

# **КОНТЕКСТНАЯ ГРАФИЧЕСКАЯ СРЕДА ПРОСТРАНСТВЕННОЙ ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ В МЕХАНИКЕ СПЛОШНЫХ СРЕД**

**В. Н. Храмушин**

*Санкт-Петербургский государственный университета*

E-mail: Khram@mail.ru

В исследовательском проектировании, построении и последующем анализе достоверности моделируемых процессов в ресурсоемких вычислительных экспериментах, что особо востребуется при изучении нестационарных процессов в механике сплошных сред, становится весьма актуальным использование открытой и легко модифицируемой программной среды для пространственной визуализации быстротекущих физических явлений непосредственно в ходе суперкомпьютерных расчетов. Важным условием такой визуализации является минимальность воздействия на вычислительные процессы, с возможностью внешнего влияния на реологические параметры моделируемой физической среды и критерии динамической или гибридной перестройки вычислительных процессов. Практически все современные вычислительные комплексы обладают встроенными графическими средствами, обеспечивающими быструю визуализацию пространственных геометрических объектов с использованием независимых многоядерных процессоров, которые в полной мере способны обеспечивать решение сформулированной задачи для параллельной визуализации текущих результатов без существенного влияния на основные вычислительные процессы. В настоящем исследовании рассматривается вариант построения программного комплекса на базе графической среды программирования OpenGL, окружаемой инструментальными средствами для работы со временем и интервальными таймерами, устройствами ввода информации и представления текстовых данных на предельно низком уровне прямого ввода/вывода информации и обработки прерываний в OS Windows.

Ключевые слова: алгоритмы, вычислительные операции, графическая визуализация, параллельное программирование, функции, числовые объекты.

© 2018 В. Н. Храмушин

## **1. Введение**

Прямые вычислительные эксперименты представляются новым и весьма эффективным инструментом для поисковых исследований физических явлений и нестационарных процессов при решении широкого круга прикладных или производственных задач, в том числе при построении сложных счетно-решающих устройств с элементами автоматизированных экспертных и управляющих систем в междисциплинарных областях научных знаний или, даже, при необходимости адекватных эмпирических оценок и прогнозов характера взаимодействия инженерно-технических объектов с окружающей природной средой. Развитие современной вычислительной техники наилучшим образом соответствует новым техническим требованиям к реализации необходимого графического и интерактивного инженерно-технического инструментария, чему в первую очередь служат автономные графические средства для визуализации и сквозного контроля вычислительных процессов, развитая система супервизоров управления множеством арифметико-логических и аналоговых моделирующих устройств, включая системы прерываний с таймерами или внешними консолями управления информационными потоками и вычислительными комплексами.

Работа выполняется в рамках проекта «Виртуальный полигон», целью которого является создание системы поддержки принятия решений для моделирования, прогнозирования и предотвращения опасных ситуаций на море. Предусматривается задействование высокоэффективных графических средств визуализации результатов моделирования в реальном времени.

## **2. Построение интерактивной графической среды**

Среди целей разработки интерактивной графической подсистемы для выполнения инженерных прикладных изысканий с использованием прямых вычислительных экспериментов видится поиск инженерных решений в механике сплошных сред, где искомый результат получается методами плавного возбуждения с последующим выходом на режимы установления исследуемых физических явлений и процессов. Фактически это означает необходимость задействования суперкомпьютерных вычислительных ресурсов с возможностью повсеместного и непрерывного визуального контроля моделируемых законов гидромеханики, когда заданные или искомые природные процессы возникают в результате продолжительных вычислений изначально в подготовительных режимах искусственного установления или релаксации негативных численных эффектов, с последующим управляемым выходом на моделирование реальной физики на основе неких гибридных схем, осуществляющих автоматическое использование чисто математических, асимптотических или даже эмпирических алгоритмов, в зависимости от реологического состояния моделируемой сплошной среды в локальных подобластях и в зонах нарушения критериев устойчивости во времени, или в подобластях явного обрушения аппроксимационной гладкости в пространстве.

Современные средства программирования во многом утратили разнообразие архитектурных решений для специальных прикладных задач, однако монополия языка C++ отчасти восполняется расширениями объектно-ориентированного подхода для ограниченного конструирования необходимых в гидромеханике языковых описаний и вычислительных операций.

### **2.1. Технические особенности построения вычислительной среды**

Рассматривается вариант построения специализированного интерактивного графического интерфейса для проектирования, разработки и реализации прикладных вычислительных экспериментов с ориентацией на задачи моделирования нестационарных процессов в инженерной гидромеханике.

