

ОПЫТ РАЗРАБОТКИ СПЕЦИАЛИЗИРОВАННЫХ СИСТЕМ НАУЧНОЙ ВИЗУАЛИЗАЦИИ

В.Л. Авербух, А.Ю. Байдалин, М.О. Бахтерев, П.А. Васёв, А.Ю. Казанцев, Д.В.
Манаков

ИММ УрО РАН, Екатеринбург

averbukh@imm.uran.ru, www.cv.imm.uran.ru

Оглавление

1. Введение	2
2. Универсальные и специализированные системы визуализации.....	2
3. Примеры специализированных систем научной визуализации.....	4
3.1. Система визуализации модели анализа загрязнения окружающей среды.....	4
3.2. Визуализация модели размножения и мутаций вирусов	6
3.3. Системы визуализации для задач оптимального управления и дифференциальных игр.....	8
3.4. Визуализация четырехмерных информационных множеств	11
3.5. Визуализация сеток	14
3.6. Система визуализации структуры области достижимости в задаче оптимального управления.....	18
3.7. Идея создания виртуального испытательного стенда.....	19
3.8. Проект среды разработки специализированных систем визуализации	20
4. Интерактивная визуализация параллельных и распределенных вычислений.....	21
4.1. Онлайн-визуализация	21
4.2. Пример: визуальное сопровождение решения обратных прикладных задач методами регуляризации и итерационной аппроксимации.....	24
4.3. Удалённая визуализация	27
4.4. Пример: удалённая визуализация результатов расчетов в пакете Deform.....	29
5. Заключение.....	30
Литература.....	31

Аннотация

Статья посвящена обобщению опыта разработки специализированных и персонализированных систем научной визуализации. Рассматриваются теоретические аспекты проблемы компьютерной визуализации и вопросы теории компьютерной метафоры и практики использования метафор при проектировании. В работе сделано обзорное описание разработанных в ИММ УрО РАН специализированных систем. В связи с этими разработками рассматривается проблематика онлайн и удаленной визуализации.

Ключевые слова: научная визуализация, компьютерная метафора, специализированные системы

Keywords: scientific visualization, computer metaphor, specialized systems

1. Введение

В статье рассматривается обобщение опыта разработки специализированных и персонализированных систем научной визуализации. При создании таких систем выделяются следующие составляющие:

- инженерия программного обеспечения;
- компьютерная графика и средства организации человеко-компьютерного интерфейса;
- аспекты, связанные с “человеческим фактором”.

В работе описывается опыт разработки специализированных систем научной визуализации. При этом значительное внимание уделяется “человеческим” проблемам, возникающим при проектировании визуальных и интерактивных систем. Задача выбора тех или иных методик генерации изображения затрагиваются только лишь в связи с разработкой специализированных систем. Также в связи с разработкой специализированных средств научной визуализации возникли задачи интерактивной визуализации супервычислений, которые относятся к проблематике инженерии программного обеспечения.

Очевидно, что **визуализация, понимаемая как зримое представление ментальных моделей**, существовала задолго до появления современной вычислительной техники. Более того, визуализацию, то есть перевод данных и информации в некоторые графические образы, можно рассматривать как составную часть нашей повседневной жизни. Официально научная визуализация как самостоятельная дисциплина оформилась после публикации в ноябре 1987 г. специального выпуска журнала Computer Graphics, содержавшего доклад “Визуализация в научных вычислениях” [1].

Визуализация, представляя результаты вычислений, обеспечивает интерпретацию и анализ полученных данных.

Под компьютерной визуализацией понимается методика перевода абстрактных представлений об объектах в геометрические образы, что дает возможность исследователю наблюдать результаты компьютерного моделирования явлений и процессов.

Традиционно выделяются следующие подобласти компьютерной визуализации:

- **научная визуализация;**
- **визуализация программного обеспечения;**
- **информационная визуализация.**

Отметим также, что иногда визуализация упрощенно понимается лишь как непосредственное отображение трехмерных образов (рендеринг) на некоторую плоскость вывода или даже как простой набор визуальных и иконических интерактивных методик. Эти (кстати, весьма, важные) вопросы следует скорее отнести к проблематике компьютерной графики и человеко-машинного взаимодействия.

Имеет место ряд связанных между собой базовых понятий, который включает в себя **метафору визуализации, язык визуализации, вид отображения, конкретный графический вывод.**

2. Универсальные и специализированные системы визуализации

Уже упоминавшийся доклад “Визуализация в научных вычислениях”, программа в области научной визуализации, официальное признание новой дисциплины и потребностей научных вычислений вызвали настоящий бум в разработке средств научной визуализации. Следует отметить две тенденции развития систем визуализации. С одной стороны - разработка универсальных средств визуализации, а с другой – специализация по всем направлениям, вплоть до создания специальных графических станций с реализацией для данного случая графическим алгоритмическим и программным обеспечением.

Характерным примером универсальной системы визуализации служит Open Visualization Data Explorer (DX), разработанный уже очень давно компанией IBM. (Первоначально эта система носила коммерческий характер, а в последствие стала доступна для свободного распространения.) Эта система характеризуется богатым набором средств визуализации и стандартных способов представления модельных сущностей - видов отображения. Кроме того, в состав DX входит набор неграфических функций, например, набор математических выражений, статистические функции, средства обработки изображений и пр. Важными являются реализованный в системе подход к описанию визуализации, основанный на потоке данных и поддерживающий конвейер визуализации (фильтрация, мэппирование, рендеринг). Для описания визуализации DX предлагает визуальный язык на базе потока данных. Кроме этой системы популярными (хотя и также устаревшими) примерами универсальных систем визуализации в учебной литературе часто служат системы AVS/Express и IRIS. До сих пор популярная система Visualization ToolKit (VTK), по сути, является библиотекой процедур, служащей для разработки специализированных систем. Средства визуализации в пакетах прикладных программ также можно рассматривать как универсальные системы визуализации. Те математические объекты, расчет которых описывается средствами пакета, можно и визуализировать, используя имеющиеся приемы и виды отображения с фиксированными параметрами. Часть пакетов для описания визуализации использует визуальные языки на базе потока данных. Важной особенностью универсальных систем является наличие типового набора видов отображения для типовых математических объектов. Тем самым, они могут дать инструмент для разработки специализированной визуализации. Задача пользователя – так описать связь между модельными сущностями, которые необходимо визуализировать, и стандартными видами отображения, чтобы увидеть интересующие его особенности изучаемых объектов. Это зачастую требует от пользователя значительных усилий, а иногда вывод важных особенностей практически неосуществим. В принципе, при помощи универсальных систем можно отобразить любые математические объекты. Другой вопрос, какие усилия требуются от пользователя, причем не только (и не столько) при самой визуализации, сколько при интерпретации результатов. Вообще отметим, что полностью универсальных систем визуализации не существует. По сути, универсальные системы содержат некоторый стандартный набор (“универсальных”) видов отображения и приемов визуализации.

Специализированные системы облегчают работу пользователя, а, в случае исследования принципиально новых модельных объектов, только за счет их использования можно получить наглядное представление об их природе и особенностях.

Существует несколько подходов к обеспечению специализации - по объектам визуализации, по использованию методик компьютерной графики, по научным направлениям, для которых разрабатываются соответствующие системы. Имеет место подробная номенклатура отраслей научной визуализации, соответствующих научным отраслям и дисциплинам. Встречаются такие подразделы научной визуализации, как биомедицинская визуализация, химическая визуализация, визуализация аэро и гидродинамики, географическая визуализация и т.п. Возможна разработка систем, пригодных для решения только одной, хотя и мощной задачи. Интересно, что такого рода специализация приводит к созданию специальных суперкомпьютеров, которые можно рассматривать как специализированные вычислители, в целом ориентированные на строго

определенные проблемы, например, на задачи газовой динамики. Изучение статей, касающихся отдельных отраслей научной визуализации, позволяет уяснить, что каждое направление содержит свои (хотя иногда и совпадающие с другими отраслями) наборы моделируемых сущностей, объектов изучения и визуализации, методик выделения этих объектов, например, методики выделения изоповерхностей. Для каждого подраздела визуализации, связанного с тем или иным научным направлением, характерны наиболее важные объекты визуализации, выбор той или иной естественной или привычной образности, набор видов отображения и т.п. (Характерные примеры всего этого можно найти в системах визуализации потоков. См., в частности, [2] или [3].)

3. Примеры специализированных систем научной визуализации

Ниже мы рассмотрим ряд примеров специализированных систем научной визуализации, спроектированных и разработанных в ИММ УрО РАН.

3.1. Система визуализации модели анализа загрязнения окружающей среды

Специализированная система научной визуализации для модели загрязнения окружающей среды разрабатывалась несколько лет назад. Первоначальная задача на визуализацию предполагала использование образности, связанной с источником моделирования. Попросту говоря, мы должны были отобразить как загрязнение, возникающее от деятельности промышленного предприятия, распространяется в воздухе (дым из заводской трубы) и в воде (сточные воды). Ставилась задача создания достаточно реалистической анимации, которая могла представлять определённый интерес для администрации завода, региональных властей или защитников окружающей среды. Именно они в данном случае становились “потребителями” модели. То есть задачей визуализации становилось привлечение внимания к проблемам экологии, а её предметом - сам процесс загрязнения.

Однако анализ показал, что пользователями системы являются специалисты в области математического моделирования, поскольку данная научная проблема связана не столько с экологией того или иного заводского посёлка, сколько с самой моделью загрязнения. Дело в том, что для обеспечения экологической безопасности проводится мониторинг выбросов промышленными предприятиями загрязняющих субстанций в окружающую среду. Как правило, прямые замеры концентрации загрязняющей субстанции вблизи источника загрязнения по ряду причин невозможны. Используются косвенные замеры (например, усреднение концентрации) в некоторых областях (где есть возможность разместить датчики). Суть задачи сводится к восстановлению интересующих величин (интенсивности выбросов) по некоторой доступной информации. Таким образом, предметом визуализации оказывается вовсе не процесс распространения загрязнения, а сама модель. В этом случае для визуализации модели такого типа вполне закономерно использование трёхмерной абстрактной образности. Метафора высоты, пожалуй, более других подходящая для описания качественной стороны процесса, в применении к двумерному полю концентрации загрязняющей субстанции, порождает трёхмерную поверхность.

Как известно, основой системы визуализации является совокупность комплексных видов отображения и системы управления, осуществляющих их взаимодействие.

Для осуществления анализа и интерпретации был необходим целый ряд дополнительных манипуляций с поверхностью, например, построение сечений, задание

изолиний и изоповерхностей. При проектировании оказалось, что одного единственного вида отображения явно недостаточно, так как ввод информации может оказаться сложным, а вывод – “засорённым”, тяжёлым для восприятия из-за перегруженности деталями. При этом визуальной информации, может оказаться недостаточно для анализа.

Для упрощения манипуляций и большей гибкости интерфейса одновременно с основным выводятся вспомогательные виды отображения, см. рис. 1.

