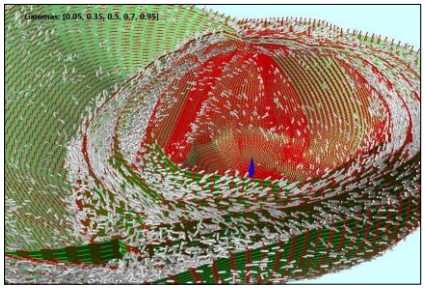
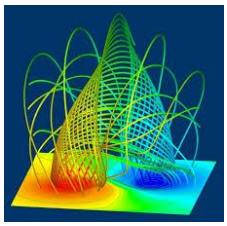
ДЕКЛАРАТИВНО-ИМПЕРАТИВНЫЙ МЕТОД КОНСТРУИРОВАНИЯ СЦЕН НАУЧНОЙ ВИЗУАЛИЗАЦИИ

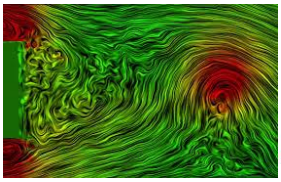


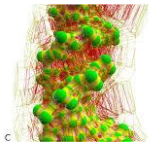
Сектор визуализации
Институт Математики и Механики УрО РАН
г. Екатеринбург

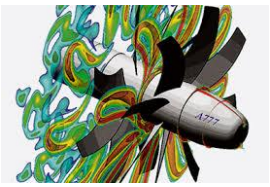
Научная визуализация

- Визуализация результатов научных и инженерных вычислений.





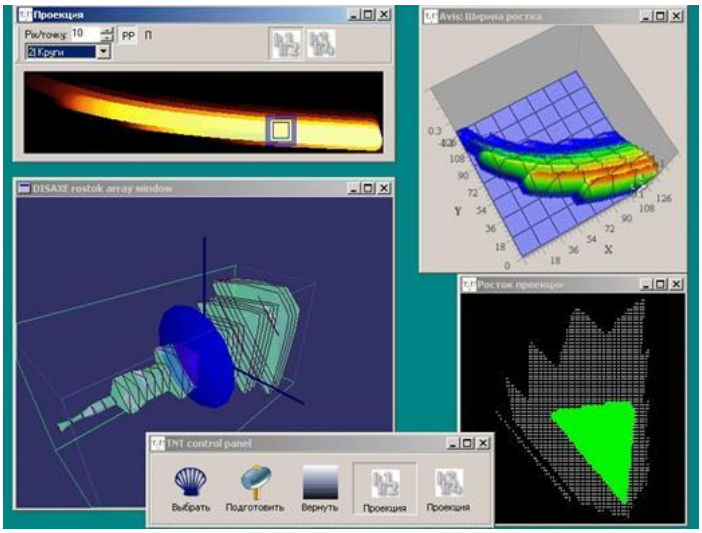




Специализированные системы научной визуализации

- Создаются, когда готовых средств визуализации не хватает. Ориентированы на конкретную задачу и пользователя.

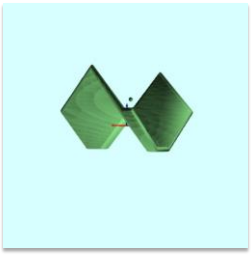
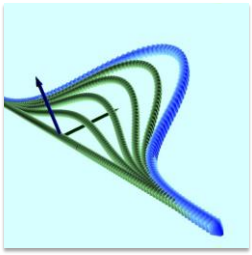
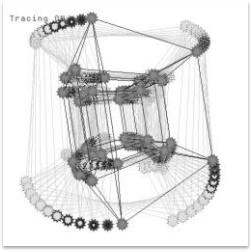
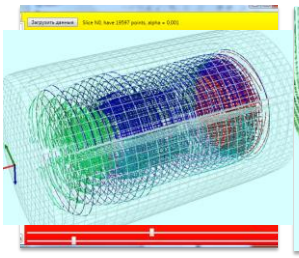
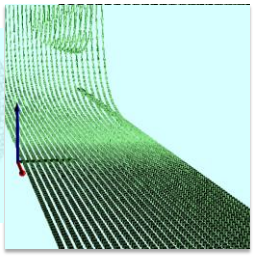
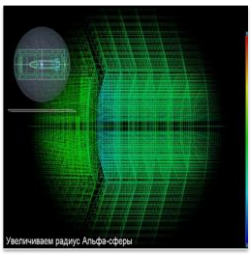
Пример специализированной системы



Свойства специализированных систем

- Преимущества: эффективность (учет всех нюансов задачи).
- **Недостаток: ресурсоёмкое, трудоёмкое создание (месяцы, годы..).**

Примеры

- ~~Подбор окружения исполнения (язык, платформа, оконная система) (MFC, .NET Forms, WPF, VCL, Qt)~~
- ~~Выбор графической библиотеки (OpenGL, DirectX, WPF, etc) или среды рендеринга (VTK, OGRE, Open Inventor etc),~~
- Программирование загрузки данных и их преобразование в визуальные сущности
- ~~Реализация алгоритмов рендеринга~~
- Взаимодействие с визуальными сущностями,
- ~~Реализация управления представлением и настройки сцены.~~
- ~~Программирование оконного интерфейса, возможностей по настройке системы.~~

Решение для случая 3D

Чем по сути отличаются специализированные системы визуализации между собой?

