

ОТЕЧЕСТВЕННЫЕ CFD КОДЫ – 2025
Москва, 29-30 ноября 2025 г.



В НОВОМ СВЕТЕ

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

*Институт теоретической и прикладной механики им. С.А. Христиановича СО РАН,
Лаборатория вычислительной аэродинамики*

Общая информация

Раньше **HyCFS-R**, теперь **SUNSHyNE** :
Structured / **U**nstructured **N**avier-**S**tokes
Solver / **H**ybrid / **N**on-**E**quilibrium

Версия: SVN r1883 (год назад r885)

Расчетный код для моделирования
сжимаемых течений ламинарных и
турбулентных течений, в т.ч. с химической
и термической неравновесностью.

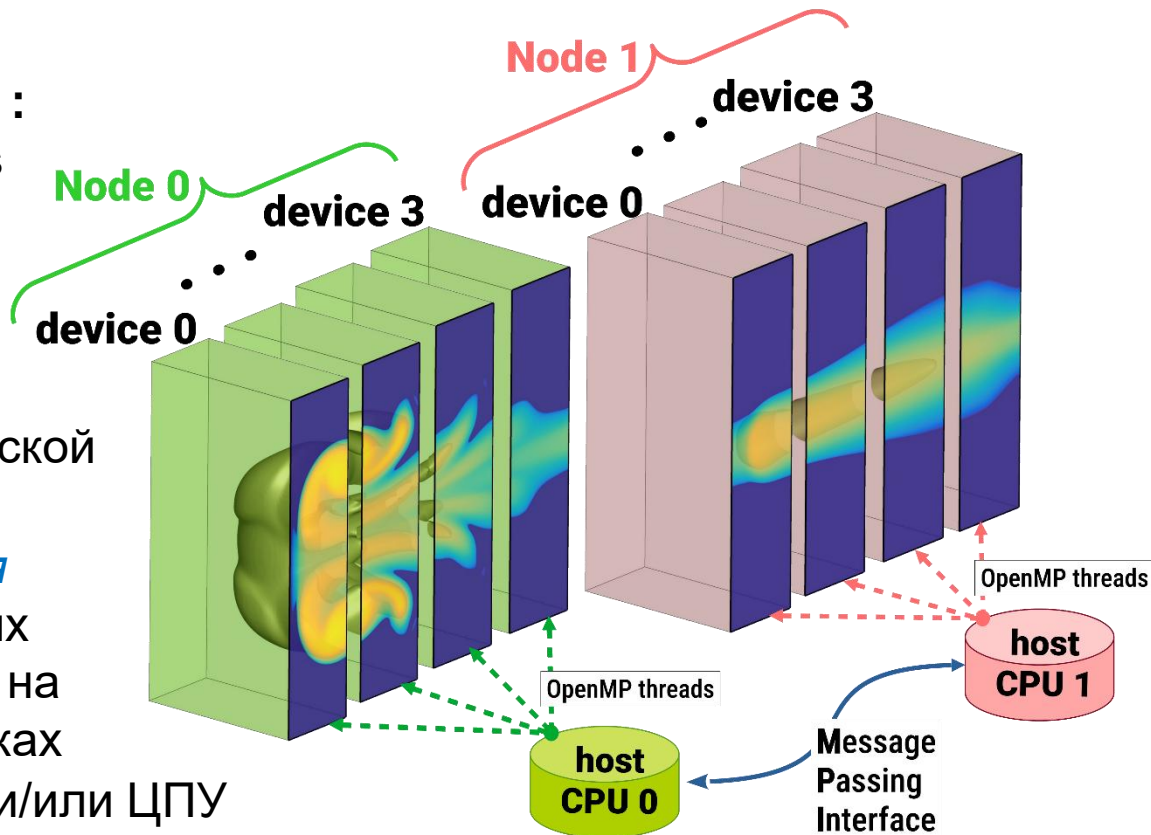
Расчетная сетка *структурированная*
многоблочная, в общих криволинейных
координатах и *неструктурированная* на
произвольных выпуклых многогранниках

Вычислительные устройства: ГПУ и/или ЦПУ

Параллелизация: многоуровневая CUDA/OpenMP/MPI

Операционная система: *Linux (можно подумать о версии под Windows, наверное)*

Графический интерфейс в тестовом режиме



Физические модели

- **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 (CO_2 с равновесием между колебательными модами)
- **Скорость химических реакций:** закон Аррениуса (в т.ч. для турбулентных течений), однотемпературные модели с зависимостью от давления (Petersen-Hansen), двухтемпературные модели (Парка, Кузнецова, Мачерета-Фридмана, Тринора-Маррона, β -модель, модель для 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 бародиффузии
- **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»

Численные методы и алгоритмы

- **Конвективная часть:** реконструкция для структурированной **MUSCL1-4**, для неструктурированной псевдо-одномерная реконструкция **1-2** порядка. Римановские солверы **HLL**, **HLLC**, **Roe**, **Rusanov**, **AUSM-Van Leer**, семейство **AUSM** (-UP, -UP2, P+, PW+, MP), гибридные **HLL/HLLC**, несколько вариантов **Steger-Warming** – всего **26 ОПЦИЙ**
- **Интегрирование по времени:** явные схемы **RKTVD1-3** / **RKG4**, полунеявная **ASIRK2c**, неявная параллельная **DPLUR**, локальный шаг по времени
- **Граничные условия:** несколько рядов фиктивных ячеек ставятся на консервативные переменные, но есть возможность явно задавать потоки через граничные грани.
- **Варианты ГУ:** сверхзвуковые вход¹ и выход², pressure inlet/outlet^{3,4}, условия в дальнем поле (Riemann far-field)⁵, условия непротекания⁶, изотермическая стенка с прилипанием⁷, в том числе каталитическая с заданным составом⁸. Специальные условия для DNS: наложение возмущений в виде белого гауссового шума⁹, в виде собственных функций¹⁰, периодические тепловые возмущения¹¹

Основные разработчики кода

- ◆ А.Н. Кудрявцев — численные схемы и методы
- ◆ Г.В. Шоев — физико-химические модели, в т.ч. неравновесность и 2Т-химия, модели турбулентности, римановские солверы, ГУ, полиэдральная сетка
- ◆ С.П. Борисов — однотемпературная химия, газовая детонация, неявные схемы
- ◆ Т.Ю. Шкредов — излучение, модели диффузии, ионизация, AUSM солверы, ГУ
- ◆ Д.П. Полевщиков — реализация многоблочности, утилиты для построения и анализа качества сеток
- ◆ А.В. Кашковский — общая архитектура программы, многоплатформенность, низкоуровневая оптимизация, отладка
- ◆ А.С. Литвинцев — GUI
- ◆ Т.А. Полянский — GUI, сетка, med/HDF утилиты, Лагранжевы частицы
- ◆ **А.А. Шершнёв — кодирование, интеграция частей в единую систему, всё остальное**

Со-разработчики / консультанты / пользователи / тестеры

- ◆ П.В. Ващенко — ConfigReader, GUI
- ◆ А.В. Зайцев — консультации по RANS
- ◆ Ю.В. Кратова — многофазность
- ◆ Д.В. Хотяновский — DNS расчеты
- ◆ А.И. Кутепова — DNS расчеты

Опции и возможности

■ Сопровождение эксперимента

- ❖ «Датчики» для записи параметров в точке по времени
- ❖ Система мониторинга параметров в расчете для контроля сходимости и прочего по АДХ, балансу массы и т.д.

