Пакет расчётных программ лаборатории аэрофизических исследований МФТИ

Оптимизации методов и рефакторинг legacy кода

Докладчик:
Погорелов И.О.
(аспирант МФТИ, инженер ЦАГИ)

CFD Weekend, декабрь 2023

ЛАФИ МФТИ

Области интереса:

– ЛТП

– Устойчивость ПС

– Развитие возмущений ПС

Состав:

- *Егоров И.В.*, д.ф.-м.н., член-корр РАН
- Фёдоров А.В., к.ф.-м.н.
- Устинов М.В., д.ф.-м.н.
- Новиков А.В, д.ф.-м.н.
- Чувахов П.В, д.ф.-м.н
- Образ А.О., к.ф.-м.н.
- Пальчековская Н.В., к.ф.-м.н.

Что в пакете? Основной солвер

- Близкий родственник HSFlow
- Решение задач транс- и сверхзвуковой газовой динамики
- NSE, RANS, Химия, 2D- и 3D- задачи
- Структурированные сетки
- Неявный метод конечных объёмов 2-го порядка
- Работа на супер-ЭВМ

Что в пакете? Вспомогательные утилиты

… для работы с сетками:

Разбиение сетки на зоны, слияние зон, преобразование 2D -> 3D, извлечение сечений, выделение скачков, …

… для работы с полями:

Интерполяция, сложение/вычитание, расчёт интегральных характеристик (C_x, C_y, C_f, …), …

… для солвера:

Анализ логов, генерация настроек, отправка задачи в slurm

И библиотеки python:

pylibcgns, pyhsfcgns

Что в пакете? Код устойчивости

- Расчёт неустойчивых мод погранслоя
- Выделение неустойчивых мод из DNS
- e-N метод
- Генерация источников турбулентности для уравнений RANS

Рефакторинг. Возникновение legacy кода

Солвер имеет длинную историю:

– **Fortran66** (1980-90-е)

→Построчно переписан на **C** (1990-2000-е)

→Плавный переход на **C++** (2000-10-е)

Рефакторинг. Тяжёлое наследие 90-х

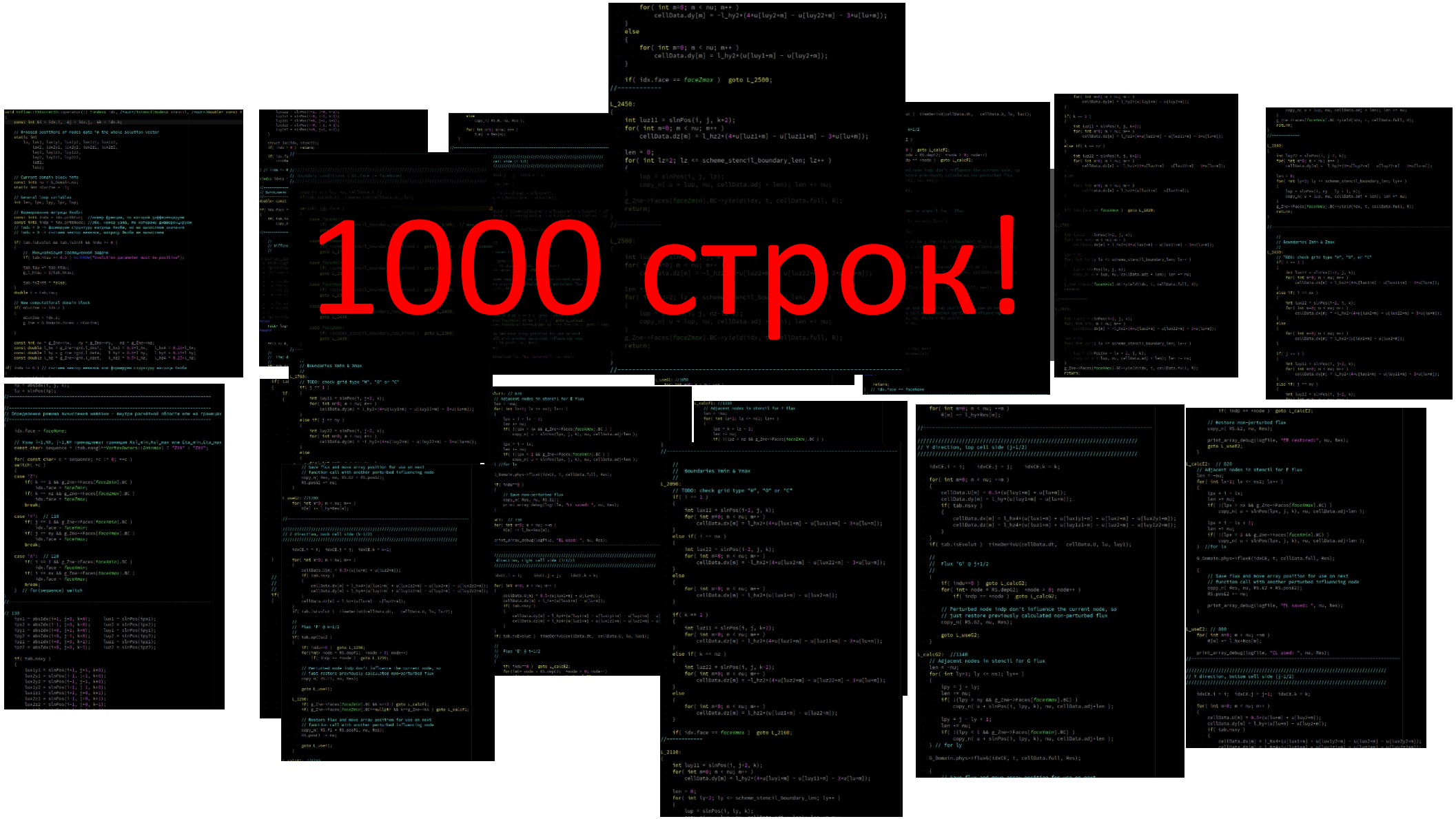
```
static int
    lu, lux1, lux1y1, lux1y2, lux1z1, lux1z2 /*, ... */;

extern int *data, *b;

if( condition1 )
    goto L_calcE2;

for (int *a = data; *a > 0; ++a)
    if ( a == b)
        goto L_calcE2;

// ...

goto L_useE2;

L_calcE2:
    // ...

L_useE2:
    // ...
```