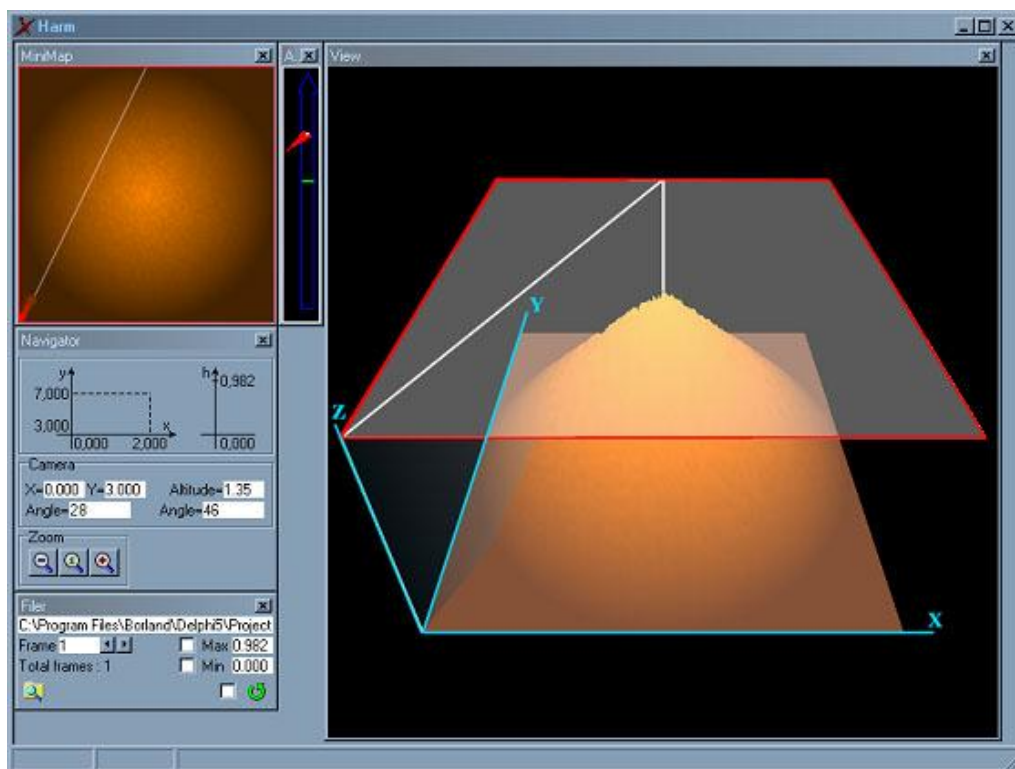


Рис. 1. Общий вид системы визуализации моделей загрязнения окружающей среды.

Один из них (в целом, идентичный первому) также отображает поверхность, однако она оказывается жёстко ориентированной в трёхмерном пространстве. Как следствие, обеспечивается более полное восприятие модели в целом, а также более удобный доступ к её элементам. Упрощается ввод и анализ вспомогательных данных, расположенных в плоскости XY (например, построение сечений, перпендикулярно к этой плоскости). Аналогом такого вида отображения является вид отображения типа MiniMap, который одно время активно использовался в пакетах обработки растровых изображений и в компьютерных играх.

Ещё один вид отображения предоставлял механизм ввода и анализа данных, расположенных в плоскости, перпендикулярной к XY. Alt-view (назовём его так), делает удобнее ввод и анализ данных, расположенных вдоль оси Z (например, построение линий уровня). Совокупность MiniMap и Alt-view позволяет однозначно задать любую точку трёхмерной модели. В данном случае такая система используется для управления камерой. Таким образом, обеспечивается навигация в двух перпендикулярных плоскостях. Это даёт распределённость управления, удобную в большинстве случаев (например, если необходимо изменить плоскостные координаты камеры, не меняя её высоты). По мнению разработчиков, такой подход, не сильно распределяя внимание, обеспечивает простоту ввода управляющей информации со стороны пользователя и адекватность реакции на неё со стороны системы визуализации (результат, выполнения операции совпадает с тем, что ожидал пользователь). Изменение положения камеры влечёт за собой соответствующие изменения в основном окне вывода.

В этом плане подход к созданию системы манипуляции отличается от других, в частности от всем известного Direct Manipulation, которая в случае многомерной модели, представляется весьма ограниченной, вследствие двумерности экрана. Раздельная схема вывода информации и система настройки интерфейса позволяют в любой момент скрыть ненужные в данный момент виды отображения, оставив на экране только необходимую информацию.

Следует отметить, что многооконность интерфейса не обязательна, но, безусловно, удобна, в первую очередь в силу привычности для пользователя (например, для того чтобы скрыть один из компонентов, достаточно закрыть его окно). Ввод информации в систему также сделан распределённым. Это даёт возможность указать необходимые точки, используя MiniMap и Alt-View, или ввести их точные координаты в соответствующие поля. В любом случае, результаты первого метода можно уточнить вторым.

В системе применены два альтернативных способа управления камерой - численный и визуальный. Такой же (мультимодальный) подход применяется и к заданию иных характеристик визуализации.

Предложенные в данной системе решения были применены при проектировании ряда специализированных систем визуализации.

3.2. Визуализация модели размножения и мутаций вирусов

Другой, на наш взгляд, интересный урок поиска метафор и, прежде всего, самой образности можно получить из опыта разработки средств визуализации для вычислительной модели размножения и мутаций в клонированной популяции вируса ЕСНО 11. (Эта работа также проводилась несколько лет назад.) Заказчиками системы являлись, математики, разработавшие методы математического описания интересующих нас процессов, а конечными пользователями – медики-вирусологи. Одной из целей работы была демонстрация медикам и биологам возможностей, как самого моделирования, так и методик визуализации. Таким образом, визуализация, в значительной мере, становилась иллюстративной. Предметом визуализации являлся сам процесс размножения и мутации, данные о котором были получены из моделирующей программ. Нужно было, во-первых, отобразить изменение количества объектов от единиц до величин порядка величин, 10^7 - 10^8 , а, во-вторых, показывать детали процесса, как-то деление вируса, появление вируса-мутанта, обратное превращение мутанта в “обычный” вирус.

Такая постановка делала основным видом отображения анимацию процесса модели размножения и мутации. Анимация использовала условные образы и набор приёмов, заимствованных из мультипликационного кино. При этом конкретные и точные параметры процесса пользователь может увидеть в текстовой таблице, простом, но, тем не менее, очень информативном виде отображения. Данные в таблице в данном случае отображаются непосредственно из входного файла. Текстовые данные не требуют никаких усилий пользователя для интерпретации. С другой стороны, анализ процесса проводить по таблице затруднительно. Динамику процесса легче отследить по динамически строящемуся графику. Так как мы имеем дело с большим диапазоном значений данных, то строить его надо в логарифмических координатах. По одной оси откладывается время, а по другой – десятичный логарифм интересующей нас величины. Используемая же “живая” картинка при всей своей схематичности, простоте образов и низкой реалистичности все же дает пользователю шанс заглянуть внутрь природного процесса, возможно даже абстрагироваться от численного описания.

Остановимся на описании методик создания анимации .

При проектировании серьёзной проблемой стал поиск адекватных образов для изображения вирусов, мутантов, а также способов отображения изменения их количественных характеристик.

Первое решение, которое было принято – это использовать в качестве портрета вируса – шарик. Вирус, по сути, является сложной молекулой, но в условиях, когда на экране показываются сотни и сотни объектов, погоня за точностью геометрических форм не имеет смысла. Тем более, во многих изданиях вирусы изображаются именно в сферической форме. То есть увиденное не войдет в противоречие с предыдущим опытом наблюдателя, и у него возникнут правильные ассоциации с изображением на экране. Простой геометрический образ, кроме того, как бы подтверждает, что не визуализация претендует на достоверность. Таким образом, визуализация не претендует на достоверность увиденного, а только лишь даёт общее описание модели.

Отличия от вируса мутанта также реализовано за счёт простых методик. Вполне естественно изображать мутант также шариком, но другого цвета (рис. 2).

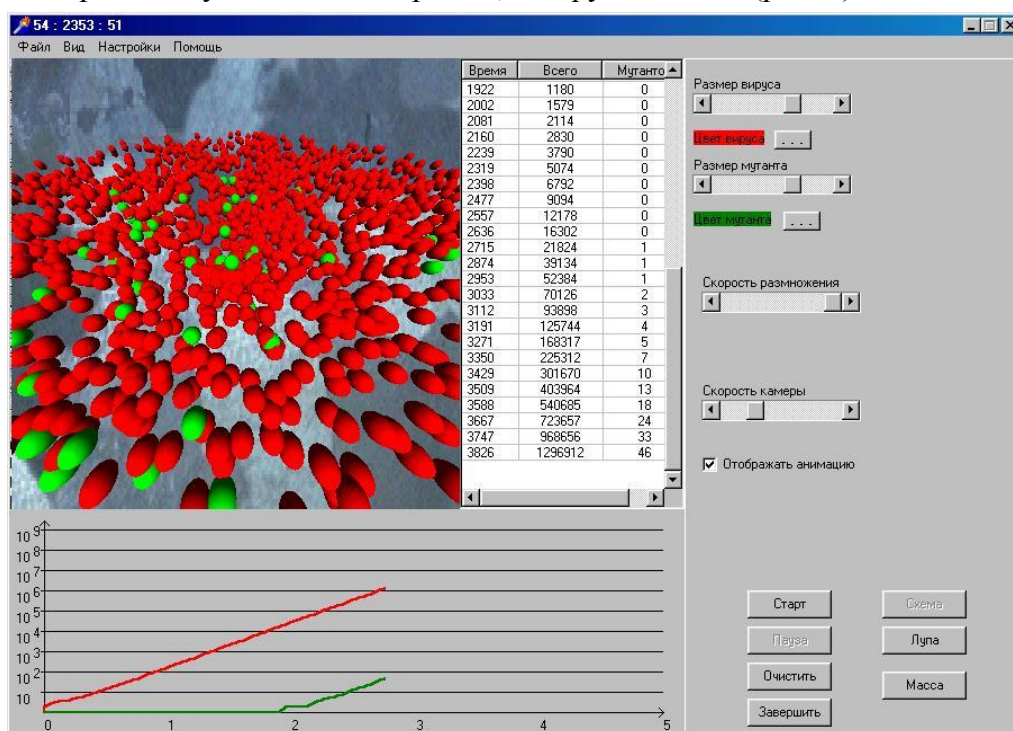


Рис. 2. Скриншот анимации размножения и мутации вирусов.

Теперь рассмотрим, каким образом строится анимация.

Отметим, что всякий раз, когда встает задача расположения на экране большого количество объектов, на помощь приходит третье измерение. В нашем случае на трёхмерной сцене можно эффективнее донести до пользователя количественные характеристики. Создается ощущение глубины, оно может быть усилено при помощи изменения угла зрения (перспективы) и наложения фона картинке. Как показали проведенные тесты, нет смысла рисовать больше 1500-2000 вирусов на экране. Уже такое количество объектов создает у зрителя ощущение массы. И вот новых объектов на экране больше не появляется. У наблюдателя мы смогли создать впечатление, что вирусов много, но дальше встает другой вопрос: как при таком количестве объектов на экране заставить его поверить, что оно (количество) по-прежнему продолжает расти? Тут был применен эффект шевелящейся массы – новые объекты не рисуются, но происходит хаотичное движение уже нарисованных. За счет этого изображение не теряет динамичность. Было использовано несколько несложных приемов, которые, тем не менее, дали неплохой

эффект. До определенного предела пользователь вообще не обманывается. На экране рисуется реальное количество вирусов. Затем, появляется разница между данными, которые считываются из входного файла (эти числа честно выводятся в текстовой таблице и отображаются на графике) и количеством прорисовываемых объектов. Кроме того, эффект увеличения массы достигается за счет того, что картинка как бы наплывает на пользователя, а также за счет того, что вирусов становится все больше - они уже не помещаются в области рисования и появляется эффект неизмеримости. Выпишем набор использованных приёмов.

1. Использование “перспективы”.

Изображалась масса вирусов в виде слоя, заполненного маленькими сферами в плоскости параллельной плоскости экрана. Эксперимент показал, что большей изобразительной силой картинка будет обладать, если изменить угол зрения.

2. Масштабирование Невозможно рисовать слишком много объектов, поэтому, чем больше их становится, тем меньше процент их отрисовывается. Таким способом достигается своего рода эффект масштабирования.

3. Эффект постоянного “приближения” или “удаления” пользователя от изображения. Когда на картинке появляются новые объекты-вирусы, а старые просто остаются на месте, то зрителю проще оценить, сколько именно объектов добавлено на данной итерации. Его внимание фокусируется только на появляющихся вирусах. В условиях, когда рисуется заведомо меньше элементов, чем объявляем пользователю, нужно создать у пользователя ощущение, что на экране прибавляется много объектов. Это достигается за счет изменения расстояния от камеры (наблюдателя) до изображения. При этом если происходит приближение к изображению, то появляется эффект наплыва, заполнения изображением экрана.

4. Эффект шевелящейся массы.

