Решение дифф. уравнений на CUDA на примере задач аэро-гидродинамики.

- Лектор:
  - Сахарных Н.А. (ВМиК МГУ, NVidia)

План

- Введение и постановка задачи
- Основные уравнения
- Численный метод расщепления
- Особенности реализации
- Результаты и выводы

Введение

- Вычислительные задачи аэро-гидродинамики
  - Моделирование турбулентных течений

- ВМиК МГУ, кафедра мат. физики
  - Пасконов В.М., Березин С.Б.

Турбулентность

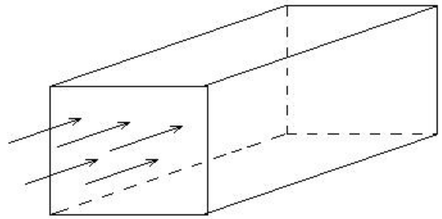
Моделирование турбулентности

- Прямое численное моделирование (DNS)
- Моделирование крупномасштабных вихрей (LES)
- Осредненные уравнения Навье-Стокса (RANS)

• все масштабы турбулентности
• очень затратный

Постановка задачи

- Течение <u>вязкой несжимаемой</u> жидкости в 3D канале



  - Канал заполнен и находится в однородной среде
  - Произвольные начальные и граничные условия
  - Неизвестные величины – скорость и температура

Основные уравнения

- Полная система уравнений Навье-Стокса в безразмерных величинах
  - Уравнение неразрывности
  - Уравнения движения (Навье-Стокса)
  - Уравнение энергии

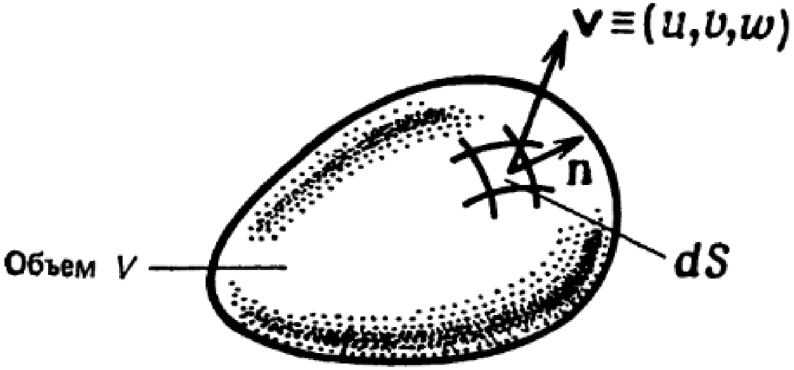
Обозначения

Плотность	$\rho = const = 1$
Скорость	$\mathbf{u} = (u, v, w)$
Температура	T
Давление	p

- Уравнение состояния $p = \rho RT = RT$

R – газовая постоянная

Уравнение неразрывности



$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

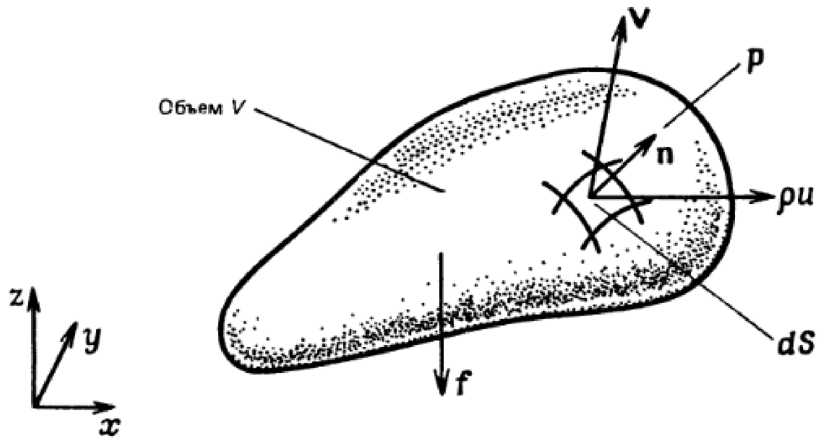
$$\rho = const$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

- Используется при выводе остальных уравнений (движения и энергии)
- Проверка точности текущего решения

Уравнения Навье-Стокса

- Второй закон Ньютона: $\int_V \rho \frac{D\mathbf{v}}{Dt} dV = \sum \mathrm{F}$