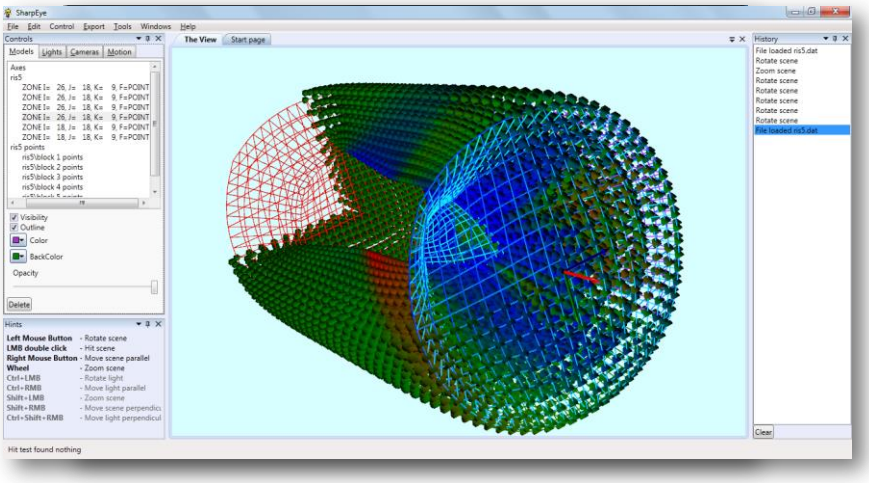
- Загрузка данных и преобразование их в визуальные сущности.
  У каждой задачи свой формат исходных данных, но в итоге они формируют 3-мерную сцену объектов.
- Управление со стороны пользователя.
  Разные задачи – разные элементы управления, управляющие загрузкой и отображением.

Остальные аспекты – могут быть универсальными.

- Подбор окружения исполнения (MFC, .NET Forms, WCF, VCL, Qt etc),
- Выбор графической библиотеки (OpenGL, DirectX , WebGL) или среды рендеринга (VTK, OGRE, Open Inventor etc),
- Реализация алгоритмов рендеринга.
- Программирование оконного интерфейса, возможностей по настройке системы.
- Настройка параметров сцены - освещение, камеры, анимация.
- Экспорт результатов визуализации.

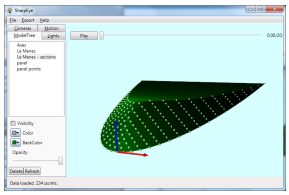
2011-2012 Проект SharpEye

Среда для создания специализированных систем визуализации.



Функции системы

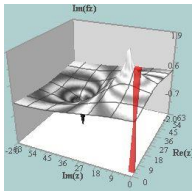
- Отображение 3D-объектов
  - Загрузка-удаление объектов, их хранение в нескольких иерархиях, операции над группами.
  - Операции с атрибутами объектов: изменение цвета, прозрачность, видимость-невидимость.
  - Геометрические операции со сценой (повороты, масштабирование, перемещение).
  - Операции со светом (перемещение источника, добавление, удаление, изменение яркости, включение-выключение) и камерами.
- Загрузка файлов с помощью модулей
- Управление сценой через API
- Экспорт изображений



Типовой сценарий разработки

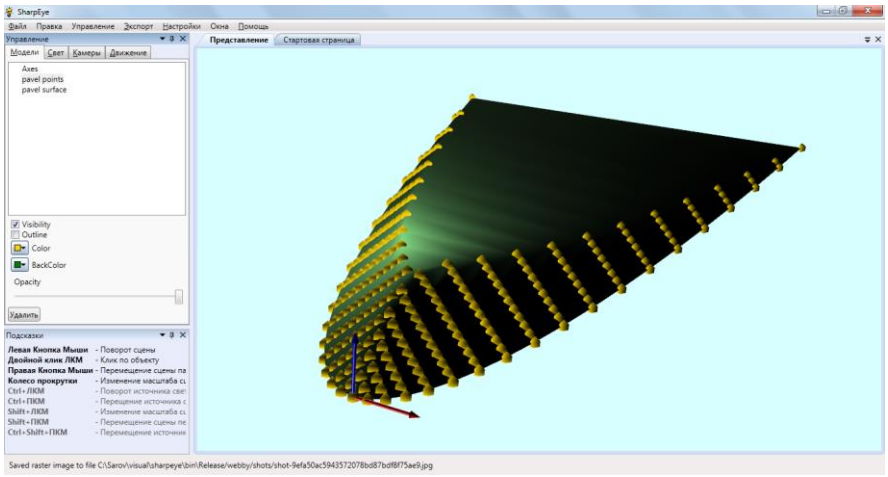
1. Реализовать функцию преобразования данных в визуальные сущности. C# (dll), Ruby (скрипт).
2. При желании, добавить элементы управления.

```
function load( f,scene )
{
  model m = scene.create(«model1»)
  while (!feof(f)) {
    double tri[9] = f.readline.split
    m.add_triangle( tri )
   }
   m.add_slider(«test1»,0,50,0.25)
   return 0;
}
```