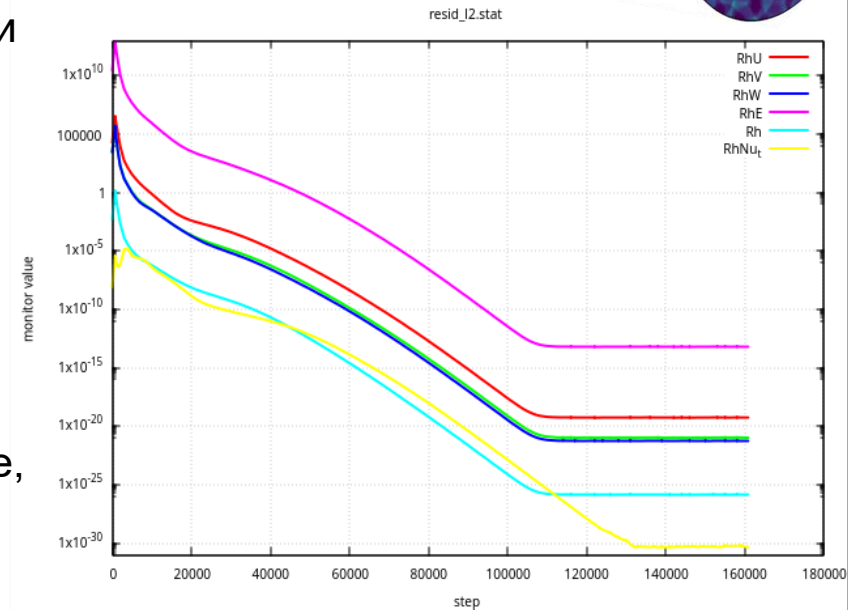
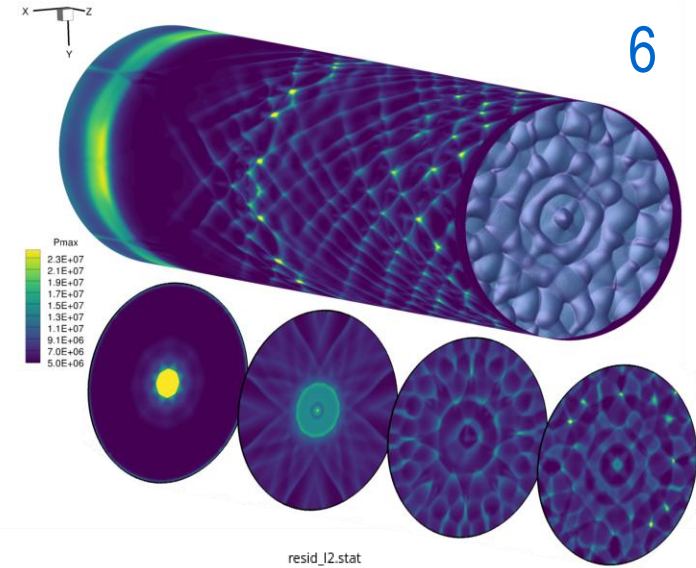
■ Фундаментальные исследования (устойчивость, ЛТП и т.д.)

- ❖ Накопление средних полей для DNS и полей истории максимального значения величины
- ❖ Отслеживание положения и формы фронта УВ/ДВ

■ Система командных файлов, для действий в процессе расчета: запись данных, корректная остановка счета

■ Инструмент сравнения файлов входных данных

■ Встроенные утилиты для сетки и полей: масштабирование, вырезание подобласти, разбиение-объединение блоков, умножение выбранных переменных в полях

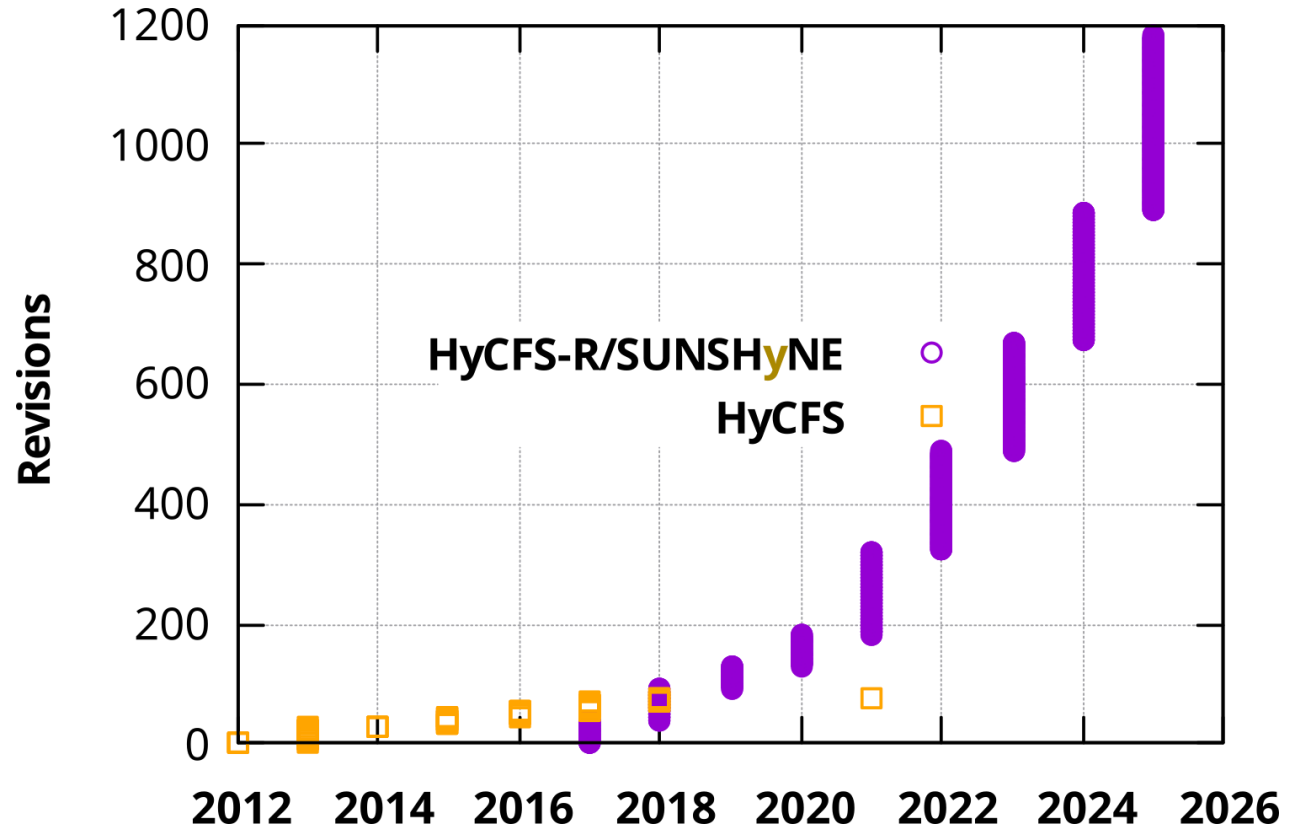


НОВОЕ ЗА 2025 ГОД

- ❖ Примерно +300 ревизий и +20 тысяч строк кода
- ❖ Но конкретные большие достижения как-то тяжело найти. Возможно кодовая база стала чуть более зрелой
- ❖ Из моделей - сделали и почти отверифицировали **kw-SST standard**
- ❖ Много работали над схемами интегрирования по времени:
 - перенесли на полиэдральную сетку DPLUR, в том числе для SA
 - сделали Full-matrix DPLUR
 - сделали схему с двойными шагами по времени (dual-time stepping)
- ❖ Доделывали юзер-экспириенс. Проведение многопараметрических расчетов заставило:
 - Напихать safeguard'ов, flow_limit'ов и прочих подпорок
 - Поработать над разбиением на партии
 - В целом исправить кучу мелочей: более активное использование бинарных файлов, например.

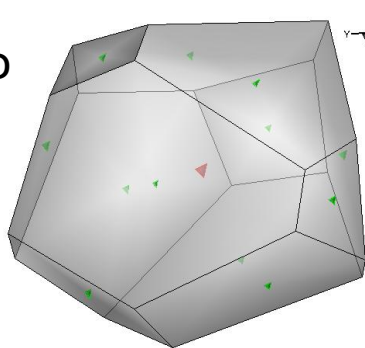
Немного статистики из системы контроля версия

- Эволюция исследовательского кода в отчуждаемый продукт
- Стали уменьшать размер изменений, чтобы быстрее искать, где сломалось
- Пришлось ввести юнит-тесты и автоматическое регрессионное тестирование (об этом чуть позже)



