Модуль 2 «Компьютерная графика»
Лекция 10 «Визуализация наборов данных»

к.ф.-м.н., доц. каф. ФН-11, Захаров Андрей Алексеевич,
ауд.: 930а(УЛК)
моб.: 8-910-461-70-04,
email: azaharov@bmstu.ru

МГТУ им. Н.Э. Баумана

2 июня 2021 г.

Введение

Визуализация научных данных позволяет преобразовать числовые данные в графические образы посредством технологии компьютерной графики. Графическое представление данных позволяет наглядно представить огромные массивы числовой информации, выявить в этих массивах наиболее интересные фрагменты и сосредоточить именно на них дальнейшие исследования. В научных приложениях данные могут поступать из множества источников — это могут быть результаты измерений, математического моделирования, имитационного моделирования, датчики спутников и космических кораблей, радиотелескопов, медицинских сканеров и т.д.
В зависимости от типа величины различают три вида задач:

скалярная визуализация — исследуемая характеристика представляет собой скалярную величину, например плотность излучения;

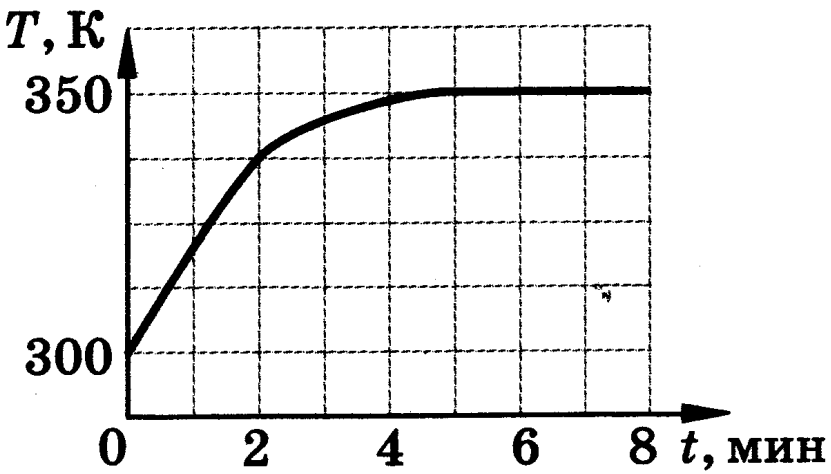
векторная визуализация — исследуемая характеристика представляет собой векторную величину, например скорость течения жидкости в каждой точке потока;

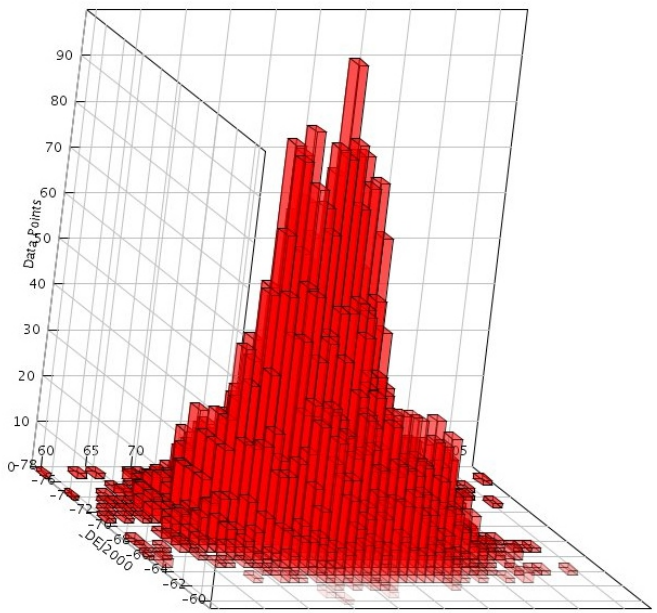
тензорная визуализация — исследуемая характеристика в каждой точке пространства имеет вид матрицы, например напряжения в механической конструкции.

Стратегия визуализации определяет, как эти данные трансформируются в графические объекты, и набор их отображаемых атрибутов.

Визуальное представление скалярных полей

Скалярной называется величина, имеющая одно значение. Наборы скалярных данных содержат значения, которые могут быть распределены по времени и по пространству. В качестве примеров физических скалярных величин можно привести энергию, плотность, температуру, давление. Распространённым методом визуализации наборов скалярных данных является использование графиков и диаграмм, демонстрирующих распределение элементов данных как функций других скалярных параметров, таких как координаты и время. Если данные распределены по поверхности, их элементы можно изобразить как вертикальные столбцы, растущие с поверхности.





Визуализация с помощью цветового кодирования

Одним из самых распространённых способов визуализации скалярных данных является использование цветовых заливок (*pseudocolor* displays). Для этого нужно отобразить диапазон данных в диапазон цветов:

$$g : \mathbb{R} \mapsto [0, 1]^3,$$

где $[0, 1]^3$ относится к кубу RGB. Требуется указать набор цветов, с которыми будут связаны конкретные значения и линейно интерполировать цвета между ними. Например, синий цвет можно сопоставить с наименьшим скалярным значением, а красный — с наибольшим. Данные, представленные с цветовым кодированием, иногда требуют внимательного изучения, поскольку определённые комбинации цветов могут привнести или скрыть некоторые особенности; они могут даже вводить визуальные артефакты, такие как разрывы там, где их нет.

Визуальное представление скалярных полей

Контурные графики используются для изображения *изолиний* (линий постоянного значения, линий уровня) набора скалярных данных, распределённых по поверхности. Изолинии размещены с некоторым постоянным интервалом. Типичной сферой применения контурных графиков является изображение возвышений над нулевой плоскостью. Изолинии обычно изображаются как прямые отрезки вдоль каждой ячейки. Иногда изолинии изображаются с помощью сплайновых кривых, но аппроксимация сплайнами может привести к противоречиям и ложной интерпретации набора данных. Например, две сплайновые изолинии могут пересекаться. Изолинии могут иметь и цветовое кодирование, отражающее величину значения, для которого они построены. Также разными цветами могут быть показаны изолинии, относящиеся к разным наборам данных, отображаемым на одном графике.

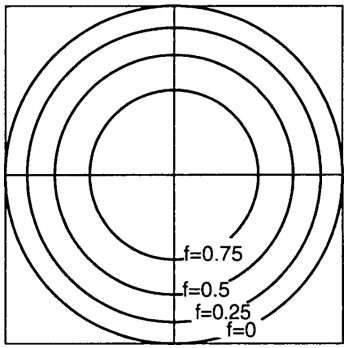


Рис.: Контурный график, построенный для четырёх значений функции $1 - x^2 - y^2$

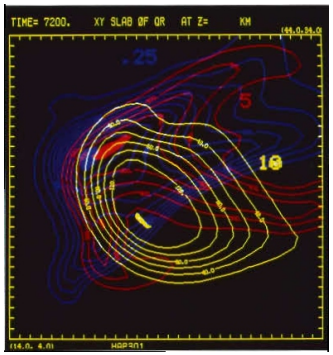


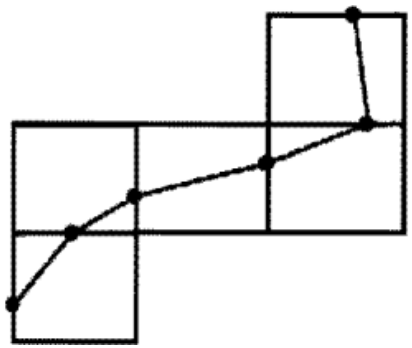
Рис.: Контурные графики с цветовым кодированием трёх наборов данных

Вычерчивание линий уровня

Если функция f задана аналитически, то для построения линии уровня необходимо найти решение уравнения

$$f(x, y) = c$$

при определённом значении параметра c. Чаще всего на одном изображении выводится несколько линий уровня, соответствующих разным значениям c. Если в нашем распоряжении имеется не аналитическая формула для $f(x, y)$, а лишь значения в узлах расчётной сетки, то мы сталкиваемся с аналогичной проблемой. Даже при аналитическом задании функции $f(x, y)$ получить аналитическое решение уравнения $f(x, y) - c = 0$ удаётся чрезвычайно редко. На практике чаще всего в этом случае формируется все тот же массив значений в узлах сетки, а уже по нему строятся линии уровня. Будем решать эту задачу с помощью метода маркированных квадратов. В этом методе изолинии строятся от ячейки к ячейке с проверкой четырёх углов сетки. Далее определяется какие стороны ячеек пересекаются определённой изолинией.



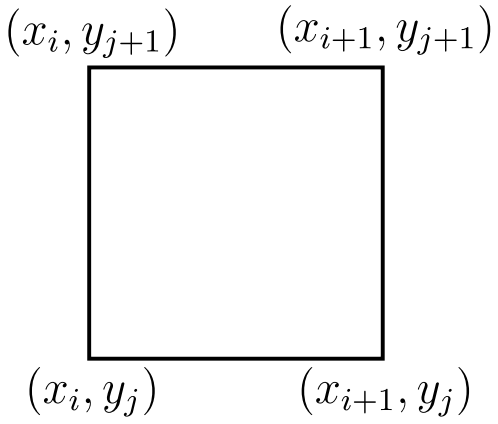
Метод маркированных квадратов

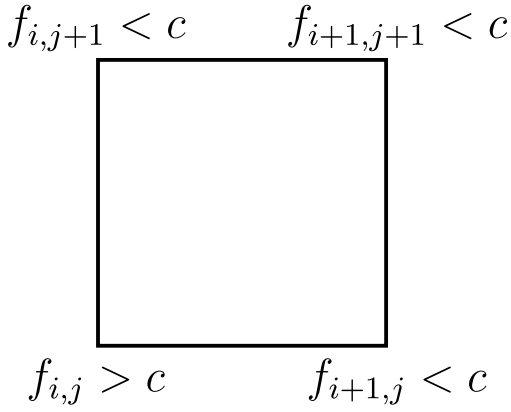
Предположим, что имеется массив $h_{ij} = f(x_i, y_j)$ точек функции $f(x, y)$, сформированный на регулярной сетке по независимым переменным x и y:

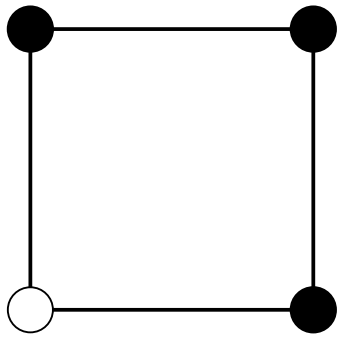
$$x_i = x_0 + i\Delta x, \quad i = 0, 1, \ldots, N-1, \qquad y_j = y_0 + j\Delta y, \quad j = 0, 1, \ldots, M-1,$$

где Δx и Δy — шаги между узлами сетки в направлении x и y. Для упрощения рассуждений предположим, что шаги являются постоянными. Массив значений в точках может быть получен и путём непосредственного измерения некоторой физической величины с постоянным шагом по пространству и во времени, а если речь идет об интенсивности какого-либо излучения, то шаг определяется расположением элементарных ячеек в матрице приемника излучения.
Предположим, что необходимо найти точки на кривой, неявно заданной уравнением $z = f(x, y)$ при $z = c$. Для определённого значения c функция $f(x, y)$ может вообще не иметь линии уровня, иметь единственную линию или множество линий. Стратегия построения линейной интерполяции линии уровня состоит в следующем.

Метод маркированных квадратов

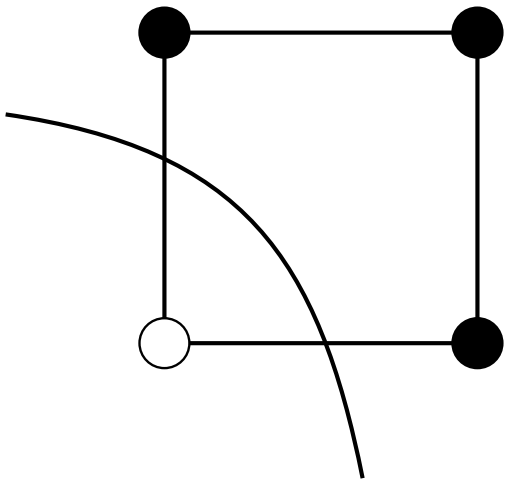


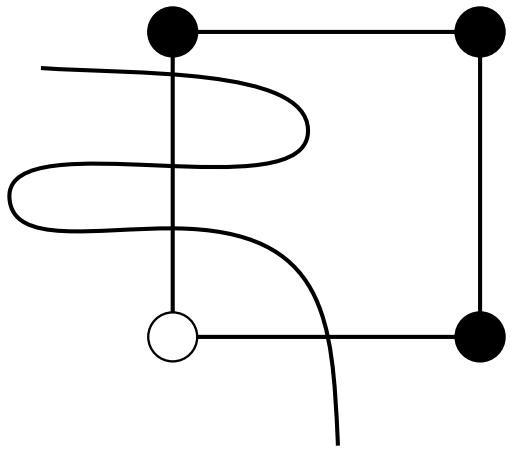




Процесс начинается с некоторой прямоугольной ячейки, определённой четырьмя узлами сетки (x_i, y_j), (x_{i+1}, y_j), (x_{i+1}, y_{j+1}), (x_i, y_{j+1}). В общем случае может оказаться так, что линия уровня проходит через ячейку, хотя ни одно из значений функции в углах ячейки не будет равно c. Рассмотрим простой случай, когда одно из значений в углах ячейки, например, f_{ij}, оказывается больше c, а остальные — меньше c.
Будем использовать раскраску (маркировку) углов ячейки: белым цветом маркируем угол, в котором значение *больше* заданного, а чёрным — углы, в которых значения *меньше* заданного. Очевидно, что в этом случае кривая линии уровня должна пересечь два ребра ячейки, прилегающих к «белому» углу. Это же должно произойти и в том случае, когда значение в одном углу ячейки будет меньше c, а в соседних с ним — больше.