← Непонятные названия переменных Fortran – style

↖ Обилие глобальных переменных

Обилие запутанных GOTO

Рефакторинг. Огромные функции



Рефакторинг. *Почти* повторяющийся код

Метод конечного объёма:

- Рассчитать потоки через левую стенку ячейки
- --//-- через правую
- Верхнюю, нижнюю, заднюю, переднюю
- То же самое для 2D

Итого:

10 (!) почти одинаковых участков
с точностью до оси и смещения

Рефакторинг. Разбиение на функции

Выделить повторяющийся код в функцию – классическое решение

Функции расчёта потоков требуется:

1. Индекс ячейки
2. Обозначение грани ячейки: [x, y, z]×[+,-]

Проблемы

Накладные расходы на вызовы

Индексы в схемах

~~Производительность~~

Рефакторинг. Параметры времени компиляции

Шаблоны + *inline* функции

- Шаблоны позволяют задавать грань
- inline функции предотвращают оверхед на вызовы

```
// axis = [X, Y, Z], edge = [+, -]
template<char axis, char edge>
void get_flux(const Tindex &index, ...)
{ /* ... */ }
```

constexpr

- Позволяет не пересчитывать одно и то же

const &

- Помогает компилятору в оптимизации

```
template<char axis>
void get_flux_diff(const Tindex& idx, ...)
{
    get_flux<axis, '-'>(idx)
            - get_flux<axis, '+'>(idx);
}
```

Рефакторинг. Кэширование индексов

Примерный вид схем для расчёта производных:

```
du_dx = U[get_index(i+1, j, k)] - U[get_index(i, j, k)];
du_dy = U[get_index(i, j+1, k)] - U[get_index(i, j, k)];
du_dz = U[get_index(i, j, k+1)] - U[get_index(i, j, k)];
```

Проблемы:

- Вычисление индекса зависит от направления сеточной линии
- Постоянные пересчёты

Решение:

- Шаблонный кэш индексов

Рефакторинг. Кэширование индексов

```
struct TIdxsCache {
    static constexpr int stencil_size = 3*3*3;
    int idxs[stencil_size];

    constexpr int mempos(int di, int dj, int dk) const {
        // ...
        return index_in_idxs;
    }

    void init(const Tindex &idx) { /*fill idxs array*/ }

#define CHOOSE_3D(axis0) \
    template<char axis>  \
    typename std::enable_if<axis==axis0, int>::type

    CHOOSE_3D('X')
    get(int d0, int d1, int d2) const
    {  return idxs[mempos(d0, d1, d2)];  }

    CHOOSE_3D('Y')
    get(int d0, int d1, int d2) const
    {  return idxs[mempos(d2, d0, d1)];  }

    CHOOSE_3D('Z')
    get(int d0, int d1, int d2) const
    {  return idxs[mempos(d1, d2, d0)];  }
#undef CHOOSE_3D
};
```

← Массив для кэша индексов
Относительная индексация

← *constexpr* – положение индекса в кеше

← Считаем индексы единожды

← SFINAE* – компилируется только вариант функции get() для нужной оси

*Substitution Failure Is Not An Error

← Просто меняем оси местами

Рефакторинг. Кэширование индексов

Вычисление производной

```
template<char axis>
inline double derivative1(const TIdxsCache &c) {
    return u[c.get<axis>(1,0,0)] - u[c.get<axis>(0,0,0)];
}
```

Вычисление производных по осям параметризовано!

```
TIdxsCache c;
c.init(idx);
double du_dx = derivative1<'X'>(c);
double du_dy = derivative1<'Y'>(c);
double du_dz = derivative1<'Z'>(c);
```

Более удобный код при нулевом overhead’е!

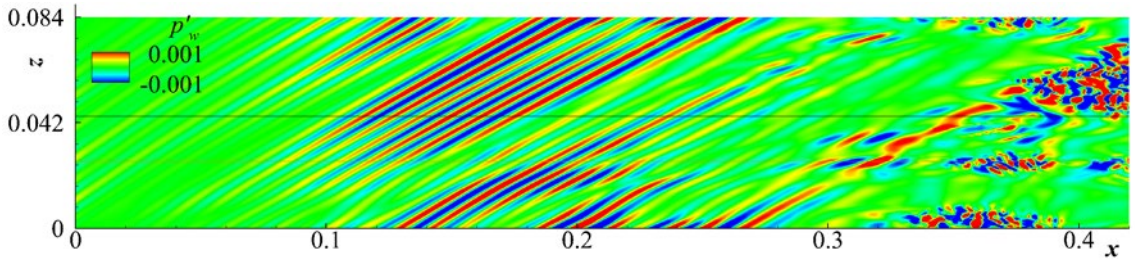
Оптимизация расчётных методов

Методом Ньютона решаются сеточные уравнения $\vec{f}(\vec{u}) = 0$ на каждом временном шаге:

1. Начальное приближение $\vec{\mathrm{u}}_{\mathrm{k}=0} = \vec{U}$ ← Чем точнее – тем лучше
2. Матрица Якоби $\boldsymbol{J}_{ij} = \frac{\delta f_i}{\delta u_j}$ ← ~ 50% времени
3. Решение СЛАУ $\boldsymbol{J}\,\Delta u_k = -\vec{f}\,(\vec{u}_k)$
4. $\vec{u}_{k+1} = \vec{u}_k + \Delta u_k$
5. До сходимости $\|\Delta u_k\| < \varepsilon$ переходить к (2) или (3)

Оптимизация методов.
«Ленивый» расчёт матрицы Якоби

Типичная задача содержит линейную стадию



Возмущения давления на поверхности крыла*

Линейное развитие

Нелинейный «хвост»

⇒матрица Якоби изменяется слабо

⇒Пересчёт раз в N шагов

⇒Ускорение 45% (N = 10) !!!