Средства отладки и контроля

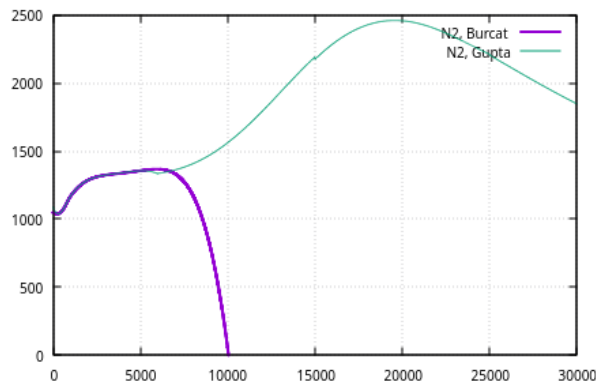
- Метод пристального взгляда уже мало помогает в идентификации источника ошибки
- Отдельные сборки исполняемых файлов с опцией **-g** для ЦПУ и ГПУ с дополнительными проверками выхода за границу массива, наличие NaN и возможностью смотреть в отладчике
- После каждой вычислительной функции есть точка «инспекции»: в ней можно вывести в Tесplot поле затронутого функцией контейнера, либо напечатать содержимое всех контейнеров в наборе точек
- Выводы ячеек по выбору в виде отдельного Tесplot файла
- Вывод полиномов в виде графика



```
Terminal
File Edit View Search Terminal Help
59 int main(
60     int nargs,
61     char** args
62 ) {
63
64 #ifdef DEBUG_BUILD
65     feenableexcept(FE_DIVBYZERO | FE_INVALID | FE_OVERFLOW);
66 #endif
67
68 /*-----*
69  *
70  * Initialize MPI only from MPI environment
71  *
72  *-----*/
73
74 MyMPI mpi;
75 mpi.Init(nargs, args);
76
/home/Antony/Progs/GPU/NS_CUDA_MIX/Src/branches/unstable/Src/ns.cu
(gdb) p mpi
$1 = {nodes_total = 0, node = 0, mpi_on = 221,
      devices_per_node = 1070707664, fpstat = 0x3ff0000000000000,
      n_nei_blocks = 0, nseg2send = 0, gather_size = 0,
      gather_ranks = 0x3ff0000000000000, gather_buf = 0x0}
(gdb) |
```

```
int e2prim_slot = mem.timers.set_timer();
KERCALL(cuEintToPrim, mp.blocks, mp.threads, "cuEintToPrim",
mem.timers.update_timer_tag("e2prim", e2prim_slot);

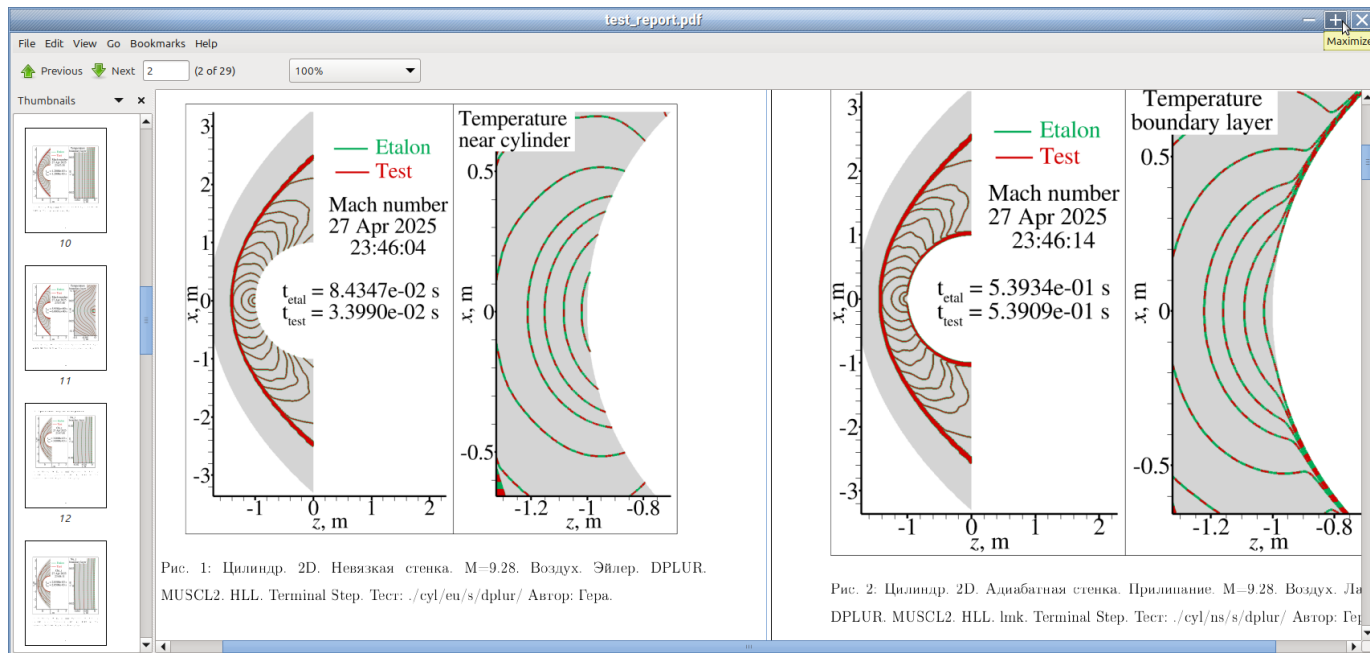
#ifdef DEBUG_BUILD
{ InspectContainer(mem, mem.fields.Q, "e2prim"); }
#endif
```



Автоматическое тестирование на регрессии

- ◆ Код довольно большой (~100 kloc), достаточно сильно связанный (предвычисления, многоэтапные процедуры и т.д.) – вынужденно пришлось сделать тестирование
- ◆ Сейчас это Unix shell скрипты, которые работают ретроактивно. Скрипт мониторит SVN репозиторий на наличие новой ревизии. Как только она появилась, запускается тестирование. Результаты собираются в виде PDF и высылаются на почту

- ◆ По-хорошему, нужно запускать проактивно, до коммита. Для этого надо сделать более модульным и универсальным. Поэтому сейчас думаем переписать на **Python** или **C++**
- ◆ Для такого тестирования, по идее, необходим достаточно стабильный формат файлов входных данных. Что, конечно, не всегда так просто

