной точке делается поворот вокруг оси Oy .

Положительным углам при этом соответствует вращение вектора против часовой стрелки в правой системе координат, и по часовой стрелке в левой системе координат, если смотреть против направления соответствующей оси.

Интерсекция модели и плоскости, ее математическая модель

Интерсекция математической модели и плоскости может быть выполнена с использованием уравнений, описывающих как саму плоскость, так и модель.

Для нахождения интерсекции воспользуемся уравнением плоскости, заданное в виде неявной функции (10):

$$Ax + By + Cz + D = 0, \quad (10)$$

где (A, B, C) – нормаль к плоскости;

(x, y, z) – координаты точек на плоскости;

D – коэффициент, расстояние от плоскости до начала координат.

Предположим, что у нас есть математическая модель, заданная уравнением (11):

$$F(x, y, z) = 0, \quad (11)$$

где F – функция, определяющая форму поверхности.

Интерсекция будет представлять собой точки, удовлетворяющие как уравнению математической модели, так и уравнению плоскости. Таким образом, мы ищем решения следующей системы уравнений (12):

$$\begin{cases} F(x, y, z) = 0, \\ Ax + By + Cz + D = 0. \end{cases} \quad (12)$$

Таким образом, общий принцип интерсекции заключается в нахождении точек, удовлетворяющих как уравнению математической модели, так и уравнению плоскости. Количество и характер этих точек зависит от конкретных уравнений модели и плоскости.

Теперь рассмотрим замечание, которое необходимо для нашей задачи. Для того чтобы определить какие части математической модели нужно скрывать, необходимо учитывать:

если $Ax + By + Cz + D > 0$, то точка находится по одну сторону плоскости;

если $Ax + By + Cz + D < 0$, то точка находится по другую сторону плоскости;

если $Ax + By + Cz + D = 0$, то точка лежит на плоскости.

В общем случае, если у нас есть модель с уравнением, то мы можем использовать значения координат точек на модели в уравнении плоскости, чтобы определить, по какую сторону от плоскости они находятся.

GLSL для скрывания полупространства

Язык программирования шейдеров OpenGL Shading Language (GLSL) предоставляет уникальные возможности для эффективной работы с графическими процессорами (GPU). Благодаря GLSL, разработчики могут использовать мощь современных GPU для сложных вычислений и визуализации. Один из шагов в использовании GLSL – создание программы на основе последней версии OpenGL API, которая интегрируется с библиотекой VTK для визуализации.

Любая программа, использующая OpenGL API, включает одну или несколько программ на языке GLSL, которые называются шейдерами. Шейдеры выполняются на GPU и проходят через различные этапы в общем конвейере OpenGL. Каждый шейдер имеет свою задачу, и в зависимости от этапа он может обрабатывать вершины, пиксели или выполнять другие вычисления [5]. На рисунке 3 приведены все типы шейдеров, доступных в OpenGL.

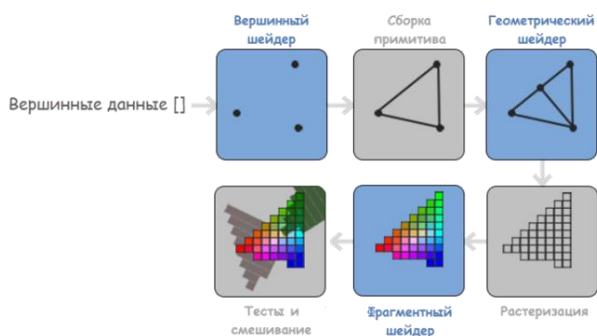


Рисунок 3. Вершинный и фрагментный шейдеры
Figure 3. Vertex and fragment shaders

Вершинные шейдеры выполняют манипуляции с координатами в 3D-пространстве и вызываются один раз для каждой вершины. Главной задачей вершинного шейдера является настройка переменной *gl_Position*, которая является встроенной глобальной переменной GLSL, используемой для хранения положения текущей вершины.