* Чувахов П.В., Погорелов И.О. Источники турбулентности на прямом крыле сверхзвукового пассажирского самолёта // Математическое моделирование, 2022

Оптимизация методов. Начальное приближение

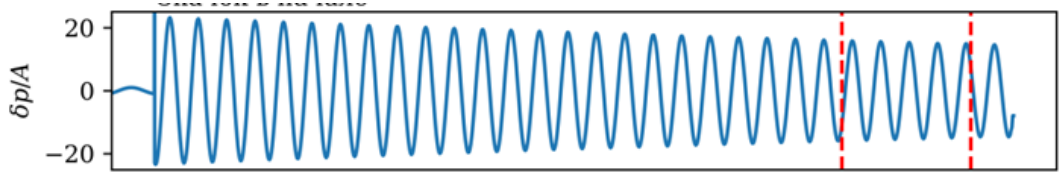
- Меньше итераций — меньше время расчёта
- Точнее начальное приближение — меньше итераций
- ⇒ Точнее приближение — быстрее расчёт

Используется полином. экстраполяция по полям на предыдущих шагах

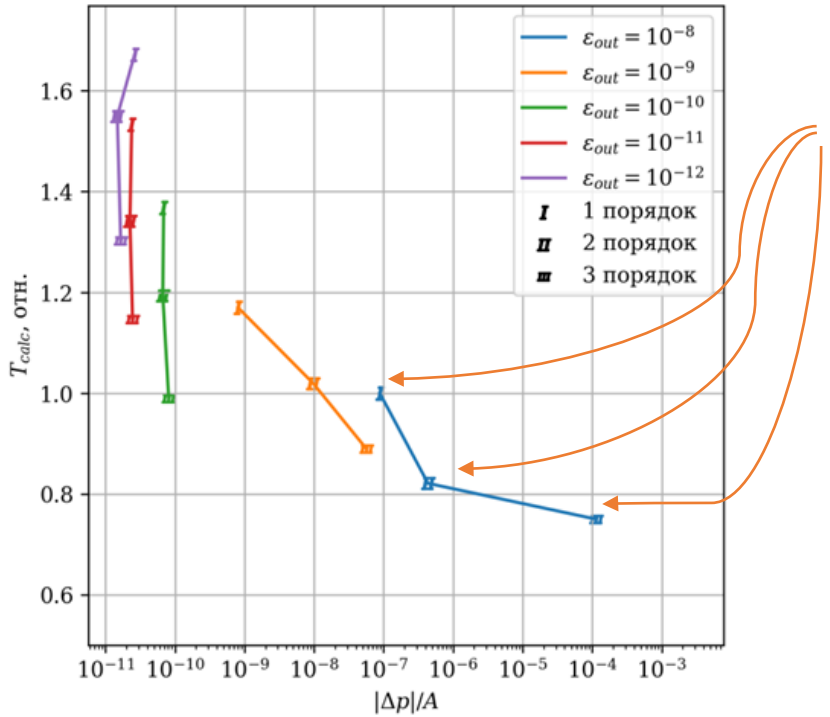
Корректно, если $\vec{u}(t)$ непрерывна.

Численная вязкость размывает скачки, поэтому корректность не нарушается.

Оптимизация методов. Начальное приближение



Тестовая задача — прохождение звука через сильный скачок



Экстраполяция увеличивает ошибку!

- Точность определяется по *худшей* ячейке
- Худшие ячейки – нелинейные
- В линейных областях точность избыточна
- Экстраполяция «уравнивает» все области
- Растут ошибки во всём поле

Оптимизация методов. Итоги

CFD Weekend - 2016

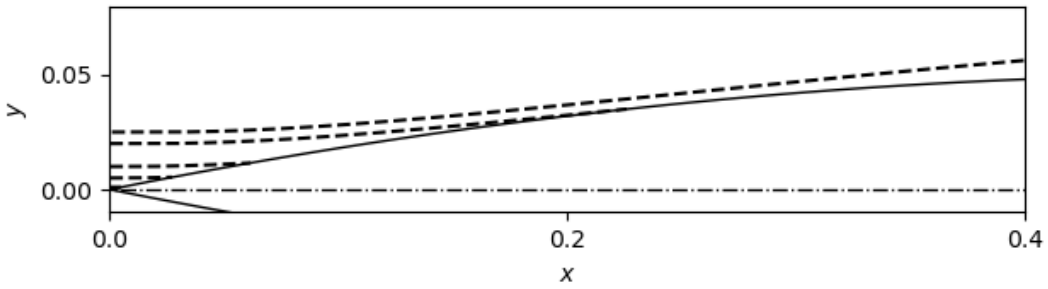
- 250 млн узлов
- Сетка 5.6 Гб, поле 9.3 Гб
- Кластер Flowmodellium МФТИ
- 1152 процессорных ядра

CFD Weekend - 2023

- 256 млн узлов
- Сетка 6 Гб, поле 10 Гб
- Кластер Flowmodellium МФТИ
- 384 процессорных ядра