<u>Невязкая</u> жидкость:

$$\sum \mathrm{F} = \int_V (\rho\mathbf{f} - \nabla p) dV$$

<u>Вязкая</u> жидкость:

$$\sum \mathrm{F} = \int_V (\rho\mathbf{f} - \nabla p + \nabla \cdot \boldsymbol{\tau}) dV$$

f – массовые силы (сила тяжести)

$\boldsymbol{\tau}$ – тензор вязких напряжений

p – давление

Безразмерные уравнения

- Параметры подобия
  - Число Рейнольдса
  - Число Прандтля

$$\mathrm{Re} = \frac{V'L'}{\mu'} \qquad \mathrm{Pr} = \frac{\mu' c_p'}{k'}$$

V', L' – характерная скорость, размер

μ' – динамическая вязкость среды

k' – коэффициент теплопроводности

c_p' – удельная теплоемкость

- Уравнение состояния для идеального газа/жидкости:

$$p = \rho RT$$

Уравнения движения

- Безразмерная форма:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla T + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}$$

  - Не рассматриваем массовые силы $\mathbf{f} = 0$
  - Уравнение состояния $p = T$

Уравнение энергии

- Первый закон термодинамики для объема V:

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = -\nabla T + \frac{1}{\Pr \cdot \mathrm{Re}} \Delta T + \frac{\gamma - 1}{\gamma \cdot \mathrm{Re}} \Phi$$

- Диссипативная функция:

$$\Phi = \Phi_x + \Phi_y + \Phi_z$$

$$\Phi_x = 2\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial w}{\partial x}\frac{\partial u}{\partial z}$$

$$\Phi_y = \left(\frac{\partial u}{\partial y}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2 + \frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \frac{\partial w}{\partial y}\frac{\partial v}{\partial z}$$

$$\Phi_z = \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 + 2\left(\frac{\partial w}{\partial z}\right)^2 + \frac{\partial u}{\partial z}\frac{\partial w}{\partial x} + \frac{\partial v}{\partial z}\frac{\partial w}{\partial y}$$

Финальные уравнения

- 4 нелинейных уравнения

$$
\begin{cases}
\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla T + \dfrac{1}{\mathrm{Re}} \nabla^2 \mathbf{u} \\
\dfrac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = -\nabla T + \dfrac{1}{\mathrm{Pr} \cdot \mathrm{Re}} \Delta T + \dfrac{\gamma - 1}{\gamma \cdot \mathrm{Re}} \Phi
\end{cases}
$$

- 4 неизвестные величины:
  - Компоненты скорости: u, v, w
  - Температура: T

Численный метод

- Расщепление по координатам

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$$

X

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

Y

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial y^2}$$

Z

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial z^2}$$

Уравнение диффузии

- 3 дробных шага – X, Y, Z
- Неявная конечно-разностная схема

$$\boxed{\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}} \longrightarrow \frac{u_{i,j,k}^{n+1/3} - u_{i,j,k}^{n}}{\Delta t} = \frac{u_{i+1,j,k}^{n+1/3} - 2u_{i,j,k}^{n+1/3} + u_{i-1,j,k}^{n+1/3}}{\Delta x^2}$$

$$q = 2 + \frac{\Delta x^2}{\Delta t} \qquad \begin{pmatrix} q & -1 & & & 0 \\ -1 & q & -1 & & \\ & -1 & q & -1 & \\ & & -1 & \ddots & -1 \\ 0 & & & -1 & q \end{pmatrix} \cdot \begin{pmatrix} u_{1,j,k}^{n+1/3} \\ \vdots \\ u_{i,j,k}^{n+1/3} \\ \vdots \\ u_{nx,j,k}^{n+1/3} \end{pmatrix} = \frac{\Delta x^2}{\Delta t} \begin{pmatrix} u_{1,j,k}^{n} \\ \vdots \\ u_{i,j,k}^{n} \\ \vdots \\ u_{nx,j,k}^{n} \end{pmatrix}$$

Уравнения Навье-Стокса

- Уравнение для X-компоненты скорости

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{\partial T}{\partial x}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)$$

X

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=-\frac{\partial T}{\partial x}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}\right)$$