Шейдеры фрагментов (или текстур) определяют цвета RGBA (красный, зеленый, синий, альфа) для каждого обрабатываемого пикселя. Один шейдер фрагментов вызывается один раз на пиксель. Основная цель фрагментного шейдера – настройка переменной *gl_FragColor*, которая является встроенной переменной GLSL, аналогичной *gl_Position*.

Между вершинным и фрагментным шейдерами существует дополнительный этап, называемый геометрическим шейдером. Этот шейдер принимает на вход набор вершин, представляющих собой единый примитив, такой как точка или треугольник. Геометрический шейдер может трансформировать эти вершины и передать их на следующий этап. Особенность геометрического шейдера заключается в том, что он способен преобразовывать исходный примитив в другие примитивы, возможно, генерируя больше вершин, чем было задано изначально.

Для получения значимых результатов необходимо как-то получить вывод с предыдущего этапа шейдера. GLSL предоставляет встроенную переменную *gl_in*, которая дает доступ к данным предыдущего этапа. Одним из важных элементов является переменная *gl_ClipDistance*, играющая ключевую роль

при пересечении геометрической модели с плоскостью.

Программная реализация интерактивного виджета плоскости

Интерактивный виджет плоскости будет иметь вид, как показано на рисунке 4.

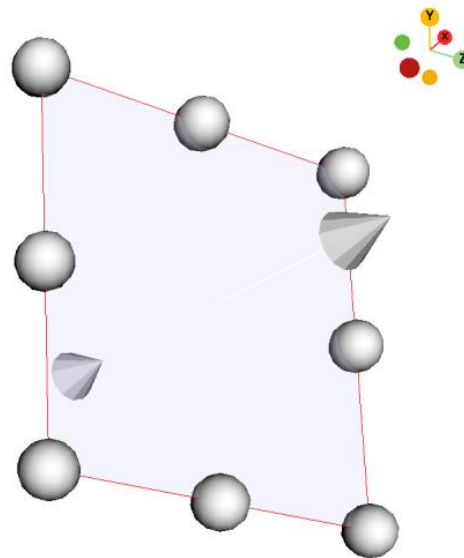


Рисунок 4. Отображение виджета на экране
Figure 4. Widget display on a screen

Управление виджетом происходит с помощью специальных объектов, выделенных серым цветом. Рассмотрим изменение размера, вращение и перемещения плоскости по отдельности.

1. Вращение плоскости будет происходить удержанием левой кнопки мыши нормали, как показано на рисунке 5а.

2. Перемещение плоскости будет происходить удержанием левой кнопки мыши самой плоскости вдоль нормали, как показано на рисунке 5б.

3. Изменение размером будет происходить с использованием, так называемых точек или сфер, которые изображены на плоскости. Каждая из этих сфер имеет свои функции, которые содержат алгоритм изменения плоскости.

Основная концепция изменения размеров плоскости заключается в том, что 3 точки, играющие основу плоскости, должны сохранять свои позиции. То есть сама плоскость не должна перемещаться при использовании

интерактивных элементов. В таком случае ставятся две задачи:

1) описать плоскость так, чтобы при взаимодействии с центральными сферами каждой стороны она изменяла свой размер только относительно двух ортогональных сторон, приложенных к данной. То есть сторона, на которой находится сфера, и параллельная ей не должны менять своего размера, а также параллельная сторона должна сохранять свое положение относительно движущейся стороны;

2) описать плоскость так, чтобы при взаимодействии с угловыми сферами она не меняла своего положения относительно угла противоположного данному. То есть при взаимодействии с угловой сферой плоскость должна менять размер всех своих сторон, но должна сохранять положение точки, находящейся на противоположном углу от выбранной интерактивной сферы.

На рисунке 5в показаны эти две задачи.