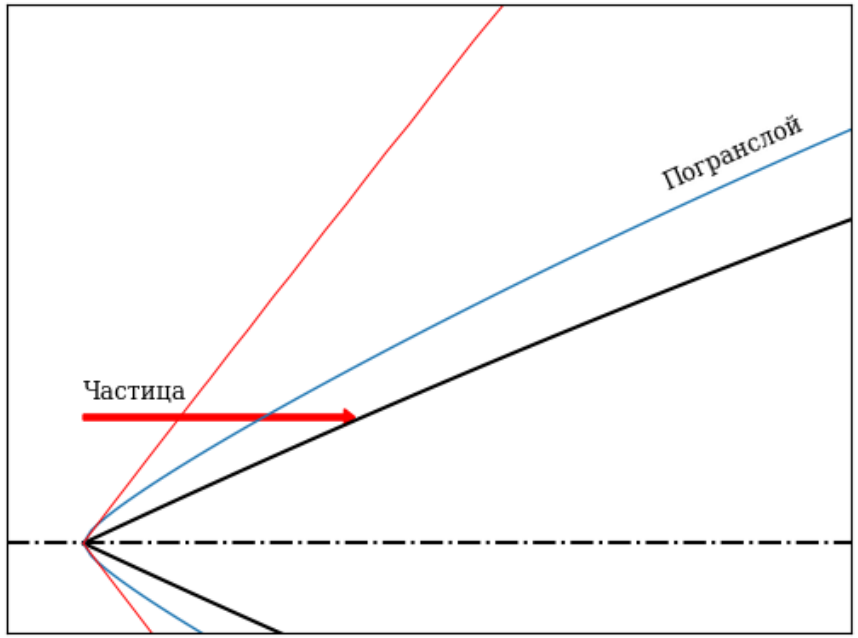
3-х кратное ускорение!

Расчёт возмущений, порождённых частицей



- «Чечевичный» профиль. Длина 5 м, толщина 5%, 10%
- Стандартная атмосфера, 20 км
- M = 3
- $Re_L = 27$ млн
- Референс – *Aerion AS2*

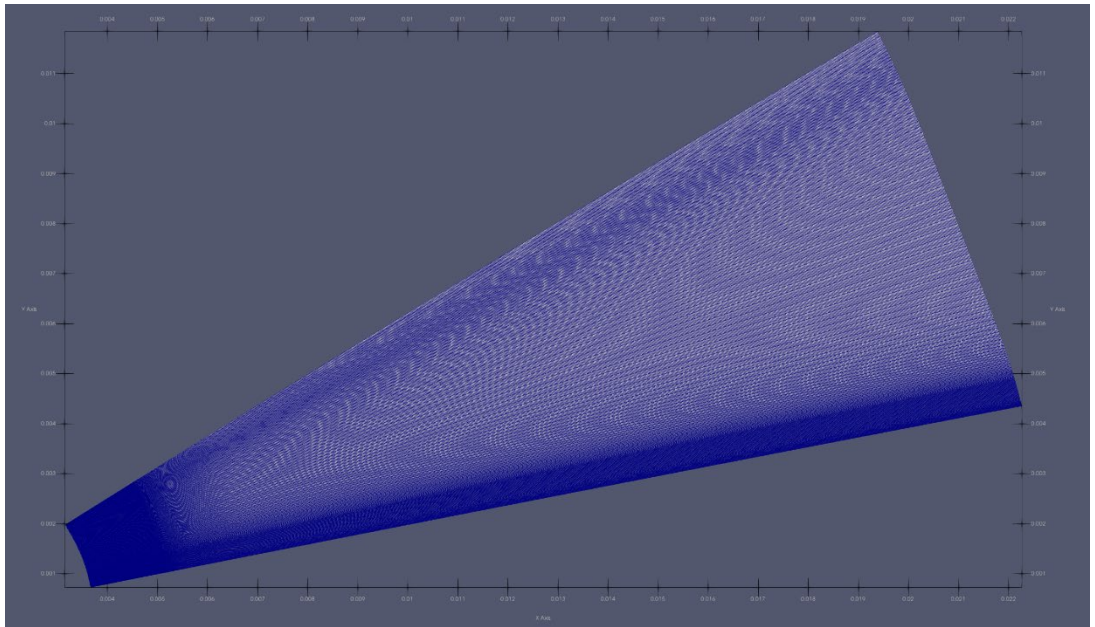
- Радиус частицы 10 мкм
- Плотность 1 г/см^3
- Покоится отн. набегающего потока
- После удара о пов-ть исчезает
- Удар при x = 0.005 (10% профиль)
- … при x = 0.05 (5% профиль)



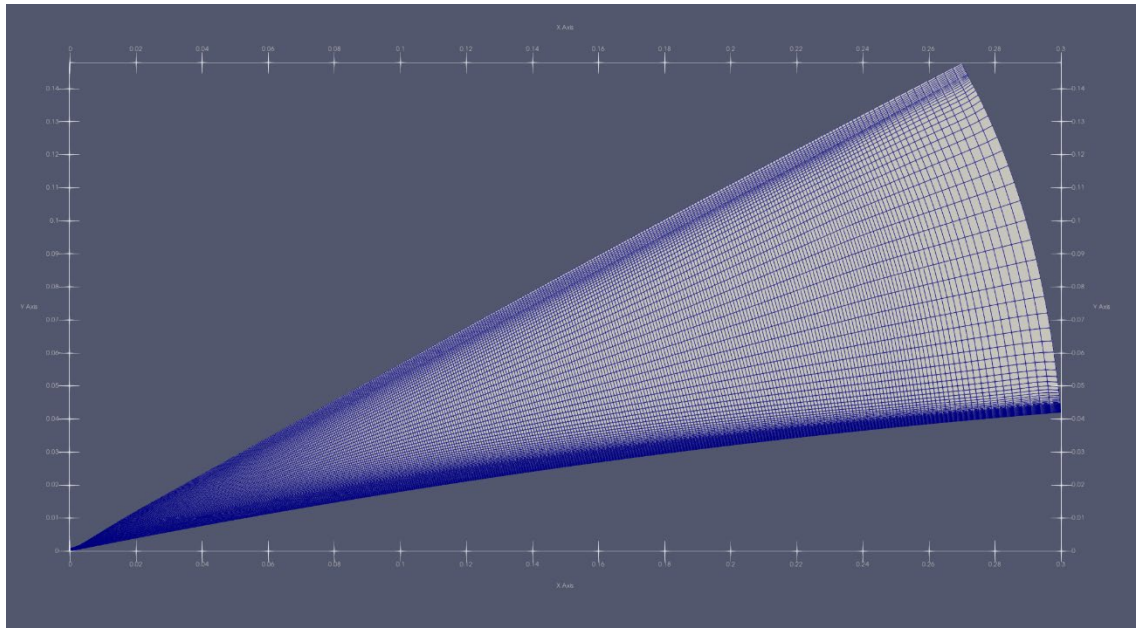
Расчёт возмущений. Сетка для 10% профиля