На определенном этапе прекращается дорисовка новых объектов. Вместо этого продолжается “наплыв” изображения, а также хаотичное движение вирусов по экрану. Передвижения совершаются с небольшой амплитудой. Поэтому у наблюдателя создается ощущение такой живой подвижной массы.

Кроме количественных характеристик процесса еще тем или иным способом необходимо демонстрировать его такие качественные характеристики как деление вируса, мутация вируса и обратная мутация. Вполне естественным решением было попробовать реализовать следующую схему: общий план остается на месте, а на интересующий объект (вирус или мутант) происходит “наезд”. И в новом окне или в выделенной части того же окна можно наблюдать более подробно за жизнью “героя”. Выбор объекта для наблюдения может происходить, как интерактивно – например щелчок мыши пользователем, так и быть заранее продуманным и прописанным в сценарий анимационного фильма. Таким образом, счёт реализации стандартной идеи “зуминга” (zooming) показываются детали процесса.

3.3. Системы визуализации для задач оптимального управления и дифференциальных игр

Наверное, наиболее важным для нас являлась разработка специализированных систем визуализации для задач оптимального управления и дифференциальных игр. Эти работы начались в середине 90-ых годов и продолжают по сей день.

Рассмотрим опыт создания системы, а точнее нескольких вариантов, систем визуализации построения максимальных стабильных мостов в линейных дифференциальных играх [6]. Для задач с фиксированным моментом окончания и терминальной функцией платы каждое множество уровня функции цены является максимальным стабильным мостом. Если функция платы зависит только от двух координат фазового вектора в момент окончания, то стандартным является переход к эквивалентной дифференциальной игре второго порядка. Построение мостов при этом ведется в трехмерном пространстве, определяемом двумерной фазовой переменной и временем. Каждая трубка (максимальный стабильный мост) соответствует определенному значению цены игры. Трубка задается набором параллельных двумерных многоугольных сечений, перпендикулярных оси времени. Сечение определяется упорядоченным списком координат своих вершин. Средства визуализации отдельных мостов или системы мостов должны дать информацию об устройстве функции цены и ее особенностях.

Пользователями системы заведомо являются специалисты-математики, которые используют полученные изображения для анализа результатов использования численных методов решения соответствующих задач, а также для демонстрации полученных результатов. При этом большое внимание уделяется именно особенностям полученных решений, а также различным вырожденным случаям.

Традиционные методы визуализации этих объектов заключались в построении проекций контуров сечений на плоскости, параллельные осям координат. При увеличении числа сечений изображение становится перегруженным. Одновременное же отображение нескольких трубок в этом случае является еще более затруднительным. А задача состоит в том, чтобы показать объект в виде поверхности, со всеми ее особенностями. Под особенностями, в основном, понимаются нарушения гладкости, которые могут зарождаться и исчезать на сечениях трубки с течением времени. Далее, чтобы получить представление о функции цены игры в целом, нужно обеспечить одновременное отображение нескольких трубок, построенных для разных значений цены. При этом на изображении должны восприниматься как конкретные мосты, так и вся структура. Необходимы интерактивные средства трёхмерной визуализации отдельных мостов или набора мостов, средства изображения объектов с разных положений, чтобы дать информацию об устройстве функции цены и ее особенностях. После нескольких попыток были спроектированы два вида отображения мостов. Первый - параллельная проекция контуров сечений на произвольную плоскость. Этот вид отображения служит для поиска такого ракурса, который наиболее удовлетворяет пользователя. Когда удачный ракурс найден, используется второй вид отображения - восстановленная с помощью триангуляции по контурам сечений поверхность, которая изображается с закраской Гуро. Разработанный алгоритм триангуляции позволяет за счет увеличения соответствующего угла подчеркивать негладкость поверхности трубки, представляющую особенности функции цены игры. Таким образом, в этом случае было использовано *визуальное утрирование особенностей* для их быстрого опознания.

Интерактивные средства управления выводом изображения и изменения видов отображения информации дают возможность изменять положение и ориентацию объекта в трехмерном пространстве, масштаб вывода изображения, положение источника света и секущих плоскостей, изменять ракурс вывода, а также отображать мосты в различных режимах прозрачности.

В результате была разработана первая версия системы, позволяющей в интерактивном режиме рассматривать трехмерное изображение трубок с разных положений. Далее было реализовано несколько версий этой системы визуализации, что вызвано как непрерывной сменой аппаратных и программных возможностей компьютерной графики, так и появлением новых подходов к визуализации изучаемых

объектов. В настоящий момент завершается очередной цикл перепроектирования и перепрограммирования этой специализированной системы, см. рис. 3.

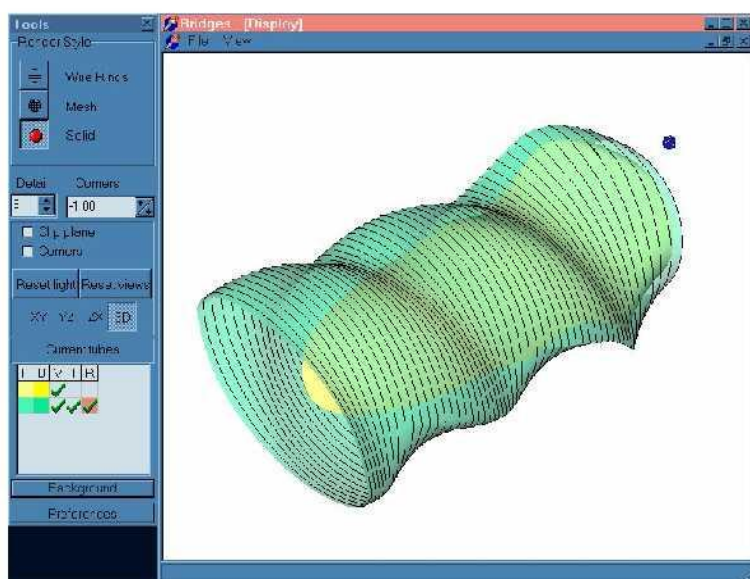


Рис. 3. Визуализация множества функции цены игры.

Разработка системы визуализации для задач построения максимальных стабильных мостов в линейных дифференциальных играх в свою очередь является существенным результатом, позволяющим активизировать исследования по данной теме. Ещё одна работа также была связана с разработкой специализированных средств визуализации решений дифференциальных игр.

При решении игровых задач быстродействия второго порядка [7] полезным является изображение графика функции цены (времени оптимального быстродействия). График функции цены - поверхность в трехмерном пространстве, имеющая достаточно сложное устройство. Возможен случай, когда информация о функции цены поступает в виде набора фронтов (линий уровня функции цены). Фронты просчитаны с определенным шагом по времени. Каждый фронт является ломаной, заданной своими вершинами. Соседние фронты могут иметь различное число компонент связности. Для восстановления поверхности был выбран метод триангуляции. Суть специализированного алгоритма заключается в следующем. На фронтах выделяются точки, в которых линии резко изгибаются. Этими точками фронты разбиваются на несколько компонент так, что в пределах одной компоненты нет резких изгибов. Затем находится соответствие между компонентами на соседних фронтах. Между соответствующими компонентами производится триангуляция. Алгоритм позволяет выделять места, где нарушается гладкость и непрерывность графика функции цены. Система включает в себя средства для изменения ракурса, "материала" поверхности, положения источника света. Имеется возможность плавного изменения цвета поверхности в зависимости от значения функции (см. рис. 4).

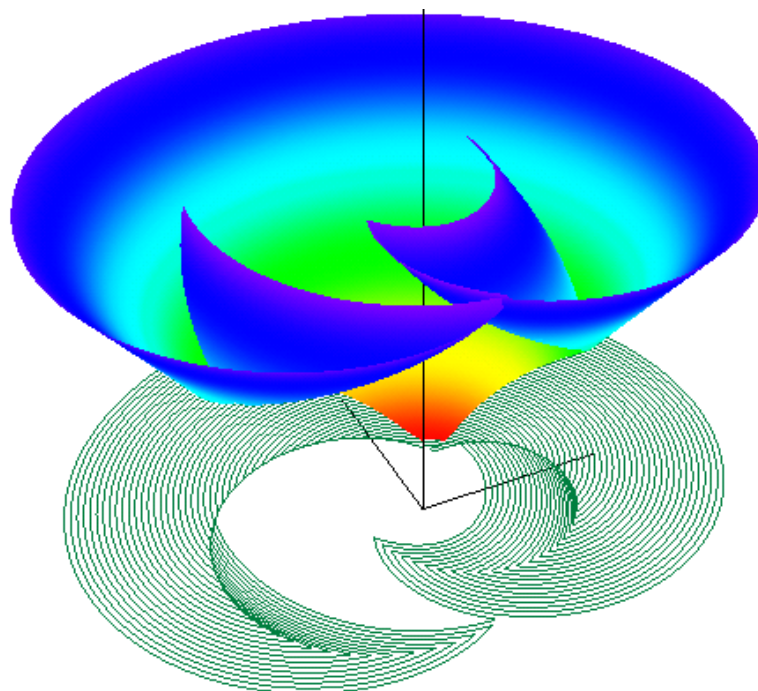


Рис. 4. Трехмерное отображение фронтов уровня цены игры.

Важное требование к системам визуализации заключается в том, что алгоритмы, заложенные в системе, должны сами находить интересные особенности в данных, описывающих математические объекты, чтобы затем на них могло концентрироваться внимание пользователей. При выборе соответствующих алгоритмов для упомянутых выше систем визуализации это требование в целом было учтено. В обоих случаях алгоритмы рисования построены с учетом “инсайдерской” информации, полученной от пользователей-математиков. По сути, они являются участниками разработок. Мы уже отмечали, что самое значительное внимание при визуализации в данном случае уделяется изучению сложных и вырожденных случаев. Как было указано, полученные от пользователей данные о природе математических объектов, в частности, позволили за счет определенного утрирования визуально выделить малозаметные нарушения гладкости и разрывы непрерывности в решениях дифференциальных игр. Важным также является учет замечаний по пользовательскому интерфейсу, включая требования по размещению источников света, манипулированию объектами и пр.

3.4. Визуализация четырехмерных информационных множеств

Во второй половине 90-ых годов ходе исследований мы столкнулись с задачей, в которой было необходимо отобразить некое подмножество четырехмерного пространства. В синтетической органической химии изучаются реакции, протекающие при процессах динамического кинетического расщепления. Важную информацию о таких реакциях можно извлечь из данных о кинетике их протекания. В частности, для оценки эффективности процесса расщепления необходимо знание скоростей его элементарных реакций. Для нахождения скоростей некоторой реакции обычными статистическими методами требуется провести большое число экспериментов. Из-за высокой стоимости компонентов, участвующих в реакции, это весьма неэкономично. Поэтому были применены методы компьютерного моделирования, что позволило обойтись малым числом экспериментов [8].

При проведении химического опыта в определенные промежутки времени измеряются концентрации интересующих веществ. Затем строится математическая модель, описывающая поведение этих концентраций. Ее параметрами служат скорости протекания элементарных реакций. Некоторый набор скоростей считается допустимым, если при моделировании процесса расщепления концентрации веществ меняются в соответствии с экспериментальными данными. При этом учитываются погрешности замеров в эксперименте. Множество всех допустимых наборов скоростей есть информационное множество. В нашем случае реакции всего четыре, поэтому информационное множество состоит из четверок чисел вида. Таким образом, задача нахождения четверки скоростей элементарных реакций свелась к задаче нахождения информационного множества. Программа нахождения этих множеств была реализована на суперкомпьютере. Выходной файл при каждом проходе этой программы насчитывал порядка миллиона четырехмерных точек. Была поставлена задача изучения геометрического строения множества, что позволяло исследовать взаимосвязь элементарных реакций, участвующих в процессе расщепления. Так встает задача изучения геометрии информационных множеств, а вместе с ней - проблема их графического представления. Таким образом, был востребован метод визуализации, с помощью которого можно было бы исследовать локальные и глобальные характеристики расположения множеств в четырехмерном пространстве. К ним относятся, например, локальное строение окрестностей, связность, выпуклость и ориентация в четырехмерном пространстве.

