



**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ)**

**Кафедра №204  
«Авиационная и  
космическая теплотехника»**



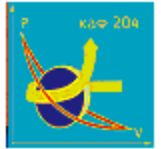
# **Программный комплекс Universe CFD: высокие скорости и сложная неравновесная кинетика**

А.М.Молчанов, д.т.н., профессор кафедры  
Д.В. Грибиненко, к.т.н., инженер кафедры  
Д.С. Янышев, к.т.н, доцент кафедры

Москва, 2023 г.



# Программный комплекс Universe CFD (краткая аннотация)



- Пакет разработан на базе кафедры «Авиационно-космическая теплотехника» МАИ
- **Основные возможности пакета:**
  - Расчет полей течений высокоскоростных потоков газа на основе решения уравнений механики сплошной среды.
  - Учет широкого спектра физико-химических процессов и явлений, свойственных высокоскоростным течениям (турбулентность, химические реакции, излучение, термическая неравновесность).
  - Расчет спектральных характеристик излучения потока газа на основе получаемого поля течения
  - Моделирование разреженных течений (в теоретическом пределе – в диапазоне до  $Kn < 10$ ) на базе использования подхода квази-гидродинамики (КГД).
- **Особенности пакета:**
  - Использование структурированных (собств.формат) и неструктурированных сеток (форматы msh/OpenFOAM) в 2d и 3d.
  - Решение уравнений Навье-Стокса в полной и параболизированной форме
  - Существуют однопоточная и НРС (параллельная) версия решателя



# Математическая основа кода



В основе пакета программ – решение в векторном виде обобщенной системы уравнений переноса.

С учетом указанной математической формулировки систему можно дополнять любым количеством уравнений переноса для моделирования различных физических процессов.

Обобщенная система уравнений в векторной форме с разделением на конвективные и вязкие потоки:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial x}(\mathbf{E}_C - \mathbf{E}_V) + \frac{\partial}{\partial y}(\mathbf{F}_C - \mathbf{F}_V) + \frac{\partial}{\partial z}(\mathbf{G}_C - \mathbf{G}_V) = \mathbf{H}$$

Конечно-объемное представление уравнений:

$$V_q \frac{\partial \mathbf{U}_q}{\partial t} = - \sum_{j \in q} (\vec{\mathbf{F}}_{C,j} - \vec{\mathbf{F}}_{V,j}) \cdot \vec{\mathbf{n}}_j S_j + V_q \mathbf{H}_q$$

Конвективные потоки:

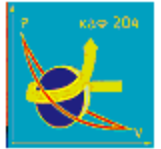
$$\vec{\mathbf{F}}_{C,j} \cdot \vec{\mathbf{n}}_j = \mathbf{F}_{C,n,j} = \mathbf{A}_L^+ \mathbf{U}_L + \mathbf{A}_R^- \mathbf{U}_R$$

Вязкие потоки:

$$\vec{\mathbf{F}}_V \cdot \vec{\mathbf{n}} = \mathbf{M} \cdot \nabla \mathbf{U}$$



# Математическая основа кода



Обозначив через  $\delta \mathbf{U}^{n+1}$  приращения вектора при переходе от n-го шага по времени к (n+1)-му шагу по времени, получим:

$$\delta \mathbf{U}^{n+1} = \mathbf{U}^{n+1} - \mathbf{U}^n$$

На каждом шаге по времени стоит задача определения  $\mathbf{U}^{n+1}$  по известным значениям  $\mathbf{U}^n$ .