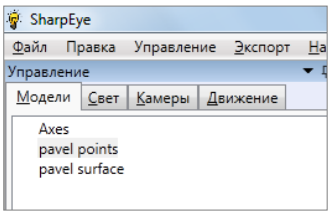


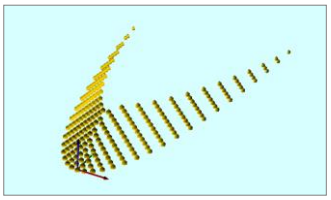
Пример

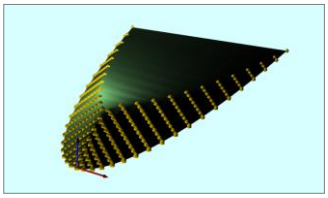
Задача. Дан набор точек {xyz} в файле D3.txt. Необходимо отобразить эти точки. А также построить по этим точкам триангуляцию функции z(x,y) и нарисовать и её.



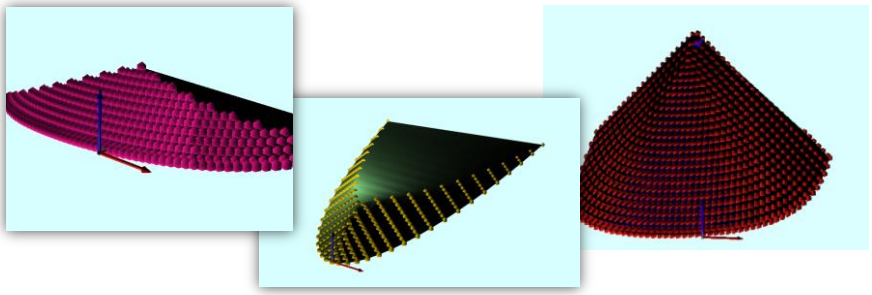
```
class LebedevSurface
    def feel(p)     # Определяем, наш ли это файл?
     p.first_lines =~ /X\s+Y\s+Z/ ? 2.5 : 0.0
    end

    def load(p)    # Загрузка файла
     m = add_model( "#{p.name} points" )

     file = File.new( p.path , "r")
     file.gets

     vertices = List.of(Triangulator::Geometry::Point).new

     while line = file.gets
        items = line.strip.split(/\s+/) [0..2]
        x,y,z = items.map{ |i| i.to_f }

        m.add_sphere( x, y, z, 0.10, 3, 2 )
        vertices << Triangulator::Geometry::Point.new(x,y,z)
     end

     tris = Triangulator::Delauney::Triangulate(vertices)

     m2 = add_model( "#{p.name} surface" )
     vertices.each{ |v| m2.add_node( v.X, v.Y, v.Z ) }
     tris.each{ |t| m2.add_triangle( t.p2, t.p1, t.p3 ) }
    end
end
```





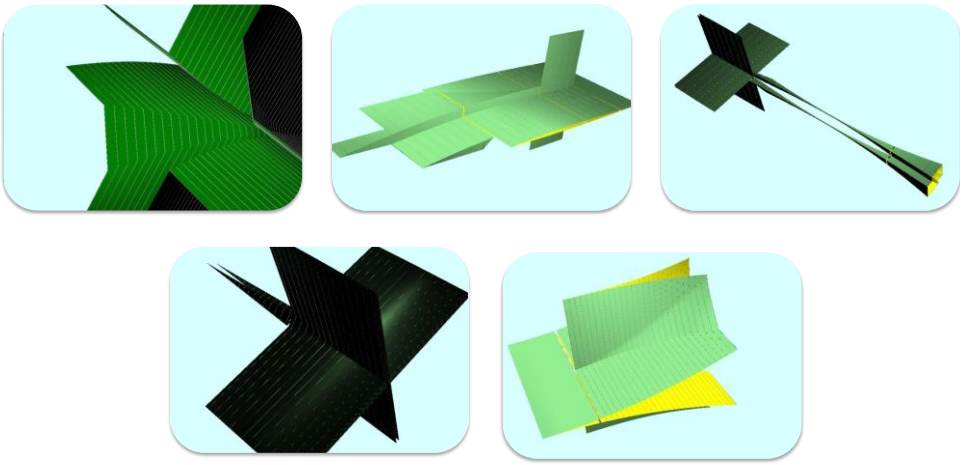


Задача быстродействия с круговой вектограммой скоростей на плоскости и связанная с ней задача Дирихле для дифференциального уравнения Айзекса – Беллмана