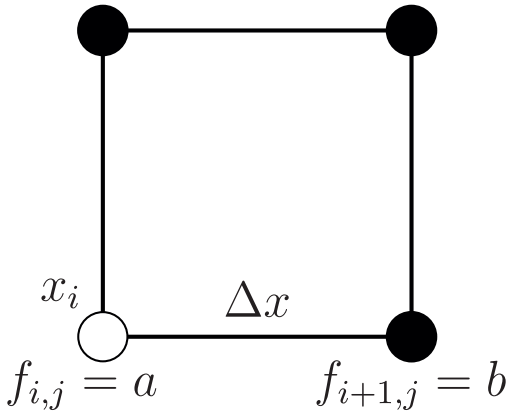
Метод маркированных квадратов





Кроме варианта, когда линия уровня пересекает ячейку один раз, возможен вариант, когда линия уровня пересекает ребро ячейки три раза, а в общем случае — любое нечетное число раз. Но мы всегда будем интерпретировать переход линии уровня через ребро ячейки как однократное, в противном случае, расчётную сетку следует измельчить.

Метод маркированных квадратов



Найдем теперь в каких именно точках ребер линия уровня пересекает ячейку. Рассмотрим, например, две соседние вершины ячейки, имеющие противоположную маркировку, т.е. значение функции в одной вершине больше c, а в другой — меньше c:

$$f(x_i, y_j) = a, \quad a > c, \qquad f(x_{i+1}, y_j) = b, \qquad b < c.$$

Если шаг независимой переменной x равен Δx, то можно вычислить положение точки пересечения, линейно интерполируя перепад между значениями a и b на интервале Δx:

$$x = x_i + \frac{c - a}{b - a} \Delta x.$$

Точку пересечения на втором ребре найдем аналогично, а затем соединим эти точки отрезком и получим один из сегментов кусочно-линейной аппроксимации линии уровня.

Метод маркированных квадратов

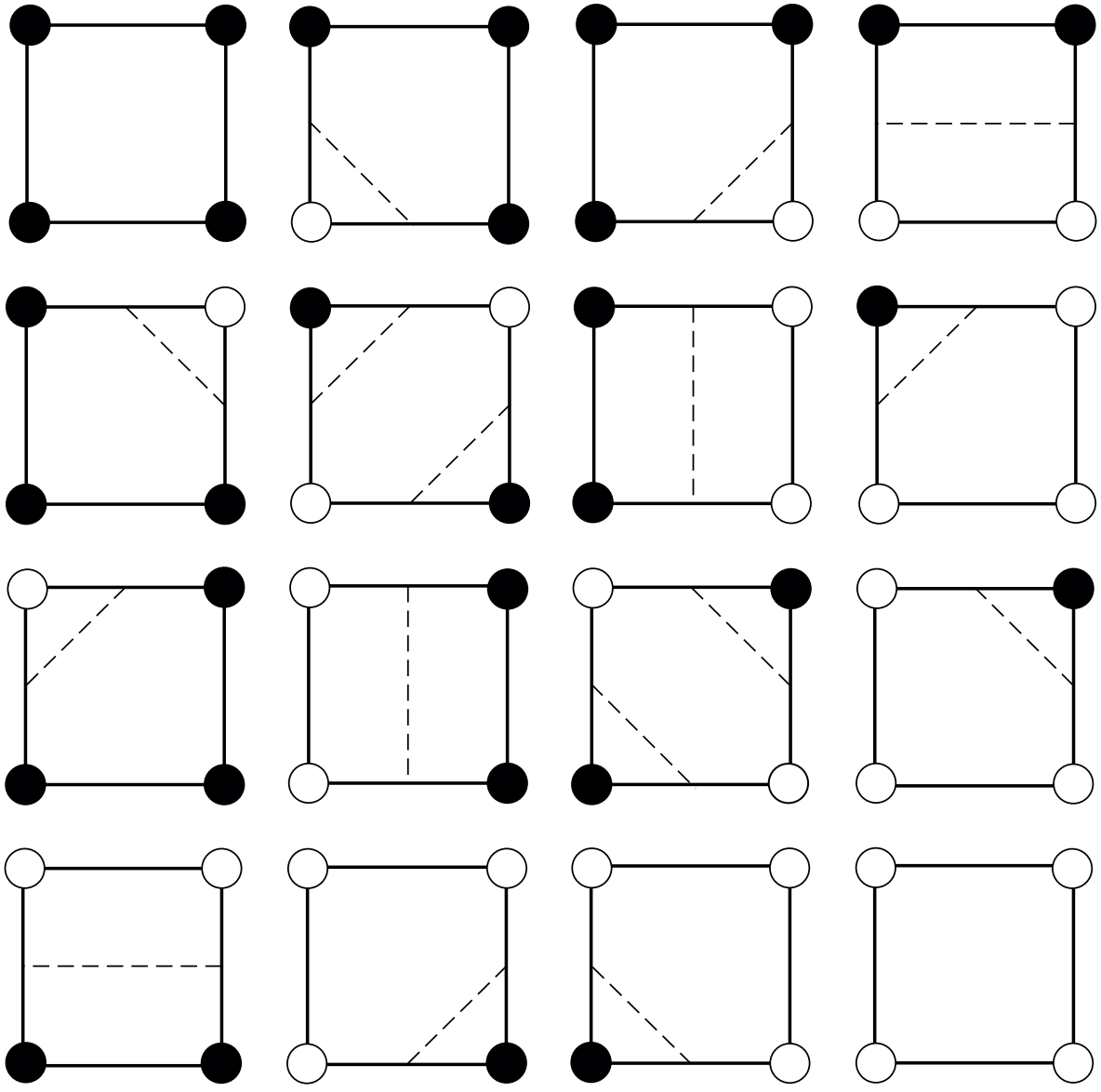


Рис.: Все $2^4 = 16$ возможных вариантов пересечения ячейки линией контура

Метод маркированных квадратов

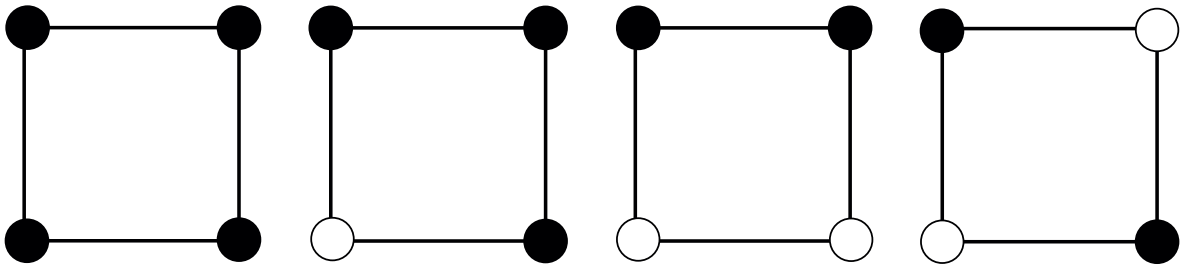


Рис.: Четыре уникальных варианта пересечения ячейки линией контура

Внимательный анализ всех возможных вариантов показывает, что в них имеются два типа симметрии. Один тип — симметрия относительно поворота. Все варианты, которые могут быть преобразованы к единому путём поворота, как, например, варианты 1 и 2, характеризуются тем, что линия уровня «отсекает» один из углов ячейки. Второй тип симметрии получается, если инвертировать маркировку вершин, — белые сделать чёрными, а чёрные — белыми, как, например, в вариантах 0 и 15. Если принять во внимание подобные типы симметрии, то 16 возможных вариантов сводятся к четырём.

Первый вариант тривиальный, поскольку в этом случае линия уровня не пересекает ячейку. Второй вариант мы уже рассмотрели. Третий вариант также довольно прост — нужно вычертить сегмент, который пересекает ячейку от одного ребра до противоположного.

Метод маркированных квадратов

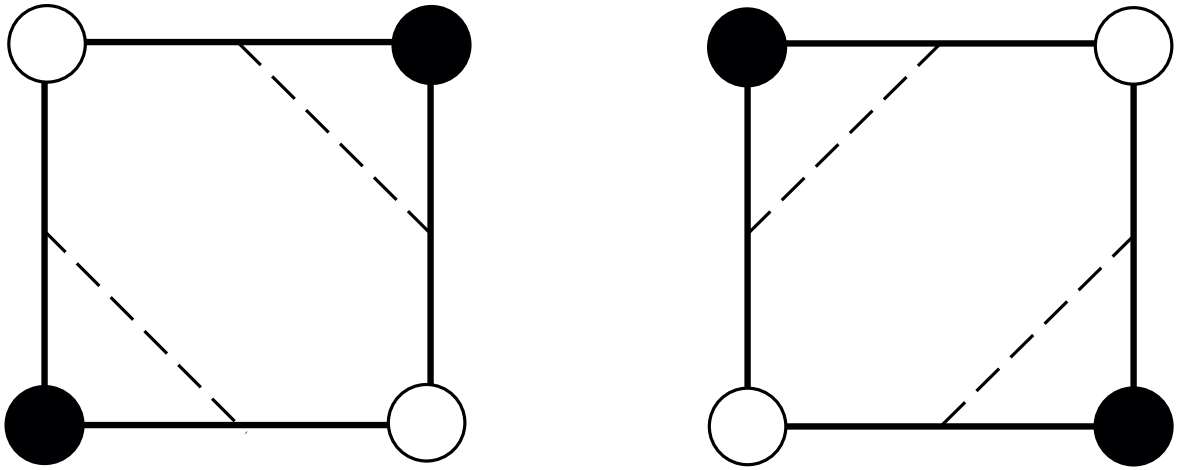


Рис.: Неоднозначная интерпретация маркировки вершин ячейки

Последний, четвертый, вариант наиболее сложен — его можно интерпретировать по-разному, и нужно выбрать одну из возможных интерпретаций. При отсутствии дополнительной информации нельзя отдать предпочтение ни одному из представленных вариантов.

Метод маркированных квадратов

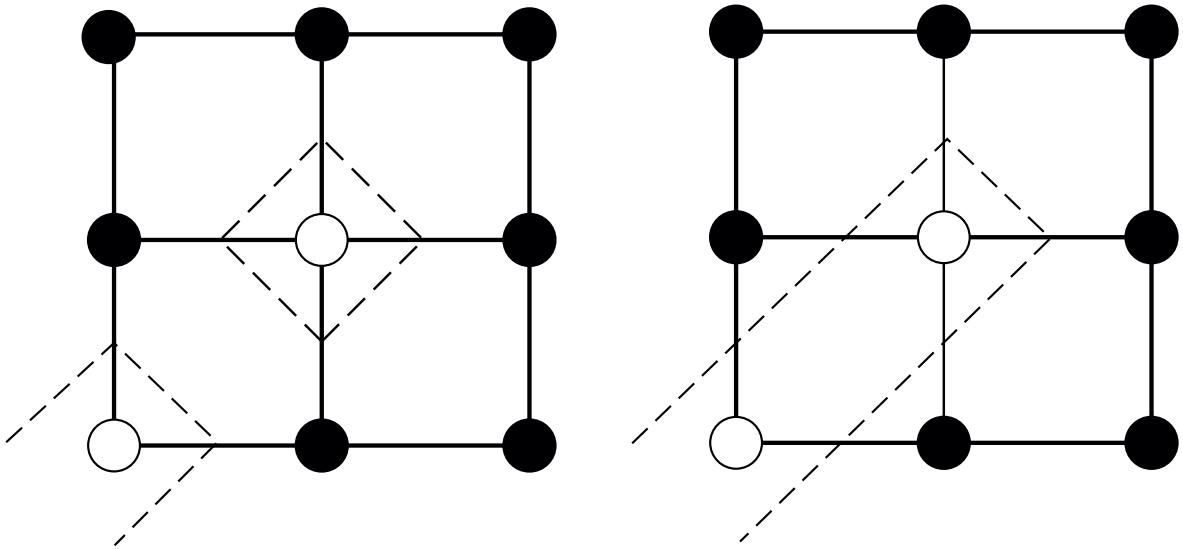


Рис.: Пример разной интерпретации одного и того же варианта маркировки вершин ячейки

Если выбрать один из вариантов наудачу, случайно, или заранее выбрать один из способов интерпретации, а о другом «забыть навсегда», то в зависимости от такого выбора результаты могут существенно отличаться.