В результате дискретизации вязких и невязких потоков получим следующую систему уравнений:

$$\mathbf{A}_q \delta \mathbf{U}_q^{n+1} + \sum_{j \in q} \mathbf{A}_{qj} \delta \mathbf{U}_{qj}^{n+1} = \Delta \mathbf{U}_q^n$$



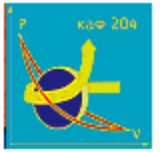
# Математическая основа кода



- Схема по конвекции – «против потока» первого/второго порядка
- Стабилизация решения путем введения искусственной вязкости в матрицу собственных значений
- Ограничитель градиента Барта-Джесперсона в схеме второго порядка
- Автоматическое обнаружение скачков уплотнения и стабилизация численной схемы на основе информации о волновой структуре течения
- Авторский численный метод для решения систем уравнений с жесткими источниками
- Алгебраические решатели – GMRES, BCGStab и др.



# Турбулентность



- Основа – уравнения Рейнольдса-Фавра (RANS)
- Модели:
  - $k-\varepsilon$
  - SST  $k-\omega$
  - Перенос напряжений Рейнольдса
  - Модель  $k-g-V_n$  (разработана авторами)
- Учет сжимаемости на базе подхода проф. А.М.Молчанова
- Взаимодействие со стенкой – Low-Re или Wall Functions



# Разреженные течения (КГД подход)



$$(1) \quad \frac{\partial \rho}{\partial t} + \operatorname{div} \vec{j}_m = 0,$$

$$(6) \quad \vec{j}_m = \rho(\vec{u} - \vec{w})$$

$$(2) \quad \frac{\partial(\rho\vec{u})}{\partial t} + \operatorname{div}(\vec{j}_m \otimes \vec{u}) + \vec{\nabla} p = \operatorname{div} \Pi,$$

$$(7) \quad \vec{w} = \frac{\tau}{\rho} \left( \operatorname{div}(\rho\vec{u} \otimes \vec{u}) + \vec{\nabla} p \right)$$

$$(3) \quad \frac{\partial}{\partial t} \left[ \rho \left( \frac{\vec{u}^2}{2} + \varepsilon \right) \right] + \operatorname{div} \left[ \vec{j}_m \left( \frac{\vec{u}^2}{2} + \varepsilon \right) \right] = \operatorname{div} \vec{A} - \operatorname{div} \vec{q}.$$

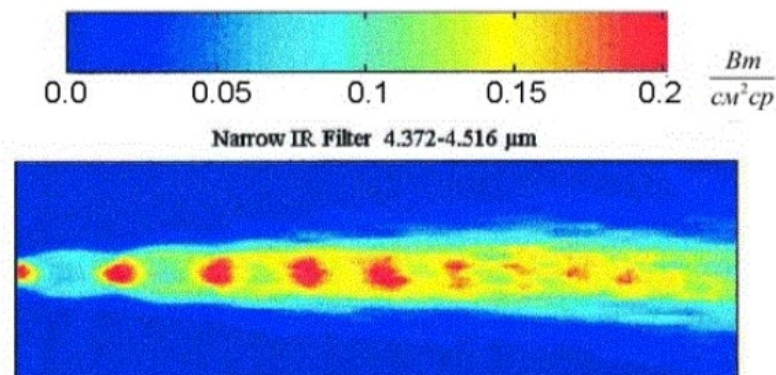
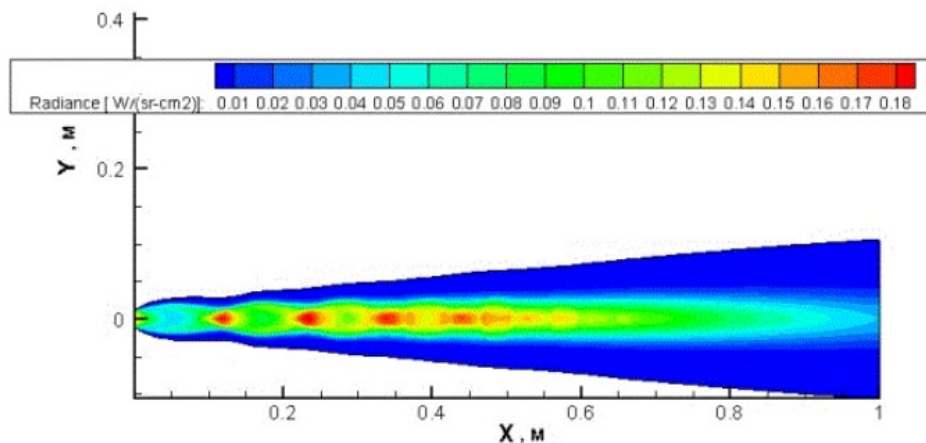
$$(8) \quad \tau = \frac{\mu}{pSc},$$