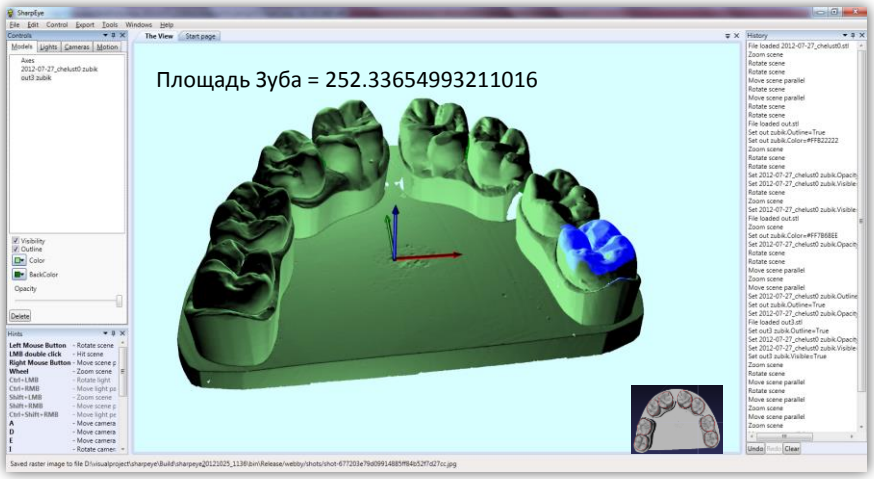
- Brykalov, S.A., Lebedev, P.D., Uspenskii, A.A., Ushakov, A.V. Symmetry Sets in Construction of a Minimax Solution for a Bellman-Isaacs Equation, Proceedings of the 18th IFAC World Congress, Edited by S. Bittanti, A. Cenedese, S. Zampieri, Milan, 2011, Vol. 18, Part 1, IFAC PapersOnLine Identifier: 10.3182/20110828-6-IT-1002.00744. http://www.ifac-papersonline.net/Detailed/51871.html
- Lebedev, P.D, Uspenskiy, A.A. "Symmetry Sets in a Solution of a Velocity Problem " 12th Viennese Workshop on Optimal Control, Dynamic Games and Nonlinear Dynamics May30th - June 2nd, 2012. P. 102
- П. Д. Лебедев, А. В. Ушаков. Аппроксимация множеств на плоскости оптимальными наборами кругов // Автоматика и телемеханика, 2012, № 3, С. 79–90.

Задача исследования максимальных стабильных мостов в линейных дифференциальных играх с двумерной эквивалентной фазовой переменной



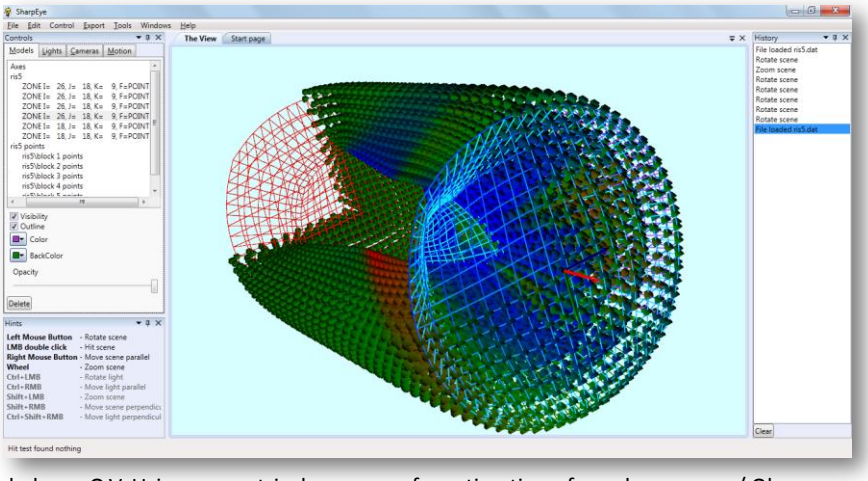
- Ganebny S.A., Kumkov S.S., Le Menec S., Patsko V.S., Model problem in a line with two pursuers and one evader // Dynamic Games and Applications, No.2, 2012, pp. 228–257.

Анализ свойств жевательной поверхности в исследовании перспективных методов восстановления зубов.



- Гимаева Зилия: Оптимизация реставрации зубов керамическими вкладками. Микропотезирование. Монография.