Y

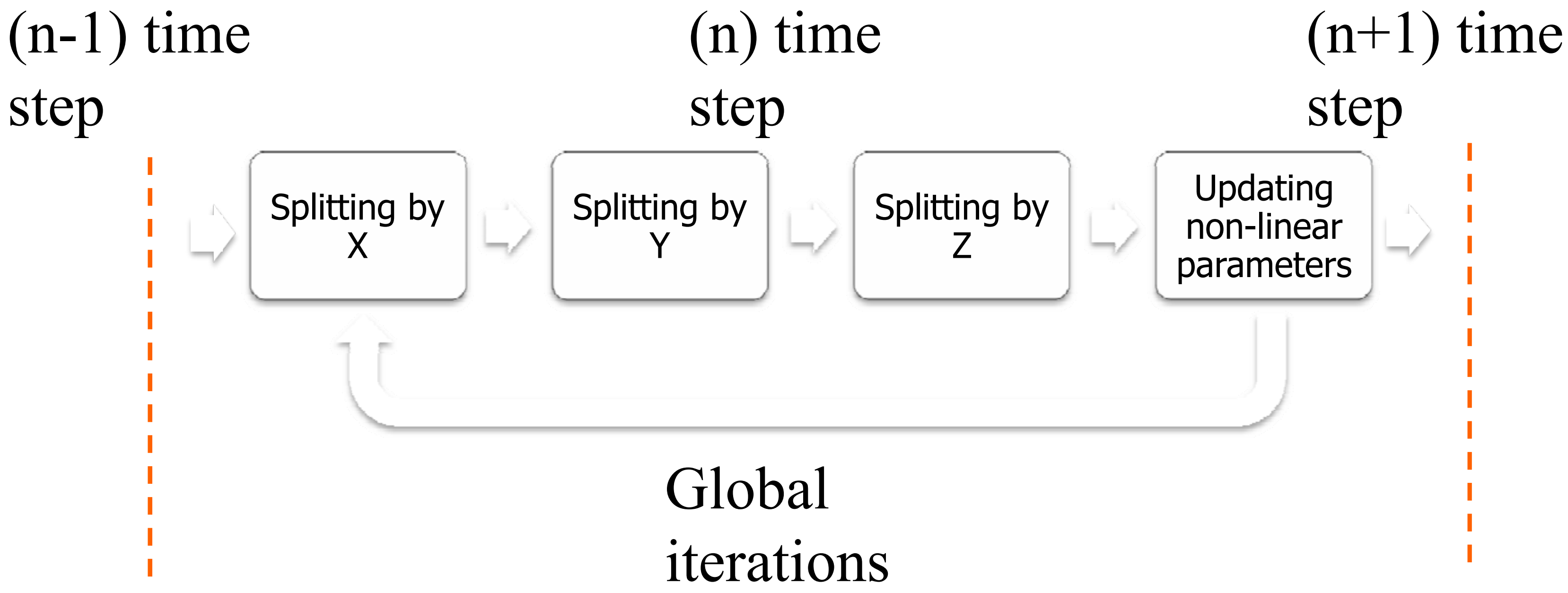
$$\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial y}=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial y^2}\right)$$

Z

$$\frac{\partial u}{\partial t}+w\frac{\partial u}{\partial z}=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial z^2}\right)$$