$$(4) \quad \Pi = \Pi_{NS} + \tau\vec{u} \otimes \left[ \rho(\vec{u} \cdot \vec{\nabla})\vec{u} + \vec{\nabla} p \right] + \tau I \left[ (\vec{u} \cdot \vec{\nabla})p + \gamma p \operatorname{div} \vec{u} \right]$$

$$(5) \quad \vec{q} = \vec{q}_{NS} - \tau\rho\vec{u} \left[ (\vec{u} \cdot \vec{\nabla})\varepsilon + p(\vec{u} \cdot \vec{\nabla}) \left( \frac{1}{\rho} \right) \right]$$

Реализовано два основных подхода:

- Детальный расчет излучения путем решения уравнения переноса излучения с применением метода  $k$ -распределения (для газов с ярко выраженным линейчатым спектром)
- Упрощенный расчет переноса излучения на основе метода P1 и статистической модели полосы





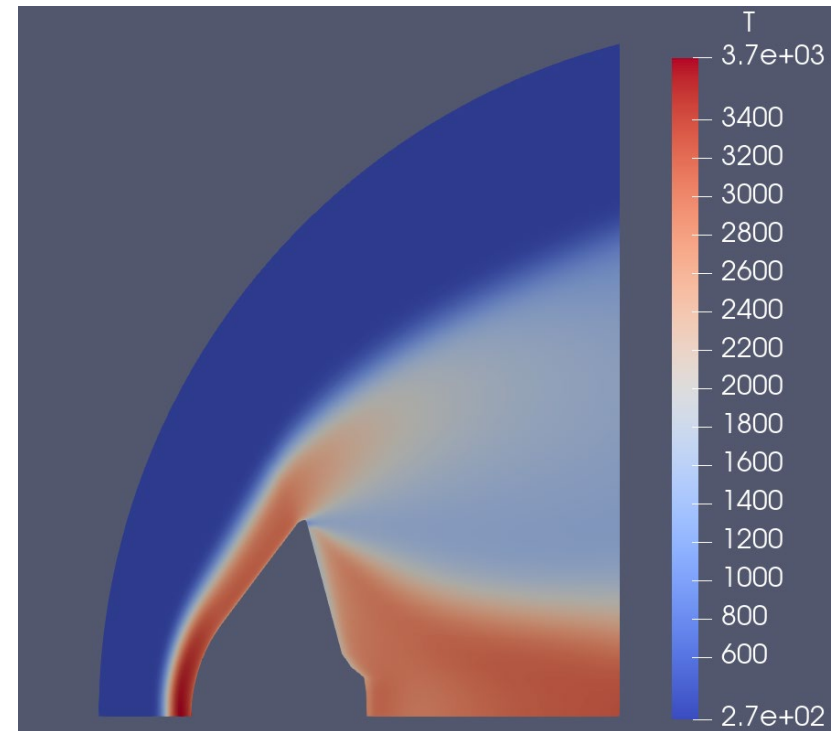
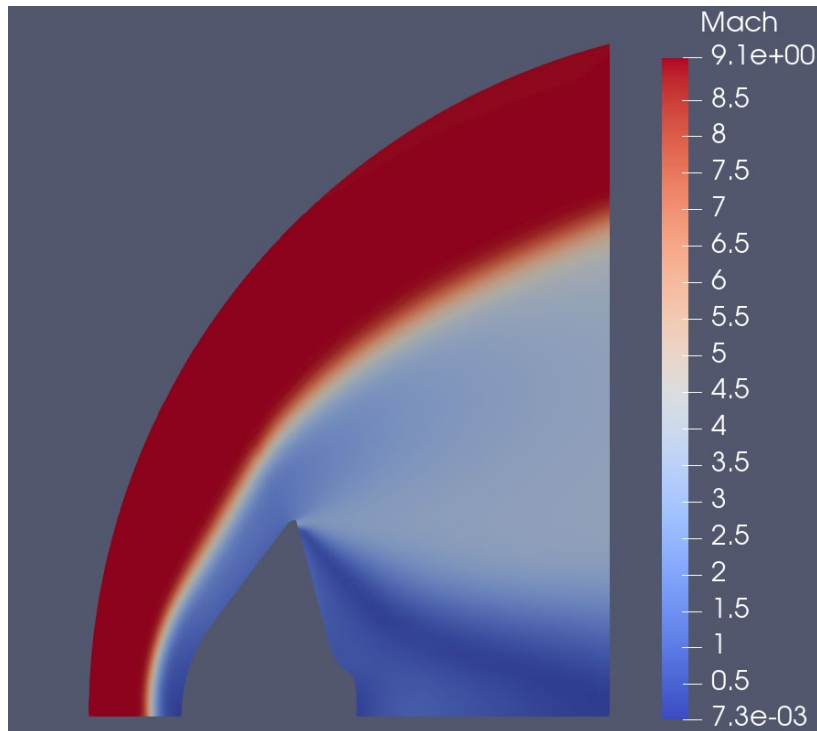