Мы уже упоминали тезис доклада по визуализации 1987 года о том, что “визуализация есть способ увидеть невидимое”. В данном случае “невидимость” четырехмерных множеств связана с возможностями восприятия человека. Очевидно, что при визуализации четырехмерного информационного множества не существует естественной образности для многомерных множеств. В литературе описаны некоторые приемы представления четырехмерных множеств, однако, все они оказались неприемлемы для данного случая. Было необходимо придумать методику хорошо интерпретируемого представления таких множеств. Исходя из знаний об априорной структуре множеств и цели, поставленной перед визуализацией, и после длительного анализа структуры множеств, целей моделирования и опыта представления многомерных объектов был предложен ряд методик визуализации четырехмерных множеств. В ходе консультаций с заказчиками-математиками среди предложенных видов отображения были выбраны приоритетные, которые легли в основу комбинированного метода и созданной системы многомерной визуализации.

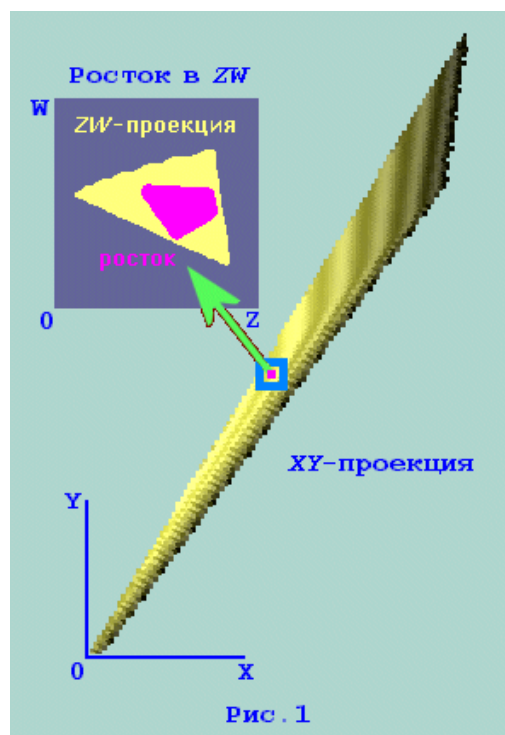


Рис. 5 Структура вида отображения «Росток».

Успех в данном случае связан с возможностью разделения координат четырехмерной точки на две сравнительно независимые пары, так, что можно построить связанные между собой наборы видов отображения, представляющих двумерные и трехмерные проекции четырехмерного множества. Для лучшей интерпретации вводится специальный визуальный объект, так называемый, росток точки, связанный с представлением двумерных сечений множества, где секущая гиперплоскость задается с помощью выбора точек из двумерной проекции (см. рис. 5). Пользователь в интерактивном режиме выбирает одну точку проекции и наблюдает строение ее ростка, тем самым, получая возможность визуально оценить его конфигурацию. Был разработан комплексный подход к визуализации, в основе которого лежит идея интерактивного взаимодействия с пользователем с использованием нескольких видов отображения (рис. 6).

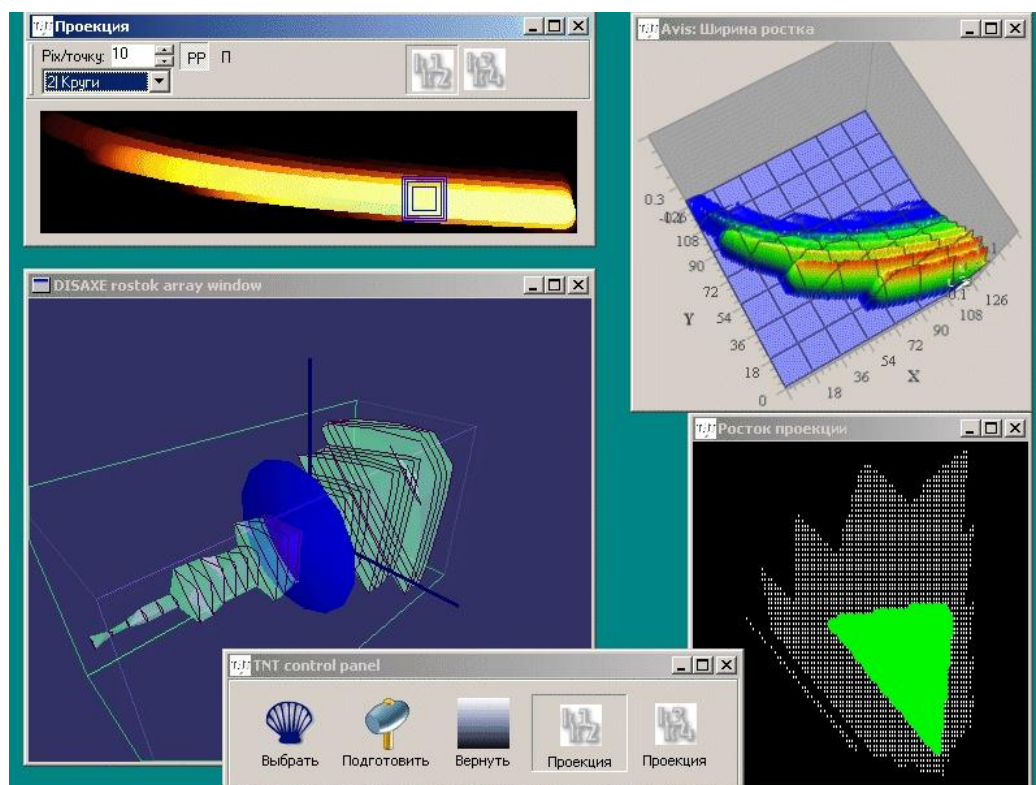


Рис. 6. Пример внешнего вида системы многомерной визуализации.

При этом каждый вид отображения связан с другими, что позволяет проводить навигацию по множеству и извлекать все его необходимые свойства. Таким образом, в ходе реализации системы был предложен новый подход для представления информационных множеств, что в принципе решает задачу их представления. Роль пользователей при решении данной задачи – очень существенна. Без передачи ими знаний о сути явлений задача в принципе не могла быть решена. Они также активно участвовали в работе по проектированию, отбирая среди значительного числа метафор визуализации и построенных на их базе видов отображения тот набор, который привёл к решению проблемы. В тоже время разработанные методики представления многомерных пространств сильно связаны с конкретной моделью и спецификой её реализации. Для других проблем, возникающих при визуализации многомерных пространств, необходимо искать свои метафоры и строить свой набор видов отображения.

3.5. Визуализация сеток

В течение ряда лет нами активно разрабатывалась система визуализации для задач математической физики, в частности, связанных с сеточными методами вычислений. Основные данные, получаемые в результате расчетов для аэро- и гидродинамических задач – это массивы значений газодинамических параметров в узлах расчетной трехмерной сетки. Поля газодинамических величин могут быть как скалярными (например, давление, плотность, число Маха, энтропия), так и векторными (поле скоростей). Кроме того, исследователей интересует не только пространственная картина течений, но и распределение величин (плотности, давления) на поверхности. Проект системы визуализации параллельных вычислений предполагал ее функционирование по следующей схеме:

- задача считается на параллельном вычислителе и оставляет (окончательный или промежуточный) результат в файлах или в памяти процессоров вычислителя;
- на параллельном же вычислителе проводится некоторая предобработка и фильтрация данных, связанная с будущей визуализацией;
- по заданию пользователя строятся модули описания параллельных фильтров и видов отображения, на вход которых подаются данные после предобработки (на первых этапах предлагается использовать predeterminedенные в системе виды отображения);
- сама визуализация может идти как на графической рабочей станции, так и на параллельном вычислителе.

Рассматривалось два варианта использования системы – в off-line (после окончания счета) и on-line (по ходу счета параллельной программы) режимах. Также рассматривается два варианта размещения средств предобработки и визуализации на процессоры параллельного вычислителя – на те же процессоры, где происходил счет или на дополнительно выделенные процессоры вычислителя.

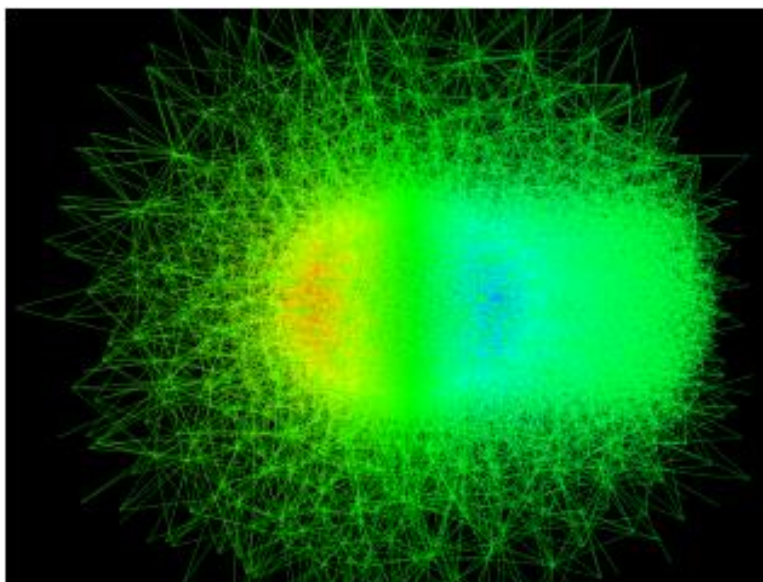


Рис.7. Фрагмент отображения тетраэдральной сетки.

В первом варианте системы (см. рис. 7) рассматривалась регулярная сетка, которая состоит из набора стандартных модулей, например, “прямоугольник”, “квадрат-цилиндр”, “полусфера”, “квадрат” и “законная область”. При этом каждый модуль состоит из нескольких блоков регулярных сеток, ограниченных границами. Программа должна наглядно отображать сеточные данные больших объемов с возможностью поиска аномалий в структуре сетки. Был разработан комплексный вид отображения с элементами интерактивного взаимодействия. При этом в один вид отображения заложена способность к изменению множества параметров, существенно влияющих на адекватность восприятия его содержимого в конкретной ситуации. Предполагалось, что может меняться размерность, метод рендеринга, набор доступных функций пользовательского интерфейса, все это будет (возможно, автоматически, в зависимости от некоторого набора внешних факторов). Так, например, при построении сечений сетки плоскостью, удобно спроецировать вид отображения на плоскость перпендикулярную секущей. Использование нескольких методов рендеринга в одном виде отображения заметно повышает его универсальность. Возможности полигональной графики позволяют очень детально описать блочную структуру сетки, иерархию составных частей и особенности ее внутреннего устройства. Воксельная графика подчеркивает прикладное значение

сеточных задач, отображая не дискретное множество точек, а трехмерную модель в виде непрерывного массива информации.

Для отображения сеток на базе полигональной графики была разработана среда, которая представляет собой специализированную систему визуализации по выводу сеточных данных с управляемой камерой и системой фильтрации данных (рис. 8). Навигация в системе представляет собой совокупность функций управления камерой. При этом используется проекционная система манипуляции данными. Функции манипуляции данными разделены на группы, такие как “перемещение”, “вращение”, “построение сечений” и т.д.

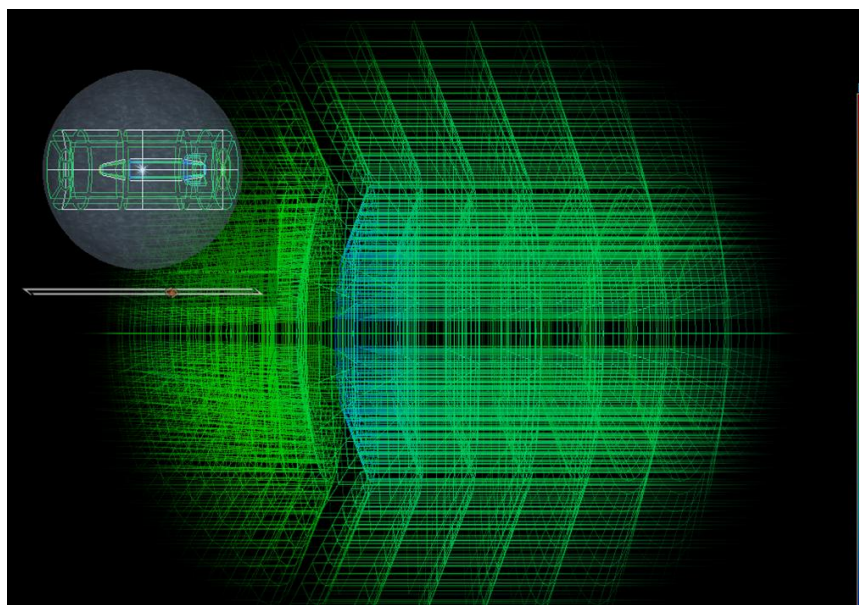


Рис. 8. Визуализация сетки порядка 10^4 примитивов.