1. Изначально предусматривается, что ресурсоемкие расчеты выполняются в параллельной вычислительной среде без избыточного влияния программно-аппаратного окружения, где несмотря на повышенный приоритет внутрипрограммных или внешних

управляющих прерываний, возникающие транзакции исполняются на независимых графических устройствах или других параллельных сопроцессорах.

2. Визуализация всех моделируемых физических явлений выполняется с задействованием произвольного количества независимых текстово-графических окон, в которых могут раздельно представляться параллельно работающие или функционально независимые подпрограммы практического моделирования. В каждом таком окне базовыми объектами устанавливаются наложенные текстовые и графические площадки, упорядочиваемые стековыми списками по уровням перекрытия в общем графическом окне.

3. В постановке проектной задачи предусматриваются различные варианты прерываний вычислительного процесса. Внутренние управляющие запросы могут возникать при нарушении критериев устойчивости или аппроксимационной гладкости численного решения, реакция на них предусматривается на этапах разработки вычислительных моделей. Внешние прерывания поступают от экранного курсора (мышь, перо и т.п.) и должны привязываться к конкретной текстовой или локальной графической площадке. Прерывания от клавиатуры передаются активному окну или игнорируются, что зависит от внешнего управления компьютером. Прерывания от таймера, так же как и от других инородных для компьютера устройств, обрабатываются специальными процедурами, которые переадресуют на себя управление соответствующими аппаратными ресурсами.

При необходимости любые прерывания могут породить интерактивные запросы на текстовых консолях, в новых графических окнах или с наложением текстово-графических площадок, встраиваемых в активные списки оконного стека, если это необходимо и возможно.

Элементы вышеописанных технических тезисов исполнены в форме многооконного контекстно-зависимого интерактивного графического интерфейса Window-Place [1] с задействованием низкоуровневого доступа к стандартной графической подсистеме OpenGL, что удовлетворяет требованиям по визуализации результатов вычислительного эксперимента с использованием внешних или параллельных графических устройств. Вычислительная среда дополняется независимым комплексом базисных объектов и операций тензорной математики [2], что служит задействованию традиционной для гидромеханики математической нотации, где пространственные перемещения задаются операциями векторного сложения, а деформации и вращения соответствующими тензорными произведениями с матрицами размерности [3x3].

### 2.3. Window-Place

Объектно-ориентированный комплекс Window-Place служит интерфейсом между программой, операционной системой Windows и внешними устройствами ЭВМ: графическим терминалом; клавиатурой и курсором «мышь», компьютерными часами и интервальным таймером, и включает управление текстами, в том числе со встроенными растровыми шрифтами. В прикладной программе не ограничивается включение других внешних устройств, как например *измерительной телеметрии*.

Универсальный базовый класс Place представляет текстово-графическую площадку, в которой устанавливается и масштабируется сцена в однородных координатах OpenGL, в производном классе Window образуется стековый список таких площадок с вершиной от базового включения.

Производный класс Window создает интерфейс одного независимого окна на основе стандартного инструментария (SDK) операционной системы Windows; и устанавливает индивидуальный пользовательский интерфейс для работы с часами и таймером; а в моменты активности окна Window обеспечивает опросы и ожидание откликов от клавиатуры и графического курсора, непосредственно управляет переадресацией прерываний по связанному списку независимых окон внутри базового класса Place::Site.

Класс Place, будучи базовым для Window, отмечается ссылкой в начале стекового списка аналогичных наложенных текстово-графических площадок Window::Peak, а обнуление этого адреса означает закрытие графического окна. Именно Place поддерживает трехмерную контекстно-зависимую графику OpenGL, устанавливает режимы и контролирует параметры пространственной сцены и плоских фрагментов для текстовых страниц, меню и справок. При изменении размеров площадок в Place выполняется внутренняя конвертация растрового

изображения без запросов на полную перерисовку пространственной сцены или текстовых площадок.

В настоящей версии графической среды Window::Place поддерживаются два варианта реагирования на внутренние прерывания для обработки изображения, так же как и для внешних команд от графического курсора и таймера. В простейшем случае предусматриваются стандартные параметры графической среды Window::Place, с последующим обращением по заранее сохраненным адресам к внешним контекстно-зависимым исполнительным процедурам:

```
void (*inPass)( int X,int Y ); // процедуры контекстной обработки
void (*inPush)( int State,int X,int Y ); // прерываний графического курсора
bool (*extTime)(); // внешняя функция обработки прерываний таймера
void (*extDraw)(); // контекстное рисование с базовыми настройками
void (*extActive)( int Width, int Height ); // построение контекстной среды
```

Здесь X, Y – растровые координаты; State – признак нажатой кнопки на графическом курсоре. Width, Height – растровые размерности графической площадки необходимы для пересчетов масштаба и пропорций графического изображения.