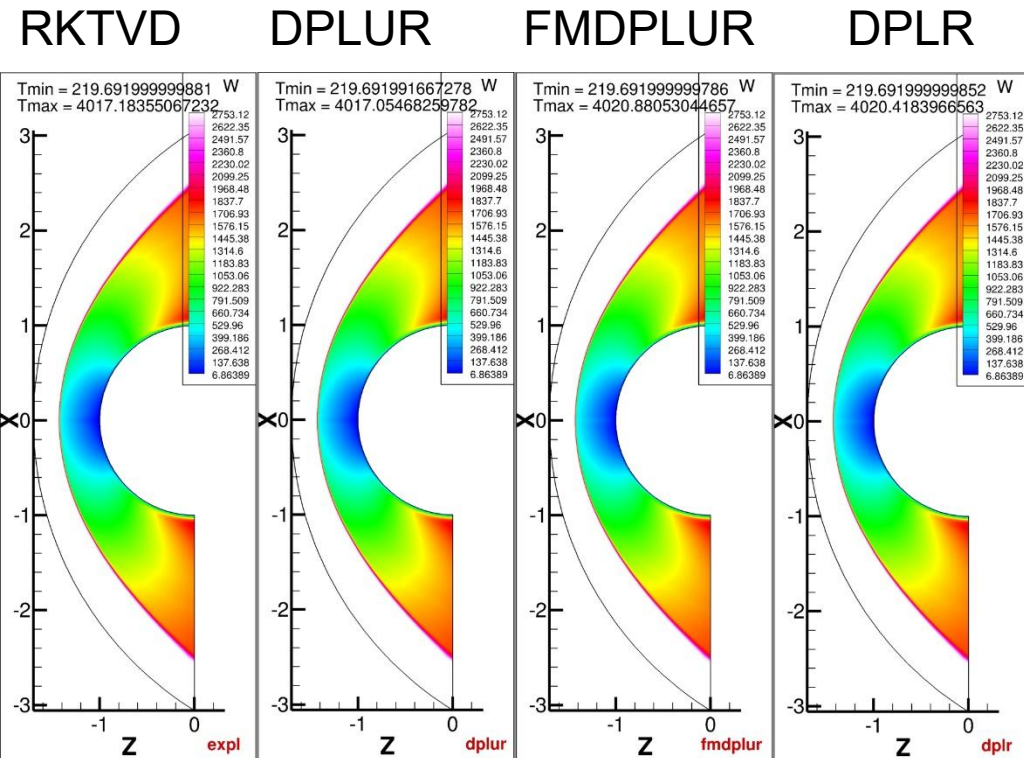


Интегрирование по времени. DPLUR/DPLR

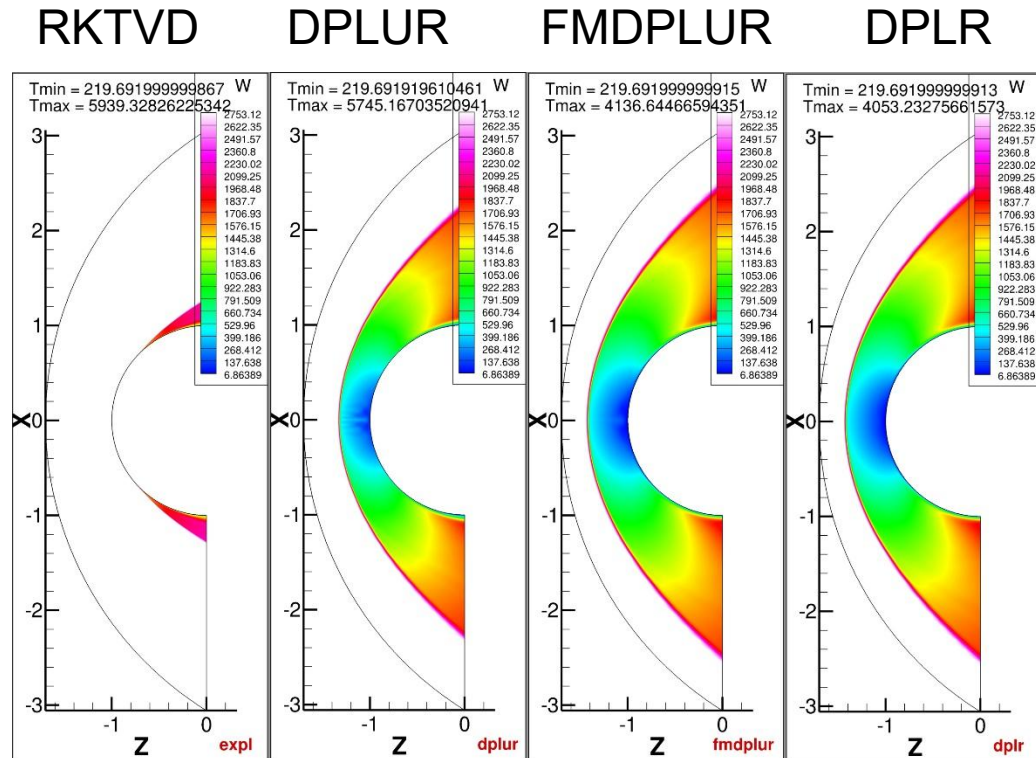
- ❖ Традиционные неявные методы (LU-SGS) используют последовательные проходы по сетке (Гаусс-Зейдель)
- ❖ Это создаёт сильные зависимости по данным и плохо параллелится
- ❖ Вместо проходов предлагается использовать точечные шаги релаксации:

$$\delta U_{i,j}^{(k)} = D^{-1} \left[R_{i,j} + \sum_{\text{соседи}} C \cdot \delta U_{\text{сосед}}^{(k-1)} \right]$$

- ❖ **DPLUR** (Data-Parallel Lower-Upper Relaxation):
 - Базовая схема, простая реализация
 - Не требует обращения матриц
 - Упрощённые якобианы (Yoon-Jameson)
 - Турбулентность: SA с явными источниками
 - Плохо сходится при больших CAR
- ❖ **FMDPLUR** (Full Matrix DPLUR):
 - Использует **точные якобианы**
 - Требуется обращения матриц $N_{eqs} \times N_{eqs}$
 - Меньше подвержена влиянию CAR
 - Улучшенная устойчивость и сходимость
- ❖ **DPLR** (Data-Parallel Line Relaxation):
 - Использует **точные якобианы**
 - Наиболее точная, но сложная в реализации
 - Решает блочно-трёхдиагональные системы
 - Не боится высоких CAR
 - Выделенные направления для шагов релаксации



CAR = 9



CAR = 10000

Интегрирование по времени. Неструктурированная сетка

❖ DPLUR:

$$\delta U_{i,j}^{(k)} = D^{-1} \left[R_{i,j} + \sum_{\text{соседи}} C \cdot \delta U_{\text{сосед}}^{(k-1)} \right]$$

D^{-1} – диагональная матрица в ячейке

Использование предположение
Yoon, Jameson, AIAA Paper 87-0600

❖ FM-DPLUR (только для лам НС)

D^{-1} – **не**диагональная матрица на гранях

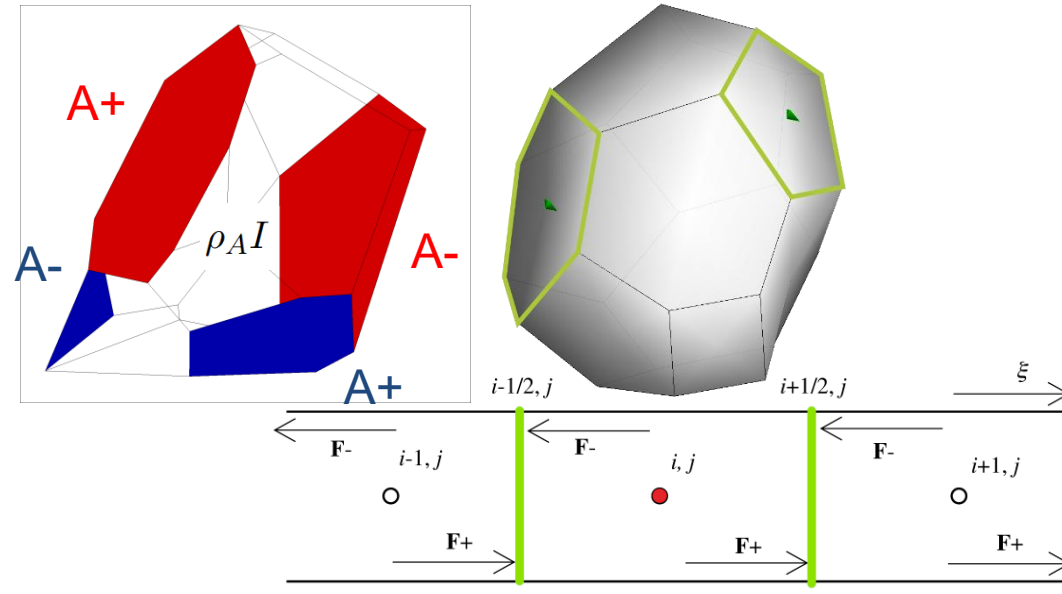
$$\delta U_{i,j}^{(k)} = D^{-1} \left[R_{i,j} + \sum_{\text{соседи}} C \cdot \delta U_{\text{сосед}}^{(k-1)} \right]$$