Система оперирует данными больших объемов. Без эффективных механизмов фильтрации данных возможно снижение производительности. Также неизбежно падение общей эффективности системы за счет перегруженности видов отображения ненужной информацией. Работа системы фильтрации данных состоит из двух частей: “пространственной” и “качественной”. Пространственная фильтрация заключается в выделении необходимого объема сеточных данных на основе их пространственного расположения. Очевидно, что одновременный вывод на экран всего слишком большого объема данных влечет за собой снижение уровня детализации. Кроме того, из-за большой некомпактности визуализируемых данных, более далекие от наблюдателя участки могут оказаться вне поля зрения, будучи скрытыми близкими участками. Пространственная фильтрация отчасти решает эту проблему, выделяя область интересующей нас информации из общего ее объема. Функции пространственной фильтрации реализованы с помощью метафоры “альфа-сферы”. Суть метафоры состоит в том, что из общего объема данных выделяется сферическая область. При этом визуальная прозрачность объектов, находящихся внутри области, прямо пропорциональна удаленности объектов от центра области. Таким образом, информация, представляющая для нас наибольший интерес, видна лучше всего, а остальная - тем меньше, чем более удалена от центра области рассмотрения. Фильтрация “по значению” необходима для отсека ненужных участков визуализируемых данных на основании значения каких-либо характеристик, заданных на этих участках. “Блочная фильтрация” является наименее универсальным методом фильтрации данных, но, тем не менее, эффективно применялась в нашей системе.

Второй вариант системы предполагал вывод нерегулярных сеток различной конфигурации и размера (от нескольких десятков тысяч до нескольких миллионов вершин).

С помощью данной системы были эффективно визуализированы сеточные данные различных типов, в частности, блочная сетка с шестигранными примитивами и сплошная тетраэдральная.

При визуализации структуры сеток и данных на них возникают задачи как непосредственно отображения сеток и данных в целом, так и задачи построения сечений и изоповерхностей на сетках. Было проведено исследование, которое показало, что возможности воксельной графики позволяют реализовать эти задачи, и что виды отображения на основе ее могут служить мощным дополнением к видам отображений на основе полигональной графики (см. рис. 9 – пример воксельного представления сетки).

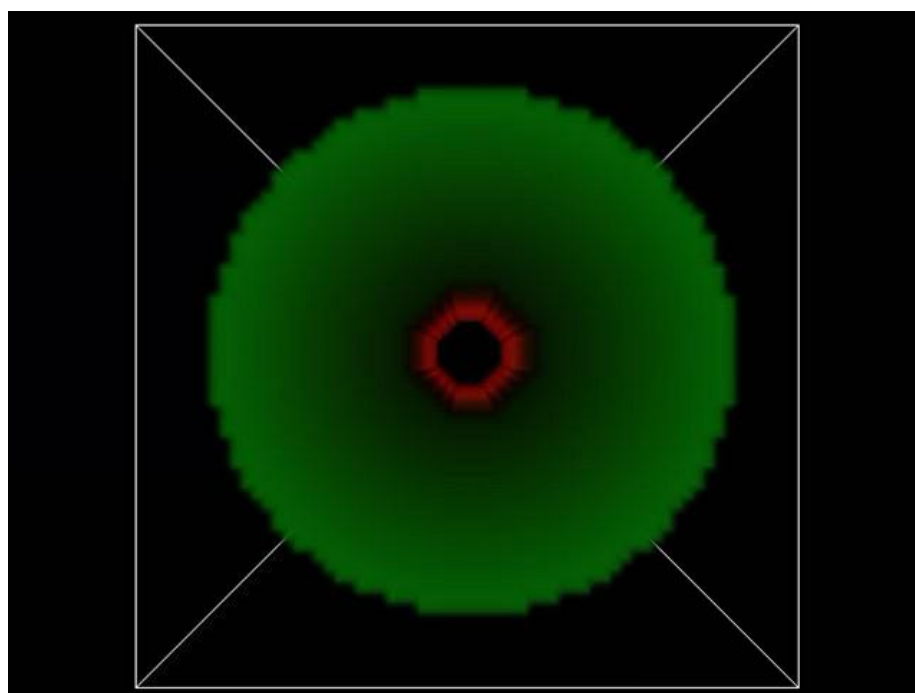


Рис. 9. Пример использования воксельной графики. См. также [ВИДЕО](#).

Возможности воксельной графики позволяют отображать внутреннюю структуру всей сетки и ее блоков (используя метафору прозрачности), а уровень современных видеокарт позволяет добиться интерактивного построения сечений и изоповерхностей как всей сетки, так и отдельных ее участков. При этом существует возможность аппаратной реализации отдельных алгоритмов. Был разработан вид отображения, позволяющий в режиме реального времени просматривать динамику сечений отображаемой сетки – “режим радара”. В нем производится перемещение секущей плоскости от одной стороны воксельного куба к другой, что и позволяет просматривать динамику сечений параллельными плоскостями.

Роль методик генерации изображений при разработке специализированных систем научной визуализации интересно рассмотреть на примерах использования различных методов визуализации объемов.

Еще в конце 90-ых годов была разработана специализированная среда визуализации (по сути, комбинированный набор нескольких графических пакетов), в которой на основе методов volume visualization, решается целый ряд задач, связанных с исследованием одного их типов турбулентности [9].

Одно время считалось, что наиболее перспективно использование этих методов при визуализации биомедицинских объектов. Действительно это применение продолжает усиленно развиваться [10]. Однако методики volume visualization применяется и в других приложениях, например, и для анализа сейсмических данных [11] или для общих задач компьютерного моделирования [12].

Можно также привести пример использования комбинированных методик воксельной и полигональной графики для получения результата в частном случае задачи визуализации.

3.6. Система визуализации структуры области достижимости в задаче оптимального управления

Источником постановки этой задачи, решенной несколько лет назад, послужил поиск областей достижимости в одной из задач оптимального управления. Ряд причин, связанных с реализацией алгоритма, привел к тому, что вычисленные данные составляют порядка десятков миллионов, а то и миллиардов точек, хранимых в bitmap-формате. Пользователю-математику необходимо изучать общий вид и внутреннюю структуру области достижимости для конкретной задачи. Очевидно, что огромный объем полученного расчетного материала потребовал использования средств трехмерной визуализации. Был создан комплекс программ, позволяющий провести фильтрацию данных не искажающую ни внешнюю и ни внутреннюю структуру объекта. Система позволяет отобразить как процесс построения области достижимости, так и сам объект в разных ракурсах.

Программный комплекс состоит из набора утилит для работы с огромным облаком точек и его последующей визуализации. Данные обрабатываются на разработанном специализированном “конвейере визуализации”, состоящем из следующих стадий:

- обработка исходных bitmap-файлов;
- вычисление освещенности;
- конвертирование в воксельный формат и создание структур хранения сцены;
- сглаживание воксельных объектов;
- конвертирование в полигональный формат.

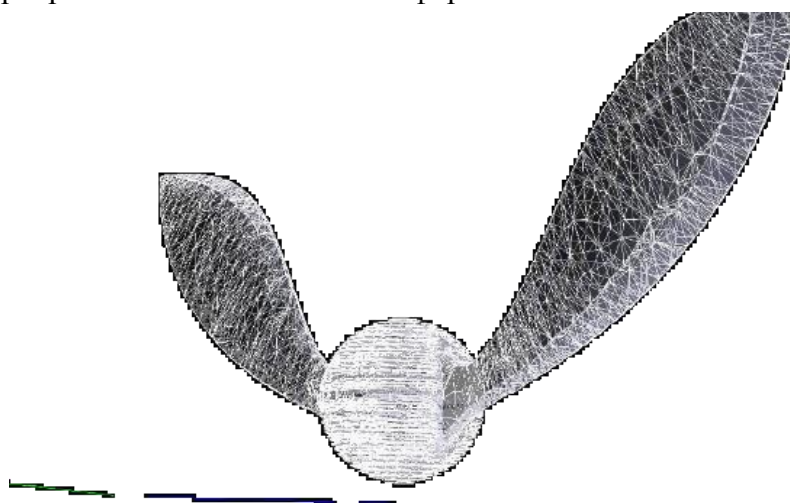


Рис. 10. Восстановление полигональной модели по сеточным данным. См. [ВИДЕО](#).

В примере была рассмотрена так называемая сфера Лоренца, при расчете которой объем исходного файла составляет примерно 6,5 миллионов точек. В итоге получен графический объект, состоящий из около 40 000 полигонов (рис. 10). Кроме того, был получен фильм, отражающий процесс построения сферы, что было весьма полезно разработчикам алгоритма. Полученные от математиков сведения о топологии “сферы”, ее однородности и непрерывности позволили применить конвертирование графических данных из формата в формат.

3.7. Идея создания виртуального испытательного стенда

Обобщение опыта разработки специализированных систем позволило использовать идею виртуального испытательного стенда. [13]. Виртуальный испытательный стенд рассматривается нами в качестве среды проведения вычислительного эксперимента. Вычислительный (компьютерный) эксперимент, известный с 70-ых годов, становится реальным инструментом исследования после появления суперпроизводительных параллельных вычислителей и мощных средств визуализации, включая средства виртуальной реальности. Идея виртуального испытательного стенда связана с проведением современного вычислительного эксперимента и возможностью многократного пропуска программы, моделирующей те или иные процессы с различными параметрами моделирования. Такие модели возникают в целом ряде исследований, связанных в частности с аэродинамикой, физикой взрыва, и в других задачах, где многократный натурный эксперимент либо сложно организовать, либо он просто не возможен. Вычисления в этих случаях, как правило, требуют использования мощных суперкомпьютерных средств. Причем для получения единичного результата требуются часы, и даже сутки счета на самых мощных суперкомпьютерах.

Очевидно, что идея виртуального испытательного стенда связана с использованием сред виртуальной реальности, хотя в принципе наличие таких сред не является обязательным при создании систем поддержки численного эксперимента.

Таким образом, предполагается, что в рамках виртуального испытательного стенда:

- 1) счет проводится на параллельном вычислителе, возможно в удаленном режиме, и требует значительного времени для получения результата
- 2) визуализация и анализ результатов проводится как по ходу вычислений (в режиме on-line визуализации), так и после их окончания возможно с использованием сред виртуальной реальности.

Анализ задач разработки виртуального испытательного стенда

Проблемы создания физических и математических моделей, также как и разработки алгоритмов и численных методов достаточно хорошо проработаны в современной науке. По мнению ряда видных исследователей, новые возможности супервычислений требуют новых математических моделей и новых алгоритмов. Конечно, при разработке первой очереди “испытательного стенда” можно опираться на фундамент уже полученных ранее реализаций, включая вычислительные методы, реализованные в современных САЕ-пакетах. Однако в дальнейшем наработки последних лет по моделям и численным методам “пойдут в ход” и будут использоваться в полном объеме.

В связи с тем, что мы собираемся имитировать ход каких-либо экспериментов, необходимо четко представлять сценарии реальных исследований. Очевидно, в состав

“бригады” разработчиков должны быть включены специалисты, способные консультировать в соответствующей области исследований.

Исследования и разработки в области параллельного программирования развиваются весьма активно. В тоже время проблемы тестирования и отладки параллельных программ остаются достаточно острыми. Для реальной работы стенда, уверенности в результате численных экспериментов необходимо обеспечить реальную надежность программ.

