

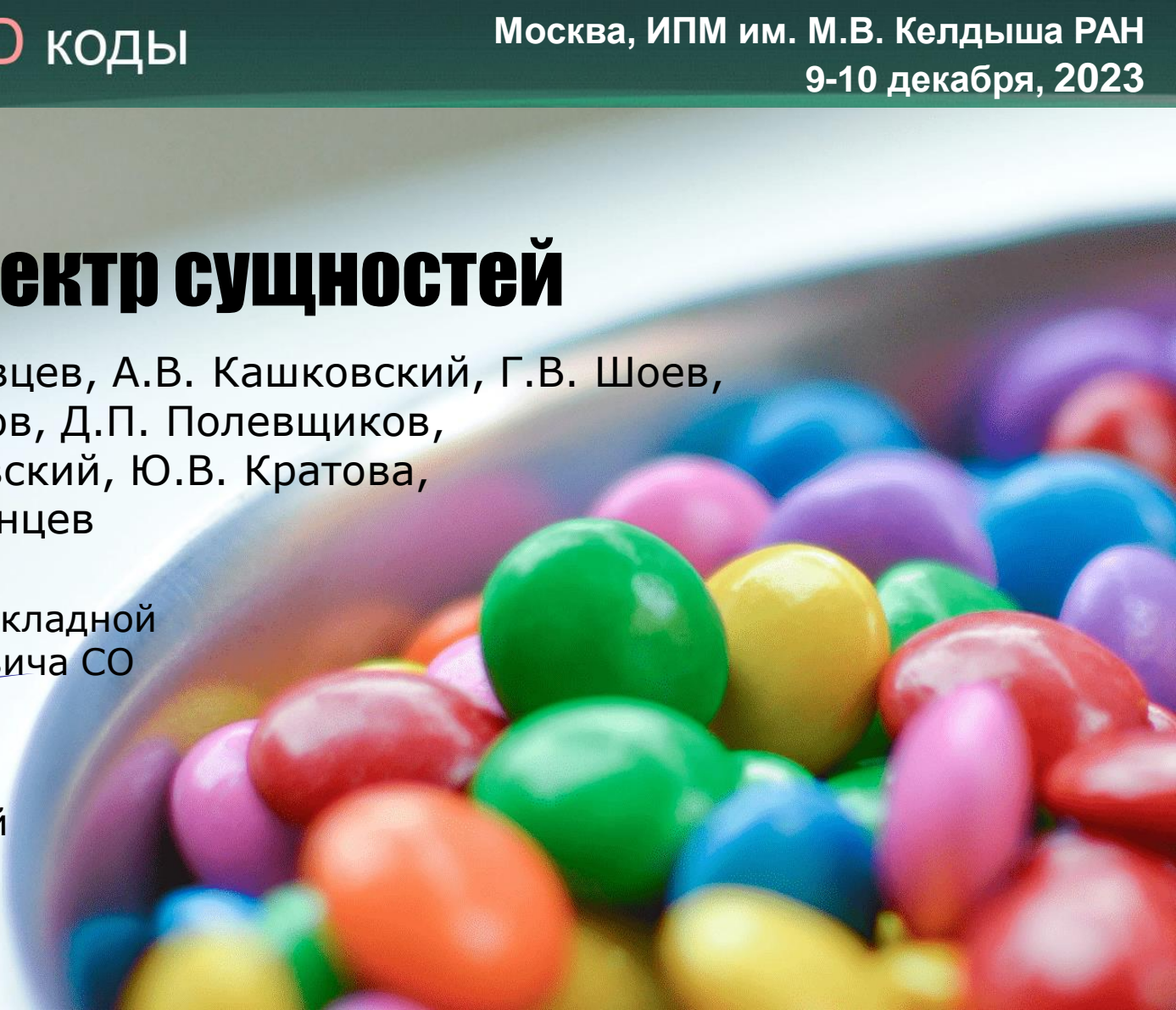
# НУСФС-R 2023: спектр сущностей

А.А. Шершнёв, А.Н. Кудрявцев, А.В. Кашковский, Г.В. Шоев,  
С.П. Борисов, Т.Ю. Шкредов, Д.П. Полевщиков,  
А.А. Королев, Д.В. Хотяновский, Ю.В. Кратова,  
П.В. Ващенко, А.С. Литвинцев

Институт теоретической и прикладной  
механики им. С.А. Христиановича СО  
РАН, Новосибирск



Лаборатория вычислительной  
аэродинамики



# Введение

- Традиционные для лаборатории задачи высотной аэродинамики + возникающие новые задачи
- Выше 80-85 км — метод ПСМ (пакеты семейства SMILE)
- Ниже необходим континуальный подход на основе полных уравнений Навье-Стокса
- CFD код «без конкретного назначения»: для нужд и лаборатории и, по возможности, всего Института в целом

# Общая информация

**HyCFS-R:** hybrid compressible flow solver

**Классы течений:** сжимаемые течения, в т.ч. с химической и термической неравновесностью

**Модели:**

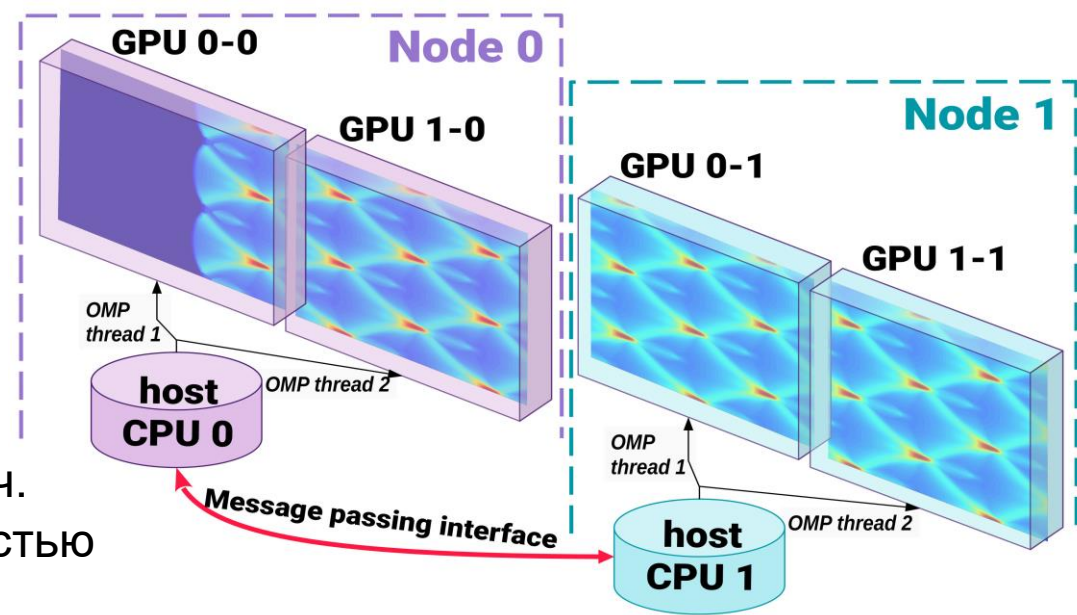
- совершенный газ (ламинарный / турбулентный режимы)
- одно- и многотемпературные модели с неравновесной химией
- химически реагирующая газовзвесь (частицы в газе) на основе эйлера подхода