Приближение матриц Якоби потоков

$$A_+ = \frac{1}{2}(A + \rho_A I), \quad A_- = \frac{1}{2}(A - \rho_A I)$$

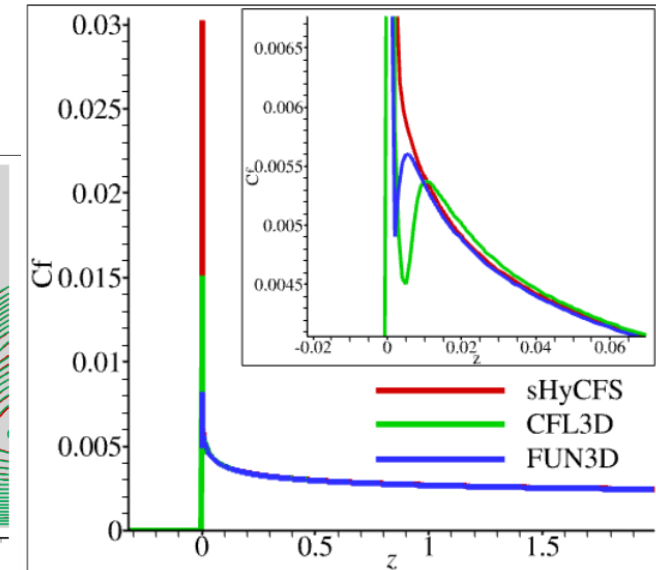
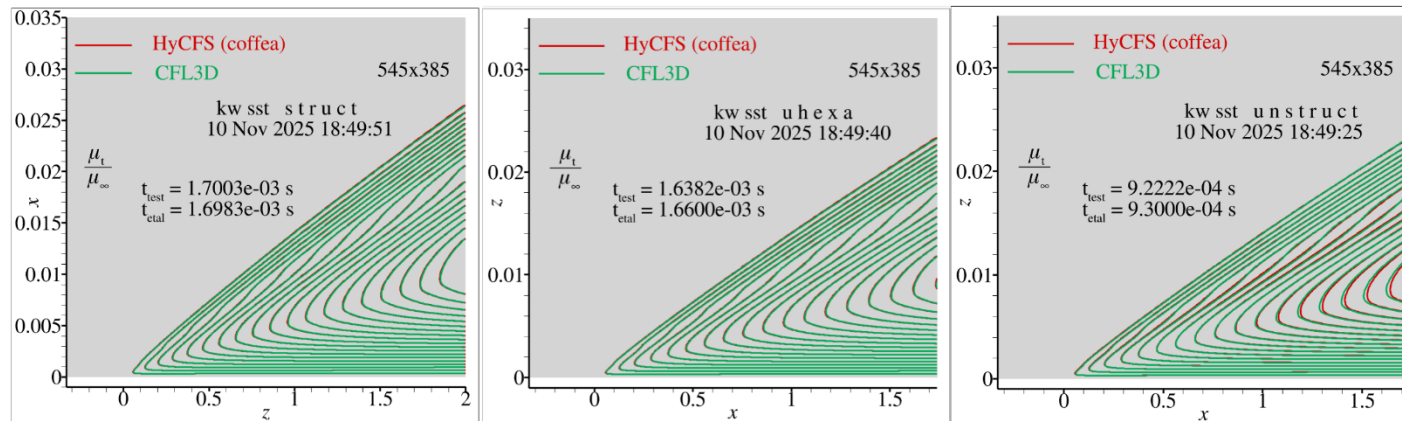
где ρ_A – спектральный радиус A

- Разность $A_+ - A_- = \rho_A I$ – диагональная матрица
- Упрощает неявную систему
- Обеспечивает диагональное преобладание



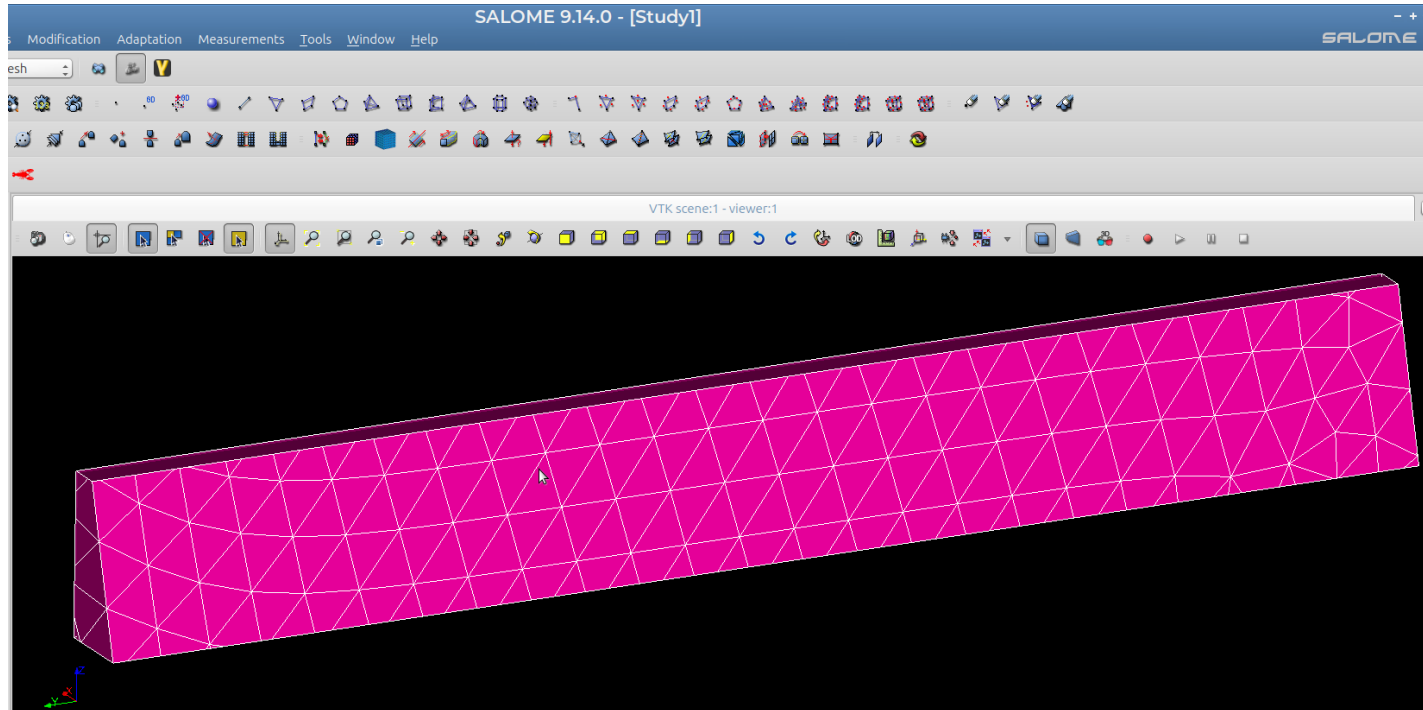
kw-SST

- ❖ Реализовали на структурированной и неструктурированной сетке **kw-SST**
- ❖ Использовали формулировки standard и VM
- ❖ Пока есть вопросы по коэффициенту трения на плоской пластине, в частности. Нет перегиба как в кодах НАСА
- ❖ Нужно разобраться подробнее
- ❖ γ -Re- θ начали, но пока не работает