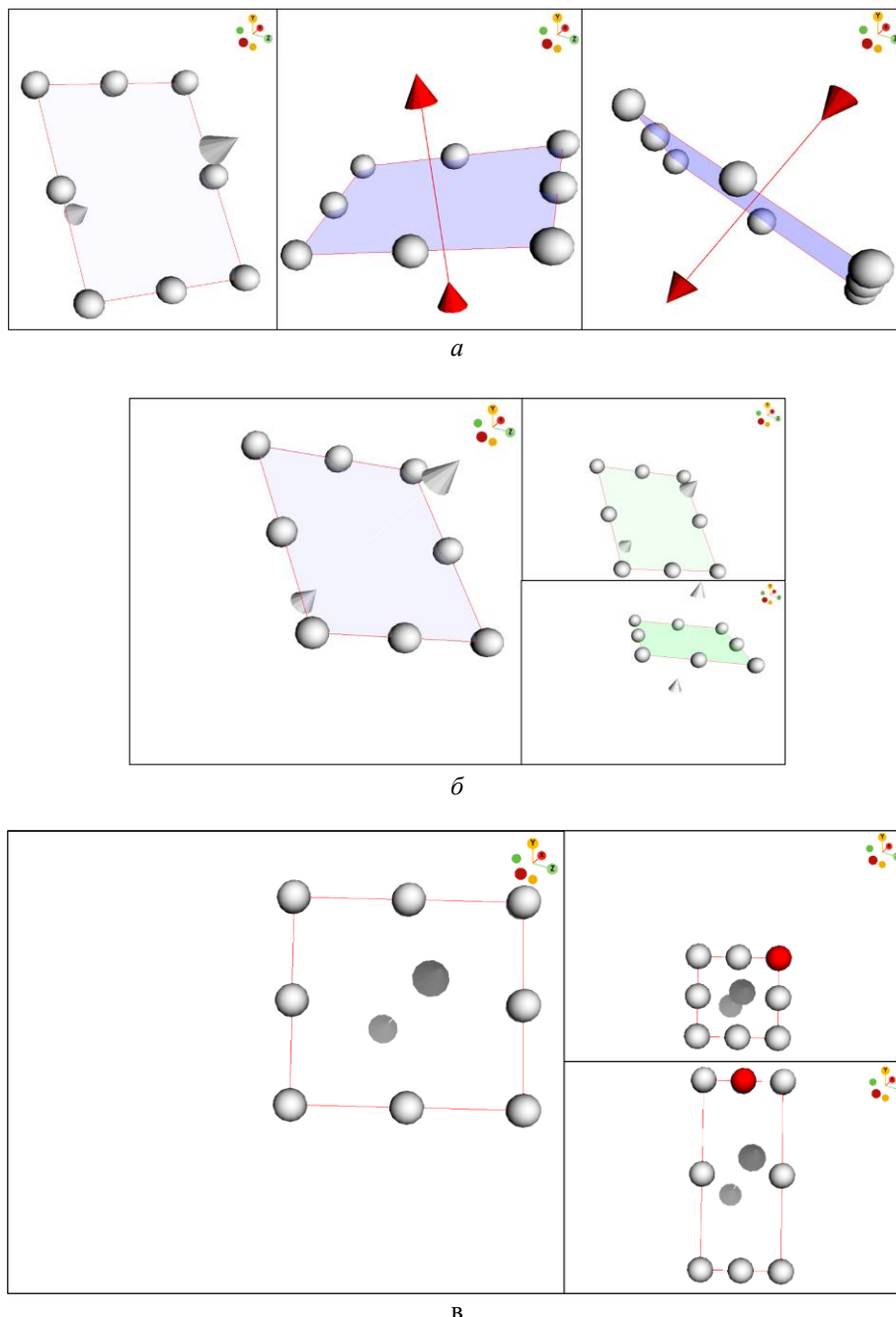


Рисунок 5. Вращение плоскости – *a*; перемещение плоскости – *б*; изменение размеров плоскости – *в*
Figure 5. Plane rotation – *a*; Plane movement – *б*; Plane resizing – *в*