**Сетка:** Структурированная многоблочная, в общих криволинейных координатах и (spoiler!) неструктурированная

**Параллельность:** многоуровневая CUDA/OpenMP/MPI

**Разработка:** C++ CUDA, Linux, SVN, GNU/make, GCC, gdb, valgrind

**Методы и схемы:** MUSCL1-5, WENO, RKTVD1-4, ASIRK2c, DPLUR



## Основные разработчики кода

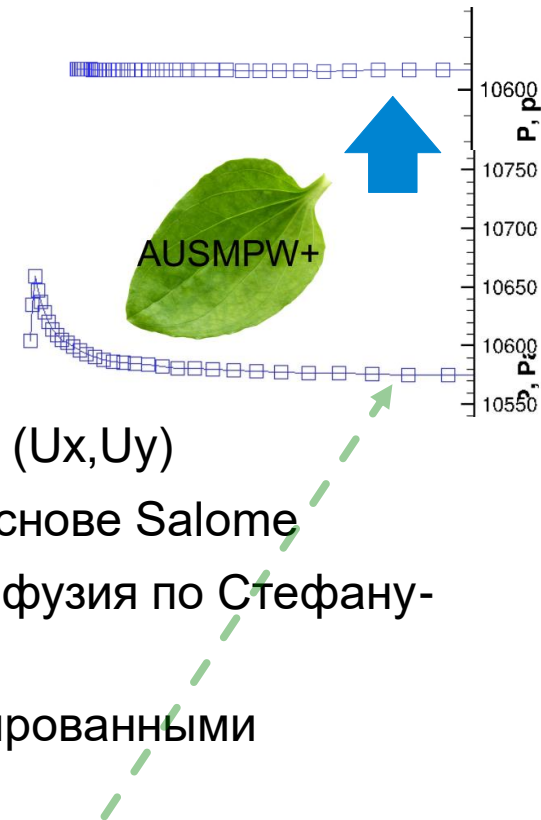
- ◆ А.Н. Кудрявцев — численные схемы и методы
- ◆ А.В. Кашковский — общая архитектура программы, многоплатформенность, низкоуровневая оптимизация
- ◆ С.П. Борисов — однотемпературная химия, газовая детонация, неявные схемы
- ◆ Г.В. Шоев — физико-химические модели, в т.ч. неравновесность и 2Т-химия, модели турбулентности, римановские солверы, ГУ
- ◆ Т.Ю. Шкредов — излучение, модели диффузии, ионизация, AUSM солверы, ГУ
- ◆ Д.П. Полевщиков — реализация многоблочности
- ◆ А.А. Королев — «подвижные» границы, демпфирующие моменты
- ◆ А.А. Шершнёв — кодирование, интеграция частей в единую систему, прочее

## Со-разработчики / консультанты / пользователи / тестеры

- ◆ П.В. Ващенко — ConfigReader, GUI
- ◆ А.С. Литвинцев — GUI
- ◆ А.В. Зайцев — консультации по RANS
- ◆ Т.А. Полянский — сетка, med/HDF
- ◆ Ю.В. Кратова — многофазность
- ◆ Д.В. Хотяновский — DNS расчеты
- ◆ А.И. Кутепова — DNS расчеты

# Новое за 2022-2023 года

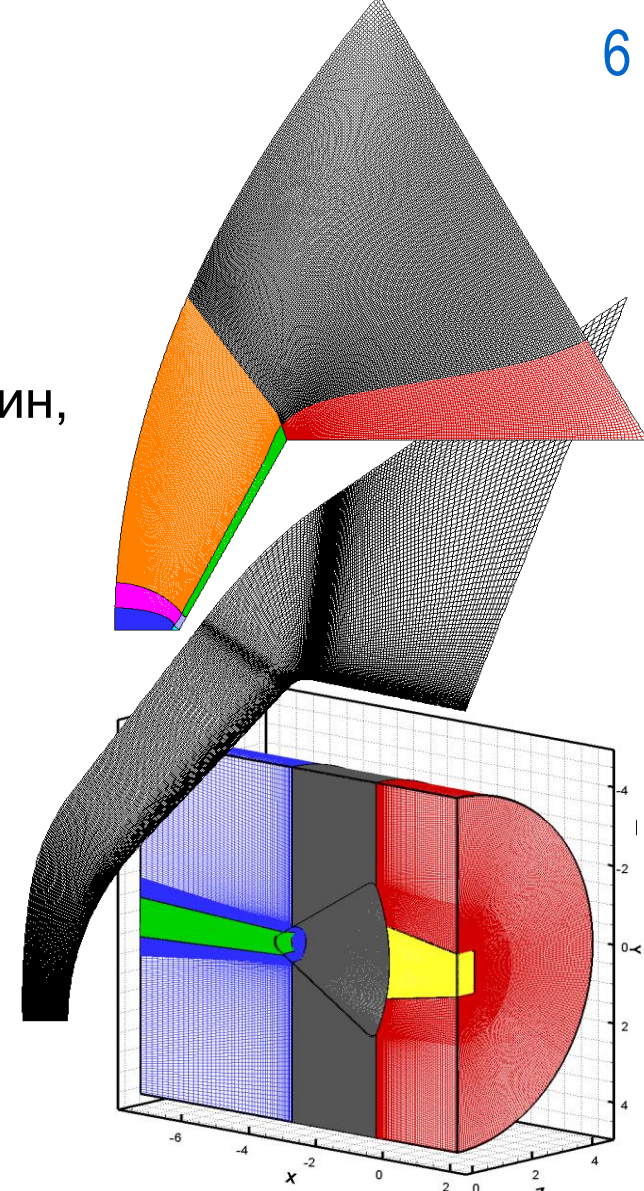
- **Неструктурированная сетка** в основной ветке!
- Новая читалка конфигурационных файлов на шаблонах и C++11
- Прототипирование расчёта на STL структурах
- Модификация MUSCL с реконструкцией на грань ( $U_n, U_t$ ) вместо ( $U_x, U_y$ )
- Начали делать графический интерфейс для пользователей на основе Salome
- Диффузия в смеси: не только Фик, но и многокомпонентная диффузия по Стефану-Максвеллу, включая баро- и термодиффузию
- Тепловое излучение газа в приближении плоских слоев с табулированными спектрами
- Много количественных изменений: новые римановские солверы (AUSM-типа и гибридные), граничные условия (pressure in/outlet, Riemann farfield, Tslip-Vjump стенка), колебательная релаксация для  $CO_2$ : VT и VV, мелочи типа патчей а ля Ansys, объемное энерговыделение, решение уравнения Эйконала для расстояния до стенки