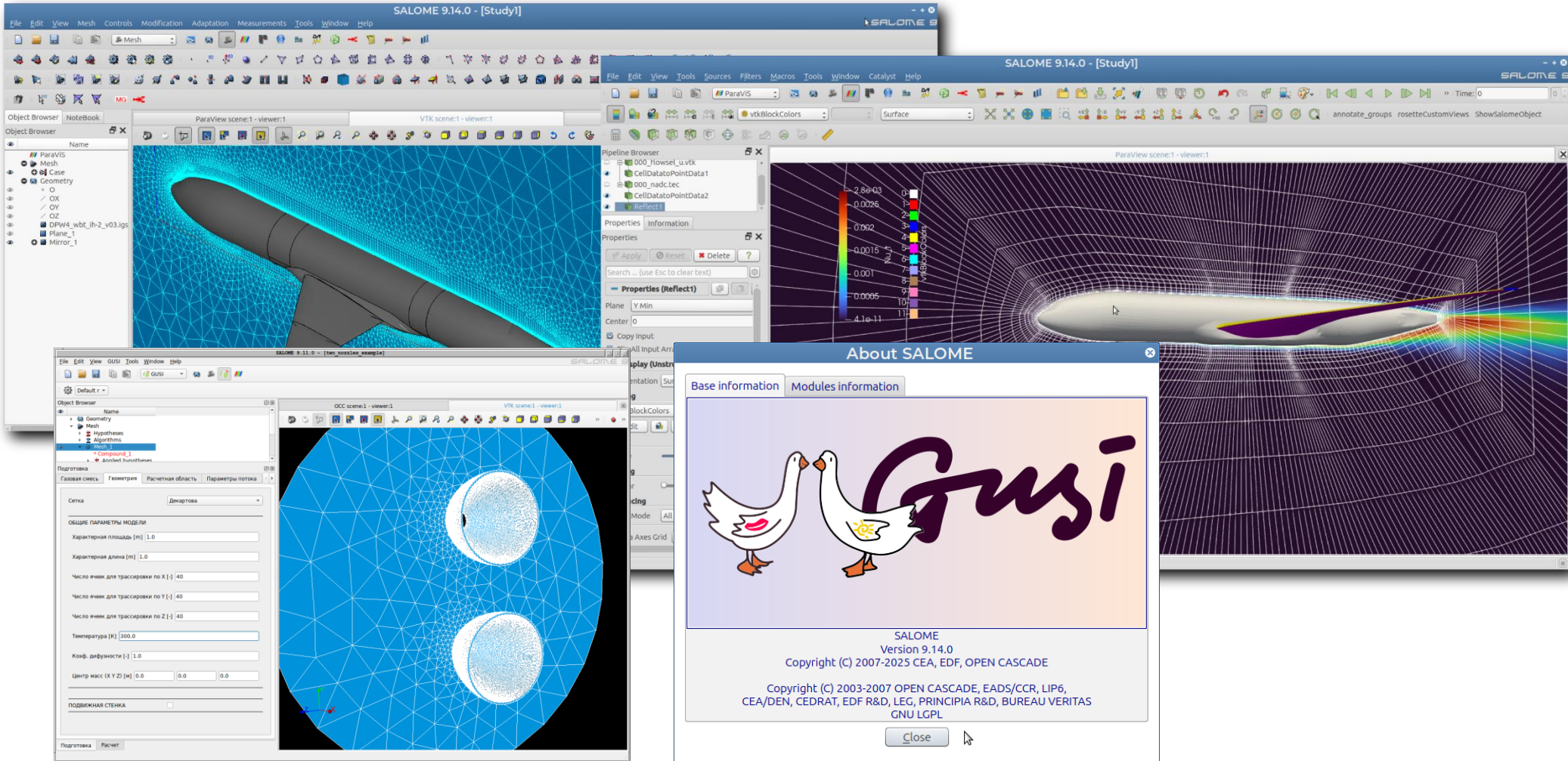


Сетка из FlowVision

- ❖ Попробовали использовать FlowVision в качестве генератора сеток
- ❖ Пока не очень получилось
- ❖ Выгружаемая сетка получается из кучи мелких граней: на рисунке одна ячейка
- ❖ Обратно в кубики мы это не смогли свернуть
- ❖ Надеемся на дальнейшее сотрудничество – генераторы сетки нам нужны



Графический интерфейс на платформе SALOME



#group Data

```
CudaBlocks (int)
CudaThreads (int)
```

```
CFL (double)
CFLcp (string)
```

```
Tfin (double)
Tout (double)
tout_cp (string)
Tsample (double)
iPrint (int)
Nwrite (int)
```

```
iAdv (int)
iSpatial (int)
iMUSCL (int)
iRS (int)
iWENO (int)
iUVRc (int)
```

```
iVisc (int)
iTherm (int)
iChem (int)
iTurb (int)
iVT (int)
iVV (int)
iDiffmod (int)
iDiffus (int)
iVibDiffusion (int)
iVibThermal (int)
```

```
iDP (int)
iDP_chem (int)
iDP_heat (int)
iDP_spa (int)
```

```
iForce (int)
grav_g (double)
iPorous (int)
iRadiation (int)
iTeq (int)
```

```
dt_forced (double)
use_local_timestep (int)
wall_flux_correction (int)
correct_yi (bool)
allow_neg_ghost temp (bool)
save_scheme (int)
save_scheme_type (int)
probe_step
muscl_unt_frame (int)
terminal_step (int)
```

#group Mesh

```
Nx (int)
Ny (int)
Nz (int)
facet_min_area (double)
iDim (int)
iaxis (int)
```

#group Umesh

```
NCells (int)
NCellsTotal (int)
NFaces (int)
NNeighbors (int)
NShare (int)
NSurfs (int)
NVerts (int)
UHEXA (bool)
facet_min_area (double)
forced_uell_type (int)
iDim (int)
iaxis (int)
ncell to face (int)
ncell to vert (int)
ncells (int)
nface to cell (int)
nface to vert (int)
nfaces (int)
nsurfs (int)
nverts (int)
```

#group BC

```
nsx (int)
nsy (int)
nsz (int)
xstart (int)
ystart (int)
zstart (int)
lev (int)
```

```
dir (int)
axis (int)
type (int)
cname (string)
```

```
mbb_id (int)
own_block (int)
nei_block (int)
```

```
body_tag (string)
```

```
_eigfuni (vector<double>)
_eigfunr (vector<double>)
_p_profile (vector<double>)
_profile (vector<double>)
```

```
is_rad.wall (bool)
is_slip.wall (bool)
```

#group UBC

```
type (int)
cname (string)
grouplen (int)
groupname (string)
```

```
mbb_id (int)
own_block (int)
nei_block (int)
```

```
body_tag (string)
```

```
_eigfuni (vector<double>)
_eigfunr (vector<double>)
_p_profile (vector<double>)
_profile (vector<double>)
```

```
is_rad.wall (bool)
is_slip.wall (bool)
```

#group Condition

```
name (string)
is_ref_cond (int)
```

```
Tw (double)
_QPin (vector<double>)
_Qin (vector<double>)
_TV (vector<double>)
_Xin (vector<double>)
_Yin (vector<double>)
_Amp (vector<double>)
```

```
wall_lin_velocity (Vector3)
wall_oscil_amp (Vector3)
wall_oscil_freq (Vector3)
wall_oscil_phase (Vector3)
wall_rot_amp (Vector3)
wall_rot_freq (Vector3)
wall_rot_phase (Vector3)
wall_rot_point (Vector3)

thpause time (double)
thspike phase (double)
thspike temp (double)
thspike_time (double)
```

```
P0 (double)
P_out_sub (double)
Pout (double)
T0 (double)
```

```
disturb_freq (double)
disturb_wavevec (Vector3)
```

#group Element

```
name (string)
Wm (double)
diam (double)
```

```
Muref (double)
Nvibmod (int)
Omega (double)
Tref_mu (double)
Tsuth (double)
```

```
DOF (double)
CHS_coef (vector<double>)
CHS_coef_Gupta (vector<double>)
Cp_tab (vector<double>)
Temp_range (vector<double>)
Temp_range_Gupta (vector<double>)
```

```
D0TM (double)
Ediss (double)
Gamma (double)
H0v (double)
Kapconst (double)
Muconst (double)
```

#group Reaction

```
equation (string)
fA (double)
fEa (double)
fb (double)
```

```
rA (double)
rEa (double)
rb (double)
```

```
ichem (int)
```

```
DisName (string)
LOW (vector<double>)
Losev_beta (double)
TBE (vector<double>)
TBEEnum (vector<int>)
TROE (vector<double>)
```

```
il (int)
ik_r (int)
in_sgs (bool)
m_A (double)
m_B (double)
park_s (double)
```

#group Postproc

```
save_plt (int)
save_tec (int)
save_vtk (int)
save_adc (int)
save_jtec (int)
```

```
aux_postproc (int)
aux_postproc_aver (int)
export_only_qc (int)
force_2d_export (int)
save_1st_ord_adc (bool)
save_aux_vars (int)
save_mlab (int)
save_xc (int)
```

#group ADC

```
xport_Q (string)
Export_QM (string)
Export_QPP (string)
Export_Qauxp (string)
Export_WQ (string)
Export_WQM (string)
Export_WQPP (string)
Export_WQauxp (string)
```

```
Lref (double)
Pref (double)
Rref (double)
Sref (double)
Tref (double)
Uref (double)
Xref (Vector3)
```

#group RANS

```
wall_function (int)
Pr_t (double)
limit_time_step (int)
first_order (int)
```

#group TurbSA

```
...
k_sa (double)
limit_Stilde (int)
limit_chi (int)
limit_fw (int)
limit_g (int)
...
```

#group TurbKWSTT

```
...
sigma_k1 (double)
sigma_w1 (double)
beta_star (double)
...
```

#group Porous

```
B (Vector3) :
Beta (Vector3) :
C (Vector3) :
K (Vector3) :
R (double) :
Xa (Vector3) :
Xb (Vector3) :
site_type_t (string)
xA (Vector3) :
xB (Vector3) :
x0 (Vector3) :
```

#group Line

```
axis (int) :
i (int)
id (int)
j (int)
k (int)
x (double)
y (double)
z (double)
```

#group Probe

```
axis (int)
i (int)
id (int)
j (int)
k (int)
x (double)
y (double)
z (double)
```

#group LLine

```
X0 (Vector3)
X1 (Vector3)
```

Расчетная область
Геометрия

Почему окошки для
нашего кода это сложно

Планы на будущее

Планы по развитию HC - x

Not Secure http://overleaf.itam.nsc.ru

90% Sign in

Menu Home Планы по развитию HC Share History Layout Chat

Code Editor Visual Editor Normal text B I ... Recompile 1 / 5 66%

```
1 \documentclass[dvipsnames,11pt]{scrartcl}
2 \usepackage{dejavu}
3 % Language setting
4 % Replace `english' with e.g. `spanish' to change the
  document language
5 \usepackage[russian]{babel}
6 \usepackage{BeamerColor}
7 % Set page size and margins
8 % Replace `letterpaper' with `a4paper' for UK/EU standard
  size
9 \usepackage[letterpaper,top=2cm,bottom=2cm,left=2cm,right=2
  cm,marginparwidth=1.75cm]{geometry}
10 \usepackage[normalem]{ulem}
11
12 % Useful packages
13
14 \usepackage{amsmath}
15 \usepackage{graphicx}
16 \usepackage[colorlinks=true, allcolors=blue]{hyperref}
17 \renewcommand{\labelenumii}{\theenumii}
18 \renewcommand{\theenumii}{\theenumi.\arabic{enumii}.}
```

5 страниц мелким шрифтом!