В плане организации вычислений возникает свой набор задач, подобных обеспечению особого (“диалогового”) режима счета на параллельных суперкомпьютерах, “on-line” управлению ходом вычислений или передачи данных, необходимых для оперативной визуализации.

Ясно, что это лишь приблизительная оценка возникающих проблем. Причем, значительная (и, быть может, основная) часть этих проблем нам пока даже неизвестна.

3.8. Проект среды разработки специализированных систем визуализации

Интересно, что в одних случаях разработанные нами специализированные системы легко адаптировались для представления объектов в иных математических задачах. В других, элементарную адаптацию провести не удалось, хотя опыт работы позволил сравнительно быстро завершить разработку новой специализированной системы визуализации. Причина этого заключается в специфике самих математических объектов, а также в отличающихся от раза к разу задачах пользователей по обеспечению анализа и интерпретации изображений. Иногда по имеющемуся образцу удаётся сравнительно быстро создать новую специализированную систему, успешно выполняющую свои функции. Но в любом случае требуется большая работа специалистов по реализации рендеринга, интерфейсов и других составляющих графической системы. В результате многолетней работы и создания целого ряда специализированных систем визуализации предлагается проект разработки инструмента, который мог бы значительно упростить решение возникающих технических проблем. Данная среда должна, в частности, содержать в себе следующие компоненты:

- Расширяемое ядро по отображению различных визуальных сущностей – от примитивов (оси, отрезки, поверхности) до высокоуровневых объектов (сетки, графы, массивы).
- Расширяемый набор средств обработки данных, например построение изоповерхностей, фильтрация, выявление особенностей.
- Скриптовый язык управления средой, в рамках которого можно реализовать загрузку данных, обработку, создание и управление визуальными объектами, взаимодействие с пользователем и так далее.
- Возможность настройки пользовательского интерфейса, включая вывод элементов управления параметрами отображения и других механизмов взаимодействия со скриптами.

Создание подобной среды позволит существенно сократить ресурсы, требуемые на разработку специализированных систем визуализации. Действительно, разработчик сможет реализовать загрузку произвольных данных на встроенном языке, обозначить визуальные объекты на этом же языке, при необходимости расширить возможности системы за счет добавления новых визуальных сущностей (описанных опять же на языке среды), настроить интерфейс пользователя и так далее.

Таким образом, разработка системы визуализации из сложного проекта, включающего в себя множество рутинных технических задач, превращается в процесс настройки, адаптации и расширения имеющегося «конструктора систем визуализации» в едином стандарте.

4. Интерактивная визуализация параллельных и распределенных вычислений

В представленном обзоре (3) мы обращали внимание прежде всего на “человеческий” фактор в разработке. Однако реализация целого ряда проектов и изучение возможности создания виртуального испытательного стенда поставила задачи, связанные с разделом, касающимся инженерии программного обеспечения. В частности было необходимо решить проблемы интерактивной визуализации параллельных и распределенных вычислений.

4.1. Онлайн-визуализация

Онлайн-визуализация – методика визуализации, обеспечивающая наблюдение и контроль за выполняющейся вычислительной программой. Наблюдение подразумевает визуальное представление текущего состояния модельных сущностей и программы с тем, чтобы оценить прогресс и качество счёта. Контроль подразумевает возможность управления вычислительной программой – изменение её переменных, влияние на распределённые данные, а также подачу особых команд (например, приостановку, создание контрольных точек, перезапуск с определённого момента и так далее).

Применение онлайн-визуализации обеспечивает следующие возможности при проведении вычислительных экспериментов:

- **Визуальная оценка прогресса счёта и его текущего состояния.** Позволяет оценить, насколько продвинулось вычисление, принять решение о целесообразности его продолжения, провести визуальный контроль над правильностью работы алгоритмов и состоянием компонент вычислительной системы.
- **Корректировка параметров счёта во время его исполнения.** Позволяет внести коррективы в начальные данные параметры вычислительных алгоритмов с непосредственным продолжением вычислений, без остановки или перезапуска.
- **Быстрый перезапуск вычислений с новыми начальными данными.** Такой сценарий работы обеспечивает проведение серий краткосрочных экспериментов без ожидания очереди вычислителя, а также одновременный запуск различных экспериментов с различными начальными данными и контроль над ними в едином визуальном пространстве.
- **Проведение интерактивных вычислительных экспериментов** в режиме «Что если?», эффективность которого многократно возрастает, когда ответ на вопрос можно получить немедленно. Дальнейшее развитие этого направления приводит к появлению больших, комплексных интерактивных вычислительных систем, позволяющих проводить масштабные вычислительные эксперименты в режиме «онлайн»,

- **Визуальная поддержка при отладке параллельных программ.** Как правило, применяется для отображения состояния объектов прикладной области, программных переменных, и характеристик вычислительных компонент (загрузка мощностей, текущие стадии вычислений и так далее).
- **Презентация результатов и программ** людям, не разбирающимся в деталях данного научного исследования. Онлайн-визуализация способна оказать поддержку в построении взаимопонимания между участниками с различной подготовкой, играя роль моста между спецификой и внешним миром.
- **Автоматизированное управление** – контроль счётной программой из другой программы, например для автоматизации поиска оптимальных решений, проведения серий экспериментов и так далее. Элементы автоматизированного управления могут применяться и при проведении интерактивных вычислительных экспериментов, выполняя часть работы по контролю над вычислениями.

Типовой сценарий проведения вычислительного эксперимента с применением онлайн-визуализации выглядит следующим образом.

Пользователь подготавливает начальные данные и запускает вычислительную программу. После запуска программа становится доступной для взаимодействия, при этом собственно вычислительный процесс может как работать, так и находиться в режиме ожидания пользовательских команд. Пользователь открывает систему визуализации, в которой отображается текущее состояние программы. При этом визуальное представление может обновляться автоматически по ходу вычислений. Кроме средств отображения и анализа, в системе визуализации могут присутствовать средства обратной связи – для изменения данных в памяти программы, подачи различных команд и так далее. В этом случае команды и изменения поступают в вычислительную программу, и она реагирует тем или иным образом. Эта реакция, в свою очередь, отображается в системе визуализации. По завершении вычислений, в зависимости от возможностей системы онлайн-визуализации, изучение результатов её работы может быть продолжено, уже без непосредственного участия вычислительной программы.

Основная проблема применения онлайн-визуализации заключается в слабой поддержке этого направления со стороны стандартных вычислительных средств. Основные применяемые в вычислительной практике инструменты – пакетные планировщики и средства программирования класса MPI/OpenMP не предоставляют возможностей по интерактивному извлечению данных из программ во время исполнения «по запросу», равно как и средств управления.

Для обеспечения таких возможностей приходится создавать особые программные комплексы – средства обеспечения и поддержки онлайн-визуализации. Такие средства разрабатываются, например: CUMULVS, CSE, SkyRun, Reality Grid, CoCoS, VISIT, CoVise, CS_Lite, Magellan, SAGA, VolVis. Однако анализ этих и других проектов показывает, что данные средства обладают рядом общих недостатков, среди которых основные – сложность в настройке и эксплуатации. Многие системы требуют наличия каких-либо специфичных системных компонент (например, CUMULVS подразумевает наличие на кластере системы параллельного исполнения PVM). Интерфейс прикладного программиста во многих системах оставляет желать лучшего, например для того, чтобы предоставить доступ к чтению одной(!) переменной параллельной программы необходимо написать до 20(!) строк кода. Более того, многие системы предлагают и полностью переписать счетные программы с применением новых программных архитектур. Дополнительно, многие системы накладывают ограничения на передаваемые типы и структуры данных (например, только массивы).

Наличие этих и других проблем в существующих средствах онлайн-визуализации приводят нас к решению по созданию собственной системы онлайн-визуализации. Такая система должна быть максимально простой в установке, легкой в использовании и при этом позволять решать широкий круг задач онлайн-визуализации. Очевидно, что хоть эти требования и можно назвать «общепринятыми» для любого программного обеспечения, однако факты показывают, что разработка в данном случае являлась нерешенной задачей.

Такая система создана. Система состоит из трёх компонент: (1) вычислительных процессов (параллельной программы), (2) сервиса-посредника и (3) программ визуализации (см. рис. 11). Вычислительные процессы – это процессы счётной программы с внедрёнными в неё специальными вызовами функций, которые соединяют её с системой визуализации. Посредник – это сердце системы – принимает команды от программ визуализации и направляет их к вычислительным узлам. Программы визуализации – это набор программ, реализующих графический интерфейс пользователя, которые отображают текущее состояние параллельной программы и позволяют управлять ею. Для каждой параллельной программы подразумевается наличие своей собственной программы визуализации (при этом возможны и унифицированные решения, одно из них представлено далее в настоящей главе).



Рис 11. Структура системы онлайн-визуализации.

Система основана на принципе *запрос-ответ*. Счётная программа после запуска работает в штатном режиме. Исследователь, производя те или иные действия, воздействует на программу визуализации, которая в свою очередь обращается к посреднику с командой или запросом на данные. Посредник направляет этот запрос к вычислительным узлам. Ответ возвращается посредством сервиса-посредника программе визуализации, которая отображает результат выполнения запроса.

Все запросы основаны на именах. Каждая параллельная программа при запуске регистрируется у посредника со своим уникальным именем. А точнее, каждый её процесс указывает это имя и свой порядковый номер. Затем каждый вычислительный процесс регистрирует у посредника свои ресурсы – данные или коммуникационные примитивы. Этот этап называется «публикацией данных».

Запросы на данные подразумевают операции чтения и записи. При этом поддерживаются запросы и на часть данных – например, можно прочитать данные массива только с *i*-го по *j*-й элемент.

По умолчанию все операции чтения и записи производятся асинхронно. Это означает, что после публикации данных вычислительный процесс продолжает своё

выполнение, производя расчёты. Когда же эти данные становятся необходимы системе визуализации, то они будут прочитаны из памяти вычислительного процесса асинхронно (из параллельного потока), вне зависимости от хода основного счёта. Этого, как пока показывает практика, вполне достаточно для большинства случаев. Конечно, в более тонких ситуациях потребуется использование примитивов синхронизации, которые также представлены в системе.

Запросы на данные могут содержать обращения к распределённым данным. Так, например, несколько вычислительных узлов могут опубликовать какой-либо массив с одинаковым именем, но с указанием различных участков этого массива. Это распределение становится известным сервису-посреднику, и запрос на «обобщённые» данные распределяется им между узлами-участниками. Затем посредник посылает программе визуализации уже объединённые результаты.

Кроме запросов на данные, программы визуализации могут создавать запросы на управление. Например, это может быть запрос на состояние счётного процесса (выполняется или ждет, какова стадия выполнения, список ресурсов и т.д.) или запрос на управление объектами синхронизации.

В рамках системы существует два программных интерфейса. Один предоставляется вычислительным узлам, а другой – программам визуализации. Интерфейс для параллельных программ – это набор высокоуровневых вызовов, реализованных в специальной библиотеке. Для программ визуализации предоставляется интерфейс на основе сетевого протокола SOAP. Сама же система внутри полностью ориентирована на применение этого протокола. При этом программный интерфейс для параллельного программиста, подчеркнём, это не SOAP, а функции Си или Фортран.

Таким образом, для того чтобы сделать параллельную программу управляемой и поддерживающей онлайн-визуализацию, программисту необходимо внедрить ряд вызовов функций и скомпоновать программу со специальной библиотекой.

Созданная система онлайн-визуализации: обладает следующими характеристиками:

- Система использует подход интеграции с параллельными программами. Отсутствует необходимость переписывать счетные коды; эти коды надо дописать – оснастить вызовами функций, которые передадут системе информацию о компоновке необходимых данных в памяти программы.
- Система использует передачу данных по запросу. Запросы инициируются программой визуализации (как правило, работающей на компьютере пользователя). Обработка запросов в счетной программе происходит асинхронно, параллельно основным счетным потокам. Дополнительно представлены примитивы синхронизации – точки сбора и барьеры.
- Разработка является **универсальной системой онлайн-визуализации**, способной работать с широким кругом прикладных задач.