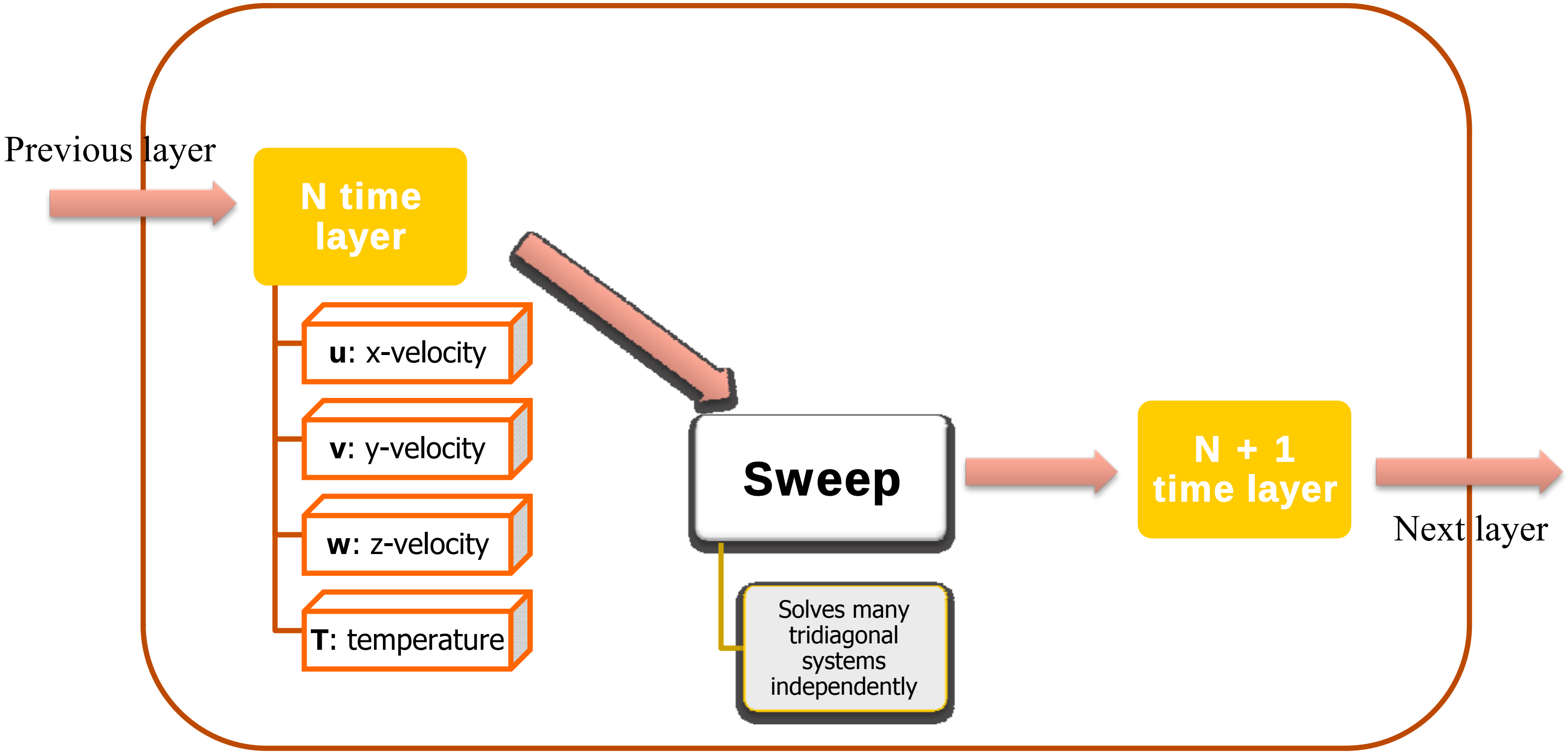
- + итерации по нелинейности

Шаг по времени



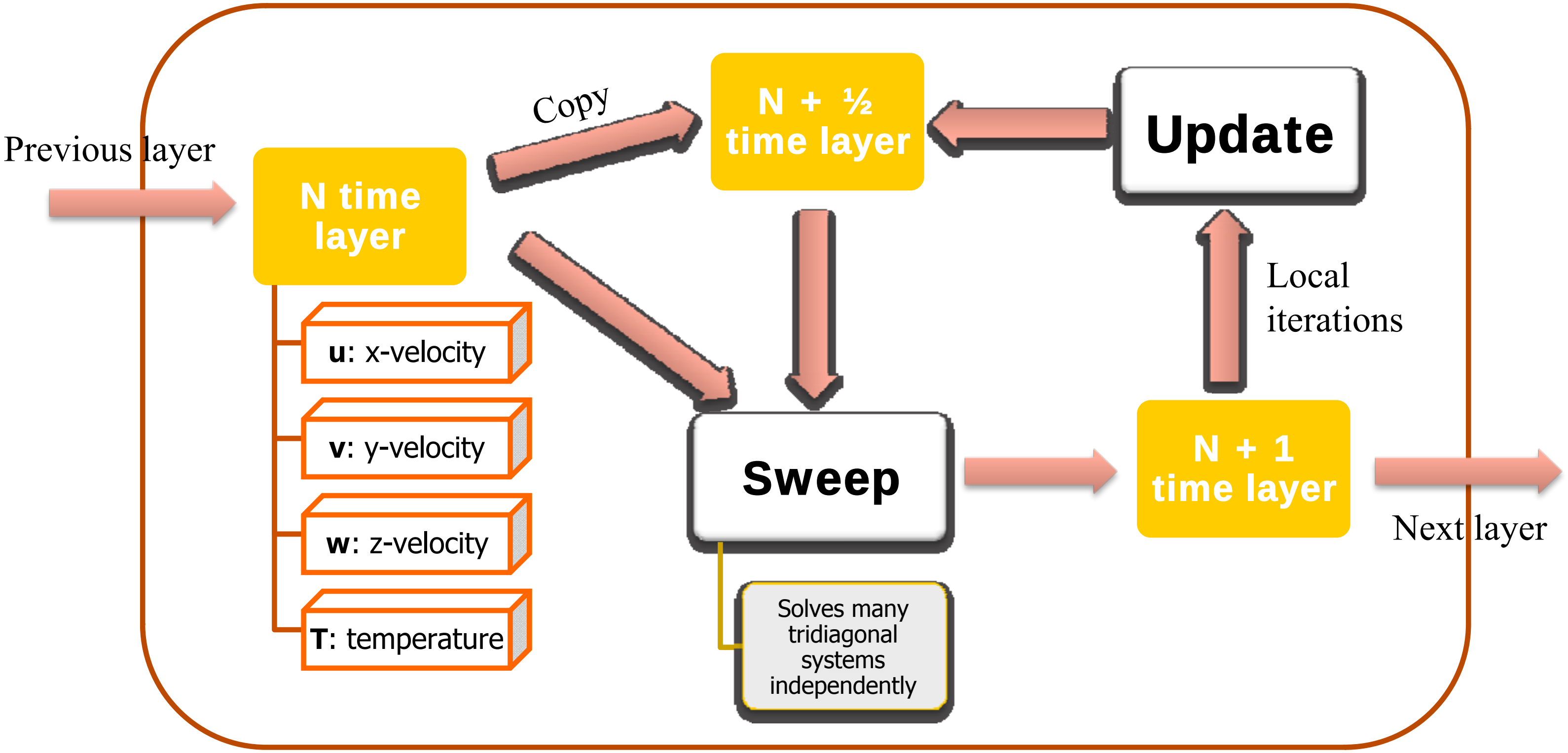
Дробный шаг

⌘Линейное PDEs



Дробный шаг

- Нелинейное PDEs



Стадии алгоритма

- Решение большого количества трехдиагональных СЛАУ
- Вычисление диссипации в каждой ячейке сетки
- Обновление нелинейных параметров

Особенности метода

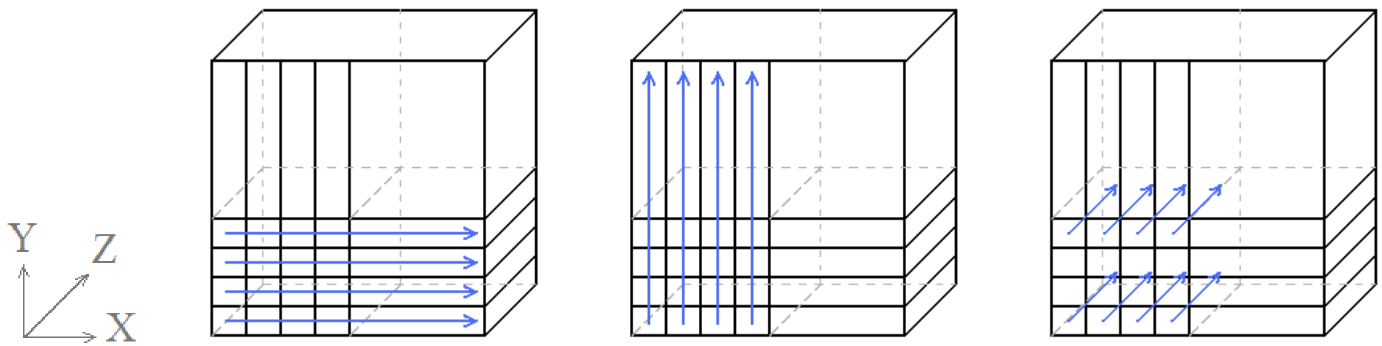
- Большой объем обрабатываемых данных
- Высокая арифметическая интенсивность
- Легко параллелится

Реализация на CUDA

- Все данные хранятся в памяти GPU
  - 4 скалярных 3D массива для каждой переменной (u, v, w, T)
  - 3 дополнительных 3D массива
- ~1GB для сетки 192^3 в double

Решение трехдиагональных СЛАУ

- Каждая нить решает ровно одну трехдиагональную СЛАУ
  - На каждом шаге N^2 независимых систем



Расщепление X Расщепление Y Расщепление Z

Метод прогонки

- Необходимо 2 дополнительных массива
  - хранение: локальная память
- Прямой ход
  - вычисление a[i], b[i]
- Обратный ход
  - x[i] = a[i+1] * x[i+1] + b[i+1]