Планы по развитию HC

все-все-все

21 ноября 2025 г.

1. Платформа
 - 1.1. Доделать `Mpi_double` \Rightarrow `Mpi_char`
 - 1.2. Автоматическое тестирование на питоне
 - 1.3. Watchdog-и для зависимых от времени/номера итерации параметров
 - 1.4. Система агрегации данных по иерархии процессоров/блоков с сегментами и прочими суб-разбиениями
 - 1.5. Прямое чтение мед или CGNS файлов. Самостоятельное вычисление геометрических параметров сетки.
 - Реализовано сохранение `000_topology`
 - 1.6. Повсеместное использование `hdptr`, в том числе
 - `Q` (убрать `h0`, `gas_p` и прочего
 - В `BSegment` убрать `hXF/dXF` и т.д. Сделано в первом приближении.
 - 1.7. Бинарные файлы для топологии [+ сделано в первом приближении]
 - 1.8. Сделать новую версию мониторов (Monitoring)
 - 1.9. Переразбиение области из N в K партишенов
 - Реализовано разбиение с сохранением карты ячеек и обработка со сборкой единого файла течения `full_flow.dat.gz`
 - Реализована возможность автоматически сохранять/переносить инклюды при разбиении. Пока нет поддержки многоуровневых инклюдов.
- 1.10. Переработать схему бэкапов для Tweakfield: сейчас легко случайно сделать вложенное дублирование исходных файлов. Нужна более сложная логика и какие-то дополнительные средства работы с файлами изнутри кода.
- 1.11. Переключение (в компайл- или рантайме) функций для химических механизмов в CGC
- 1.12. Экспорт полидральной сетки в legacy VTK
- 1.13. Повторное чтение конфигов и обновление параметров в процессе расчета. (Настройка шага по времени, числа Куранта на лету.

Планы на будущее – над чем уже работаем

- ❖ Многофазные течения с дискретными частицами. *Пылевые бури на Марсе нас ждут*
- ❖ Ориентация на пользователей в рамках института (*приблизиться к коммерческим пакетам*):
 - Доделать графический интерфейс на базе платформы SALOME
 - Отработать схему распространения и установки кода
 - Аналог UDF функций – proof-of-concept есть, нужно дальше пробовать
 - Документация, включая пошаговые инструкции-тutorials с примерами некоторых задач
- ❖ Улучшение самого кода
 - Максимум абстракций: например, общая структура 1D линия для неявной схемы DPLR, излучения, численной схемы и др. **(это пока планы)**
 - в части встроенной документации, схем и комментариев для упрощения привлечения новых разработчиков – студентов, аспирантов и т.д. **(опция «скопируй кусок и сделай по аналогии» как выяснилось не работает даже с опытными программистами)**

Планы на будущее – до чего еще не добрались

- ❖ Несжимаемые течения
- ❖ Неинерциальная система (демпфирующие моменты)
- ❖ Каталитическая стенка «как в SMILE++»
- ❖ Излучение газа в виде модуля, общего с SMILE++
- ❖ Ну и проблемы с трудностями, которые надо решить.
О трудностях чуть ниже.

Текущие трудности и проблемы

- ❖ Salome и его библиотека medcoupling ヽ_(ツ)_/┐
 - Довольно неэкономичная по потреблению памяти: чтобы сконвертировать полиэдральную сетку на 27 млн. 64 Гб памяти может и не хватить
 - Библиотечная функция падает где-то (например, при вычислении барицентра) по причинам, которые невозможно узнать
 - Хочется – читать из CGNS минимальную информацию и довычислять всё необходимое самостоятельно. Но это требует большой работы
- ❖ В целом сеткопостроители – не хочется целиком полагаться на ICEM и Ansys Meshing, но по-другому пока не получается
- ❖ Механизмы хим. реакций в СГС и других системах. *«Сконвертировать нельзя оставить»*

Текущие трудности и проблемы

- ❖ Нормальное разделение на stable и unstable ветки. Пока либо просто устаревшая ревизия, либо bleeding edge, где что-то наверняка сломано. Требуется квалификация и размышлений – поэтому пока некогда
- ❖ Из-за CUDA нельзя нормально пользоваться механизмами наследования, виртуальными методами и прочими благами цивилизации. Это усложняет организацию кода.
- ❖ Сделать динамически линкуемый аналог UDF для ГПУ тоже нельзя, по всей видимости.

Но это пока только жаловаться и терпеть

УДК 533.6
DOI: 10.15372/PMTF202515704

ПРОГРАММНЫЙ КОМПЛЕКС SUNSHYNE ДЛЯ МОДЕЛИРОВАНИЯ ТЕЧЕНИЙ СЖИМАЕМОГО ГАЗА НА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ ГИБРИДНОЙ АРХИТЕКТУРЫ

А. А. Шершнев, А. Н. Кудрявцев, А. В. Кашковский,
Г. В. Шоев, С. П. Борисов, Т. Ю. Шкредов,
Д. П. Полевщиков, Д. В. Хотяновский, Ю. В. Кратова,
П. В. Ващенко, А. С. Литвинцев, Т. А. Полянский, Е. А. Бондарь

Институт теоретической и прикладной механики им. С. А. Христиановича СО РАН,
Новосибирск, Россия
E-mails: antony@itam.nsc.ru, alex@itam.nsc.ru, sasa@itam.nsc.ru, shoev@itam.nsc.ru,
borisov@itam.nsc.ru, shkredov@itam.nsc.ru, polevshchikov@itam.nsc.ru,
khotyanovsky@itam.nsc.ru, sci_itam@itam.nsc.ru, vashen@itam.nsc.ru,
litvintsev@itam.nsc.ru, polyansky@itam.nsc.ru, bond@itam.nsc.ru

Представлен программный комплекс SUNSHYNE, предназначенный для численного моделирования течений сжимаемого газа на современных многопроцессорных вычислительных системах, включающих высокопроизводительные графические ускорители. Комплекс обладает графическим интерфейсом пользователя и позволяет выполнять расчеты в областях со сложной геометрией для промышленных приложений за счет различных способов задания граничных условий, использования неструктурированных и блочных структурированных сеток, а также сопряжения с системами автоматизированного проектирования. Описываются реализованные в коде физические модели и численные методы, функциональные возможности пакета, приводятся примеры расчетов, выполненных с его помощью, включая расчеты неравновесных течений, прямое численное моделирование перехода к турбулентности и моделирование гетерогенной детонации.

Ключевые слова: уравнения Навье — Стокса, неравновесные и химически реагирующие течения, схемы сквозного счета, графические процессоры, пакеты прикладных программ

Введение. В последнее время численное моделирование представляет собой один из основных инструментов научного исследования. Важной областью, в которой численное моделирование применяется наиболее широко, является аэрогидродинамика. При решении

Разработка общей архитектуры программного комплекса и базовых физико-химических моделей выполнена в рамках государственного задания Института теоретической и прикладной механики СО РАН (№ 24021400040-4). Реализация и валидация моделей для учета термодинамической неравновесности проведена при финансовой поддержке Российского научного фонда (код проекта 25-79-30031). Численные эксперименты выполнены с использованием ресурсов центра коллективного пользования “Механика” (Институт теоретической и прикладной механики СО РАН).