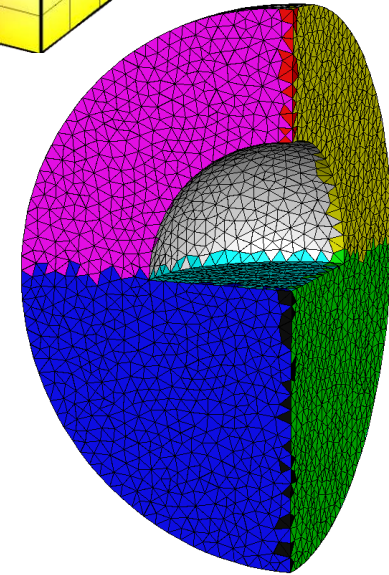
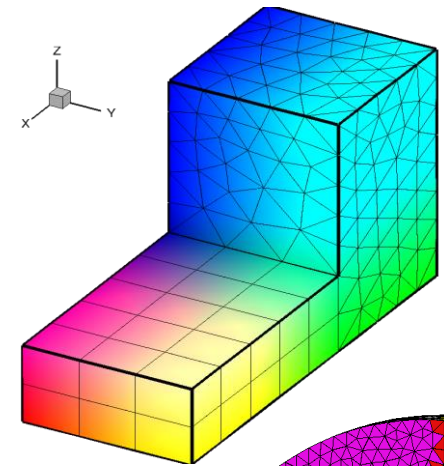
# Структурированная сетка

- **Многоблочная** в общих криволинейных координатах
- Опции для построения:
  - ◆ Самописные утилиты для простых конфигураций: клин, цилиндр, сфера, осесимметричные тела с 2D сеткой
  - ◆ Экспорт из ICEM CFD в формате Flowlogic
  - ◆ Утилита для авто-построения 3D многоблочной сетки для конфигураций с фюзеляжем/крылом из *триангулированной* поверхности
  - ◆ Утилита на Питоне для экспорта из Cadence Pointwise



# Неструктурированная сетка

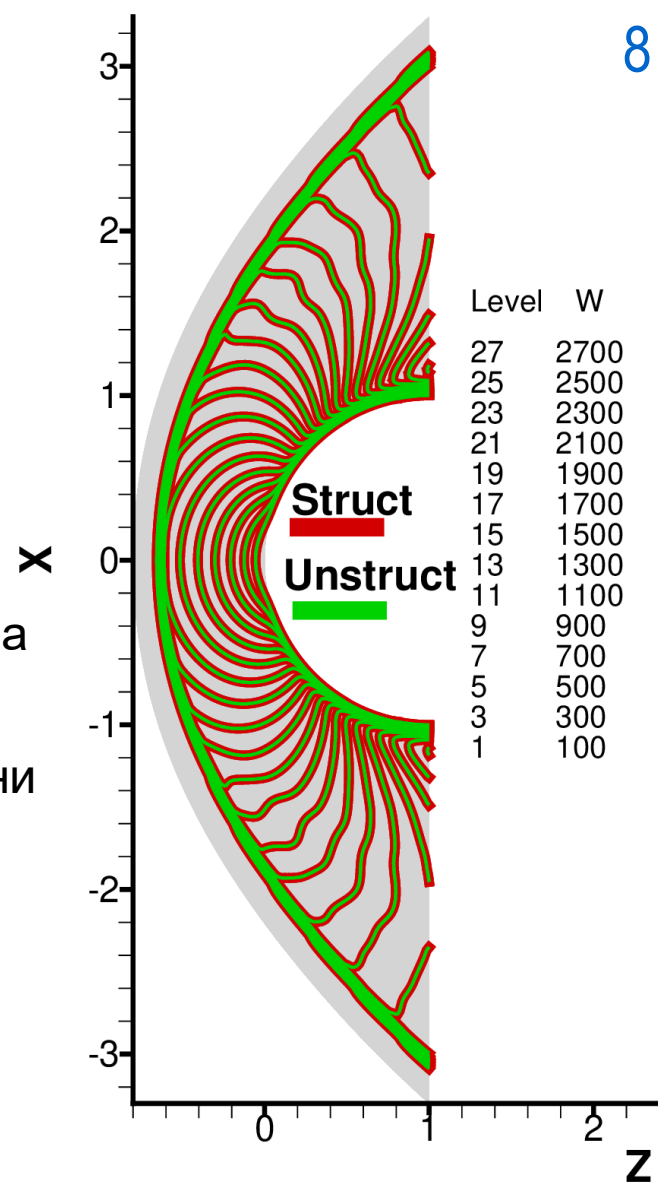
- С произвольными ячейками (mixed-elements).
  - ◆ Универсально — MUSCL\_1, на чистых тетра — MUSCL\_2, на чистых гекса MUSCL\_2/3
  - ◆ Сейчас только один блок сетки, необходимо добавить обмен между блоками. Нужно использовать задел – обмен между структурированными блоками
  - ◆ Конвертер из .med на базе библиотеки medcoupling
    - Salome => .med => HyCFS-R
    - Ansys / ICEM => CGNS => Salome => .med => HyCFS-R
  - ◆ Хочется уйти от med, потому что ненадежно и проблемно, но пока не решили куда идти
  - ◆ Пока что в попытках ослабить зависимость от сеткопостроителей добавляем в код процедуры пересчета ссылочных и геометрических массивов



# Унификация по типам сетки

Общий код, где это возможно:

- ◆ **OD** процедуры: вычисление коэффициентов переноса, химические правые части, явные шаги по времени, источниковые члены межфазных взаимодействий
- ◆ **Риман** и **ГУ**: в обоих случаях рассматривается пара ячеек, решается правкой индекса
- ◆ **MUSCL** реконструкция на гекса сетке также унифицирована с помощью небольших модификаций
- ◆ **Чтение** полей и узлов сетки, **экспорт** полей в Tecplot (как ни странно) делаются одними и теми же функциями
- ◆ **Диффузионные** производные не унифицируются
- ◆ **АДХ** в постпроцессинге также пока дублируют реализации