# Примеры получаемых результатов: спускаемый аппарат OREX



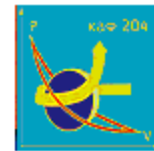
## Распределение числа Маха

## Распределение температуры



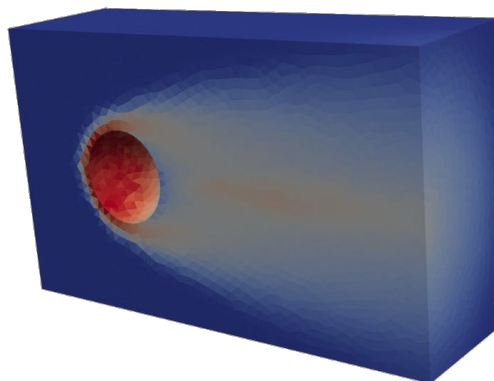
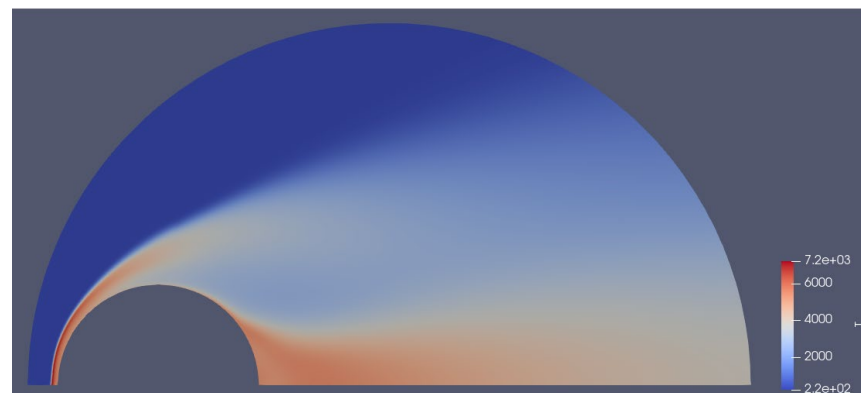
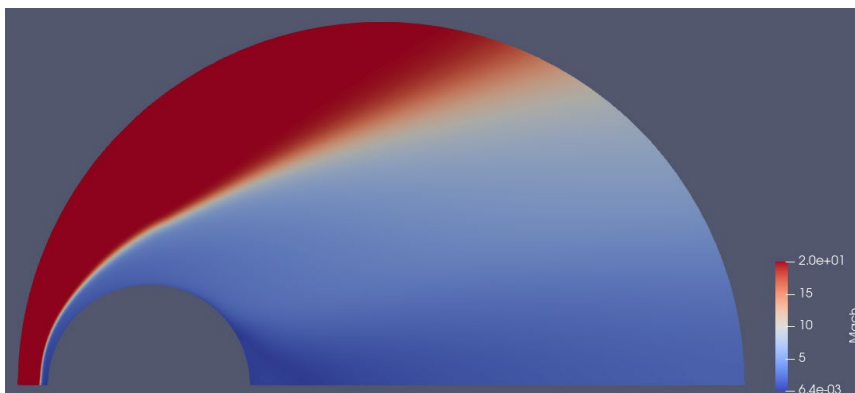