Визуализация расчетных сеток



Ushakova, O.V. Using geometrical measures for estimation of nondegeneracy / Olga Ushakova, Nataliya A. Artemova (Artyomova) // IMACS/ISGG Int. Workshops MASCOT&ISGG 2012, Las Palmas de Gran Canaria.- p. 201

Резюме по SharpEye

- Создан конструктор систем визуализации.
- Хорошее управление через пользовательский интерфейс и API.
- Быстрая разработка модулей.
- Языки модулей – C# и Руби.
- Присутствует документация.
- Решено несколько задач визуализации.

Выявленные возможности развития

- Вовлечение ученых в разработку.
- Краткость модулей.
- Технологичность.

Новый проект: Scheme2

- Цель: разработать подход к лаконичному описанию сцен научной визуализации

22

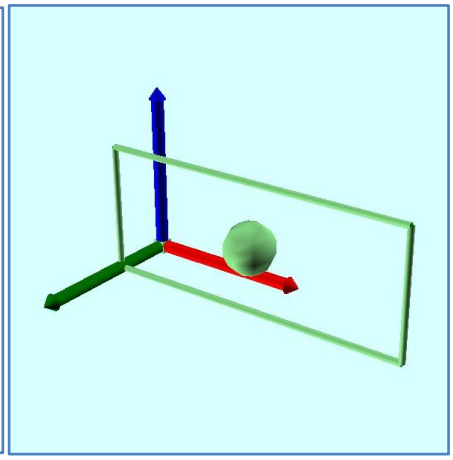
Структура

- Язык описания сцены
- Сервер данных
- Программы визуализации

23

Задача: нарисовать прямоугольник и сферу.

```
set scene.a.linestrip = array 5 3
1 0 0
1 5 0
1 5 2
1 0 2
1 0 0

set scene.a.spheres = array 1 3
1 2.5 1

set scene.a.spheres.radius = 0.4
set scene.a.linestrip.radius = 0.05
```

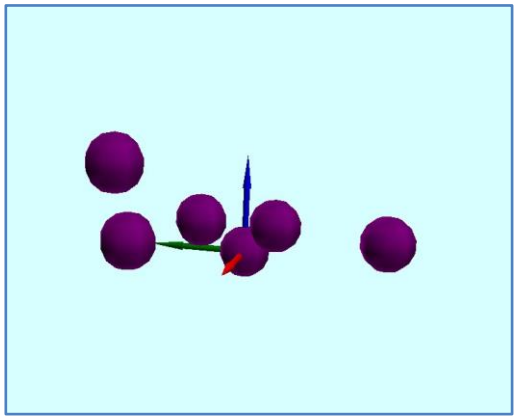


24

Задача: нарисовать сферы с координатам из файла data2.txt.

```
set scene.a.spheres.color = purple
set scene.a.spheres.radius = 0.7
set scene.a.spheres = array 6 3 <<file data2.txt
```

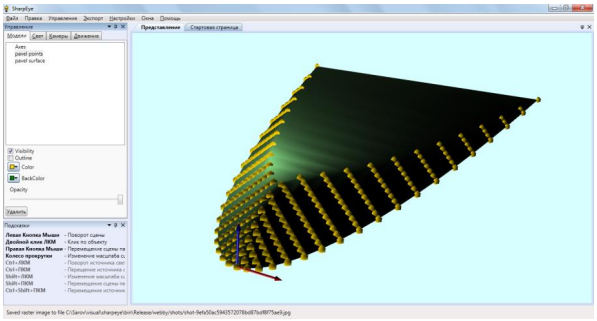
```
*** data2.txt
0 0 0
1 1 1
-1 1 1
2 5 3
-4 2 1
2 4 1
```



25

Задача. Дан набор точек {xyz} в файле D3.txt. Необходимо отобразить эти точки. А также построить по этим точкам триангуляцию функции z(x,y) и нарисовать и её.

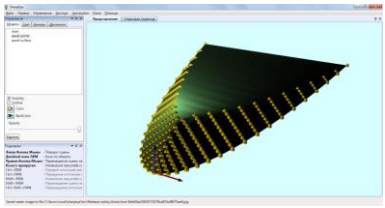
```
set scene.d3.spheres = array * 3 <<file D3.txt
set scene.d3.spheres.radius = 0.03

set scene.d3.trimesh = &scene.d3.spheres.triangulate
set scene.d3.trimesh.nodes = &scene.d3.spheres
set scene.d3.trimesh.color = green
```



Было. Стало.

```
class LebedevSurface
        def feel(p)     # Определяем, наш ли это файл?
          p.first_lines =~ /X\s+Y\s+Z/ ? 2.5 : 0.0
        end

        def load(p)    # Загрузка файла
          m = add_model( "#{p.name} points" )

          file = File.new( p.path , "r")
          file.gets

          vertices = List.of(Triangulator::Geometry::Point).new

          while line = file.gets
             items = line.strip.split(/\s+/) [0..2]
             x,y,z = items.map{ |i| i.to_f }

             m.add_sphere( x, y, z, 0.10, 3, 2 )
             vertices << Triangulator::Geometry::Point.new(x,y,z)
          end

          tris = Triangulator::Delauney::Triangulate(vertices)

          m2 = add_model( "#{p.name} surface" )
          vertices.each{ |v| m2.add_node( v.X, v.Y, v.Z ) }
          tris.each{ |t| m2.add_triangle( t.p2, t.p1, t.p3 ) }
        end
end
```