Метод маркированных квадратов

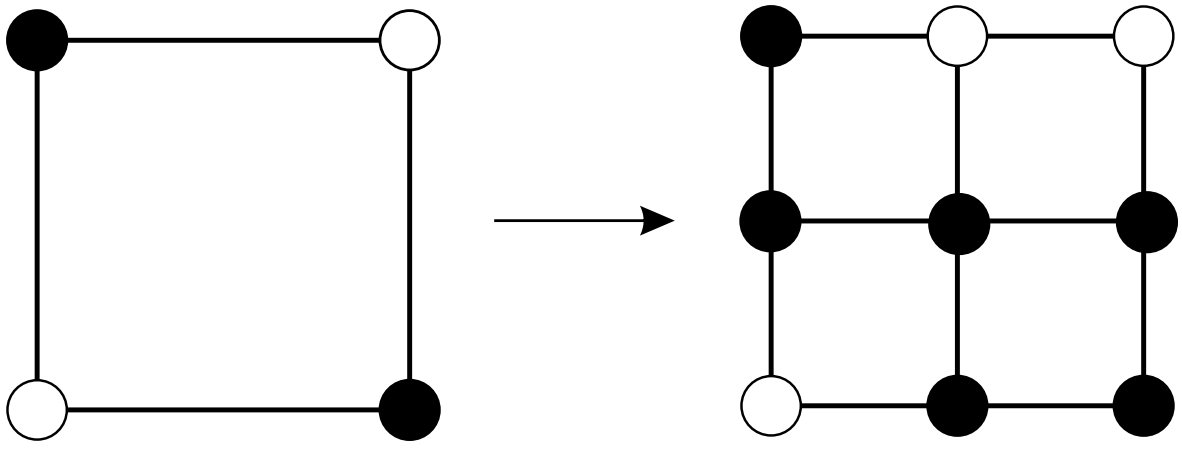


Рис.: Разбиение ячейки

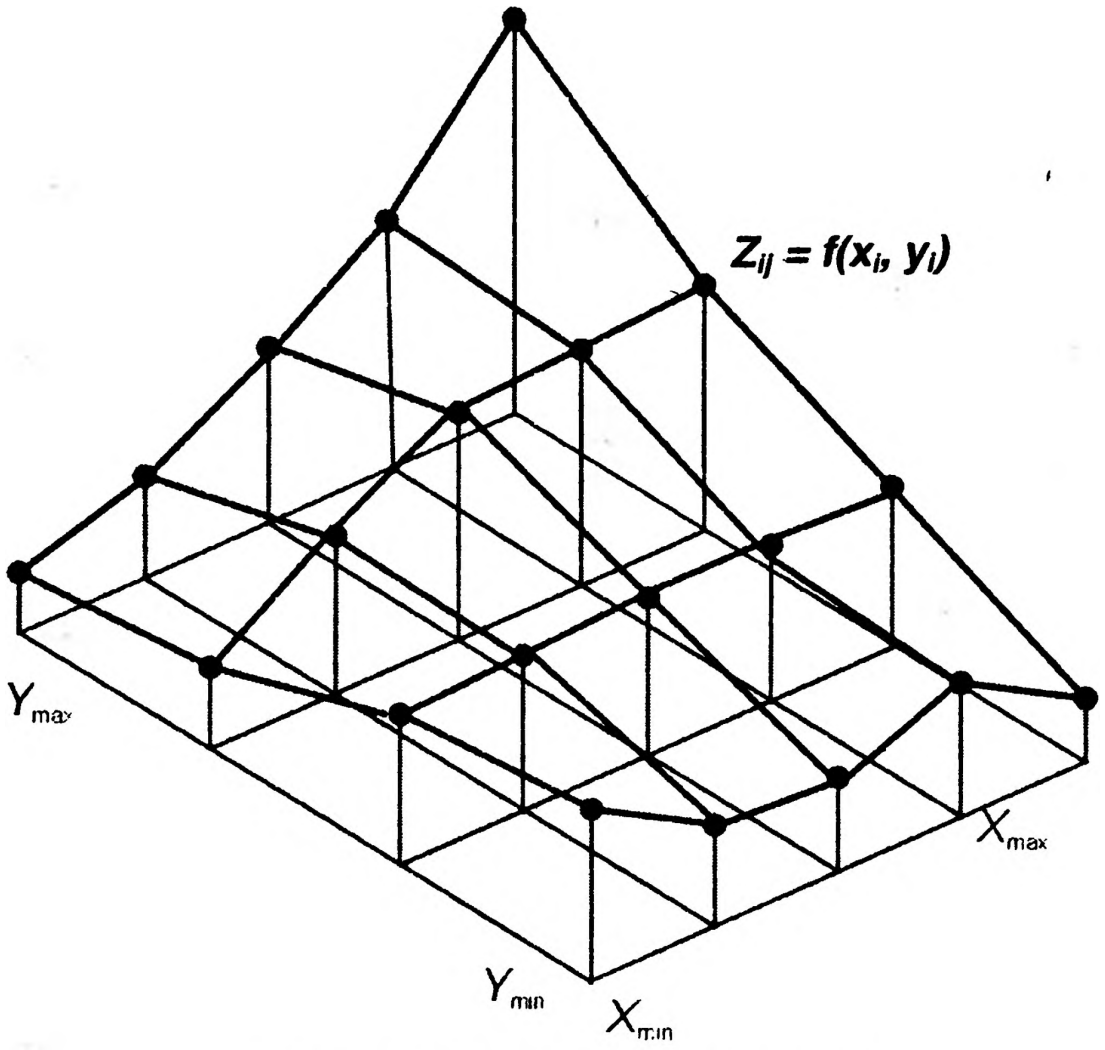
Чтобы разрешить неопределённость, требуется выполнить дополнительные расчёты. Разделим «подозрительную» ячейку на четыре части и рассчитаем значение функции в центральной точке. Если функция задана аналитически, это делается очень просто, а если имеются только данные в ячейках, то можно получить промежуточное значение, интерполируя по какому-либо закону уже имеющиеся данные в вершинах исходной ячейки. Если ситуация в новых ячейках вновь будет неоднозначной, то разделение следует повторить.

Метод маркированных квадратов

Существуют и другие методы построения линий уровня. Один из них состоит в том, что сначала отыскивается любая ячейка, через которую проходит линия, а затем отслеживается её траектория, целенаправленным переходом из текущей ячейки в ту, куда «ушла» линия контура. Этот метод кажется более производительным, поскольку анализируются не все ячейки подряд, а только те из них, через которые обязательно проходит линия уровня (раз уж она «нырнула» в ячейку, то обязательно должна из неё и «вынырнуть»), но он таит опасность упустить один из контуров линии уровня. Кроме того, достоинство рассмотренного выше метода в том, что анализ одной ячейки не зависит от другой, а следовательно, при наличии соответствующих средств ячейки можно анализировать параллельно.

Построение линий уровня на поверхности

Пусть поверхность некоторой функции $z = f(x, y)$ задана массивом значений $z(x, y)$, рассчитанных на сетке $x = (X_{\text{min}}, \ldots, X_{\text{max}})$, $y = (Y_{\text{min}}, \ldots, Y_{\text{max}})$. Для нахождения линии уровня L требуется найти пересечение поверхности $z = f(x, y)$ с плоскостью $z = L$.



Построение линий уровня на поверхности

В этом случае общий подход к построению линии уровня заключается в следующем:

1. Выполняется триангуляция поверхности.
2. Для каждого треугольника находится пересечение с плоскостью L.

Этот метод достаточно прост и позволяет получить хорошее изображение линии уровня. Линии уровня могут быть разрывными. Отдельные составные части линии называют *сегментами*.

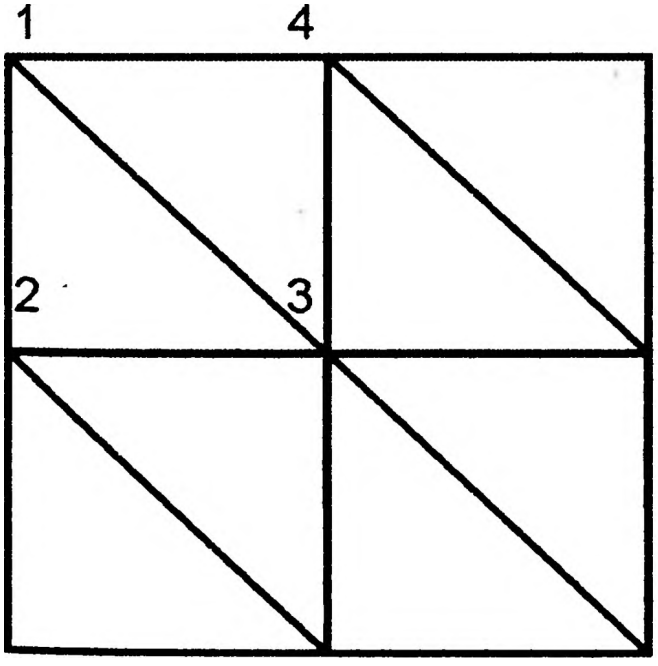


Рис.: Триангуляция поверхности

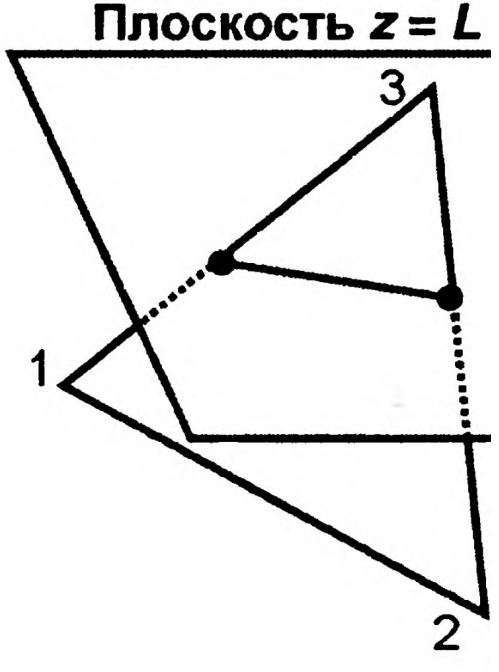


Рис.: Пересечение треугольника с плоскостью $z = L$

Визуальное представление 3D-скалярных полей

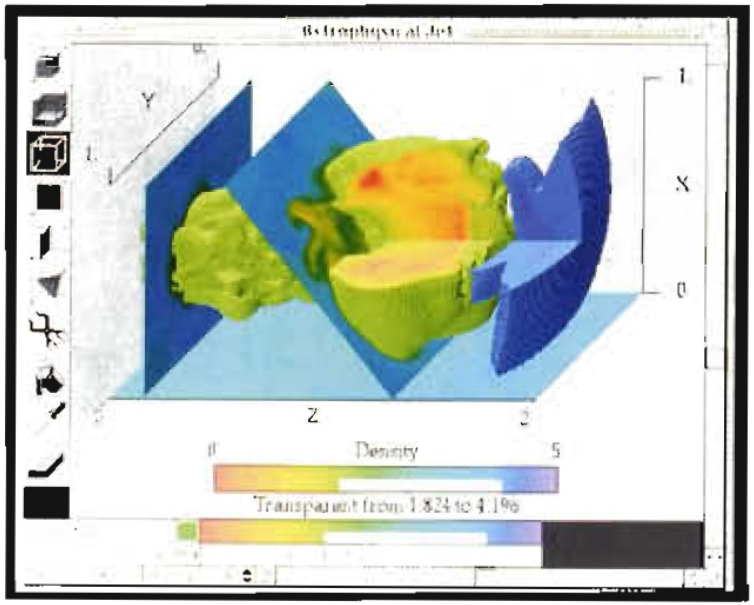


Рис.: Поперечные сечения трёхмерного набора данных

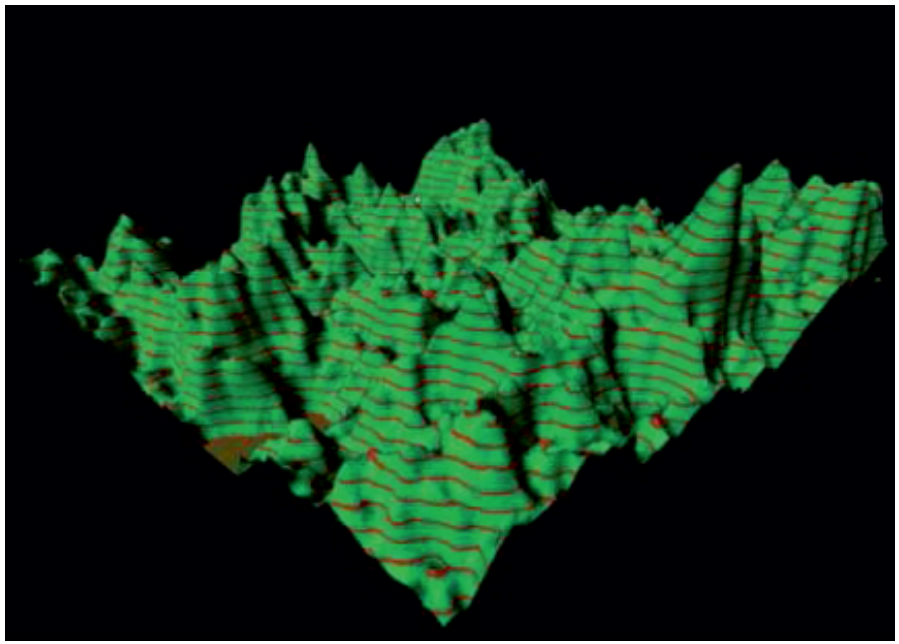


Рис.: Контурный график, показывающий линии постоянной высоты

Для трёхмерных скалярных полей данных можно взять поперечные сечения и изобразить двухмерные распределения данных в сечениях. Можно либо использовать цветовое кодирование значений данных сечений, либо отобразить линий уровня на поверхности.