4.2. Пример: визуальное сопровождение решения обратных прикладных задач методами регуляризации и итерационной аппроксимации

Работам по методам приближённого решения некорректно поставленных задач и их применению к решению обратных прикладных задач в математическом сообществе уделяется пристальное внимание. Эти работы имеют большое значение для автоматизации обработки наблюдений, решения проблем управления, в задачах математической экономики, геофизической разведки и т.д..

В работе [14] предложена программная система решения обратных задач методами регуляризации и итерационной аппроксимации. Вычислительная программа, написанная на языке Фортран, представляет собой однопоточный процесс. Работа с программой заключается в подборе исходных параметров, проведении расчётов и последующем анализе результатов. Анализ необходим как для изучения степени пригодности исходных параметров, так и, прежде всего, для отладки программы.

Авторами программы было поставлено три задачи, которые оказалось возможным решить с помощью онлайн-визуализации.

1. Процесс вычисления занимает от нескольких минут до нескольких часов. Это время исследователь проводит в ожидании возможности увидеть и интерпретировать результаты вычислений. Специфика программы такова, что выводы о целесообразности продолжения счета с введенными параметрами возможно делать буквально через несколько минут после начала вычислений. Поэтому для повышения эффективности проведения вычислительных экспериментов необходимо создать средство для оперативного визуального анализа хода вычислений.
2. Благодаря наличию параллельной инфраструктуры можно осуществлять несколько независимых запусков, таким образом одновременно проводя сразу несколько вычислительных экспериментов. Необходимо создать средства по упорядочиванию и отслеживанию всех параллельно исполняющихся экспериментов.
3. Работа с вычислительной установкой требует специфических навыков и повышенного внимания к деталям системного окружения. Даже постановка задачи в очередь вычислителя – это ручной труд, требующий определённых временных затрат и внимания. Таким образом, исследователь вынужден большую часть своего внимания уделять непрофильным вопросам, не связанным с математической спецификой работы. Задача заключается в создании средств, по возможности минимизирующих и автоматизирующих взаимодействие исследователя с системными инструментами при проведении вычислительных экспериментов.

Для решения обозначенных задач был разработан программный комплекс с применением системы онлайн-визуализации. На вычислительном кластере запускается сразу несколько независимых вычислительных программ. Каждая программа считывает свои исходные параметры из файла (сообразно своему порядковому номеру). Также программа публикует свои параметры и массивы, содержащие текущую итерацию построенных графиков функций.

Созданная программа визуализации отображает состояние всех исполняемых вычислительных программ. Состояние каждой программы характеризуется графическим представлением текущих рассчитанных данных. Исследователь визуально наблюдает на экране монитора за всеми одновременно исполняющимися вычислительными программами, анализируя представленную графическую информацию. Также исследователь имеет возможность изменять параметры вычислений и подавать команду на перезапуск вычислений в любой программе.

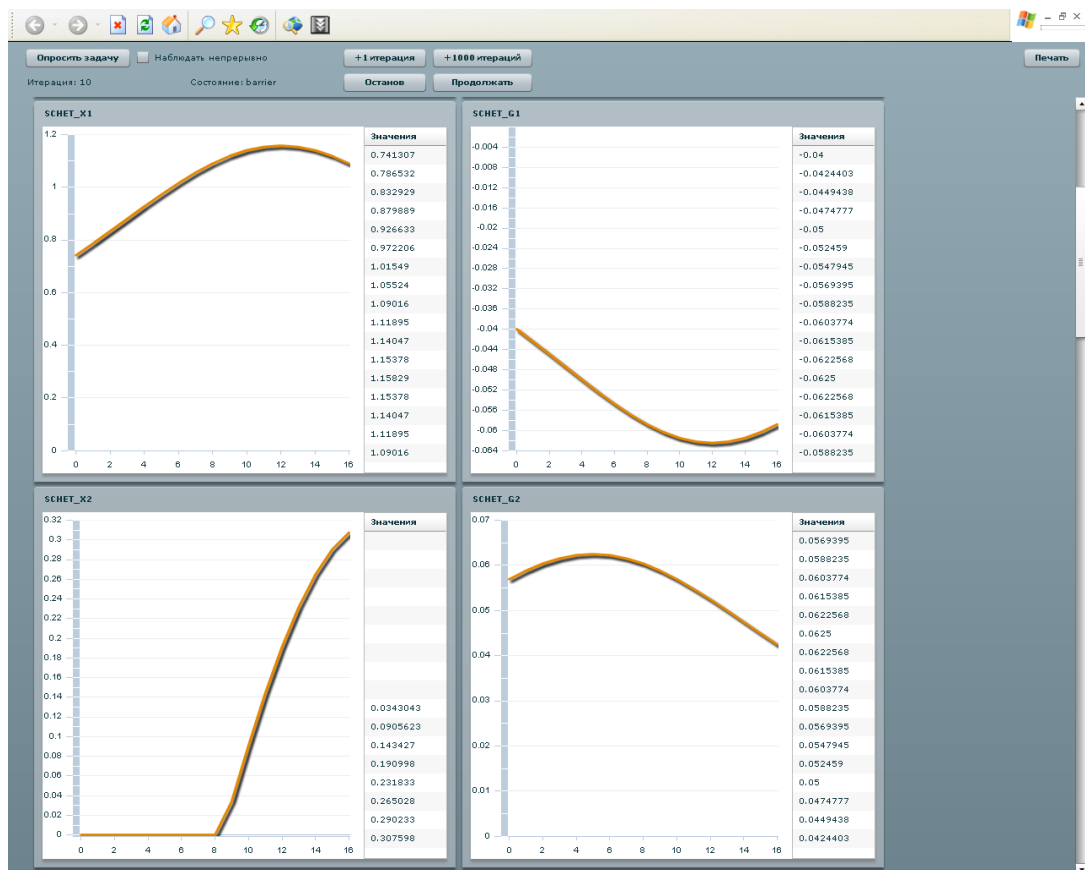


Рис. 12. Пример внешнего вида интерфейса пользователя онлайн-визуализации.

На рис. 12 представлено состояние одной вычислительной программы. Показан рассчитанный график функции, а также график вспомогательной функции, участвующей в расчетах. Кроме того, отображается состояние первых десяти итераций счета, как наиболее существенных с точки зрения анализа работы алгоритма.

Совместное представление состояний нескольких программ строится схожим образом. На экране располагается по представлению с одним графиком на каждую программу. Пользователь может выбрать, какой именно график он хотел бы видеть в данной программе (основной, служебный, или один из первых десяти итераций).

Отдельный интерес представляют технологии, примененные при разработке визуальных компонент комплекса. Программа визуализации реализована на платформе Adobe Flex и работает в веб-браузере. Реализация взаимодействия с сервисом-посредником по протоколу SOAP, как и ожидалось при проектировании системы, оказалась в данном случае довольно простой задачей. Дополнительным преимуществом использования веб-технологий оказалась возможность вести наблюдение и управление вычислительным экспериментом сразу несколькими исследователями одновременно с разных компьютеров. Вместе с тем, авторы не рекомендовали бы использование Adobe Flex в других проектах по онлайн-визуализации в связи с тем, что Adobe Flex является весьма запутанной технологией, в которой крайне сложно проводить качественную компоновку программных кодов.

Созданный комплекс показал свою работоспособность и способность решить обозначенные выше задачи. Он находится в стадии эксплуатации с 2007 года. Комплекс продолжает использоваться при проведении вычислительных экспериментов и в настоящее время.

4.3. Удалённая визуализация

Удалённая визуализация – методика визуализации, при которой система визуализации создаёт графическое изображение на отдельном компьютере и передает его по сети на компьютер пользователя. Пользователь может взаимодействовать с данным изображением. Воздействия пользователя передаются по сети обратно, в результате чего изображение перерисовывается и повторно отправляется на компьютер пользователя.

Необходимость применения удалённой визуализации в научных и инженерных вычислениях обуславливается следующим фактором.

Как правило, процесс вычислений происходит на специальной вычислительной системе. Результаты вычислений сохраняются в этой же системе. Чтобы проанализировать результаты, необходимо провести процесс их визуализации. Визуальное отображение должно проводиться на устройстве, непосредственно доступном исследователю – станции визуализации (персональном компьютере, системе виртуальной реальности и т.д.). Чтобы начать процесс визуализации, программа визуализации должна считать результаты вычислений. Проблема заключается в том, что процесс передачи данных от вычислительной системы к станции визуализации может вызвать значительную задержку по времени.

Например, если расчёты проведены в вычислительном центре в одном городе, а исследователь находится в другом и обладает соединением с относительно невысокой пропускной способностью, то задержки, вызванные передачей результатов вычислений, могут оказаться сопоставимы с длительностью самих вычислений. Кроме того, такая передача может оказаться просто невозможной, если объём данных настолько велик, что не может быть размещён в локальной дисковой памяти станции визуализации.

Именно для таких случаев применяется методика удалённой визуализации, благодаря которой необходимость передачи результатов вычислений от вычислительной системы к станции визуализации отпадает.

На настоящий момент сообществом разработан целый ряд систем поддержки удаленной визуализации, например VirtualGL, X Window System, IBM Deep Computing Visualization, SGI VisServer, Microsoft Remote Desktop Services (MS RDS), RemoteApp / RemoteFX, семейство VNC, Citrix HDX 3D и другие. Однако анализ существующих систем показывает, что они не могут быть применены в следующей постановке:

– Необходимо реализовать удалённую визуализацию результатов расчётов, которая могла бы использоваться в рамках семейств как Windows, так и Linux, не накладывая ограничений на форматы данных и с использованием открытых технологий. Дополнительно требуется учесть следующие условия:

1. Максимально ограниченный доступ к внутренней инфраструктуре систем, на которых выполняются программы рендеринга.
2. Отсутствие привязки к конкретным оконным системам. Крайне желательно, чтобы система могла поддерживать и графические программы, генерирующие изображения в пакетном (не оконном) режиме;
3. Максимально широкий спектр поддерживаемых клиентских устройств (станций визуализации), на которых смогут работать пользователи.
4. Возможность мягкой интеграции в современные среды управления вычислительными заданиями на базе веб-интерфейсов.

Данная постановка является естественным комплексом требований при создании инфраструктуры удалённой визуализации для вычислительных центров.

Рассмотрим вопрос создания системы, удовлетворяющей данным требованиям. Для этого задействуем имеющийся задел – описанную выше систему онлайн-визуализации.

Система онлайн-визуализации содержит функции по чтению и записи переменных в удалённых программах (вычислительных программах в терминах онлайн-визуализации). Таким образом, если программу рендеринга встроить в эту систему, то вопрос передачи а) глобальных параметров (имени пользователя, имен файлов данных, и других) и б) параметров отображения (углов поворота камеры, фактора приближения, и других) будет решен. Кроме того, параметры можно передавать и на этапе запуска программ рендеринга. Также система поддерживает возможность вызова особых функций в теле удалённой программы. Это позволит вызывать в программе функцию рендеринга, на вход которой можно подавать требуемое разрешение, а на выходе получать результат рендеринга согласно заданному разрешению и параметрам отображения.

Таким образом, решается вопрос взаимодействия системы с программами рендеринга. Следует подчеркнуть, что существующая архитектура системы позволяет сервису-посреднику взаимодействовать сразу с несколькими клиентскими интерфейсами и программами рендеринга одновременно. Проблема выбора технологий создания клиентских приложений и интерфейсов пользователя может быть решена за счет использования протокола SOAP, который в свою очередь базируется на HTTP. Таким образом, в систему может быть встроен веб-сервер, реализующий хостинг веб-интерфейсов и взаимодействие с ними.

Система удалённой визуализации, удовлетворяющая поставленным выше требованиям, состоит из трёх уровней: (1) **модулей визуализации**, (2) **сервиса-посредника** и (3) **интерфейсов пользователя** (рис. 13). Каждый модуль визуализации (программа рендеринга) – это один или несколько процессов, ответственных за загрузку и графическое представление экспериментальных данных определённого типа. Сервис-посредник это серверный процесс, он принимает команды от интерфейса пользователя и направляет их к программам рендеринга. Интерфейсы пользователя – в настоящее время построенные на веб-технологиях – отображают построенные графические представления данных и взаимодействуют с пользователем.