Полиморфизм производных классов более высокого уровня над базовой структурой Window для прикладных вычислительных объектов допускает подмену виртуальных функций в базовых классах нижнего уровня, что важно для интеграции графической подсистемы Window::Place в основной информационный интерфейс вычислительного эксперимента.

```
virtual void Place::Mouse( int X,int Y );
virtual void Place::Mouse( int State, int X,int Y );
virtual bool Window::Timer();
virtual Place& Place::Draw();
virtual Place& Place::Active( int Width, int Height );
```

В комплексе Window-Place клавиатура не оказывает прямого воздействия на исполняющиеся алгоритмы, и коды клавиш просто передаются в кольцевой буфер активного окна. По необходимости сохраненные коды могут выбираться из кольцевого буфера, либо с блокированием выполнения программы по запросу с ожиданием: WaitKey(), либо в режиме считывания-опроса без остановки вычислительного процесса: ScanKey() и ScanStatus().

### **3. Числовые объекты и методы тензорной математики.**

Все геометрические построения графического интерфейса OpenGL выполняются в однородных координатах, где векторы перемещения в пространстве и параметры перспективных проекций оконтуривают трехмерные тензоры деформации и вращения, что образует унифицированные матрицы с размерностью [4x4], участвующие в произведениях как для выборки глобальных координат – для настройки сцены графической визуализации, так и для локальных отсчетов и деформаций отдельных объектов – трехмерных геометрических моделей.

В решении задач механики сплошных сред использование однородных координат отчасти избыточно, и в то же время допускает некоторые неопределенности в прямых и обратных пространственных преобразованиях. Главная или линейная часть уравнений гидромеханики, так же как и уравнения движения деформируемых частиц строятся из операций сложения векторов при описании перемещений в пространстве, и операций произведения с трехмерными матрицами-тензорами для моделирования вращения и деформации в локальных системах отсчета. Такие вычислительные операции гидромеханики имеют наглядное представление в случае использования алгоритмического языка на базе тензорной математики, где, по крайней мере в рамках объектно-ориентированного подхода компилятору может быть передан контроль корректности преобразований в дуальных пространствах: взаимных преобразованиях в глобальных координатах и в локальных базисах в окрестности крупных частиц сплошной среды.

Ключевые объекты тензорной математики определяются в одномерных функциях, в специальных векторных построениях и с помощью операторов матричных (тензорных) пространственных преобразований.

1. Event – время, абсолютная и вездесущая скалярная величина:

```
struct Event{ long D; Real T } // Юлианская дата и время суток в часах
```

В операциях со временем `Event` допускается добавление нового интервала `T`, но компилятор будет блокировать сложение двух полных абсолютных отсчетов времени, а разность абсолютных отсчетов времени может дать только обычное число в качестве интервала времени. Таким образом строится некая арифметика для работы с любыми отсчетами времени, в которой на языковом уровне поддерживается корректность методов работы с числовым объектом.

2. `Point` и `Vector` – координаты в абсолютной и локальной системах отсчета:

```
struct Point // Точка отсчета в масштабе единого глобального пространства
{ Real X,Y,Z; } // прописные буквы для абсолютно зависимых величин
struct Vector // Локальные отсчеты в связанном трехмерном пространстве
{ Real x,y,z; } // необходимы для записи локальных законов гидромеханики
```

По аналогии со временем здесь контролируются операции сложения векторов в абсолютных и локальных системах координат, а также как и в случае пространственных преобразований с использованием тензорных/матричных операторов деформации и вращения.

3. `Matrix` и `Tensor` – характеристики пространства в локальных измерениях смежных континуумов, необходимы для записи законов механики, реологического состояния жидкости, для прямых и обратных преобразований в абсолютных и локальных пространствах.

```
struct Matrix // матрица односторонних пространственных преобразований
{ Vector x,y,z; } // триекторы Евклида для косоугольной системы координат
class Tensor: public Matrix // базисные векторы ненулевые и некомпланарные
{ Vector xi,yi,zi; // обратный тензор служит ускорению произведений
Real d; } // определитель - смешанное произведение базисных векторов
```

Объект `Matrix` может быть диадой, вырожденным и даже нулевым. Определенность числового объекта `Tensor` служит критерием существования решения и корректности в определении реологического состояния моделируемой сплошной среды в локальной подобласти вычислительного эксперимента.