Построение изоповерхностей

Трёхмерным аналогом изолиний является *изоповерхность (isosurface)* — поверхность постоянного значения поля. Изоповерхности удобны для визуализации трёхмерных скалярных полей. Изоповерхности обычно моделируются с помощью треугольных сеток. Несколько изоповерхностей можно изобразить используя разные цвета, так же, как множество изолиний могут быть нарисованы на одном контурном графике. Чтобы можно было видеть изоповерхности, вложенные в другие изоповерхности, используется прозрачность или отсечение.
Построение изоповерхности подобно изображению изолиний, только в этом случае есть трёхмерные ячейки сетки, и нужно проверять элементы данных в восьми углах ячейки, чтобы определить положение участков изоповерхности.

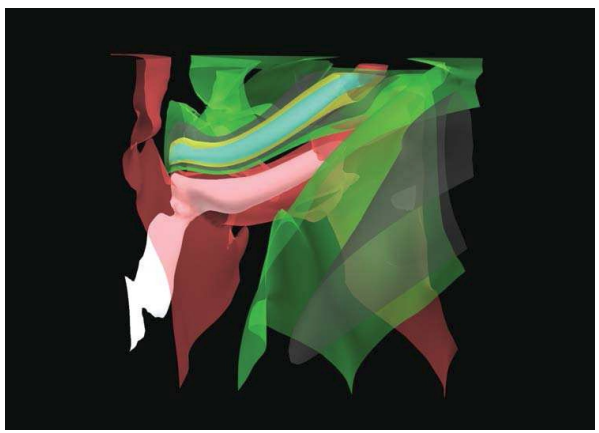
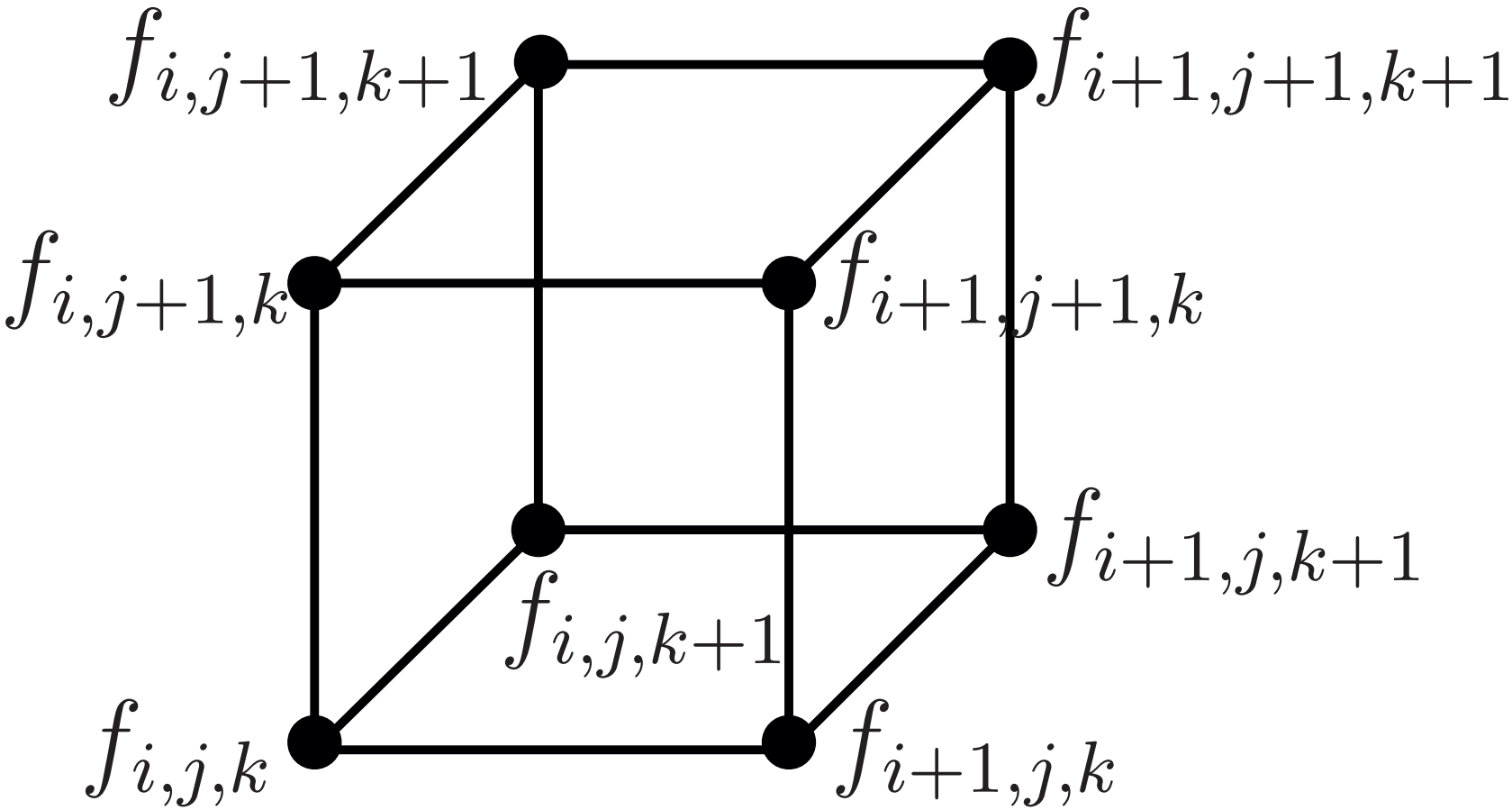


Рис.: Изоповерхности, сгенерированные по значениям содержания воды, полученным из численной модели грозы

Изоповерхности и метод маркированных кубиков

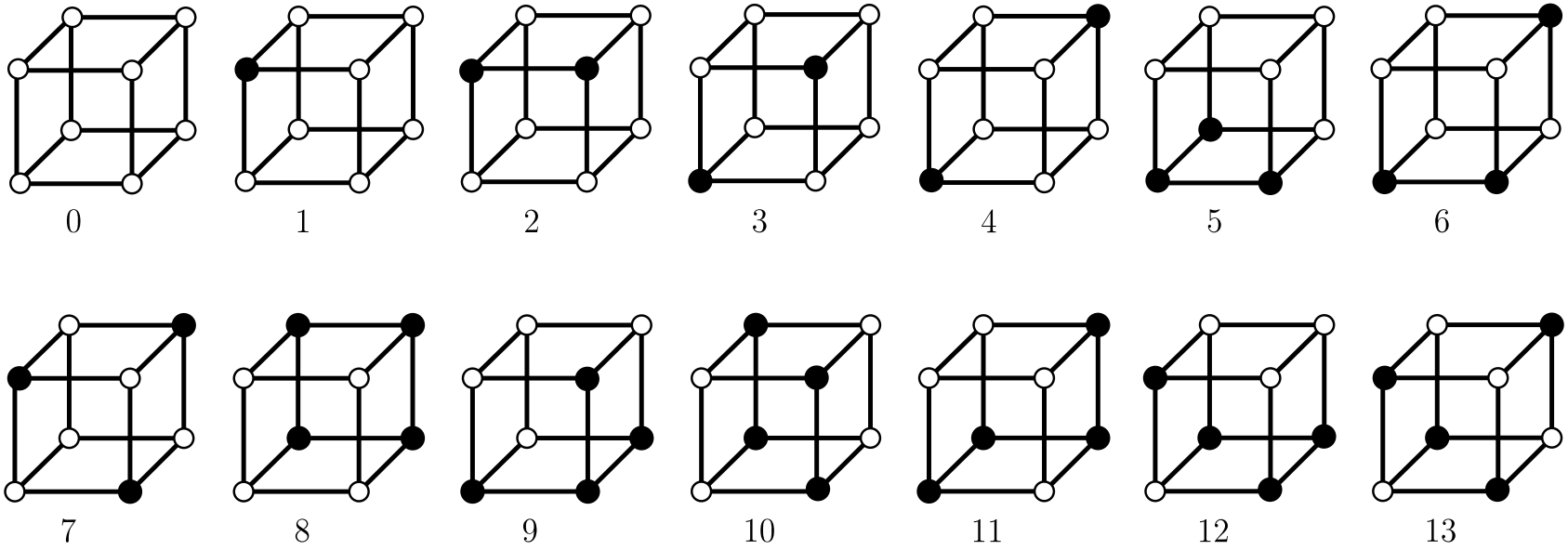
Пусть имеется массив значений f_{ijk} скалярного поля $f(x, y, z)$ в узлах расчетной сетки. Требуется по этим значениям построить изоповерхность $f(x, y, z) = c$. Для заданного значения c может существовать единственная изоповерхность, ни одной изоповерхности или несколько. Учитывая, что графическая система лучше всего справляется с задачей отображения трехмерных треугольников, разработаем метод, который позволит отыскать вершины изоповерхности и аппроксимировать её множеством треугольников. Этот метод получил наименование метода маркированных кубиков.

Изоповерхности и метод маркированных кубиков



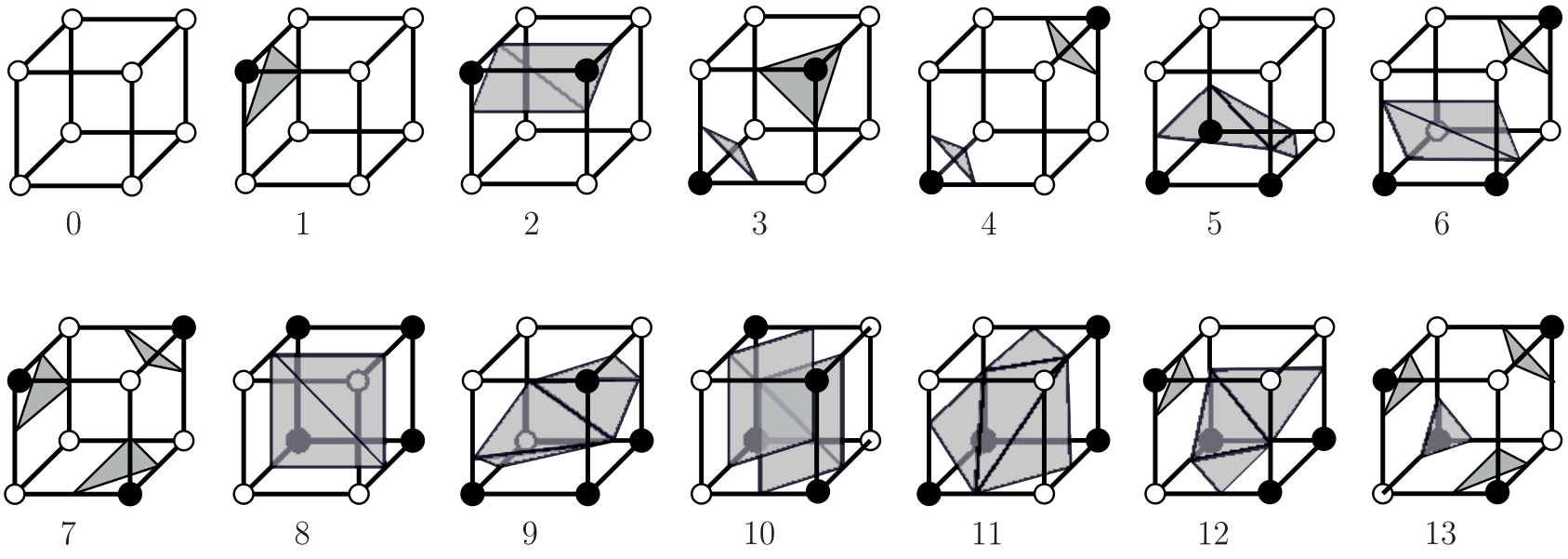
Для простоты изложения будем считать, что расчётная сетка является равномерной. Восемь соседних узлов пространственной сетки образуют одну ячейку в форме кубика.