Программная реализация скрытия сеточных моделей

Для реализации программной части скрытия сеточных моделей создадим такой класс, который будет представлять собой фильтр для обрезки полигональных данных с использованием плоскости разреза. Принцип работы этого фильтра будет заключаться в том, чтобы отсечь часть полигональных данных, находящуюся с одной стороны плоскости разреза, и перестроить данные, находящиеся на пересечении с плоскостью.

Рассмотрим алгоритм работы фильтра подробнее:

1. Фильтр требует определения плоскости разреза в виде неявной функции. Плоскость задается нормалью (вектором, указывающим направление плоскости) и точкой, через которую проходит плоскость.



Рисунок 6. Скрытие полупространства: *a* – первой сеточной модели; *б* – второй сеточной модели
Figure 6. Half-space hiding: *a* – the first mesh model; *б* – the second mesh model

Программная реализация скрытия геометрических моделей

Переменная *gl_ClipDistance* обеспечивает прямой совместимый механизм для управления пользовательским отсечением. Элемент *gl_ClipDistance[i]* определяет расстояние отсечения для каждой пользовательской плоскости отсечения *i*. Расстояние 0.0 означает, что вершина находится на плоскости, положительное расстояние означает, что вершина находится внутри плоскости отсечения, а отрицательное расстояние означает, что точка находится вне плоскости отсечения. Расстояния отсечения будут линейно интерполированы по примитиву, а часть примитива с интерполированными расстояниями менее 0.0 будет обрезана.

2. Устанавливается входной объект (полигональные данные) и параметры плоскости разреза.

3. Выполняем проверку на пересечение. Если его нет и все ячейки над плоскостью, то отбрасываем все данные. Если под плоскостью, то оставляем все ячейки. Если пересечение есть, то оставляем ячейки, находящиеся под плоскостью, целыми. Данные над плоскостью отбрасываем. А для ячеек, которые пересекают плоскость, находим их пересечение и перестраиваем.

4. Обрезанный результат может быть визуализирован или использован в дальнейшем анализе.

Для демонстрации работы алгоритма использованы модели, полученные из доступных интернет-источников. На рисунке 6 показан результат выполнения алгоритма.

Однако алгоритм скрытия полупространства для геометрических моделей будет работать некорректно на всех устройствах. Так как при выполнении функции *gl_ClipDistance* неправильно обрабатывается драйвер. Из-за него нарушается отображение.

В связи с этим была поставлена задача написания своей функции в шейдере для скрытия полупространства. Алгоритм будет иметь следующий вид:

1. Будем передавать в вершинный шейдер параметры плоскости: направление нормали и центр плоскости.

2. В шейдере известны координаты вершин *vertexP*, поэтому найдем вектор от вершины до плоскости (13):

$$to0 = plane0 - vertexP. \quad (13)$$

3. Затем вычислим расстояние, то есть найдем скалярное произведение вектора и нормали (14):

$$\text{dot} = (\text{toO}, \text{planeN}). \quad (14)$$

4. Передадим dot в фрагментный шейдер. Определяем по какую сторону от плоскости находится вершина, для это выполним проверку скалярного произведения:

$\text{dot} < 0$, то вершина за плоскостью;

$\text{dot} > 0$, то вершина перед плоскостью;

$\text{dot} = 0$, то вершина на плоскости.

5. Так как вершины, находящиеся за плоскостью нам не нужны, то будем отбрасывать их. Для этого существует функция отбрасывания discard . Команда discard используется во фрагментном шейдере для указания, что текущий фрагмент (пиксель) не должен быть записан в буфер кадра

(framebuffer). Как только команда discard выполняется, шейдер прекращает обработку текущего фрагмента. Значение, рассчитанное для gl_FragColor или других выходных переменных, игнорируется.