# ConfigReader на шаблонах

- В HC и PCM используется собственный формат конфигов
- Была библиотека, но накопились вопросы и наш коллега П.В. Ващенко написал новую версию. На шаблонах и C++14.
- Её опробовали на HyCFS-R, а после вставили и в SMILE++
- Теперь можно узнать, что прочиталось и что только пыталось прочитаться, ставить ограничения на значения и добавлять описания параметров, в HyCFS-R это позволяет автоматически выводить список всех параметров
- Есть макросы для часто повторяющихся значений. Юз-кейс – изменить «переключатель» в описаниях нескольких десятков реакций или что-то подобное
- Есть возможность сравнить конфигурации через утилиту на уровне групп и ключевых слов

```
#group DataGroup
  int_param    1 ! some test param
  float_par    3.141592 ! float val
  name         "Helium"
  array_inp    1. 11. 111.

  #group Subdata
    debug_mode 1
  #end
#end
```

Parameter	Inputplane/000ns.in	Inputtaxi/000ns.in	Input3D/000ns.in
> :BC[0]			
> :Data[0]			
> :Element[0]			
∨ :Mesh[0]			
├ :Mesh[0]:Nx	600	500	200
├ :Mesh[0]:Ny	1	1	200
├ :Mesh[0]:Nz	2000	2000	340
├ :Mesh[0]:iDim	2	2	3
├ :Mesh[0]:iaxis	0	1	
> :Postproc[0]			

Quit

# Текстовый прототип структуры переменных. Мотивация

*Примерный список физических моделей в коде:*

- **Термодинамика:** политропный газ, полиномиальные аппроксимации для  $c_{p,i}$ ,  $c_{v,i}$  и  $H_i$  (А. Буркат, Гупта), кусочно-линейные аппроксимации
- **Колебательная релаксация:** уравнение Ландау-Теллера и модель Кустовой-Облапенко, полученная строгими методами кинетической теории газов, VV-обмен (G. Candler), простая модель для многоатомных молекул ( $\text{CO}_2$  с равновесием между колебательными модами)
- **Скорость химических реакций:** закон Аррениуса, однотемпературные модели с зависимостью от давления (Petersen-Hansen), двухтемпературные модели (Парка, Кузнецова, Мачерета-Фридмана, Тринора-Маррона,  $\beta$ -модель, модель для  $\text{O}_2$  на основе данных стат.физики)
- **Диффузия:** закон Фика с эффективными коэффициентами диффузии для массовых или молярных концентраций, модель многокомпонентной диффузии (Стефан-Максвелл), учет баро- и термодиффузии
- **Излучение газа:** приближение плоского слоя с оптической моделью на основе табулированных спектров из Spcair
- **Многофазность:** взаимопроникающие континуумы,  $C_d$  формула Хендерсона, упрощенная химия (Федоров, Хмель)
- **Турбулентность:** модель Спаларта-Аллмареса, в основной формулировке из статьи Allmaras, Johnson, Spalart (1996), «SA-standard»

# Текстовый прототип структуры переменных

- Большое число физических моделей затрудняет инициализацию: нужно помнить их порядок в контейнере поля течения, а также в каком подмножестве уравнений величина участвует, какие вспомогательные массивы нужно завести
- Раньше всё было на ходу, сразу и контейнеры, и вспомогательные структуры
- Сейчас сначала прототип из текста и STL-контейнеров:

```
std::map<std::string, std::vector<std::pair<std::string, std::set<std::string>>>>
```

имя переменной

имя контейнера

```
[ "Q" ] -> < "RhU", "RhE", "Rh" >
           |
           { "GAS", "INVISC", "VISC" }
```

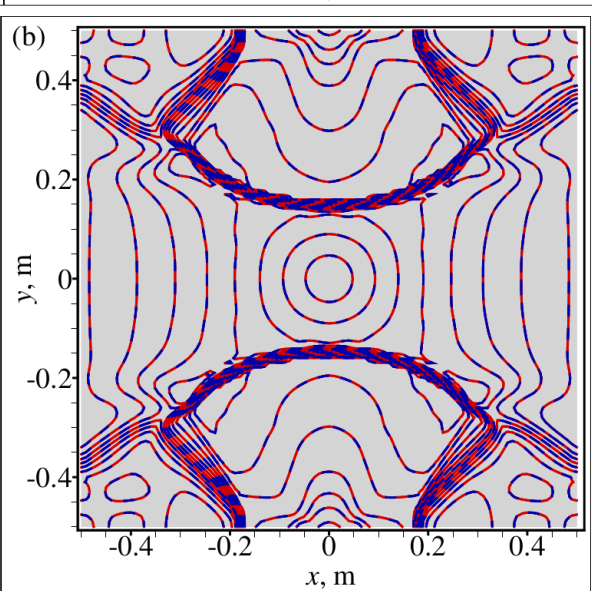
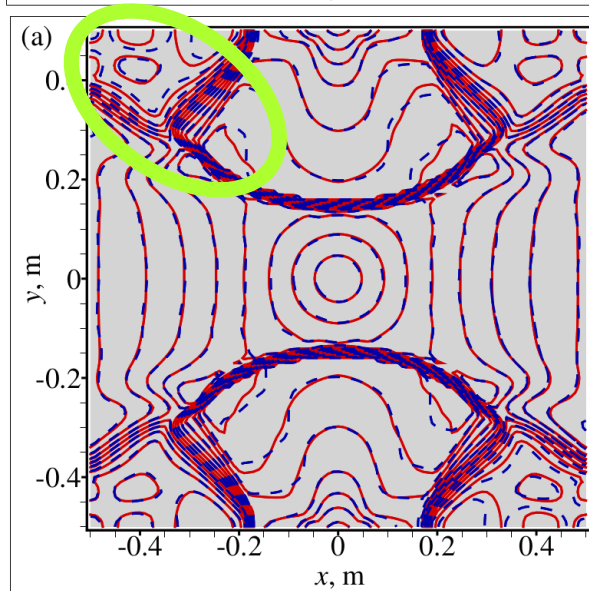
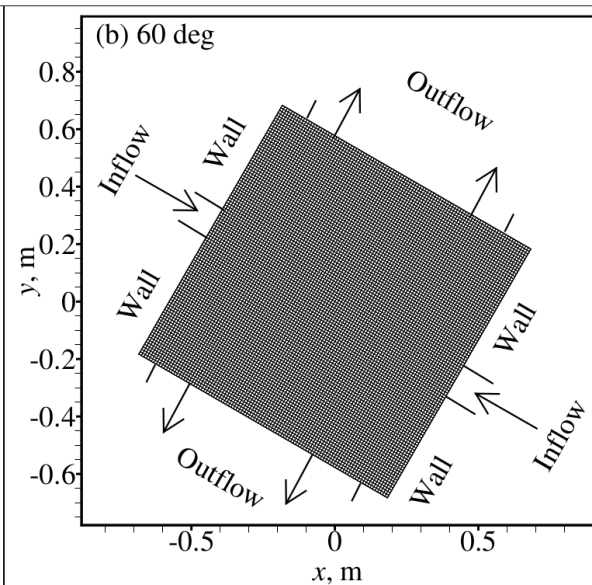
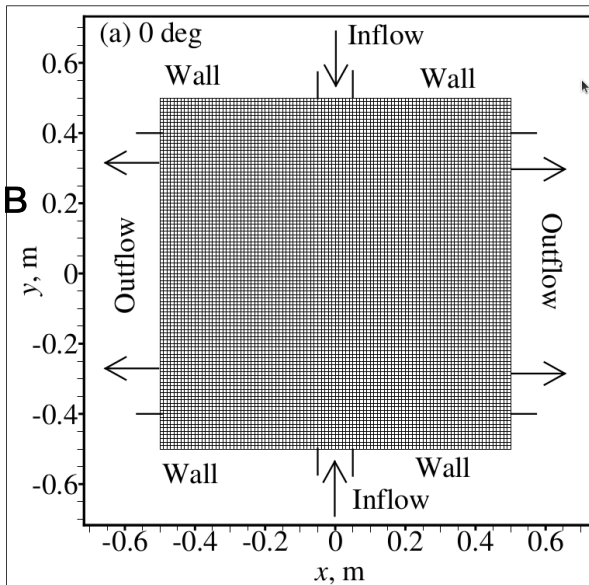
неупорядоченный набор групп уравнений