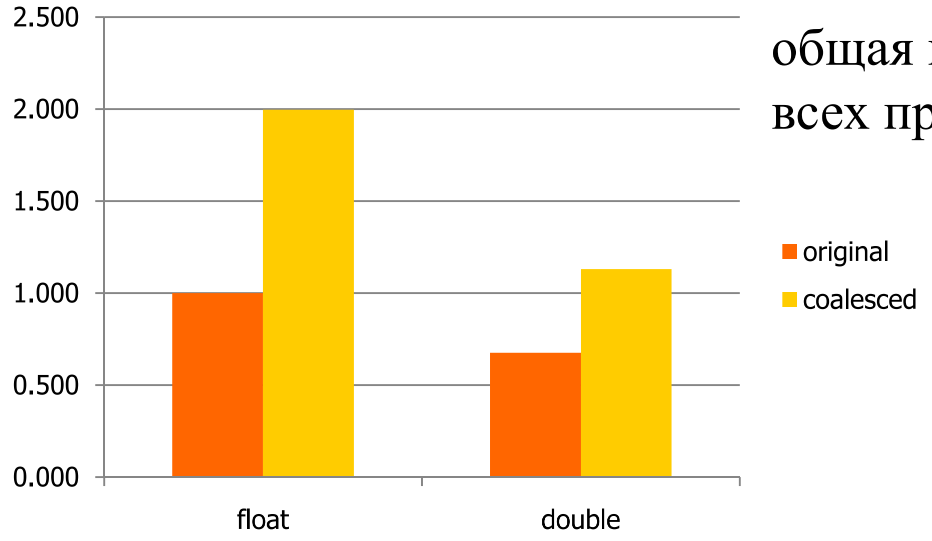
Проблемы реализации

- Каждая нить последовательно читает и пишет столбец 3D массива
  - Коэффициенты и правая часть

- Y, Z – прогонки coalesced
- X – прогонка uncoalesced!

Оптимизация прогонки

- X – прогонка
  - Транспонируем входные массивы и запускаем Y-прогонку



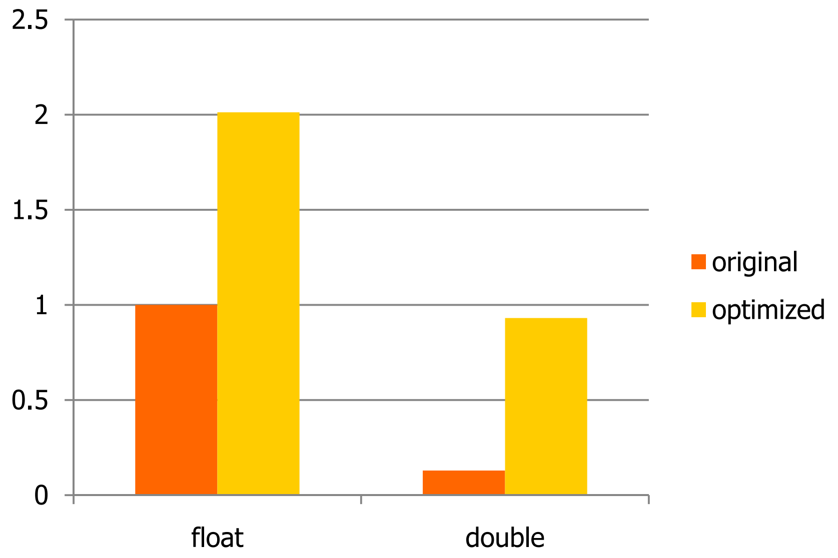
общая производительность всех прогонок

Расчет диссипации

- Расчет частных производных по трем направлениям
  - Локальный доступ к памяти
- Каждая нить обрабатывает столбец данных
  - Переиспользование расчитанных производных
- Использование разделяемой памяти (?)