Для закрашивания сечения будем использовать буфер трафарета. Буфер трафарета обеспечивает простой способ добавления шапки к объектам, которые пересекаются плоскостью отсечения. Ограничивающий полигон встраивается в плоскость отсечения, а буфер трафарета используется для обрезки полигона до внутренней части твердого тела.

Таким образом, выполняется скрытие полупространства для геометрических моделей, как показано на рисунке 7. Модели также были получены из доступных интернет-источников.

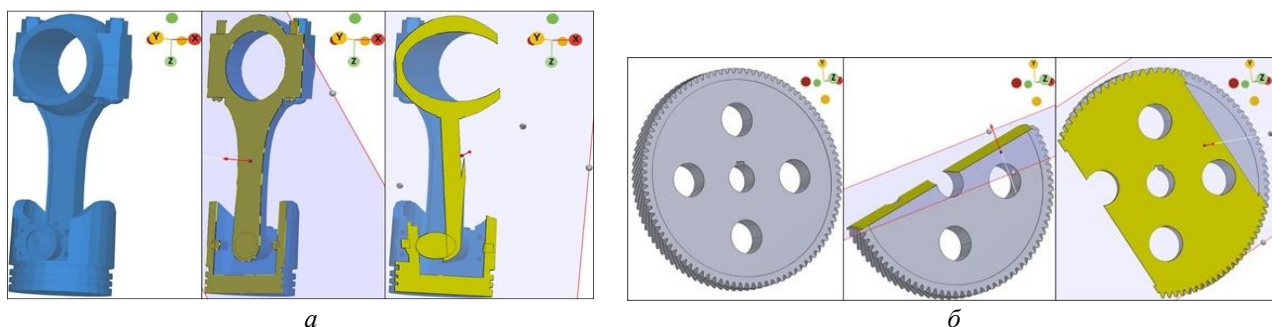


Рисунок 7. Скрытие полупространства: a – первой геометрической модели; b – второй геометрической модели

Figure 7. Half-space hiding: a – the first geometric model; b – the second geometric model half-space hiding

Заключение

В ходе работы было выполнено следующее:

- реализация виджета плоскости для интерактивного скрытия полупространства;

- разработаны и реализованы алгоритмы интерактивного скрытия полупространства моделируемой области для геометрических и сеточных моделей, а также закрашивание сечения;

- продемонстрирована работоспособность реализованных алгоритмов на тестовых моделях.

Таким образом, поставленная задача по реализации инструментов интерактивного скрытия полупространства моделируемой области была выполнена.