Эквивалентный  
результат

```
prototype->add_items("Q", "RhU", "GAS INVISC VISC");
prototype->add_items("Q", "RhU", "VISC");
prototype->add_items("Q", "RhU", "GAS");
prototype->add_items("Q", "RhU", "INVISC");
```

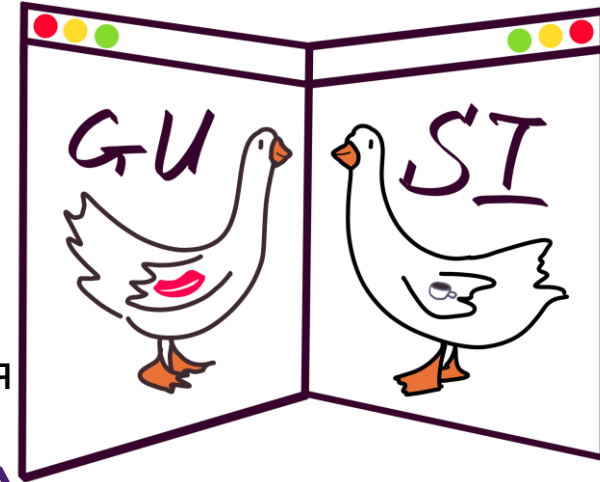
# MUSCL в $(U_n, U_t)$

- Проверили зависимость результатов MUSCL реконструкции от системы координат
- Оказалось есть зависимость, хотя ее не должно быть
- Но если использовать  $(U_n, U_t)$  вместо  $(U_x, U_y)$ , то всё становится хорошо.
- Демонстрационный тест: квадрат, в который из центров сторон втекают струи. При вращении картина течения меняться не должна



# Графический интерфейс в рамках платформы SALOME

- С появлением неструктурированной ветки открылся путь к полноценному графическому интерфейсу, с созданием геометрической модели, генерацией сетки, заданием ГУ и пр.
- **G**raphical/**U**ser/**S**alome/**I**nterface — поэтому внутреннее условное название GUSI
- Документация по созданию модулей для Salome обычно такая
- Частично разобрались как рисовать окошки, следующий шаг — сделать двустороннюю связку с расчетными кодами (делаем параллельно SMILE++ и HyCFS-R)
- Концепцию самих окошек (что, как, в каком порядке) тоже надо будет дорабатывать