Изоповерхности и метод маркированных кубиков



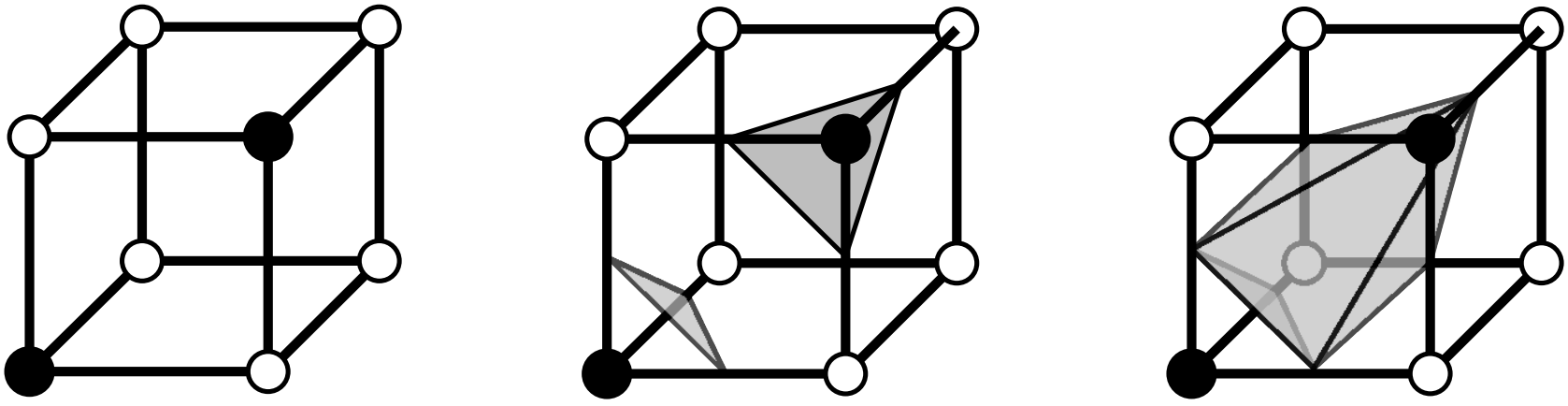
Для данного значения уровня изоповерхности c можно промаркировать чёрным и белым цветом вершины ячейки, в зависимости от того, превышает значение функции f этот уровень или нет. Существует $2^8 = 256$ возможных вариантов маркировки отдельной ячейки. Но приняв во внимание имеющуюся симметрию отдельных вариантов, все разнообразие сводится к 14 уникальным случаям.

Изоповерхности и метод маркированных кубиков



Точки пересечения изоповерхности с ребрами ячейки можно вычислить, используя тот же метод интерполяции, который применялся при обработке квадратных ячеек в методе маркированных квадратов. После определения координат точек пересечения формируется участок изоповерхности в виде одного или нескольких треугольников.

Изоповерхности и метод маркированных кубиков



При работе с маркированными кубиками возникает та же проблема неоднозначности, которую мы рассматривали выше применительно к маркированным квадратам. Неоднозначность появляется в том случае, когда узлы с одинаковой маркировкой расположены по диагонали грани ячейки. Очевидно, что выбор различных вариантов приводит к разному виду изоповерхности в районе этой ячейки. Неверный выбор проявится в виде разрыва изоповерхности, которая в других местах будет выглядеть довольно гладкой. Как и при работе с маркированными квадратами, для решения этой задачи требуются дополнительные разбиения исходной ячейки для получения однозначного случая.

Изоповерхности и метод маркированных кубиков

Как и квадратные ячейки, которые раньше использовались для формирования линий уровня, кубические ячейки можно обрабатывать независимо друг от друга. После обработки очередной ячейки сформированный в ней участок изоповерхности в виде одного или нескольких треугольников передаётся в конвейер визуализации графической системы. Поскольку этот алгоритм довольно просто можно распараллелить, метод маркированных кубиков довольно широко используется для визуализации трёхмерных данных.

Визуальное представление скалярных полей

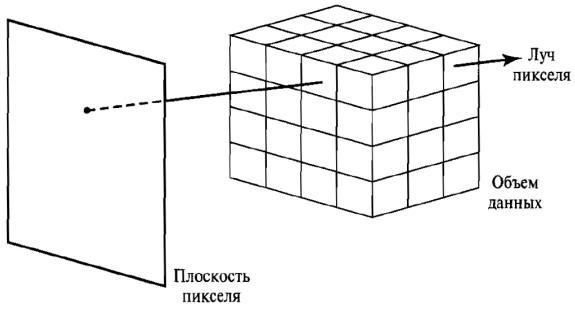


Рис.: Объёмная визуализация с использованием расчёта луча для изучения внутренних значений данных

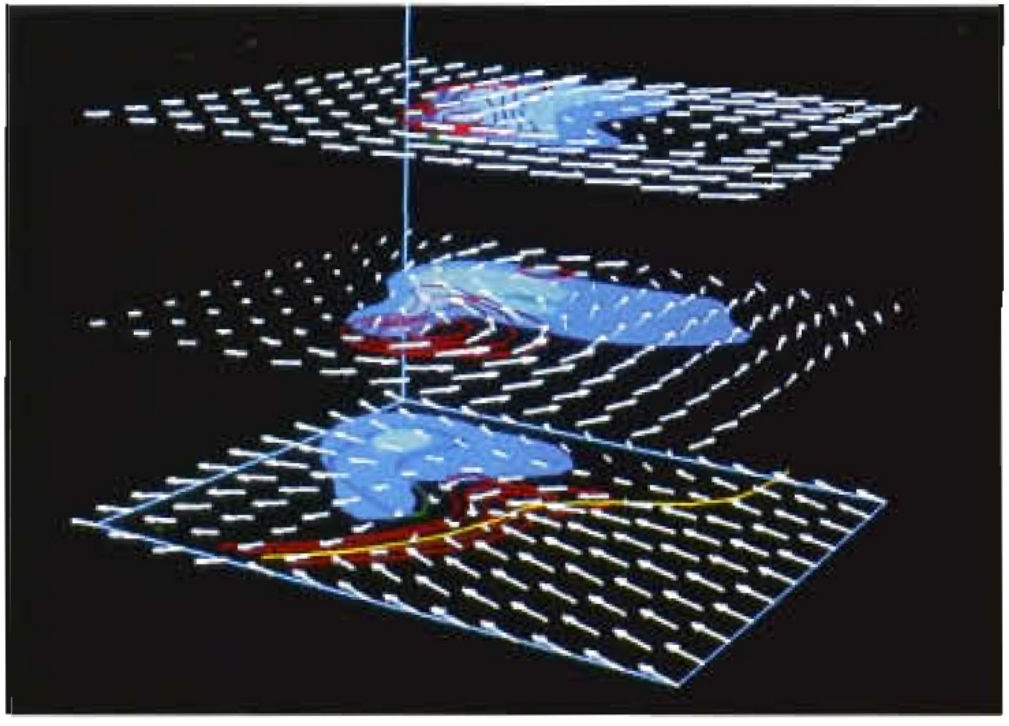
Рис.: Объемная визуализация набора данных структуры сердца собаки с цветовым кодированием расстояния до максимального значения вокселя

Другим методом визуализации трёхмерного набора данных является *объёмная визуализация (volume visualization, volume rendering)*, которая иногда даёт изображения, похожие на рентгеновские снимки. Внутренняя информация о наборе данных проектируется на экран с использованием методов расчёта луча. Вдоль траектории луча от каждого пикселя экрана элементы данных исследуются и кодируются для отображения цветом. Часто элементы данных в точках сетки усредняются, так что для каждого вокселя пространства данных хранится одно значение. Сейсмические данные часто исследуются на максимум и минимум вдоль каждого луча. В медицинских приложениях элементами данных являются коэффициенты непрозрачности из диапазона от 0 (ткани) до 1 (кости).

Визуальное представление векторных полей

В то время как скалярные поля имеют одно значение в каждой точке сетки, векторная величина $\mathbf{v}$ в трёхмерном пространстве имеет три скалярных значения (v_x, v_y, v_z), по одному на каждое координатное направление, а двумерный вектор имеет два компонента (v_x, v_y). Другой способ описания векторной величины — задать её модуль $|v|$ и направление как единичный вектор $\mathbf{u}$. Векторные величины, как и скалярные, могут быть функциями координат, времени и других параметров. Векторные поля возникают в таких приложениях, как вычислительная гидродинамика и, как правило, представляют собой поток газа или жидкости в виде поля скорости. Другие примеры векторных величин: ускорение, сила и др. Визуализация поля обеспечивает лучший анализ и понимание закономерностей потока.
Методы визуализации векторных полей могут быть сгруппированы в классы, например, визуализации векторов в узлах, визуализации траекторий частиц или линий тока и вихря и другие. Одни методы (Streak Lines и Time Lines) были получены из методов визуализации потока в экспериментальных установках. Другие являются математическими абстракциями.

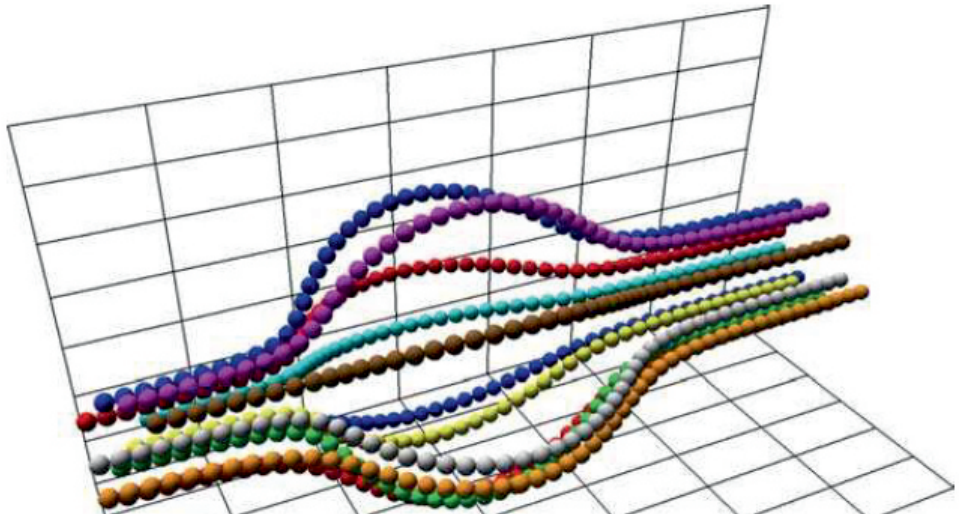
Визуализация векторов в узлах



Векторное поле, можно, например, изобразить в виде маленьких стрелок для 2D случая или конусов в 3D случае, указывающих направление и модуль вектора в узле сетки. Модули векторных значений в узле сетки можно представить, варьируя длины стрелочек, или же можно изобразить разными цветами стрелочки одинакового размера. Данный метод может дополняться изображением поперечных сечений, поскольку данные трудно увидеть в трёхмерной области, заполненной накладывающимися стрелочками.

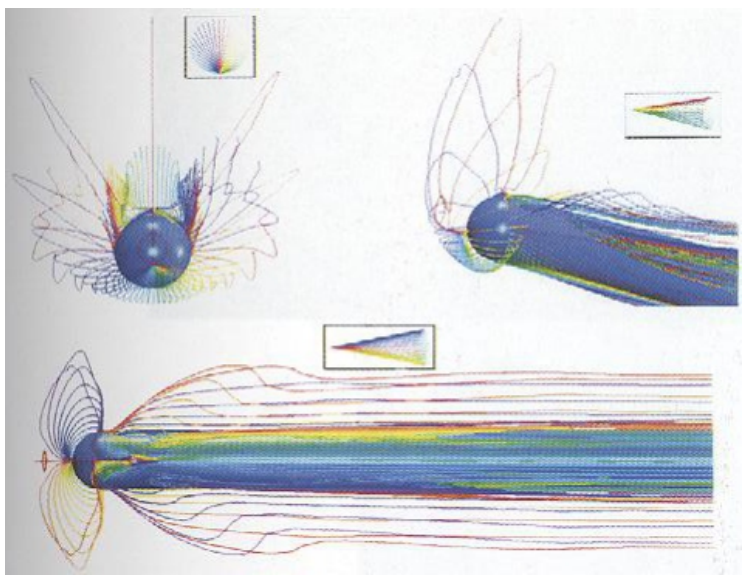
Метод трассировки частиц

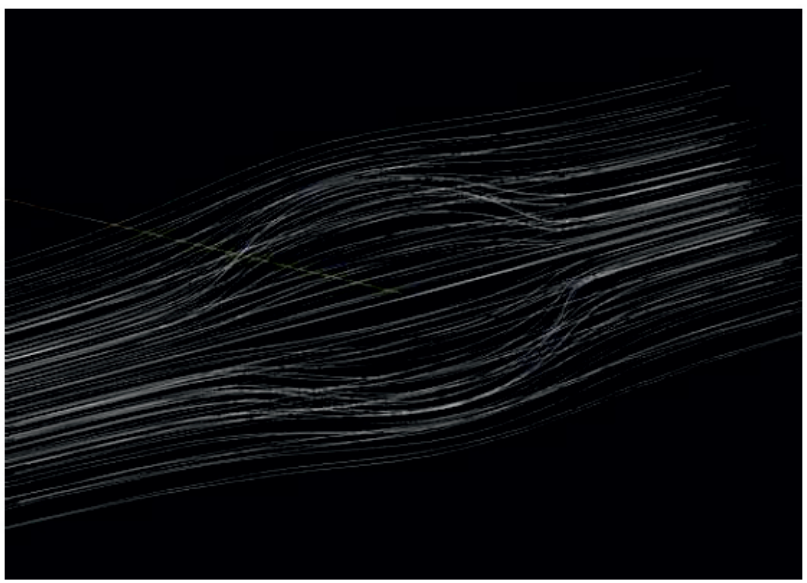
Метод трассировки частиц (particle traces) прослеживает путь невесомых частиц через векторное поле скорости. Частицы отображаются в виде маленьких сфер, направленных конусов или треугольников в 2D случае. Направление стрелок совпадает с направлением векторного поля в месте расположения частицы. Частицы перемещаются вдоль соответствующих *траекторий (pathlines)* с течением времени, чтобы передать динамику потока. Скорость движения частицы определяется модулем вектора. Цвет частицы также часто выбирается в соответствии с величиной вектора. Для небольшого числа частиц положения частиц могут пересчитываться каждый кадр. Для большого числа частиц может потребоваться предварительное вычисление положений частиц. Приложение может позволять пользователю интерактивно размещать новые частицы в системе и следить за потоком.