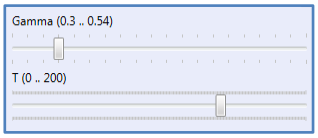
```
set scene.d3.spheres = array * 3 <<file D3.txt
set scene.d3.spheres.radius = 0.03

set scene.d3.trimesh = &scene.d3.spheres.triangulate
set scene.d3.trimesh.nodes = &scene.d3.spheres
set scene.d3.trimesh.color = green
```



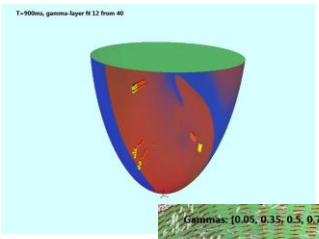
Некоторые свойства языка

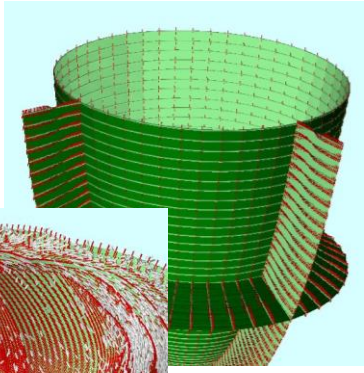
- Сцена описывается с помощью набора объектов в текстовом виде
- Возможность загрузки внешних файлов и вызова программ
- Автоматическое выявление параметров

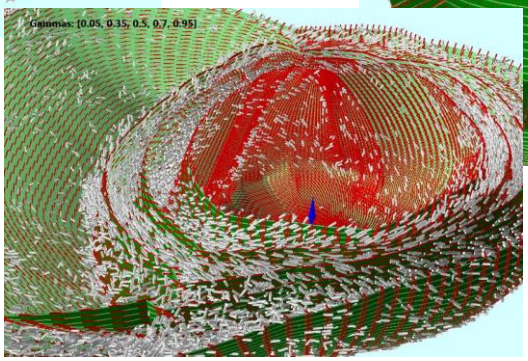
set scene.my.spheres[t=0] = …
set scene.my.spheres[t=1] = …
….



Задачи моделирования сердца



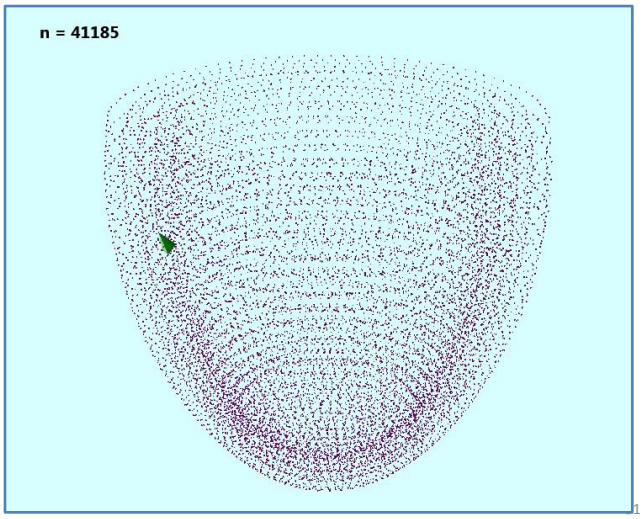




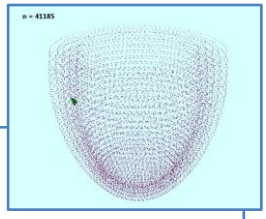
Задача:
Отобразить все точки сетки для момента времени T.
Отобразить тетраэдр с номером N при заданном T.