- Три части.
- Носовая – 2D, 77 тыс. улов
- Средняя – шаг по X 10^{-7}
- Дальняя – шаг по X 10^{-5}
- Перенос возмущений узел-в-узел

Средняя сетка:
1916x285x121 (66 млн)



Дальняя сетка:
4896x285x121 (169 млн)



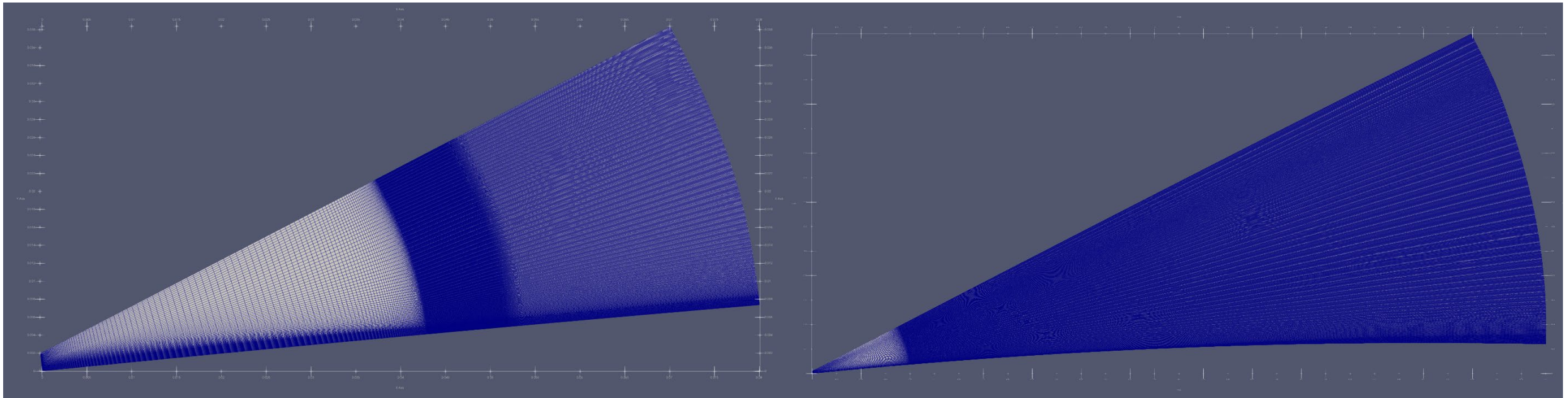
Расчёт возмущений. Сетка для 5% профиля

- Две части.
- Носовая – шаг по X 10^{-7}
- Дальняя – шаг по X 10^{-5}

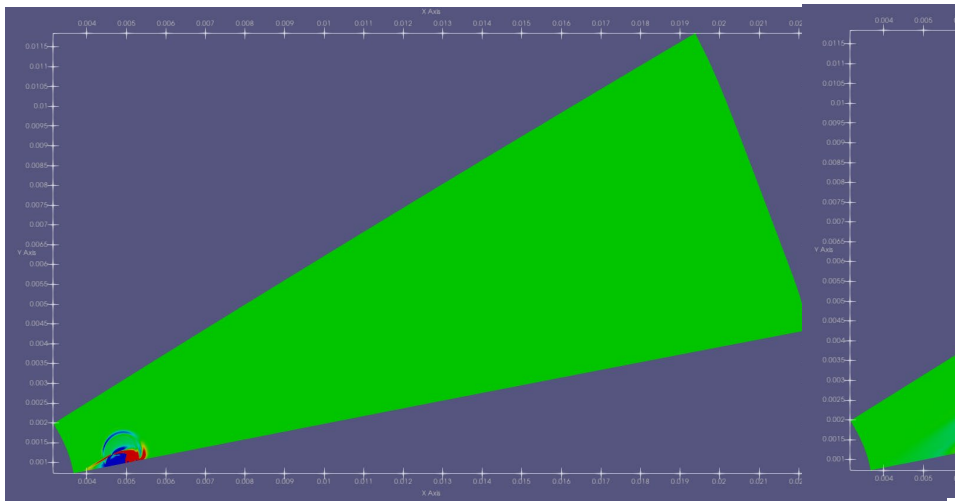
- Перенос возмущений узел-в-узел

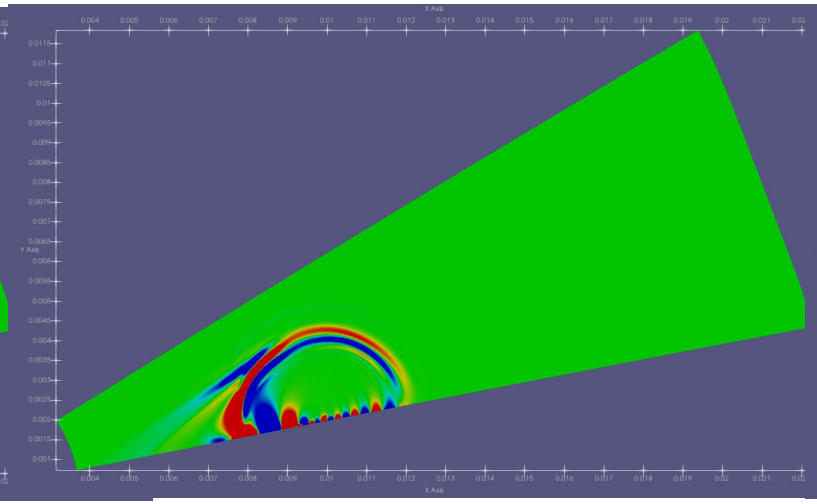
Носовая сетка:
1892x278x101 (56 млн)

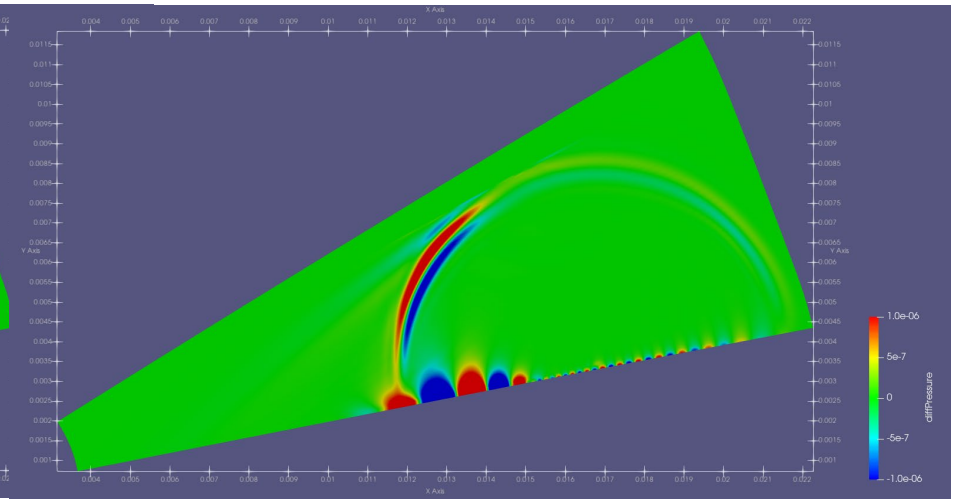
Дальняя сетка:
9106x278x101 (256 млн)