4. `Base` – простейший из сложных числовых объектов, и связанные с ним методы пространственных преобразований для величин в дуальных координатных системах:

```
struct Base: Tensor,Point // тензор формы с местоположением объекта
{ Base& operator = ( Point r ){ Point::operator=( r ); return *this; }
Base& operator = ( Matrix m ){ Tensor::operator=( m ); return *this; }
Base& operator +=( Vector v ){ Point::operator+=( v ); return *this; }
Base& operator -=( Vector v ){ Point::operator-=( v ); return *this; }
Point out( Vector a ){ return (Point)*this + (Tensor)*this * a; }
Vector in( Point A ){ return Vector( A-(Point)*this ) * (Tensor)*this; }
};
```

Строго ограниченный набор методов работы с числовым объектом `Base` соответствует его внутренней сложности и логической уникальности. Функция `Point out( Vector a )` – осуществляет переход из локальной системы отсчета к абсолютным или глобальным координатам; функция `Vector in( Point A )` – позволяет получить проекции смежных объектов в собственных локальных отсчетах через преобразования в глобальном пространстве вычислительного эксперимента.

## 4. Заключение

В реализации унифицированного объектно-ориентированного комплекса программ, названного `Window-Place`, представляется многооконный интерфейс `Window` для работы с внешними устройствами и компьютерной аппаратурой, с наложением текстовых и графических страниц `Place`, непосредственно поддерживающих контекстно-зависимые примитивы и операции функциональной графической среды `OpenGL`. Встроенные виртуальные процедуры для масштабирования наложенных страниц, для обработки программных и аппаратных прерываний и др., образуют интуитивно понятные правила построения пространственных объектов и операций с ними, предопределяя визуальное естественное поведение графических площадок с наложением текстовых фрагментов. При этом полиморфизм производных классов для пользовательских вычислительных объектов допускает подмену базовых виртуальных

функций, что может служить ускорению вычислений и повышению эффективности ресурсоемкого суперкомпьютерного эксперимента. Включение в единую систему числовых объектов тензорной математики также служит эффективности построения, отладки и практической реализации вычислительных моделей гидромеханики.

## **Благодарности**

Работа выполнена в рамках гранта Санкт-Петербургского государственного университета №26520170.

## **Список литературы**

[1] Храмушин В. Н. «Контекстная графика» – (Window-Place) Контекстно-зависимая среда построения трехмерной графики OpenGL с использованием виртуальных процедур C++ и многооконного интерфейса Windows со стекowym наложением графических и текстовых фрагментов. Сертификат Роспатента, СахГУ № 2010615850 от 8 сентября 2010 г.

[2] Храмушин В. Н. «Программа для построения числовых объектов и функций трехмерной тензорной математики при реализации вычислительных экспериментов в гидромеханике» (Tensor). Сертификат Роспатента, СПбГУ № 2013619727 от 14 октября 2013 г.

# **CONTEXTUAL GRAPHICAL ENVIRONMENT FOR SPATIAL VISUALISATION OF NUMERICAL EXPERIMENTS IN CONTINUUM MECHANICS**

**V. N. Khramushin**

*Saint Petersburg State University*

E-mail: Khram@mail.ru

The usage of open-source and easy-to-modify programme code for spatial visualisation of fast-evolving physical phenomena directly in the course of supercomputer calculations becomes relevant in design, research, construction and subsequent analysis of the validity of the simulated processes in resource-intensive numerical experiments, which is especially important in the study of non-stationary processes in continuum mechanics. Such visualisation has to have minimal impact on computational processes and allow for setting rheological parameters of simulated physical environment and criteria for dynamic and hybrid restructuring of computational processes. Almost any computational platform have embedded graphical means for visualisation of spatial geometrical objects using multi-core processors, which may provide solution to the formulated problem in full measure for parallel visualisation of the current results without substantial impact on the core computational processes. In the present research work we study one possible version of software suite which is based on graphical programming environment OpenGL, surrounded by tools for working with time and interval timers, input devices and text data views at the lowest possible level of direct input/output and interrupt processing in Windows OS.

Keywords: algorithms, computational operations, functions, graphical visualization, numeric objects, parallel programming.

© 2018 Vasily N. Khramushin