Узлы:
0mech.dat
1mech.dat
…
223mech.dat

Номера вершин тетраэдров:
tet_mesh.dat

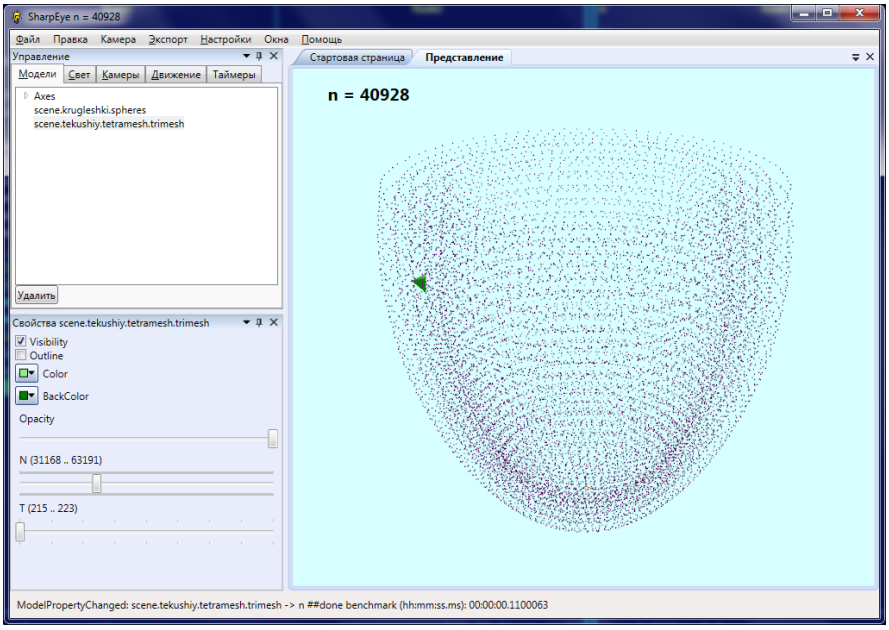


Решение

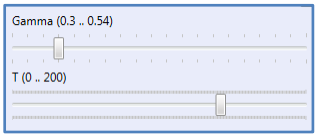


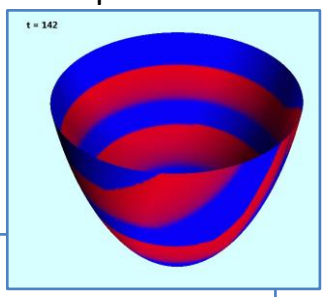

```
set koordinaty[t=any] = array * 3 <<file {get t}mech.dat

set scene.a.spheres = &koordinaty
set scene.a.spheres.color = purple

set idx = array * 4 <<file tet_mesh.dat

set params.t.values  = &range[min=0][max=223]
set params.n.values = &range[min=0][max=&idx.d1]

set scene.tekushiy.tetramesh[n=any] = &idx.skiplines[skip={get n}][len=1]
set scene.tekushiy.tetramesh.nodes = &koordinaty
```

32



Задача. Загрузить файл с координатами точек {gamma psi fi x y z}. Построить сечение по gamma. Раскрасить его сообразно значениям из файлов allpoints{T}.txt.gz

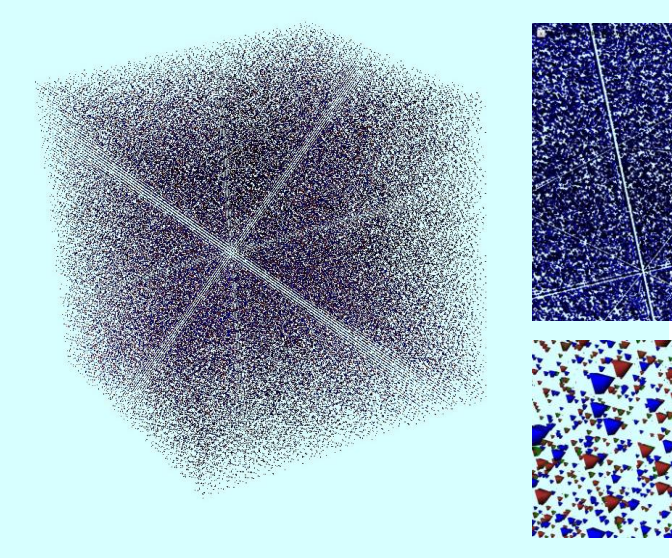


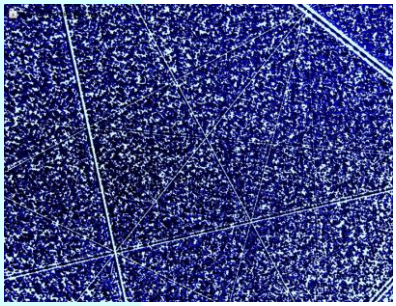


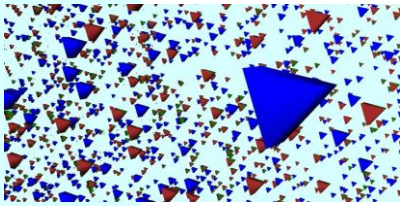

```
set heartdata = array 1260000  9 <<file el20gam03055.hrt

set params.gamma.values = &heartdata.columnvalues[column=0]
set params.t.values     = &range[min=0][max=200][step=1]

set gammalines[gamma=any] = &heartdata.findlines[column=0][value={get gamma}]

set d1 = &heartdata.getcolumns[columns=1,2,3,4,5]
set scene.gamma_slice.lattice[gamma=any] =  &d1.getlines[lines=&gammalines]

set allpoints[t=any] = array 1260000 1 <<file allpoints{get t}_.txt.gz
set scene.gamma_slice.lattice[gamma=any].scalar_values[t=any] = &allpoints .
            getlines[lines=&gammalines]
```

35

Компьютерное моделирование атомной структуры нестехиометрического монооксида титана TiOy







Особенности Scheme2

- Краткость.
- Содержание объектов: как в описании сцены, так и во внешних файлах или программах .
- Возможность работы с различными программами визуализации.
- Возможность работы по сети.