Рис 13. Структура системы удалённой визуализации.

В рамках разрабатываемой системы модуль визуализации не привязан к конечному типу оборудования – это может быть отдельное приложение на сервере визуализации или параллельное приложение, работающее на вычислителе. Более того, функции модуля визуализации может выполнять часть узлов вычислительной программы. Таким образом, система позволяет варьировать архитектуру для модуля визуализации и может применяться как для офлайн- так и для онлайн-визуализации.

4.4. Пример: удалённая визуализация результатов расчетов в пакете Deform

Система поддержки удалённой визуализации, описанная выше, обеспечила возможность создания следующего приложения удалённой визуализации результатов инженерных расчётов.

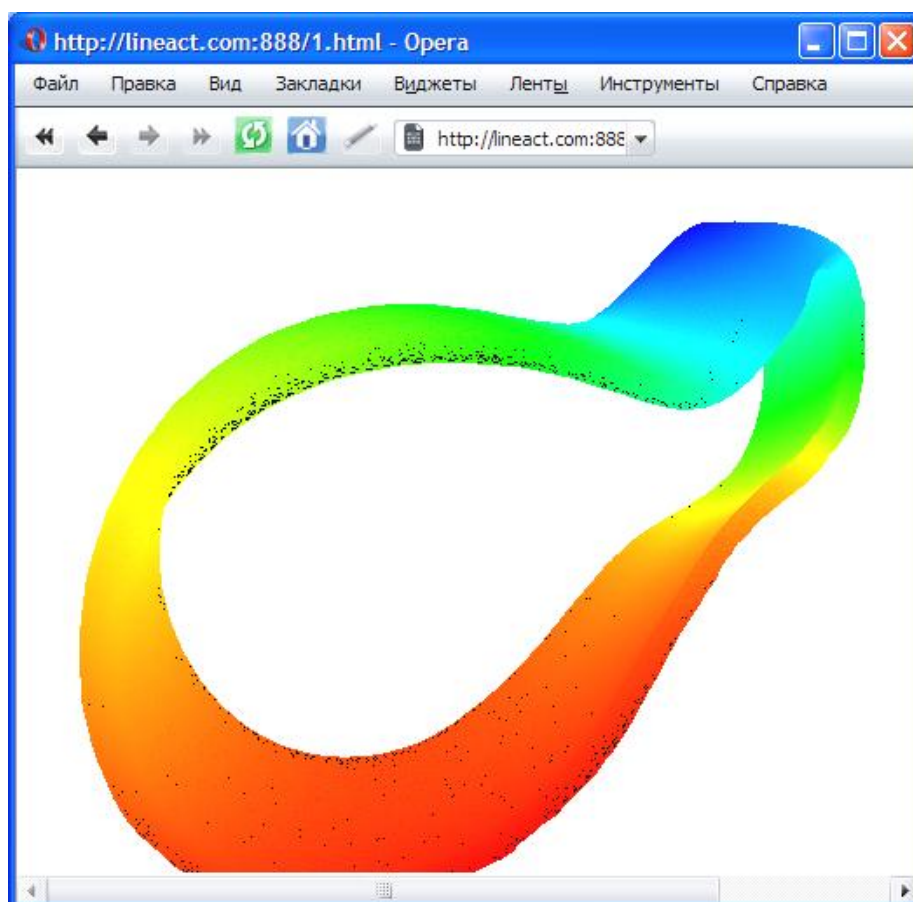


Рис. 14. Пример использования системы удалённой визуализации.

На рис. 14 показан один из шагов расчёта процесса определённого воздействия на сечение трубы нефтепровода [15], проведённого в инженерном пакете DEFORM. Пользователь может нажать кнопку мыши на изображении и перемещать её, при этом изображение будет немедленно обновляться, показывая вращение объекта. Кроме того, пользователь может управлять и процессом масштабирования трехмерной сцены.

Рендеринг осуществляется на удалённом сервере визуализации аппаратно специальной программой, реализованной на языке C++ с применением технологии DirectX. Программа считывает файл результатов вычислений в формате key, строит трехмерную сцену, соединяется с сервисом-посредником и ожидает управляющих команд и запросов на рендеринг.

Для реализации пользовательского интерфейса применена технология (d) HTML, обеспечивающая работу на максимально широком спектре конечных устройств. Интерфейс реализован полностью согласно описанию, приведённому в разделе 3.6, без применения каких-либо дополнительных компонент, требующих отдельной установки.

Кроме того, применение веб-интерфейса позволило вывести приложение на такой уровень, при котором оно может быть прозрачным для пользователя образом интегрировано в систему управления вычислительными заданиями.

В локальной сети 10 Мбит удалось достичь скорости работы 15 кадров в секунду. В эксперименте с визуализацией через Интернет по каналу Челябинск (станция визуализации) – Москва (промежуточный узел) - Екатеринбург (программа рендеринга), система показала скорость работы около 3 кадров в секунду. В эксперименте на дистанции Н. Новгород (станция визуализации) – Екатеринбург (программа рендеринга) система показала аналогичные результаты. Необходимо отметить, что в работе могут быть применены особые технологии сжатия видеопотоков (например, [16]), что может дополнительно улучшить приведенные скоростные показатели в 1.5-2 раза.

5. Заключение

В заключении хотелось бы обратить внимание на вопросы теории компьютерной визуализации.

В небольшой работе [17] ставится вопрос о целях разработки теории информационной визуализации (одного из разделов общей области компьютерной визуализации). Авторы отмечают, что часто высказывается мнение о ненужности теории – ведь практика идет сама по себе уже в течение нескольких десятков лет. Одним из обоснований необходимости поставить в основание практики разработки систем ту или иную теорию визуализации рассматривается необходимость оценки систем и обучение новых разработчиков. Авторы предлагают в качестве основания визуализации использовать и теорию восприятия визуальных образов, и когнитивные науки вообще.

Процесс визуализации можно рассматривать как построение визуального (геометрического) образа на основании абстрактных представлений об объекте. Эти абстрактные представления есть модель (исследуемого объекта, явления, или процесса), как-то связываемая с имеющимся у пользователя когнитивными структурами, описывающими данную сущность. Визуальные образы, представляющие моделируемую сущность, служат для того, чтобы создать или восстановить по нему когнитивные структуры. Цель исследований в области визуализации состоит в создании таких методик и принципов, которые обеспечат восстановление по визуальным образам адекватных когнитивных структур. **Порождение когнитивных структур по визуальным образам и есть процесс интерпретации. Этот процесс является обратным или, точнее, двойственным визуализации.** Аналогично принципам визуализации, существуют принципы, на которых базируется интерпретация.

Изучение процесса интерпретации приводит к необходимости использовать понятийный аппарат семиотики, а также теорию **метафоры визуализации, понимаемой как основной идеи сближения понятий прикладной области с той или иной образностью.** Именно метафоры визуализации лежат в основе видов отображения, проектирование которых, в свою очередь, составляет базу проектирования когнитивной составляющей конкретной специализированной системы визуализации.

В нашем обзоре делается попытка показать, как осуществляется конструирование видов отображения на основе тех или иных метафор.

Литература

1. Visualization in Scientific Computing, Special Issue, ACM SIGGRAPH Computer Graphics, V. 21, N 6, November 1987.
2. Van Wijk J.J. Image Based Flow Visualization // Proceedings of ACM SIGGRAPH 2002. ACM Transactions on Graphics vol 21, no 3, p. 745-754.14.
3. Potiy O.A. Anikanov A.A. GPU-Based Texture Flow Visualization // THE 14th International Conference on Computer Graphics and Vision GraphiCon'2004, September 6-10, 2004 Moscow, Russia. Conference Proceedings. Pp. 155-158
4. Самарский А.А. Математическое моделирование и вычислительный эксперимент // Вестник АН СССР 1979, N 5. Стр. 38-49.
5. Reed D., Scullin W., Tavera L., Shields K., Elford Ch. Virtual Reality and Parallel Systems Performance Analysis // IEEE Computer, V.28, N 11, (November 1995), pp. 57-67.
6. Красовский Н.Н., Субботин А.И. Позиционные дифференциальные игры. М.: Наука, 1974.
7. Patsko, V.S., and Turova, V.L., Homicidal chauffeur game: Computation of Level Sets of the Value Function / Annals of the International Society of Dynamic Games, vol. 6: Advances in Dynamic Games and Applications, Altman, E., and Pourtallier, O., Eds., Birkhauser, Boston, MA, 2001, pp.295-322.
8. Иванов А.Г., Краснов В.П., Кумков С.И. Расчет констант скоростей элементарных реакций процесса динамического кинетического расщепления // Алгоритмы и средства параллельных вычислений, вып. 4, Екатеринбург, ИММ УрО РАН, 2000. Стр. 99-110.
9. Clyne J., Scheitlin T., Weiss J. B. Volume Visualizing High-Resolution Turbulence Computations // Theor. Computational Fluid Dyn., 11, 1998, pp. 195—211.
10. Haidacher M., Patel D., Bruckner S., Kanitsar A., Groller M.E. Volume Visualization based on Statistical Transfer-Function Spaces // Proceedings of the IEEE Pacific Visualization 2010, pp. 17-24. March 2010.
11. Patel D., Bruckner S., Viola I., Groller M.E. Seismic Volume Visualization for Horizon Extraction // Proceedings of IEEE Pacific Visualization 2010, pages 73-80. March 2010.
12. Толоч А.В., Полякова Е.Н. Рекурсия параллельных вычислений в системе аналитического проектирования “РАНОК” // Труды Второй международной конференции «Трехмерная визуализация научной, технической и социальной реальности. Технологии высокополигонального моделирования» 24-26 ноября 2010. Ижевск. Том 1. Пленарные доклады. Секции 1, 2.. УдГУ, Ижевск, 2010. Стр. 18-22.
13. Bryson S., Levit C. The Virtual Wind Tunnel // IEEE Computer Graphics and Applications, Volume 12 , Issue 4 (July 1992), P. 25 34
14. Васин В.В., Сержникова Т.И., Шарф С.В. Васёв П.А. О распараллеливании и визуализации при решении некорректных задач методами регуляризации и итерационной аппроксимации на вычислительном комплексе МВС-1000 // Параллельные вычислительные технологии / Труды научной конференции. Челябинск. Издательство ЮУрГУ, 2007. Том 2. Стр. 227-233.
15. Дорохов В.А., Маковецкий А.Н., Соколинский Л.Б. Разработка проблемно-ориентированной GRID-оболочки для решения задачи оваллизации труб при закалке // Параллельные вычислительные технологии: тр. междунар. науч. конф. (28 янв. - 1 февр. 2008 г., г. Санкт-Петербург), 2008. - С. 520.

16. Lietsch, S., and Marquardt, O., A CUDA-Supported Approach to Remote Rendering // Proceedings of the International Symposium on Visual Computing, Lake Tahoe, Nevada, United States, 2007, Springer, ISBN 978-3-540-76857-9, pp. 724-733.
17. Liu Zhicheng, Stasko John T. Theories in Information Visualization: What, Why and How // Workshop on The Role of Theory in Information Visualization, InfoVis '10, Salt Lake City, UT, October 2010.

Благодарности

Работа выполнена в рамках программы фундаментальных исследований Президиума РАН: Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация. Направление 3: Вычислительные технологии и системы.

Авторы выражают благодарность коллегам, математикам А.Г. Иванову, А.В. Киму, С.С. Кумкову, В.С. Пацко, В.Л. Розенбергу, Т.И. Серёжниковой, Л.Б. Соколинскому, В.Л. Туровой, В.Н. Ушакову, И.А. Цепелеву, М.В. Якобовскому.

В тексте описаны системы, в проектировании и реализации которых активно участвовали Д.Ю. Горбашевский, А.И. Зенков, Д.Р. Исмагилов, Т.Р. Исмагилов, А.В. Мошков, Д.С. Перевалов, А.Н. Петров, О.А. Пыхтеев, В.О. Решетняк, Т.А. Флягина, Р.М. Чернин, А.Н. Шинкевич, Д.А. Юртаев.