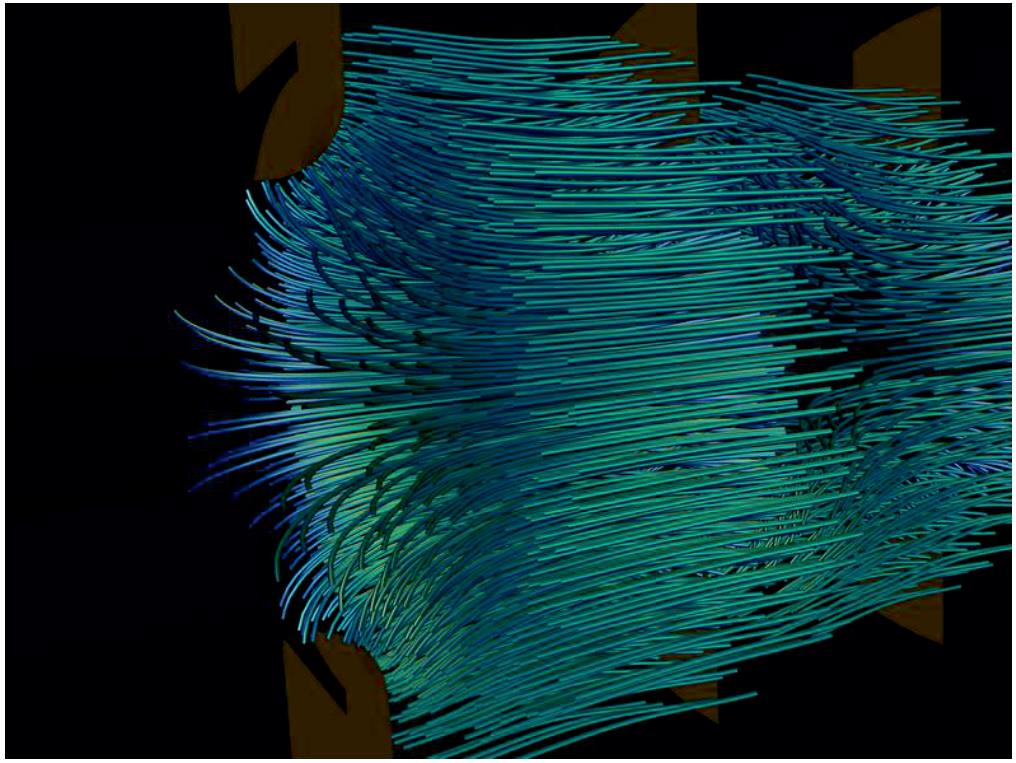
Линии тока

Линии тока получаются также интегрированием векторного поля, но при фиксированном моменте времени. В каждой точке векторное поле представляет собой касательную к линии тока. Для статических векторных полей, линии тока совпадают с траекториями частиц. Однако они отличаются в случае изменяющегося от времени поля. Линии тока позволяют отобразить только направление векторного поля. Для отображения его величины можно, например, использовать цвет. Примитивы линий могут быть отрисованы очень быстро и позволяют рисовать очень большое количество линий тока. Недостатком является то, что линии отображаются как плоская геометрия с одной нормалью в каждой конечной точке, поэтому результат обладает гораздо меньшей точностью затенения по сравнению с использованием полигонов цилиндра.





Линии тока с подсветкой



При визуализации трёхмерных полей освещение и затенение обеспечивают улучшенный просмотр, особенно для большого количества линий тока. Для визуализации линии тока используются геометрия в виде трубки, построенной из сегментов цилиндров вдоль траектории. Для получения гладкого затенения приходится использовать большое число полигонов, что приводит к большой нагрузке при отображении большого количества линий тока.

Линии тока

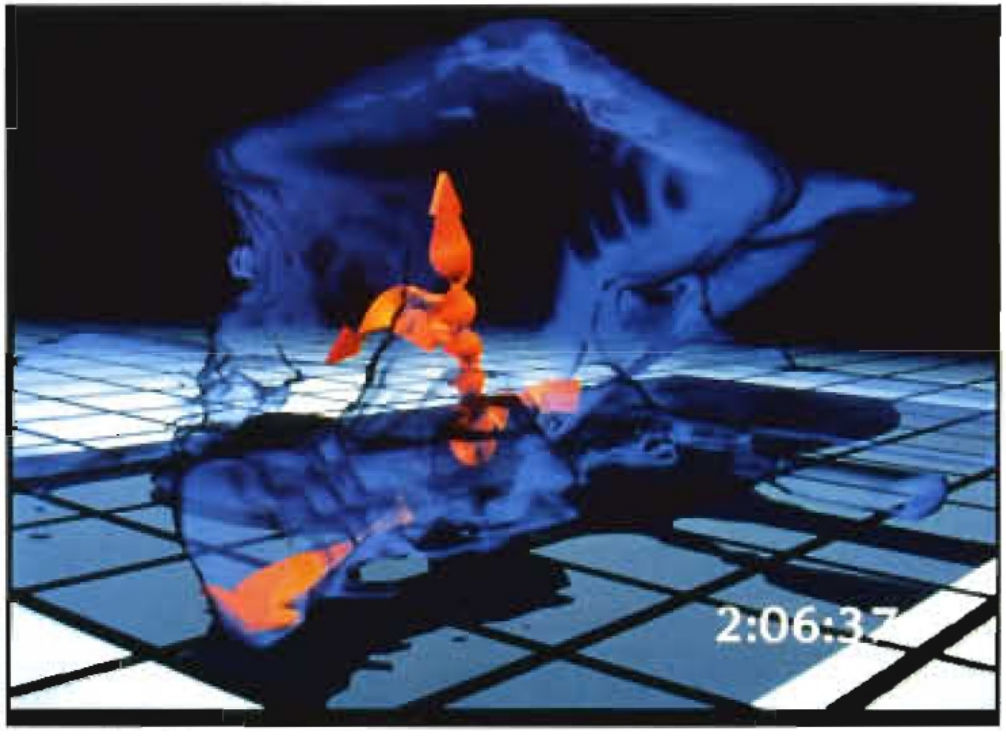


Рис.: Завихряющийся поток воздуха, визуализированный с широкими линиями тока

Также можно использовать геометрию с переменным поперечным сечением или вращением (скручиванием) для кодирования других локальных характеристик поля: расхождение или сходимость (дивергенция) модулирует ширину, а угловая скорость вращения скручивает геометрию.

Streak Line

Streak Line. Линия, соединяющая позиции всех частиц, прошедших через определённую точку. Она прослеживается в потоке жидкости путём впрыскивания в этой точке цветного материала, такого как дым или краситель, который разносится вместе с потоком и используется для его визуализации.
Для статических векторных полей Streak Lines совпадают с траекториями частиц и линиями тока. В нестационарном случае все эти кривые различны.

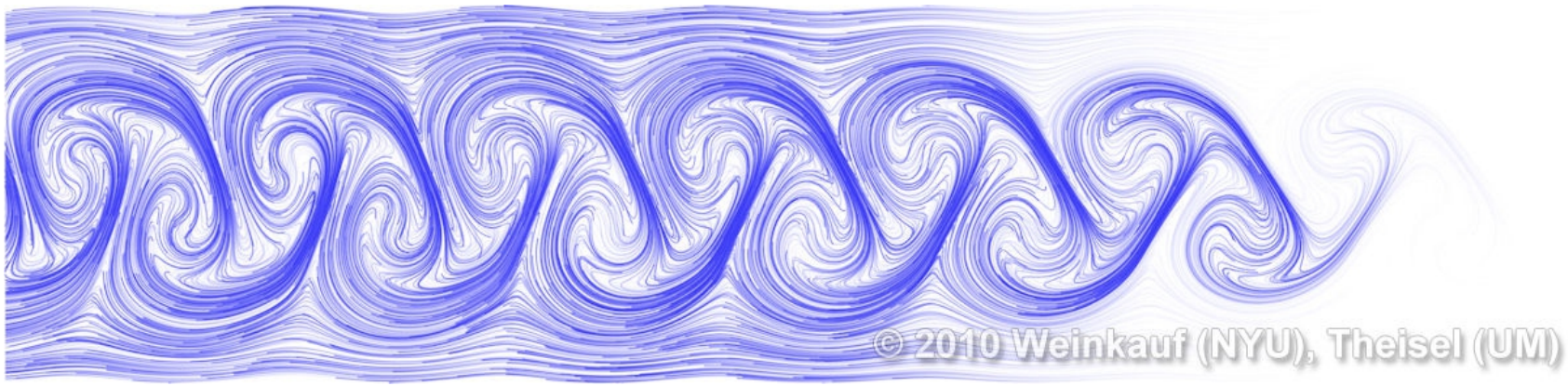
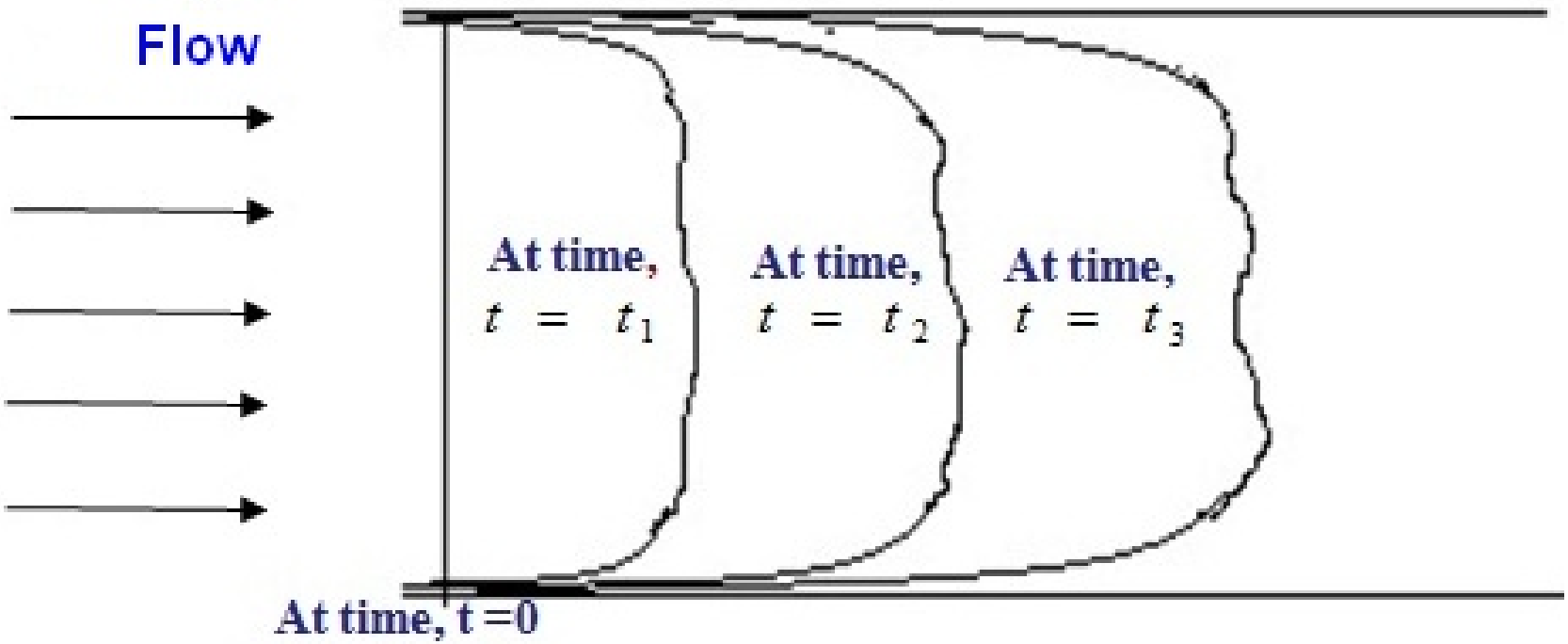


Рис.: 5000 streak lines в 2D нестационарном потоке вокруг цилиндра. Streak lines отображаются так, что они становятся менее непрозрачными по мере приближения к своей исходной точке.

Time Line

Time Line. Линия, соединяющая все частицы, которые в начальный момент времени находились на некоторой прямой, перпендикулярной направлению потока. С течением времени эта прямая линия движется и деформируется вместе с потоком в соответствии с изменением поля скорости.