# Примеры получаемых результатов: обтекание сферы гиперзвуковым потоком $M=20$



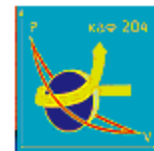
Распределение числа Маха

Распределение температуры

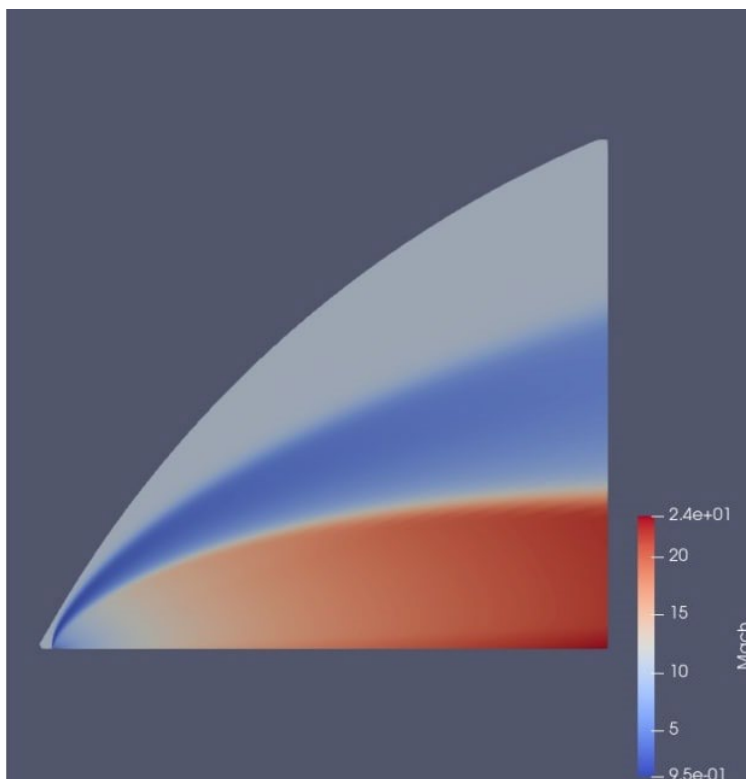




# Примеры получаемых результатов: высотная термический неравновесная струя (100 км, степень нерасчетности $10^6$ )



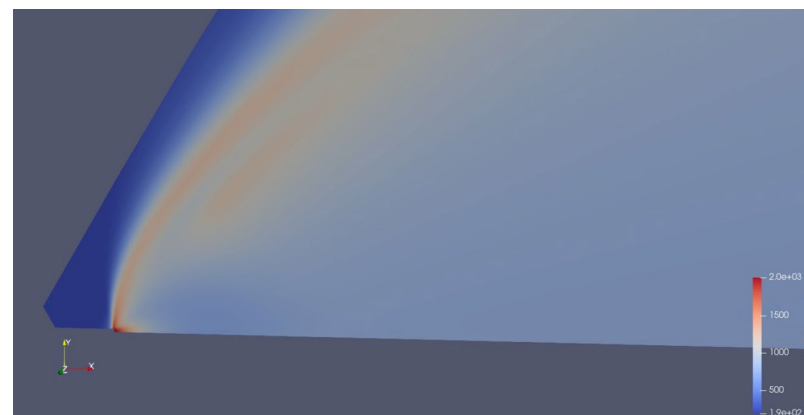
## Распределение числа Маха



## Распределение поступательной температуры



## Распределение колебательной температуры молекулы CO



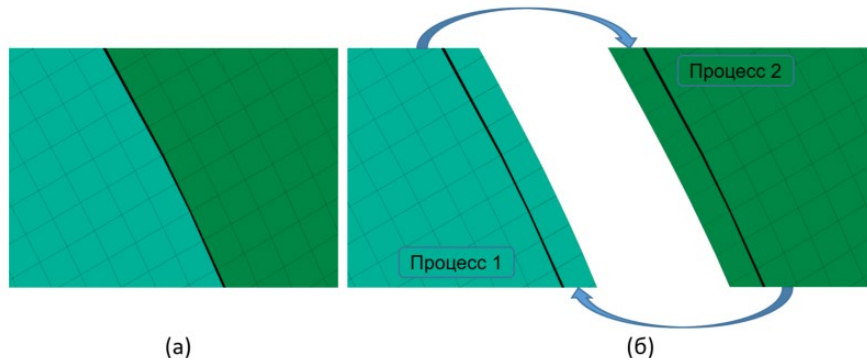
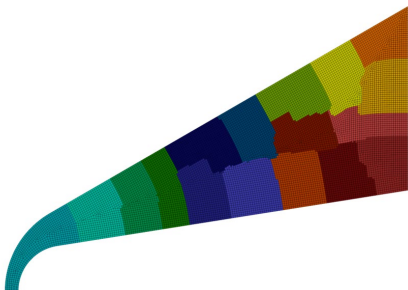
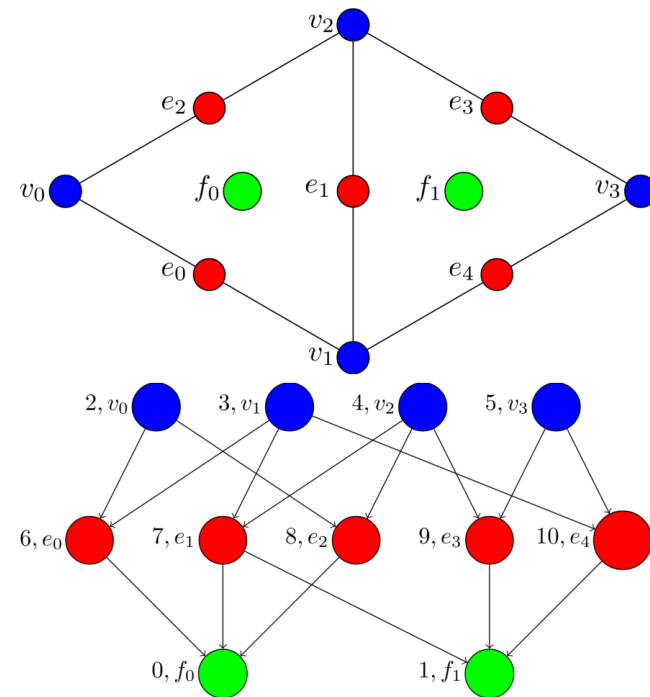