**Installation instructions**

**NOT UP TO DATE AND INCOMPLETE  
DOCUMENT BUT CAN BE HELPFUL**



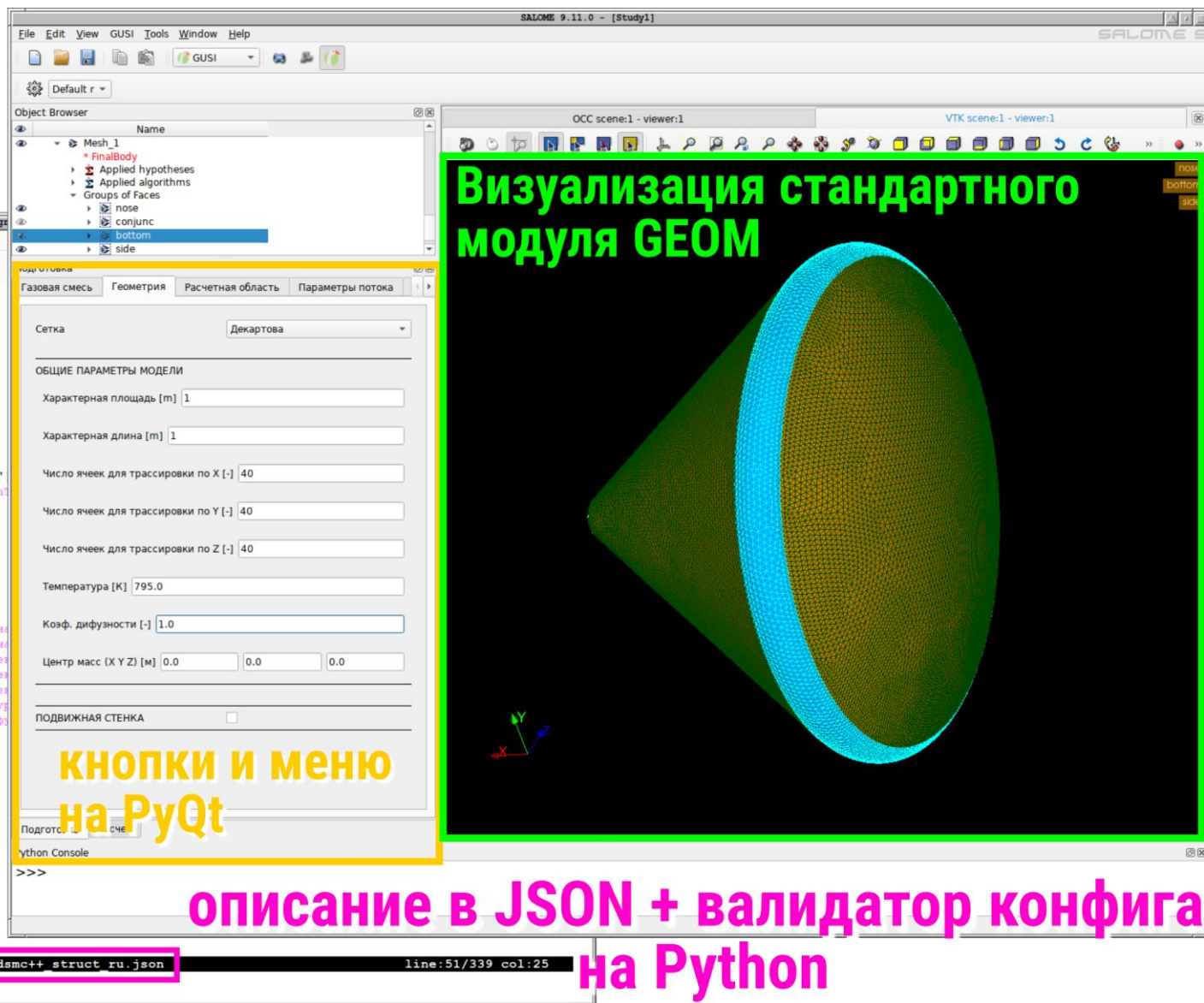
# Примерный вид (для SMILE++)

```
damc++ struct_ru.json + (~/Prog...
Файл Правка Инструменты Синтаксис Буферы Окно cprcomplete

"name": "Геометрия",
"action": "PreprocessingSetting",
"widgets":
{
  {
    "objectType": "list",
    "varType": "str",
    "name": "Сетка",
    "varName": "switch_geom",
    "items":
    {
      {"name": "Декартова", "value": "MeshRect"},
      {"name": "Тетраэдральная", "value": "MeshTetra"},
    },
    "defaultValue": "MeshRect"
  },
  "objectType": "group",
  "name": "ОБЩИЕ ПАРАМЕТРЫ МОДЕЛИ",
  "widgets":
  {
    {"objectType": "real", "name": "Характерная площадь [м]", "value": 1},
    {"objectType": "real", "name": "Характерная длина [м]", "value": 1},
    {"objectType": "int", "name": "Число ячеек для трассировки по X [-]", "value": 40},
    {"objectType": "int", "name": "Число ячеек для трассировки по Y [-]", "value": 40},
    {"objectType": "int", "name": "Число ячеек для трассировки по Z [-]", "value": 40},
    {"objectType": "real", "name": "Температура [К]", "value": 795.0},
    {"objectType": "real", "name": "Коэф. диффузности [-]", "value": 1.0},
    {"objectType": "real", "name": "Центр масс (X Y Z) [м]", "value": [0.0, 0.0, 0.0]},
  },
  "objectType": "group",
  "name": "ПОДВИЖНАЯ СТЕНКА",
  "isOptionalAndDefaultValue": false,
  "widgets":
  {
    {"objectType": "checkbox", "name": "ПОДВИЖНАЯ СТЕНКА", "value": false}
  }
}
}
}

"objectType": "group",
"name": "ПОДВИЖНАЯ СТЕНКА",
"isOptionalAndDefaultValue": false,
"widgets":
{
  {"objectType": "checkbox", "name": "ПОДВИЖНАЯ СТЕНКА", "value": false}
}
}
}

[1] ~/Programs/GUSI/GUSI_SRC/src/GUSI_CORE/Structure/dsmc++_struct_ru.json line:51/339 col:25
E73: Стек меток пустой
```



**Визуализация стандартного модуля GEOM**

**КНОПКИ И МЕНЮ на PyQt**

**описание в JSON + валидатор конфига на Python**

The image displays the SALOME 9.11.0 software interface. On the left, the 'Object Browser' shows a tree structure for a mesh object named 'Mesh 1', including sub-objects like 'FinalBody', 'Applied hypotheses', 'Applied algorithms', and 'Groups of Faces' (nose, conjunc, bottom, side). The main window shows a 3D visualization of a cone with a blue mesh surface. Below the 3D view is a configuration panel with tabs for 'Газовая смесь', 'Геометрия', 'Расчетная область', and 'Параметры потока'. The 'Геометрия' tab is active, showing a 'Сетка' (Mesh) dropdown set to 'Декартова'. Below this are various input fields for model parameters: 'Характерная площадь [м]' (1), 'Характерная длина [м]' (1), 'Число ячеек для трассировки по X [-]' (40), 'Число ячеек для трассировки по Y [-]' (40), 'Число ячеек для трассировки по Z [-]' (40), 'Температура [К]' (795.0), 'Коэф. диффузности [-]' (1.0), and 'Центр масс (X Y Z) [м]' (0.0, 0.0, 0.0). There is also a 'ПОДВИЖНАЯ СТЕНКА' (Movable Wall) checkbox which is unchecked. At the bottom, a Python console shows a prompt '>>>'. A terminal window at the very bottom shows the file path and line/col information.

# Текущие трудности и проблемы

- Неявное интегрирование по времени. Пока оно используется очень ограниченно.
- Производительность: как всегда не хватает скорости, но хочется оставить читабельный код, без микроуровневой оптимизации. Низкая эффективность OMP кода, если не использовать GPU
  - Здесь же ускорение учета теплового излучения газа (пока это на порядки медленнее всего остального вместе взятого: газовой динамики, химии и диффузии)
- Синхронизация функциональности структурированного и неструктурированного кода либо без использования наследования, либо с ним — но чтобы работала CUDA
- Совместная разработка — стандартизация стиля программирования. Сейчас многие вещи я вливаю в ветку лично, попутно корректируя. Нужно как-то сформулировать общие правила и условности

# Планы, планы, планы

## ◆ Солвер

- ◆ Доделать неструктурированный код
- ◆ Неявное интегрирование: собрать всё в единую схему для любой модели газа
- ◆ Пристеночные функции для RANS
- ◆ Неинерциальная система координат
- ◆ Адаптивная сетка (как у R. Deiterding)

## ◆ Пре/со/пост-процессинг

- ◆ Утилиты для импорта/экспорта сетки
- ◆ Конвертирование между структурированным и неструктурированным форматом входных файлов