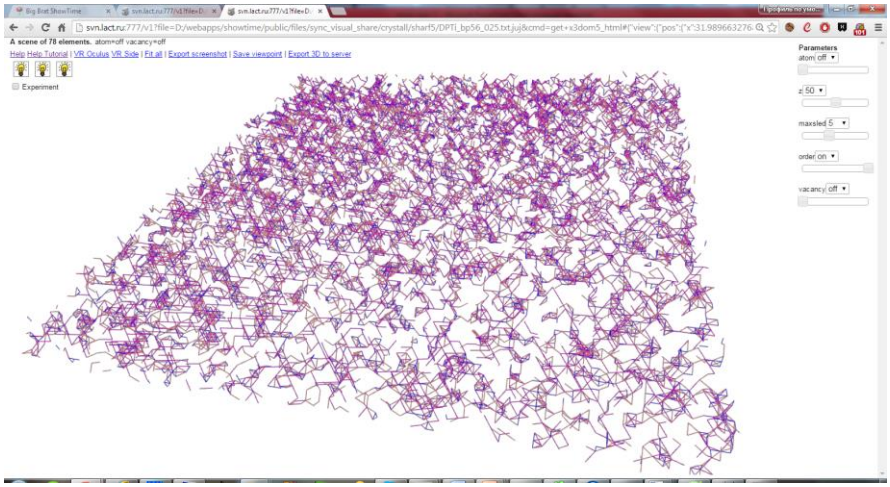
- Декларативный язык.
- Императивные возможности.

Не зависит от программ визуализации. Одни и те же данные можно смотреть разными программами.

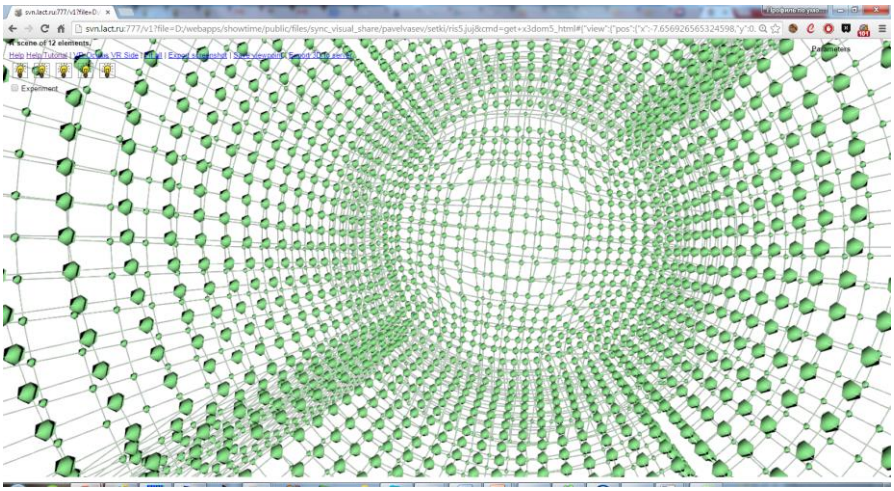
На текущий момент подключены:

- SharpEye (для Windows)
- X3Dom (для веб-браузеров)

Работа в браузере

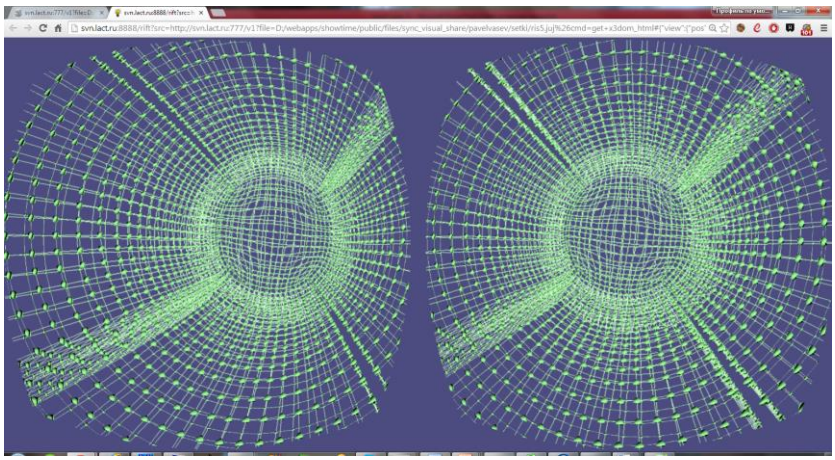


Работа в браузере



Виртуальная реальность

- Oculus Rift, Side by side



http://view.lact.ru