Построение векторных линий

Построение большинства кривых, описанных выше (траекторий частиц, линий тока, и т.д.) основано на том же принципе: они генерируются путём интегрирования векторного поля. Для линий тока, траекторий, streaklines и timelines интегрируется поле скорости $\mathbf{v}$; для вихревых линий используется поле вектора вихря ($\omega = \nabla \times \mathbf{v}$).
Все кривые генерируются с помощью численного интегрирования уравнения:

$$\mathbf{r}(t) = \int_t \mathbf{v} d\tau.$$

Начальное условие для этого уравнения определяет положение начальной точки $\mathbf{r}_0$. Решение представляет собой последовательность точек $(\mathbf{r}_0, \mathbf{r}_1, \ldots)$, через которые может быть построена кривая.

Построение векторных линий

Для численного интегрирования могут применяться, например, методы Эйлера и Рунге-Кутты.
В методе Эйлера новое положение частицы $\mathbf{r}_{i+1}$ вычисляется из текущей позиции, $\mathbf{r}_i$ как $\mathbf{r}_{i+1} = \mathbf{r}_i + \Delta t\, \mathbf{v}_i$. Вектор $\mathbf{v}_i$ является приближением вектора $\mathbf{v}$ в точке $\mathbf{r}_i$. Для 2D-поля по четырём значениям вектора $\mathbf{v}$ в вершинах ячейки строится билинейная интерполяция. Для 3D-поля и 8 векторов в вершинах объёмной ячейки используется трилинейная интерполяция. Приближение с помощью метода Эйлера имеет точность порядка $O(\Delta t^2)$, что может привести к существенной ошибке.
Более точная аппроксимация получается с использованием методов Рунге-Кутта. Например, метод Рунге-Кутта четвёртого порядка вычисляется как

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \frac{1}{6}\Delta t \left(\mathbf{v}_i + 2\mathbf{v}^1_{i+1} + 2\mathbf{v}^2_{i+1} + 2\mathbf{v}^3_{i+1}\right),$$

где $\mathbf{v}^k_{i+1}$ — вектор, вычисленный в точке $\mathbf{r}_i + 0.5\Delta t\, \mathbf{v}^{k-1}_{i+1}$ и $\mathbf{v}^0_{i+1} = \mathbf{v}_i$.

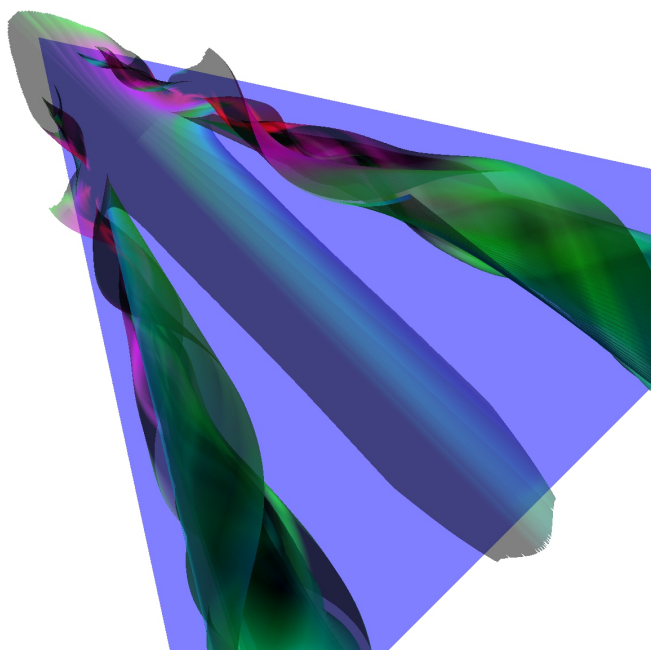
Построение векторных линий

Шаг Δt может быть постоянным или адаптивным.
В случае визуализации анимации (например, при трассировке частиц) используется фиксированный шаг, поскольку для плавной анимации требуются равномерные интервалы времени.
Если важна точность построения линии, то более эффективным оказывается использование адаптивного шага, который рассчитывается исходя из размера расчётной сетки и кривизны кривой в текущей точке. В местах, где сеточные ячейки малы и узлы расположены близко друг к другу, могут возникать большие градиенты, поэтому лучше выбирать меньший шаг интегрирования, чтобы не перешагнуть через маленькие ячейки сетки и, таким образом, пропустить рассчитанные данные. Кроме того, в областях, где кривизна большая, лучше располагать последующие точки кривой ближе друг к другу, в то время как в областях, где кривизна мала, расстояние между точками может быть увеличено для экономии вычислительного времени.

Построение поверхностей тока

В трёхмерном пространстве визуализация векторного поля с помощью рассмотренных кривых не позволяет оценить пространственную картину течения по глубине. Эту проблему можно решить в интерактивных приложениях с помощью поворота изображения или просмотра изображения в стерео режиме с помощью устройств виртуальной реальности. Для улучшения визуализации с помощью статических рисунков, рассмотренные кривые могут быть превращены в поверхности. Касательная кривая может быть расширена до касательной поверхности, т.е. поверхности, которая повсюду касается направления вектора.

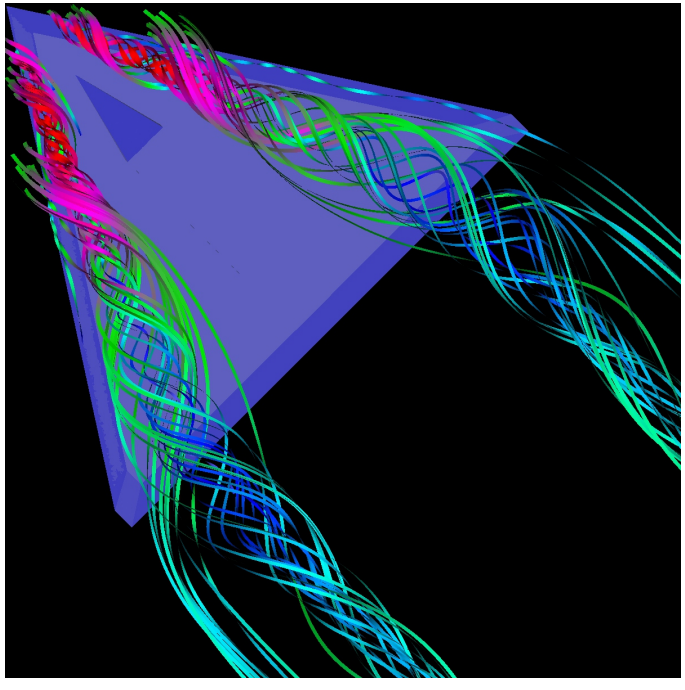
В стационарном поле скоростей касательная поверхность называется поверхностью тока. Поскольку направление скорости всюду касается поверхности потока, составляющая скорости, нормальная к поверхности, везде равна нулю. Это означает, что среда не течёт через поверхность тока. Time lines могут быть обобщены до time surfaces, связывающих частицы, которые одновременно исходят с определённой плоскости. Аналогичным образом обобщаются и остальные кривые.



Построение поверхностей тока

Самый простой тип поверхности тока — это лента тока. Помимо локального направления потока, она может показать и его локальное вращение. Ленты могут быть созданы по-разному.

Во-первых, две соседние линии тока могут быть построены от двух начальных точек расположенных близко друг к другу, а затем построения на их основе сетки треугольников. Ширина такой ленты зависит от формы обоих линий тока, и в сильно расходящемся поле она может стать достаточно большой.

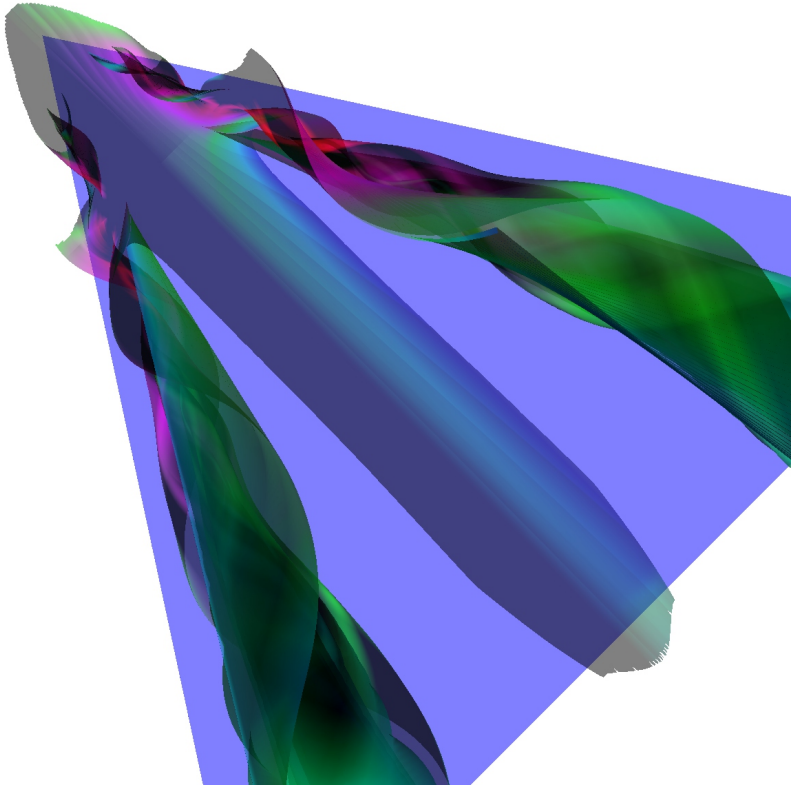


Второй способ заключается в построении ленты постоянной шириной, центрированной относительно одной линии тока. Ориентация полосы зависит от угловой скорости потока, полученной из вектора вихря.

Угол поворота можно найти путём интегрирования по времени вдоль линии тока. В начальной точке определяется начальная ориентация. Первый способ может показать вихревое поведение потока и дивергенцию в зависимости от изменения ширины ленты. Второй метод показывает только локальное вихревое поведение на центральной линии тока. В обоих случаях поверхность не является точной поверхностью тока, и условие касания верно только для исходных линий тока.

Построение поверхностей тока

Общая поверхность тока может быть построена путём генерации линий тока от каждой из нескольких точек на заданном *начальном отрезке (Rake)*. Если для всех линий тока используется один постоянный шаг по времени, то линии, соединяющие точки на всех линиях тока для одного момента времени, являются time lines. Таким образом, линии тока и временные линии образуют четырёхугольную сетку, из которой можно легко получить треугольники для визуализации.



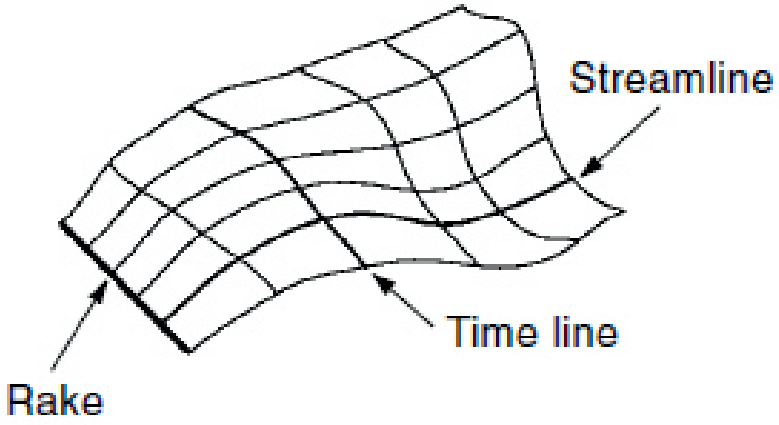
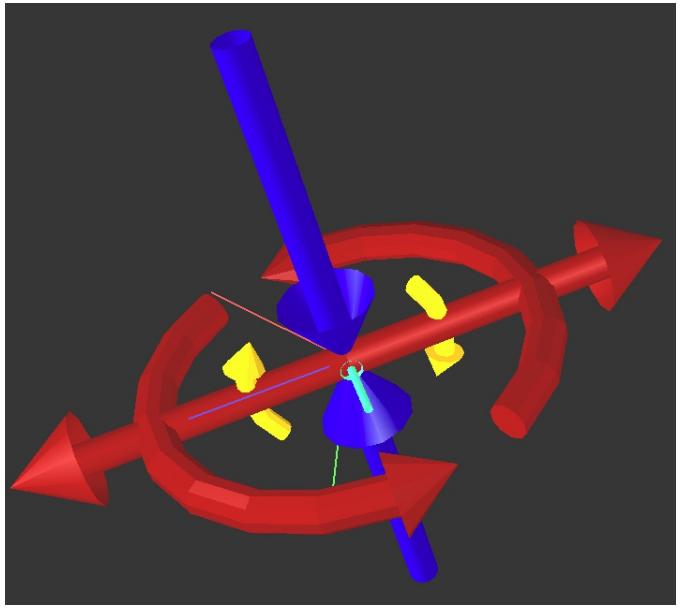


Figure 11. Mesh for a stream surface.