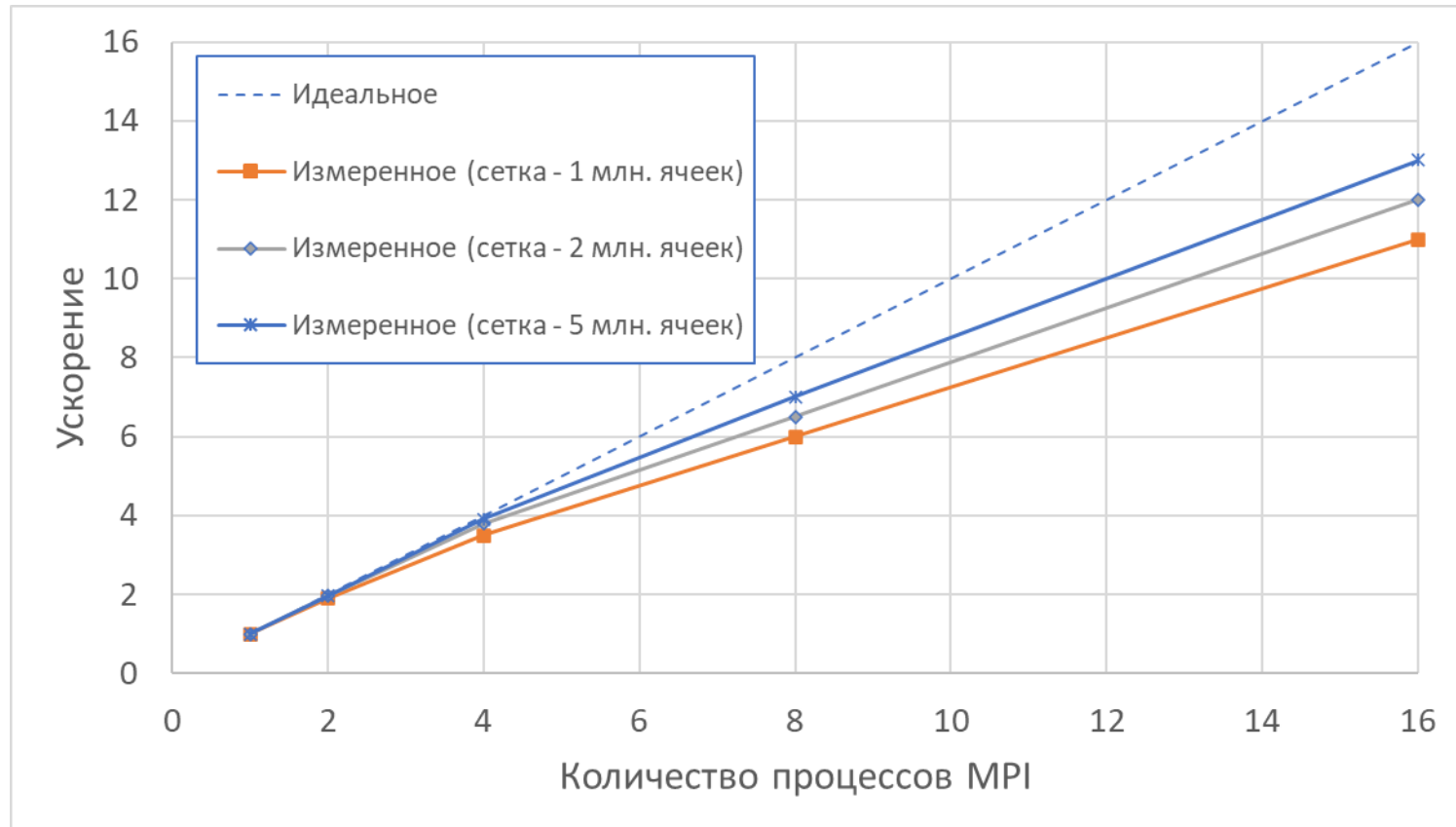
# Программная реализация



- Fortran 2008, C++17.
- Взаимодействие Fortran  $\leftrightarrow$  C++ через C-интерфейс “BIND(C)”.
- Data-oriented design, Structure of arrays (SoA).
- Полностью модульная архитектура (Вся функциональность реализована в виде модулей).
- Набор модулей определяется файлом задания.
- Каждый модуль при загрузке регистрирует реализуемые и используемые интерфейсы. После загрузки всех модулей происходит увязывание их между собой.
- Возможность реализации любой дополнительной функциональности в виде модулей без изменения уже реализованных частей решателя.