Практическая значимость результатов работы заключается в возможности интеграции инструментов интерактивного скрытия в специализированное программное обеспечение, используемое на предприятиях атомной отрасли. Внедрение разработанного виджета и алгоритмов позволит инженерам-конструкторам проводить визуальную верификацию внутренней структуры изделий, а инженерам-расчетчикам – корректно задавать начальные и граничные условия (нагрузки, крепления и т.д.) как на поверхностных, так и на внутренних элементах. Использование данных инструментов интерактивного скрытия существенно повышает эффективность процесса подготовки расчетных моделей и снижает риск возникновения ошибок.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Лисейкин В.Д. Технология построения разностных сеток: Монография. Новосибирск: Наука, 2014. 208 с. Режим доступа: <https://search.rsl.ru/ru/record/01007499394> (дата обращения: 12.11.2025).
Liseykin V.D. Technology of building difference schemes: monograph. Novosibirsk: Nauka, 2009. 414 p. (in Russ.). Available at: <https://search.rsl.ru/ru/record/01007499394> (accessed: 12.11.2025).
2. Avila L.S. The VTK user's guide: 11th edition. Kitware, 2010. 536 с. Available at: <https://vtk.org/wp-content/uploads/2021/08/VTKUsersGuide.pdf> (accessed: 12.11.2025).
3. Schroeder W., Martin K., Lorensen B. The visualization toolkit an object-oriented approach to 3D graphics. Edition 4.1. Kitware, 2018. 541 с. Available at: <https://gitlab.kitware.com/vtk/textbook/raw/master/VTKBook/VTKTextBook.pdf> (accessed: 12.11.2025).
4. Аммерал Л. Принципы программирования в машинной графике. Монография. Москва: Сол Систем, 1992. 224 с. Режим доступа: <https://djuv.online/file/CcaeTUO2bjGaa?ysclid=mkwclmw8es933611894> (дата обращения: 12.11.2025).
Ammeral L. Programming principles in computer graphics. Monograph. Moscow: Sol System, 1992. 224 p. (in Russ.). Available at: <https://djuv.online/file/CcaeTUO2bjGaa?ysclid=mkwclmw8es933611894> (accessed: 12.11.2025).
5. Вольф Д. Open GL 4. Язык шейдеров. Книга рецептов. Пер. с англ. А.Н. Киселева. Москва: ДМК Пресс, 2015. 368 с. Режим доступа: https://download.blackball.lv/data/library/OpenGL_4_Jazyk_shejderov_Kniga_receptov_%282013%29.pdf (дата обращения: 12.11.2025).
Wolf D. OpenGL 4. Shader language. Cookbook, translated from English by A.N. Kiselev. Moscow: DMK Press, 2015. 368 p. (in Russ.). Available at: https://download.blackball.lv/data/library/OpenGL_4_Jazyk_shejderov_Kniga_receptov_%282013%29.pdf (accessed: 02.11.2025).

ВКЛАД АВТОРОВ:

Копейкин А.Э. – изучение теоретических источников по проблеме рассматриваемой проблеме, разработка и реализация программы, написание текста статьи;
Кузнецов М.Г. – предоставление материала по исследуемой проблематике, внесение корректив и уточнений в работу, консультирование, редактирование текста.

AUTHORS' CONTRIBUTION:

Kopeikin A.E. – studying theoretical sources on the subject matter, developing and implementing the program, writing a manuscript;
Kuznetsov M.G. – providing material on the subject under investigation, making corrections and clarifications to the work, supervising, editing the text.

ИСТОЧНИКИ ФИНАНСИРОВАНИЯ

Авторы заявляют об отсутствии внешних источников финансирования.

FUNDING:

The authors state that there are no sources of external funding.

КОНФЛИКТ ИНТЕРЕСОВ:

Авторы заявляют об отсутствии конфликта интересов.

CONFLICT OF INTEREST:

No conflict of interest.

ИНФОРМАЦИЯ ОБ АВТОРАХ:

Артем Эдуардович Копейкин, магистр, Саровский физико-технический институт – филиал НИЯУ МИФИ, г. Саров, Нижегородская обл., Российская Федерация.
e-mail: copeikin.artem92@gmail.com

INFORMATION ABOUT THE AUTHORS:

Artyom E. Kopeikin, Master, Sarov Institute of Physics and Technology – Branch of NRNU MEPHI, Sarov, Nizhny Novgorod region, Russian Federation.
e-mail: copeikin.artem92@gmail.com

Михаил Геннадьевич Кузнецов, начальник научно-исследовательского отдела, Институт теоретической и математической физики, Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт экспериментальной физики, г. Саров, Нижегородская обл., Российская Федерация.
e-mail: mihansarov@bk.ru

Mikhail G. Kuznetsov, Head of the Research Department Institute of Theoretical and Mathematical Physics, Russian Federal Nuclear Center – All-Russian Scientific Research Institute of Experimental Physics, Sarov, Nizhny Novgorod region, Russian Federation.
e-mail: mihansarov@bk.ru

Поступила в редакцию / Received 01.12.2025

После доработки / Revision 12.02.2026

Принята к публикации / Accepted 17.02.2026