Расчёт возмущений. Результат для 10% профиля

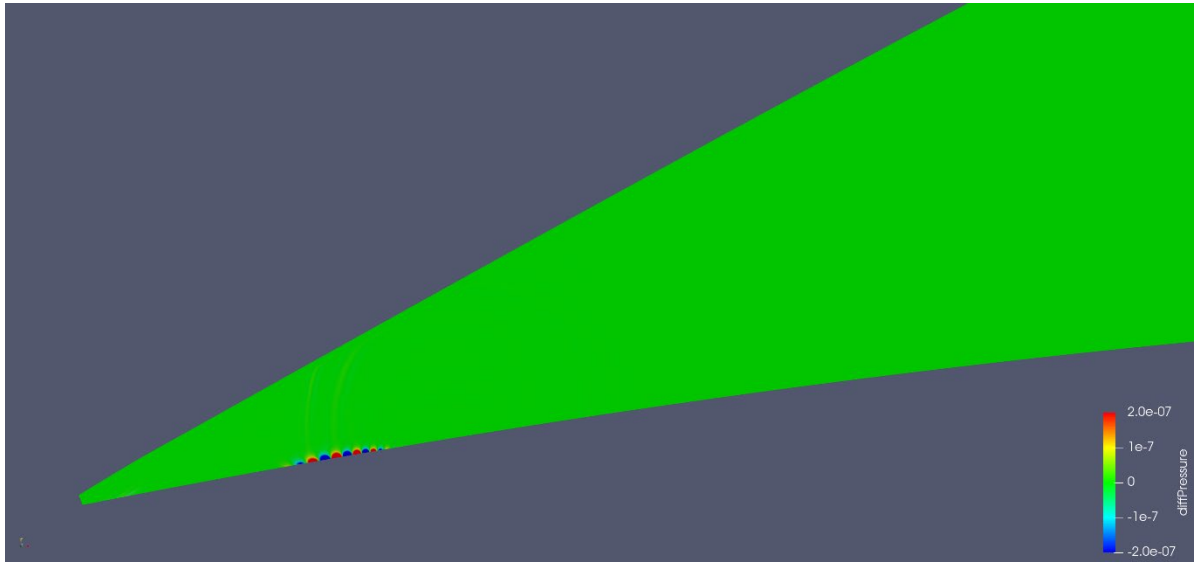




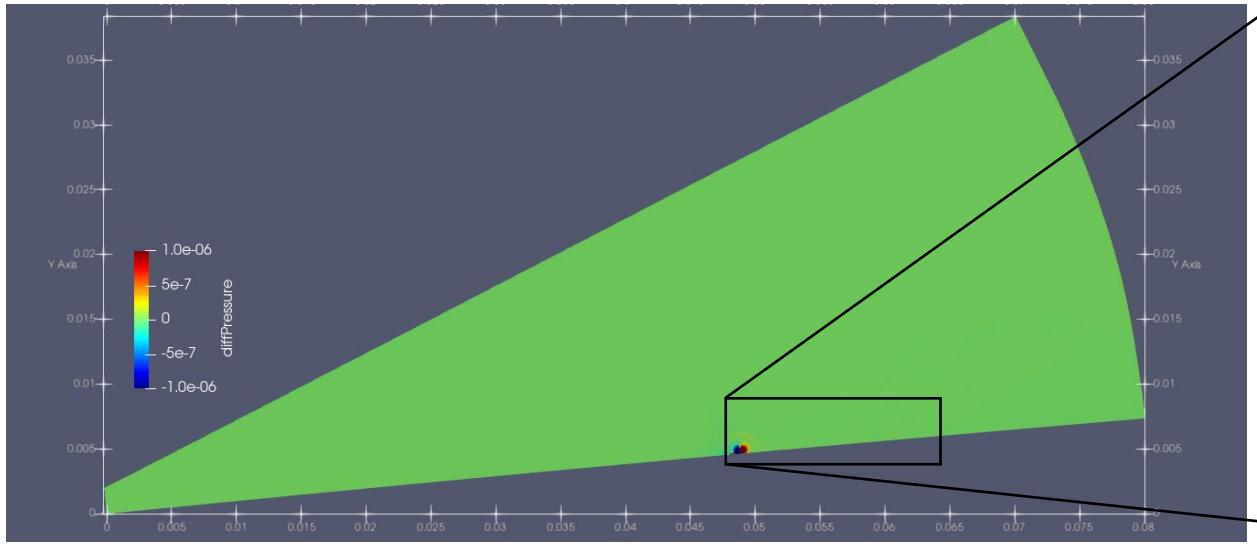


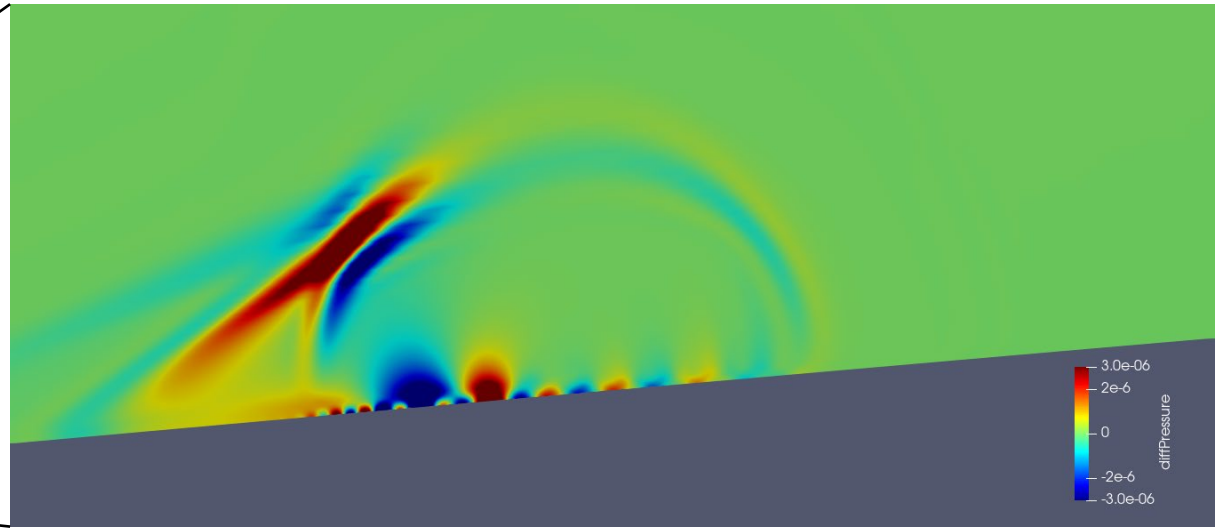
Удар частицы о поверхность → Выделение 