- Классическая MPI архитектура (каждый процесс рассматривается как однопоточное приложение).
- Для распределения расчётного домена между процессами и решения системы уравнений используется библиотека PETSc (DMPLex) и PT-Scotch.
- Каждый процесс заполняет свою часть матрицы, обмен данными между процессами сведён к минимуму.
- Для снижения влияния задержек сети на производительность используется перекрытие во времени между обменом данными и вычислениями.

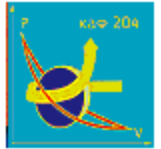




Сильная масштабируемость на AMD Ryzen 9 5950X, 16 ядер, SMT off.



# Целевые архитектуры вычислительных комплексов и операционные системы



Протестированные архитектуры:

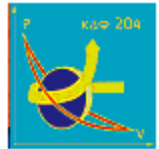
- x86-64
- ARMv8-A + VFPv4

Операционные системы:

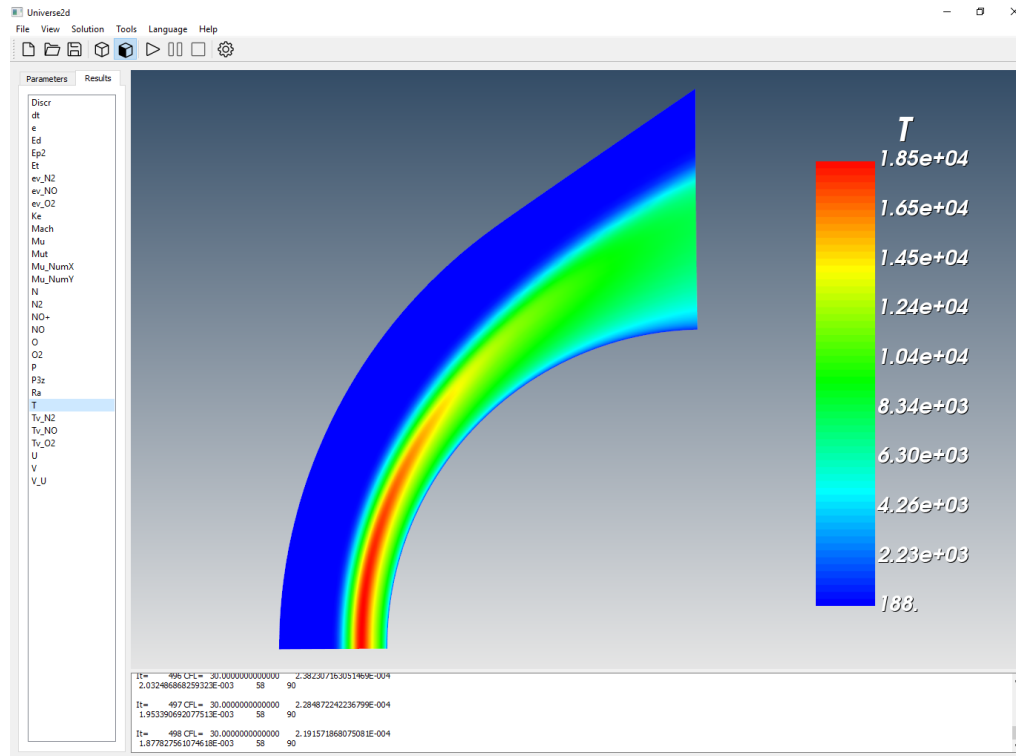
- Linux: Astra (x86-64), RHEL7-9 (x86-64), Ubuntu (x86-64, ARM)
- Windows 10



# Пре-процессор и постпроцессор



- Пре-постпроцессор на Qt5+VTK для однопоточной версии.



- Для параллельной версии разрабатывается пре-постпроцессор на основе custom-сборки ParaView с использованием Catalyst2 для in-situ обработки данных и визуализации.





СПАСИБО ЗА  
ВНИМАНИЕ!