## ◆ Модели

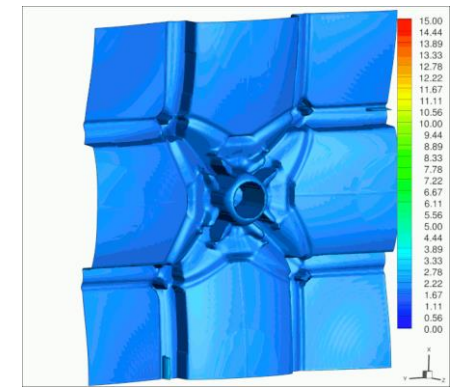
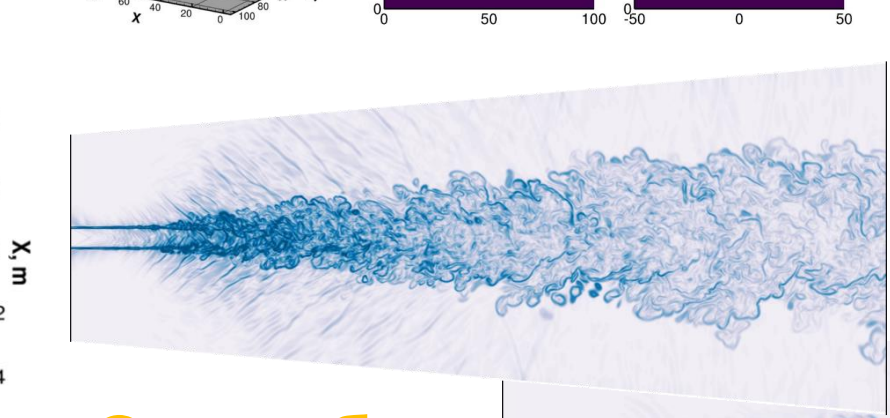
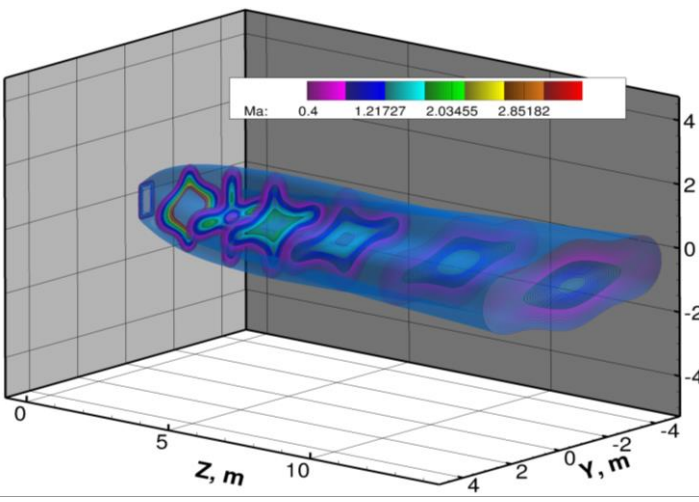
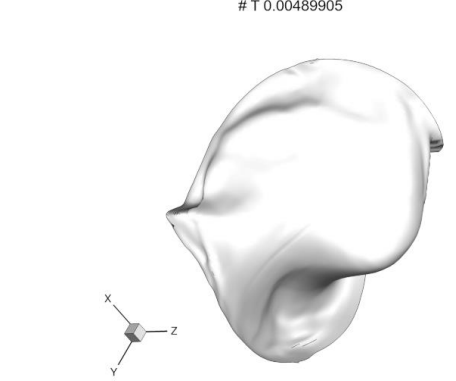
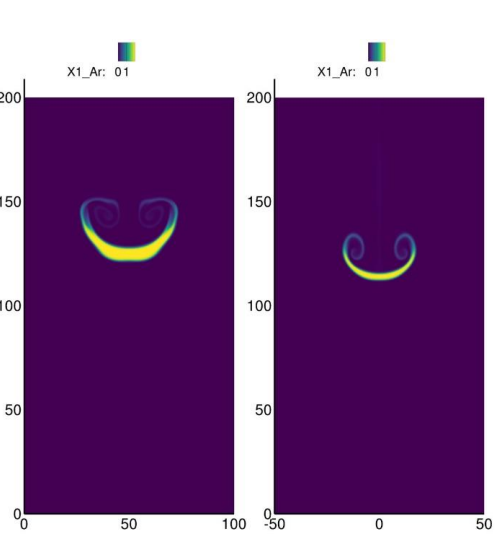
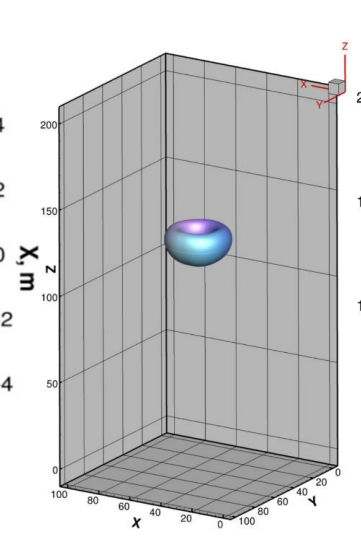
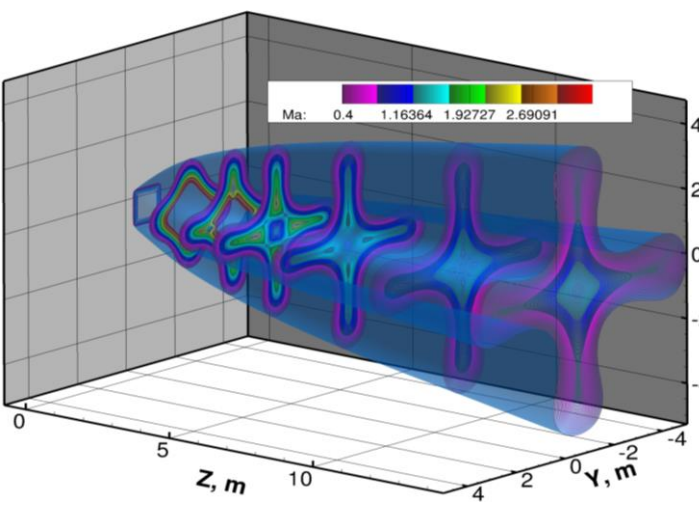
- ◆ Menter SST для турбулентности
- ◆ Модель перехода к турбулентности в SA
- ◆ Лагранжев подход
- ◆ Каталитическая стенка «как в SMILE++»

## ◆ Рефакторинг «ядра»

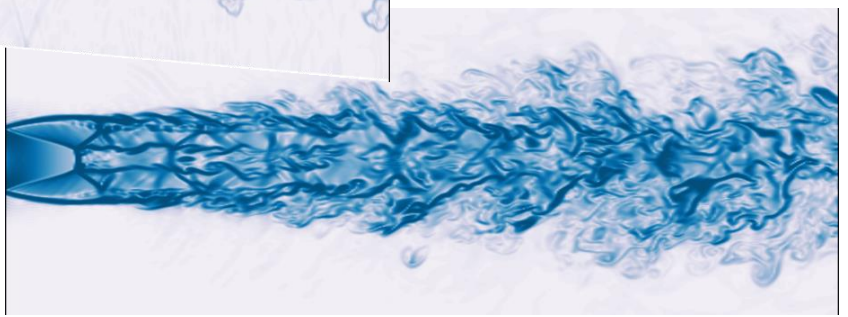
- ◆ Основной контейнер хочется сделать шаблонным классом: хранить int и bool для удобства