1-й моды → Развитие волнового пакета →

→ Перенос возмущений на 3-ю сетку → Постепенное затухание волнового пакета



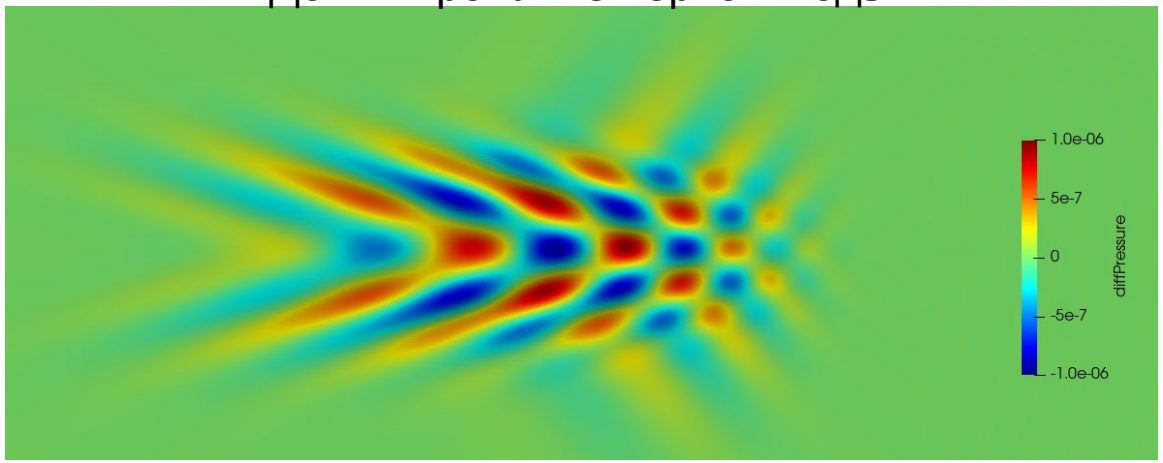
Расчёт возмущений. Результат для 5% профиля