Оптимизация диссипации

- Рефакторинг кода
  - Предварительный расчет некоторых констант, избавление от лишних if
- C++ шаблоны для X, Y, Z-диссипации
  - Уменьшение числа регистров, нет лишних обращений к памяти



Нелинейные итерации

- Необходимо посчитать полусумму двух 3D массивов
- Каждая нить считает сразу для столбца данных – N^2 нитей
  - Все чтения/записи coalesced
- Оптимальный выбор размера блока
- **80%** от пиковой пропускной способности на Tesla C1060

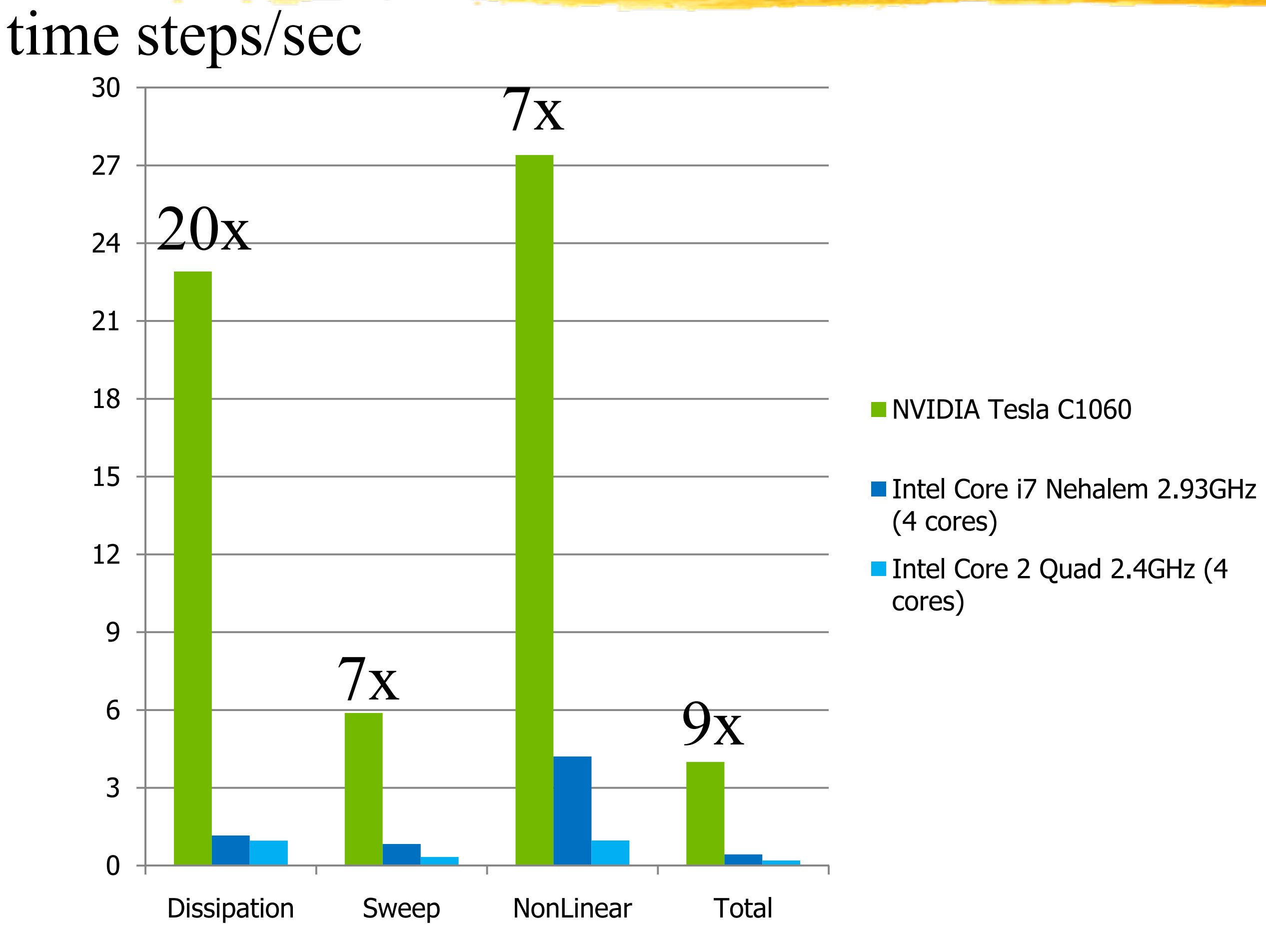
Пример кода

```
// boundary conditions
switch (dir)
{
            case X: case X_as_Y: bc_x0(…); break;
            case Y: bc_y0(…); break;
            case Z: bc_z0(…); break;
}
a[1] = - c1 / c2;
u_next[base_idx] = f_i / c2;

// forward trace of sweep
int idx = base_idx;
int idx_prev;
for (int k = 1; k < n; k++)
{
            idx_prev = idx;
            idx += p.stride;

            double c = v_temp[idx];
            c1 = p.m_c13 * c - p.h;
            c2 = p.m_c2;
            c3 = - p.m_c13 * c - p.h;

            double q = (c3 * a[k] + c2);
            double t = 1 / q;
            a[k+1] = - c1 * t;
            u_next[idx] = (f[idx] - c3 * u_next[idx_prev]) * t;
}
```