## ◆ Прочее

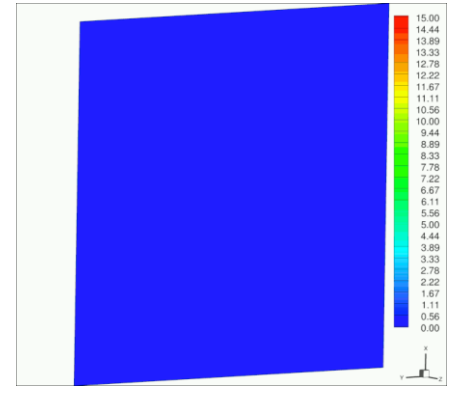
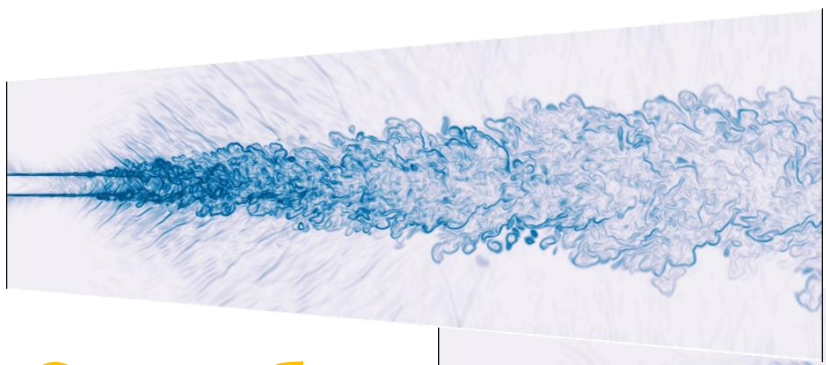
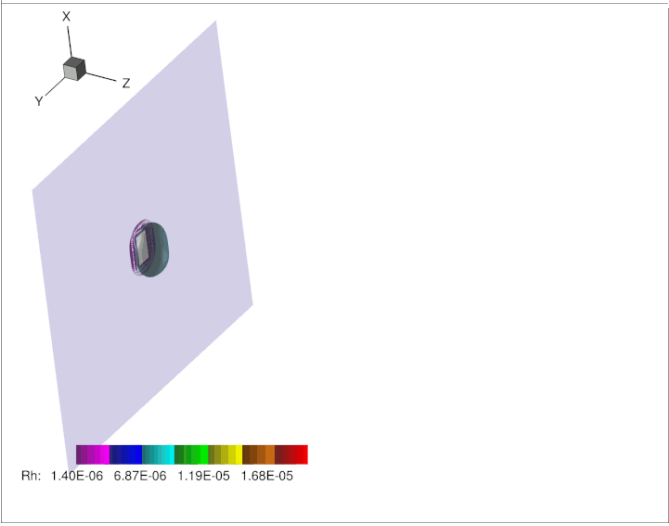
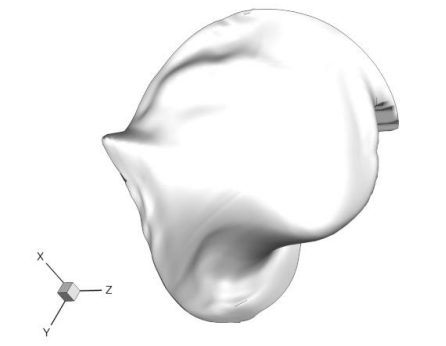
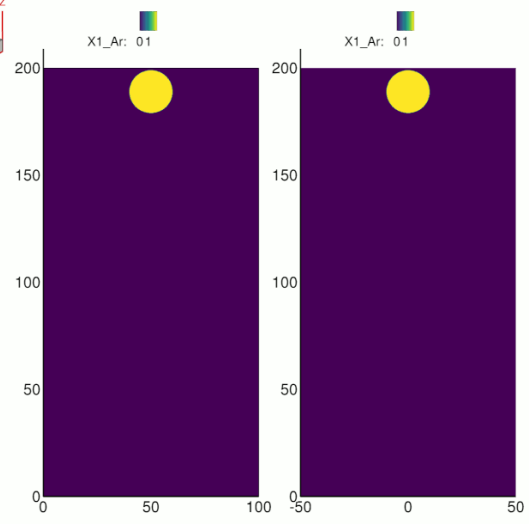
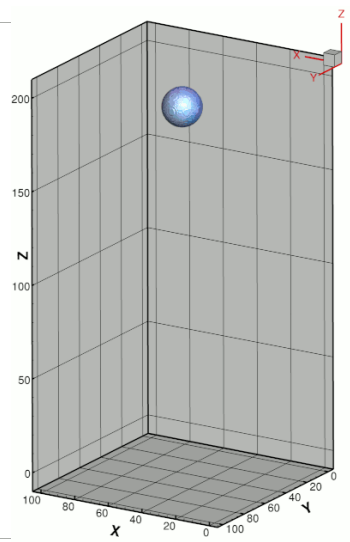
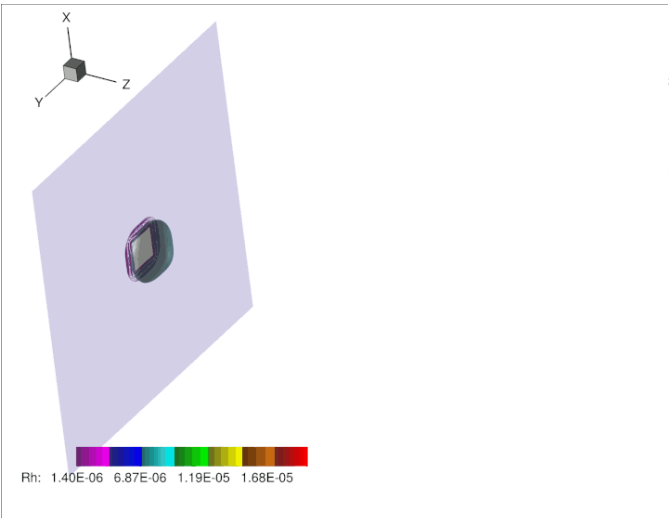
- ◆ Выделить излучение газа в модуль, общий с SMILE++
- ◆ Юнит-тесты на базе **cxxtest** библиотеки
- ◆ Графический интерфейс (что-то помимо Salome?)



Спасибо  
за внимание!







Спасибо  
за внимание!