Построение поверхностей тока

При построении общей поверхности тока следует помнить о нескольких вещах. Если поток имеет большую дивергенцию, то соседние линии тока могут разойтись слишком далеко. Если в потоке есть объект и линии тока расходятся от его поверхности в разные стороны, то поверхность тока нужно разрезать так, чтобы она не наложилась на объект. Наконец, если градиент скорости в направлении потока будет достаточно большим, то ячейки сетки могут сильно искажаться, в результате чего будут получаться треугольники неправильного размера и плохой формы.
Для решения этих проблем был предложен *алгоритм продвижения фронта (advancing front algorithm)*. Поверхность строится в направлении потока путём добавления спереди полосы треугольников. Для учёта величины градиента в направлении потока используются адаптивные временные шаги. В этом случае все точки на передней границе продвинутся вперед примерно на одинаковое расстояние. Кроме того, если две соседние точки на передней границе расходятся слишком далеко друг от друга вследствие большой дивергенции, то в середине между ними следует начать новую линию тока. И наоборот, если две точки оказываются слишком близко друг к другу, одну из линий тока следует прервать. Если в потоке обнаруживается объект, то фронт следует разделить, и далее обе части следует строить раздельно.

Визуализация тензорного поля



Тензоры являются расширением векторных объектов до k-мерных массивов в каждом узле сетки. Тензорные поля часто возникают в инженерных и физических расчётах. Например, тензоры второго ранга используются для описания скоростей деформации и напряжения. Визуализация тензорных полей представляют собой трудную задачу даже для тензоров второго ранга. Простой тензор второго ранга может состоять из массива 3×3 скалярных значений, определённых в 3D-области. Трудно визуализировать такие данные напрямую, поэтому такие поля часто визуализируются с использованием скалярных или векторных методов визуализации.

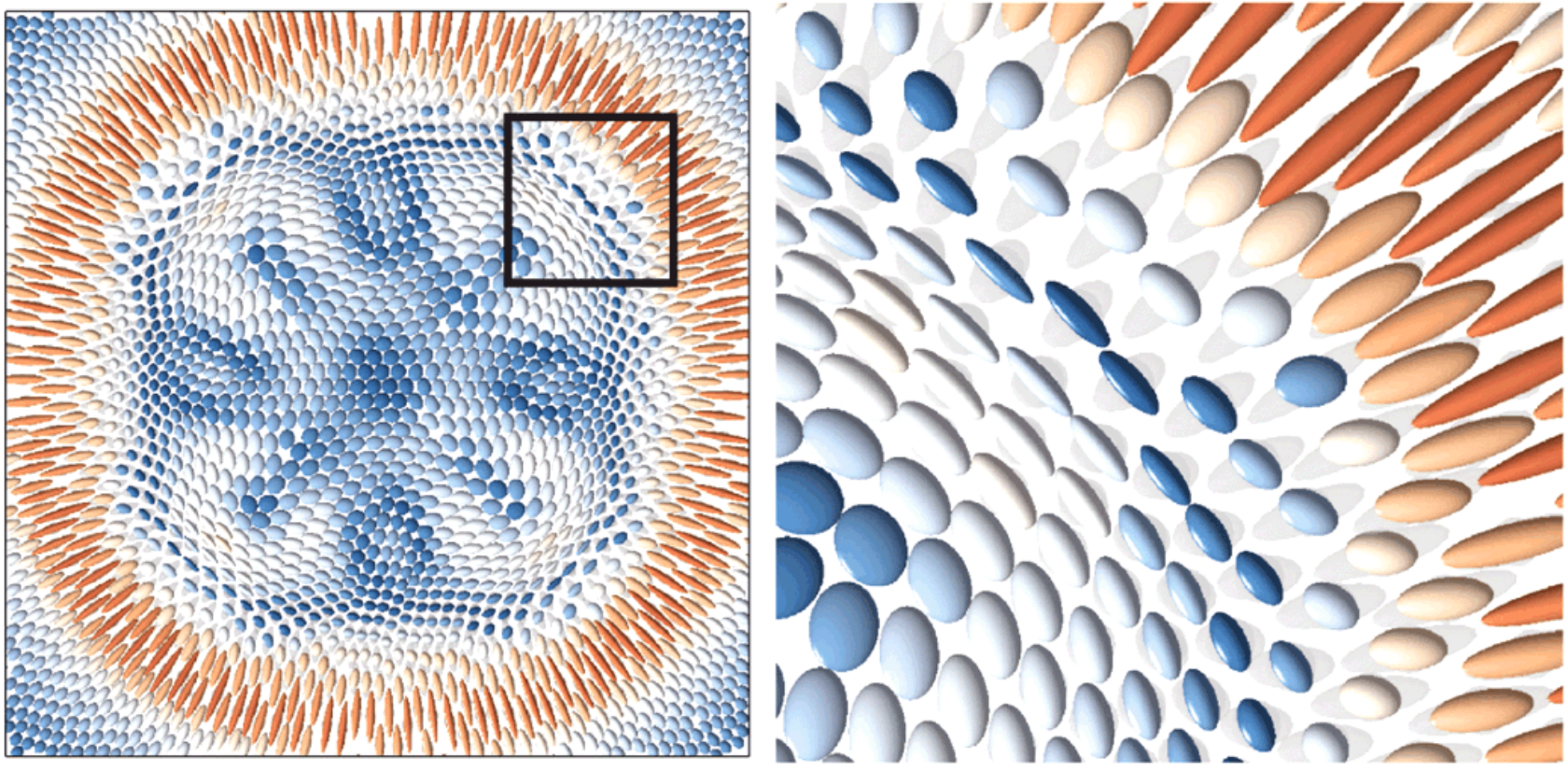
Визуализация поля симметричного тензора

Один часто встречающийся класс тензорных полей — поля симметричных тензоров, имеет специальные свойства, которые упрощают задачу визуализации. В частности, трёхмерные поля тензора второго ранга можно рассматривать как три ортогональных векторных поля. Каждое значение тензора может быть сведено к набору из трёх вещественных векторов, определяемых вещественными собственными значениями $(\lambda^{(i)})$ и единичными собственными векторами $(\mathbf{e}^{(i)})$ тензора:

$$\mathbf{v}^{(i)} = \lambda^{(i)}\mathbf{e}^{(i)}, \qquad i = 1, 2, 3.$$

Векторы упорядочены так, что $\lambda^{(1)} \geqslant \lambda^{(2)} \geqslant \lambda^{(3)}$ и $\mathbf{v}^{(1)}$ называют главным (*major*) собственным вектором, $\mathbf{v}^{(2)}$ средним (*medium*) собственным вектором среды и $\mathbf{v}^{(3)}$ — малым (*minor*) собственным вектором. Методы визуализации строятся на представлении этих трёх векторов различными способами.

Визуализация поля симметричного тензора



Самый простой метод — сопоставить каждый такой тензор с 3D-фигурой, например эллипсоидом или прямоугольной призмой. Главные оси эллипсоида совмещаются векторами $\mathbf{v}^{(i)}$. Модули векторов могут использоваться для задания длин осей фигуры и цветов, которые показывают сжатие или растяжение (отрицательные или положительные собственные значения).

Визуализация поля симметричного тензора

Поля симметричного тензора можно визуализировать по аналогии с трассировкой линий тока для векторных полей — путём прослеживания линий тензорных полей касательных к одному из трёх векторов $\mathbf{v}^{(i)}$ в каждой точке. Будем визуализировать линию тока в виде цилиндрической трубки с эллиптическим поперечным сечением. Это позволит всем трем векторам участвовать в построении линии тока. Каждая трубка, называемая гиперлинией тока (*hyperstreamlines*) строится по одному из собственных векторов, а два оставшихся вектора участвуют в формировании осей эллиптического сечения. Обычно трубка кодируется цветом в соответствии с величиной продольного вектора.

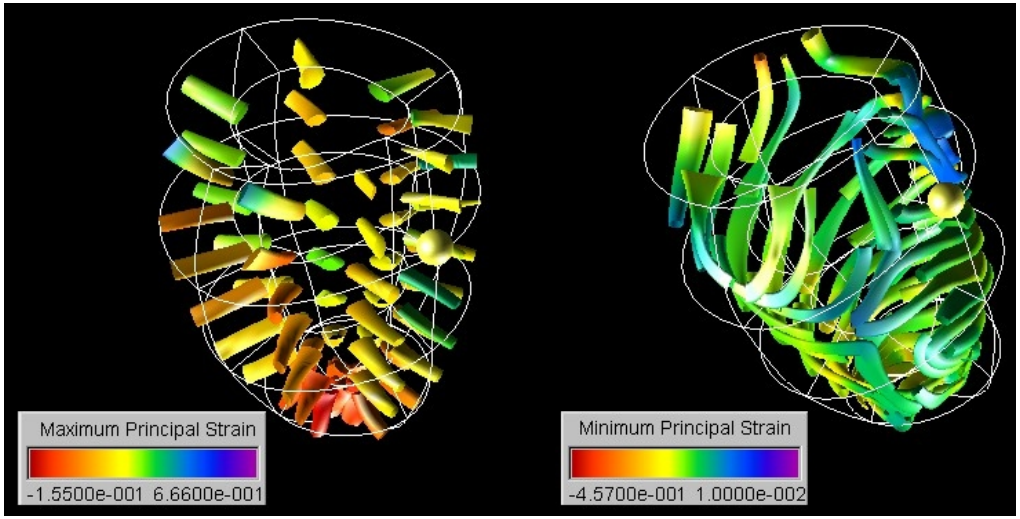
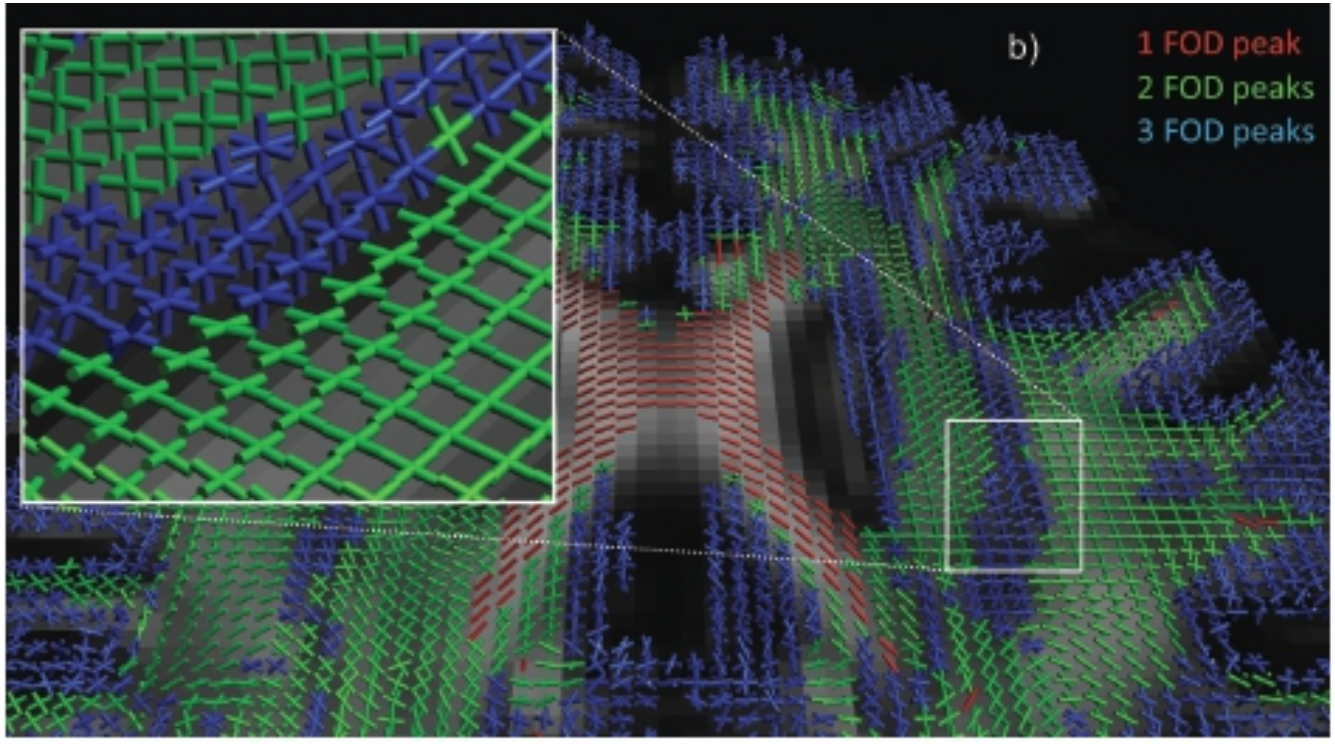


Рис.: Визуализация деформации миокарда

Визуализация поля симметричного тензора



В общем случае, можно использовать произвольный геометрический примитив, который строится вдоль траектории выбранного векторного поля, в то время как для выбора его размера используются другие два вектора. Использование креста в качестве формы профиля позволяет кодировать как величину, так и направление поперечных собственных векторов. Результат может выглядеть как спираль (*helix*), так как поперечные векторы меняют направление вдоль линии.

Заключение

Всегда следует помнить, что подобно тому, как целью вычислений является получение знаний, а не только набора цифр, для визуализации можно также сказать, что целью визуализации является получение знаний, а не только картинки. То есть результатам визуализации должен являться анализ происходящего процесса, а не только красивая картинка. Зачастую простой картинки может быть вполне достаточно, если она позволяет увидеть суть происходящего. Более того, простая картинка зачастую может быть даже лучше, чем очень зрелищная, поскольку артефакты могут скрыть важные детали из-за техники рендеринга или заскраски.