© Шершнев А. А., Кудрявцев А. Н., Кашковский А. В., Шоев Г. В., Борисов С. П., Шкредов Т. Ю., Полевщиков Д. П., Хотяновский Д. В., Кратова Ю. В., Ващенко П. В., Литвинцев А. С., Полянский Т. А., Бондарь Е. А., 2025

УДК: 519.688

А. В. Кашковский, П. В. Ващенко, А. А. Шевырин, А. Н. Магачова, А. С. Литвинцев, Л. В. Ярков,
Т. А. Полянский, Д. П. Полевщиков, А. А. Шершнев, Г. А. Жукова, Е. А. Бондарь

СЕМЕЙСТВО ПРОГРАММНЫХ СИСТЕМ SMILE ДЛЯ ПРЯМОГО СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ ТЕЧЕНИЙ РАЗРЕЖЕННЫХ ГАЗОВ

Представлен обзор семейства отечественных программных систем SMILE, предназначенных для численного моделирования разреженных газовых течений методом прямого статистического моделирования Монте-Карло. Все программы, входящие в это семейство, являются продуктами общего назначения, используемыми для решения как фундаментальных, так и прикладных задач, в частности, для исследования высотной аэродинамики космических станций, спутников и спускаемых аппаратов. Системы семейства SMILE были созданы М. С. Ивановым и его учениками в ИТПМ СО РАН в годы, когда институтом руководил В. М. Фоми, и продолжают развиваться в настоящее время. В исторической перспективе представлены основные подходы и концепции, лежащие в основе этих программных продуктов. Описаны возможности и особенности каждой программной системы, входящей в семейство SMILE.

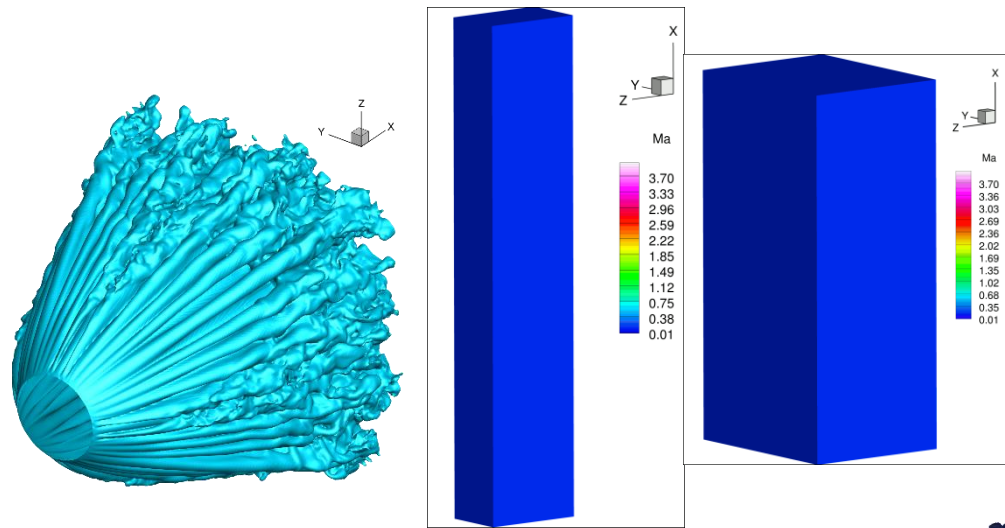
Ключевые слова: динамика разреженных газов, метод прямого статистического моделирования, методы Монте-Карло, высотная аэродинамика, пакеты прикладных программ.

Введение. При полете в верхних слоях атмосферы, в вакуумной технике и различных микроустройствах встречаются газовые течения, в которых средняя длина свободного пробега молекул сопоставима с размером изучаемого объекта. Такие течения называются разреженными. Для их теоретического описания не может быть использован подход, в рамках которого газ считается сплошной средой. В частности, такие модели как уравнения Навье — Стокса — Фурье дают существенную ошибку в определении параметров разреженных течений [1]. Для описания таких течений необходимо использовать кинетический подход, основанный на системе обобщенных уравнений Больцмана для смеси газов [2]. Отметим, что даже в самом простейшем случае одноатомных бесструктурных газов численное решение этой системы требует вычисления пятимерных интегралов для каждой точки фазового пространства, размерность которого превышает на три размерность задачи (т. е., например, для трехмерных задач размерность фазового пространства равна шести). По этой причине подход, основанный на прямом численном решении системы уравнений Больцмана, хотя и развивается активно [3–5], но пока еще не может быть использован в реальных промышленных приложениях. Основным методом численного расчета трехмерных многокомпонентных химически реагирующих разреженных течений в сложных геометриях является метод прямого статистического моделирования (ПСМ) [6, 7].

Метод ПСМ был предложен Г. Бёрдом в 1963 г. [8]. Традиционно он рассматривается как метод компьютерного моделирования эволюции ансамбля 10^5 – 10^{10} частиц, отображающих реальное газовое течение. Каждая из частиц представляет большое число (10^{13} – 10^{17}) реальных молекул. Непрерывная эволюция ансамбля частиц распадается на два независимых этапа на шаге по времени — свободномолекулярный перенос и столкновительная релаксация, которая моделируется независимо в каждой из ячеек расчетной области (при этом сталкиваться могут все частицы, находящиеся в одной ячейке, независимо от их положения). Процессы межмолекулярных столкновений, отражения частиц от поверхности и сброса частиц в расчетную область с границы моделируются с помощью компьютерной реализации значений случайных величин, имеющих заданные распределения, которые

Институт теоретической и прикладной механики имени С. А. Христиановича СО РАН. 630090, г. Новосибирск, ул. Институтская 4/1; э-почта: bond@itam.nsc.ru. Поступила 26.08.2025.

Научные картинки с благодарностями



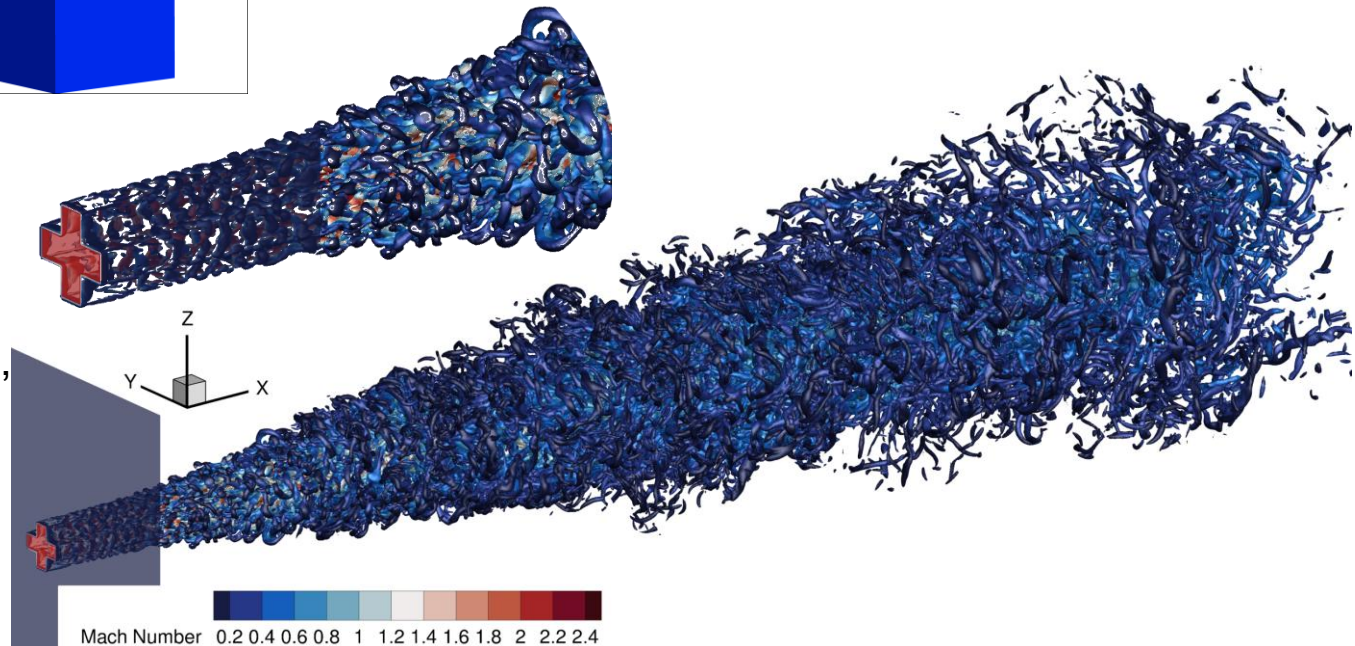
Грант РФФ 23-11-00258

Численное моделирование разреженных газовых струй на основе кинетического и континуального подходов

Грант РФФ 25-79-30031 (лаб. мирового уровня)

Неравновесные течения газа и плазмы в приложении к аэрокосмическим, экологическим и медицинским технологиям

- Вихри Гёртлера в круглой струе
- Струи из сопел различной формы, натекающие на преграды
- Ламинарно-турбулентный переход в струе из сопла крестообразного сечения





**Спасибо за
внимание!**