Удар частицы о поверхность → Выделение 1-й моды →

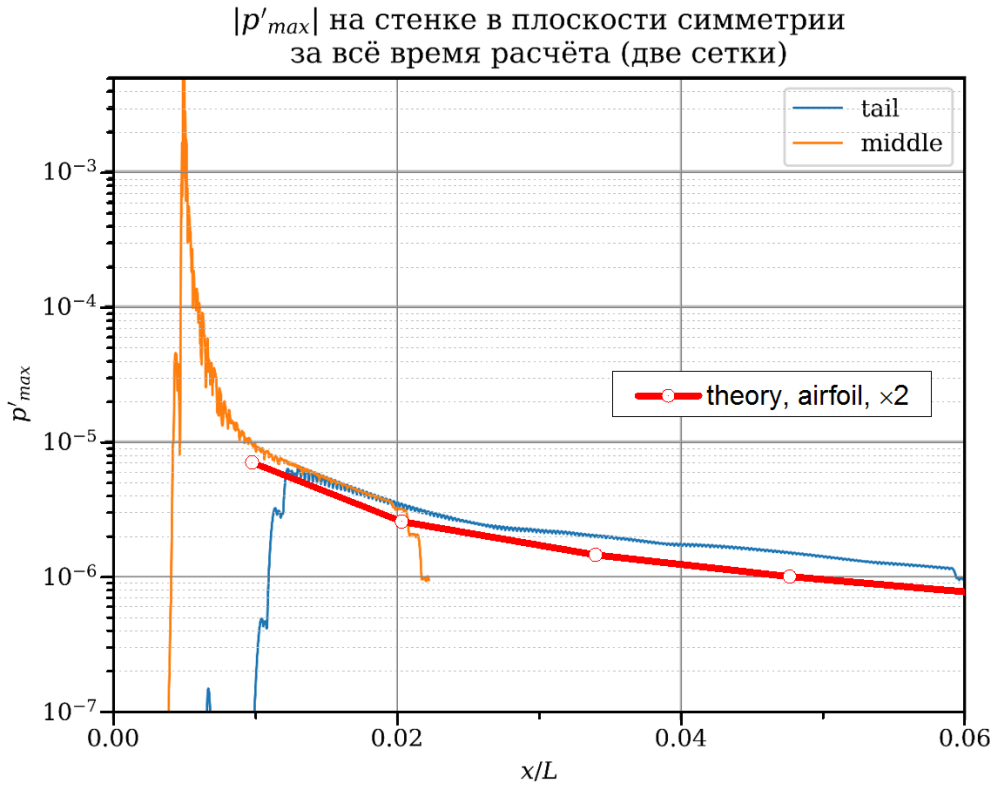
Доминирование первой моды



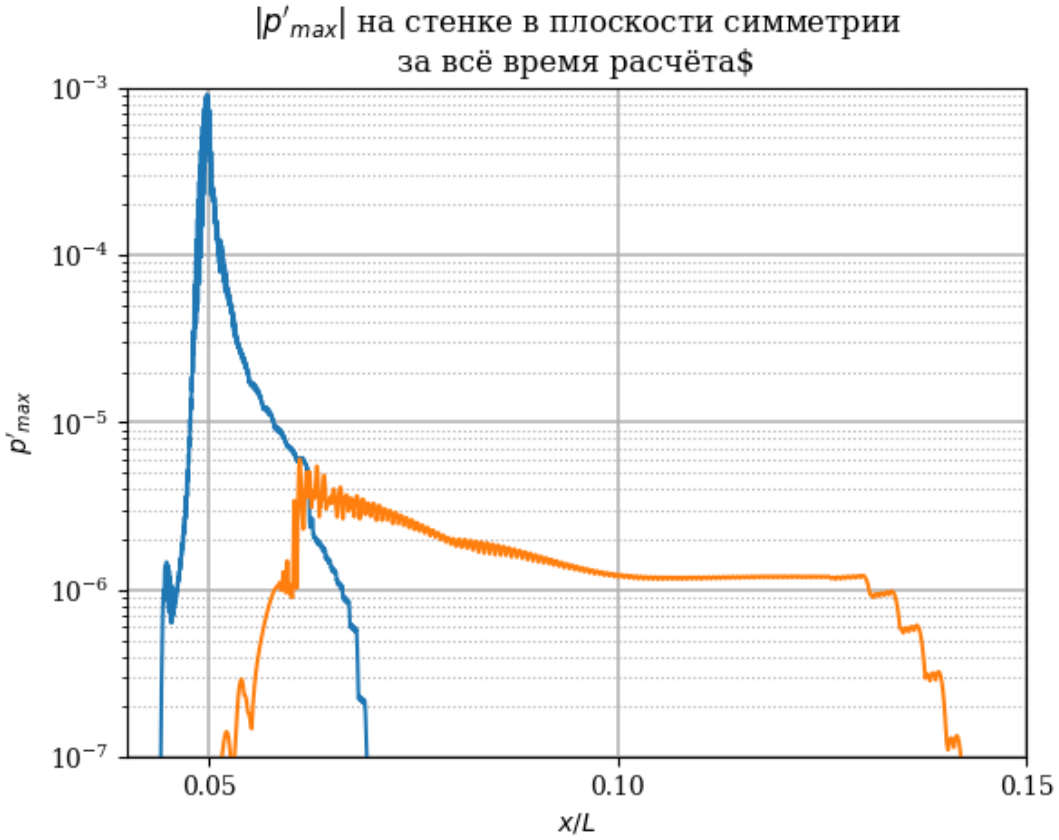
→ В процессе расчёта…

Расчёт возмущений. Сравнение с теорией

Профиль 10%



Профиль 5%



Пакет расчётных программ лаборатории аэрофизических исследований МФТИ

Оптимизации методов и рефакторинг legacy-кода

Докладчик:
Погорелов И.О.
(аспирант МФТИ, инженер ЦАГИ)

CFD Weekend, декабрь 2023