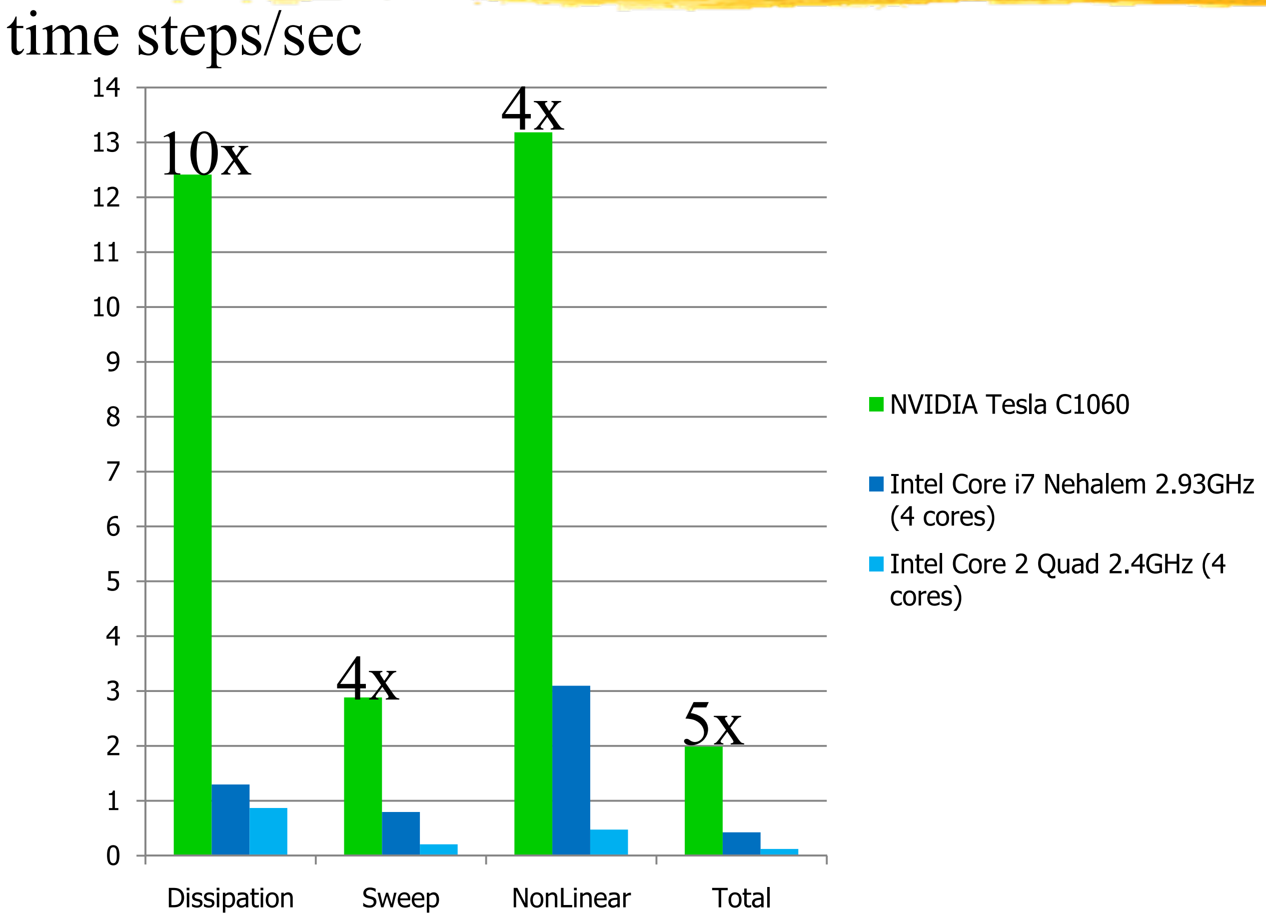
Тест производительности

- Тестовые данные
  - Сетка 128^3, 192^3
  - 8 нелинейных итераций
- Сравнение CPU и GPU
  - Абсолютное время работы

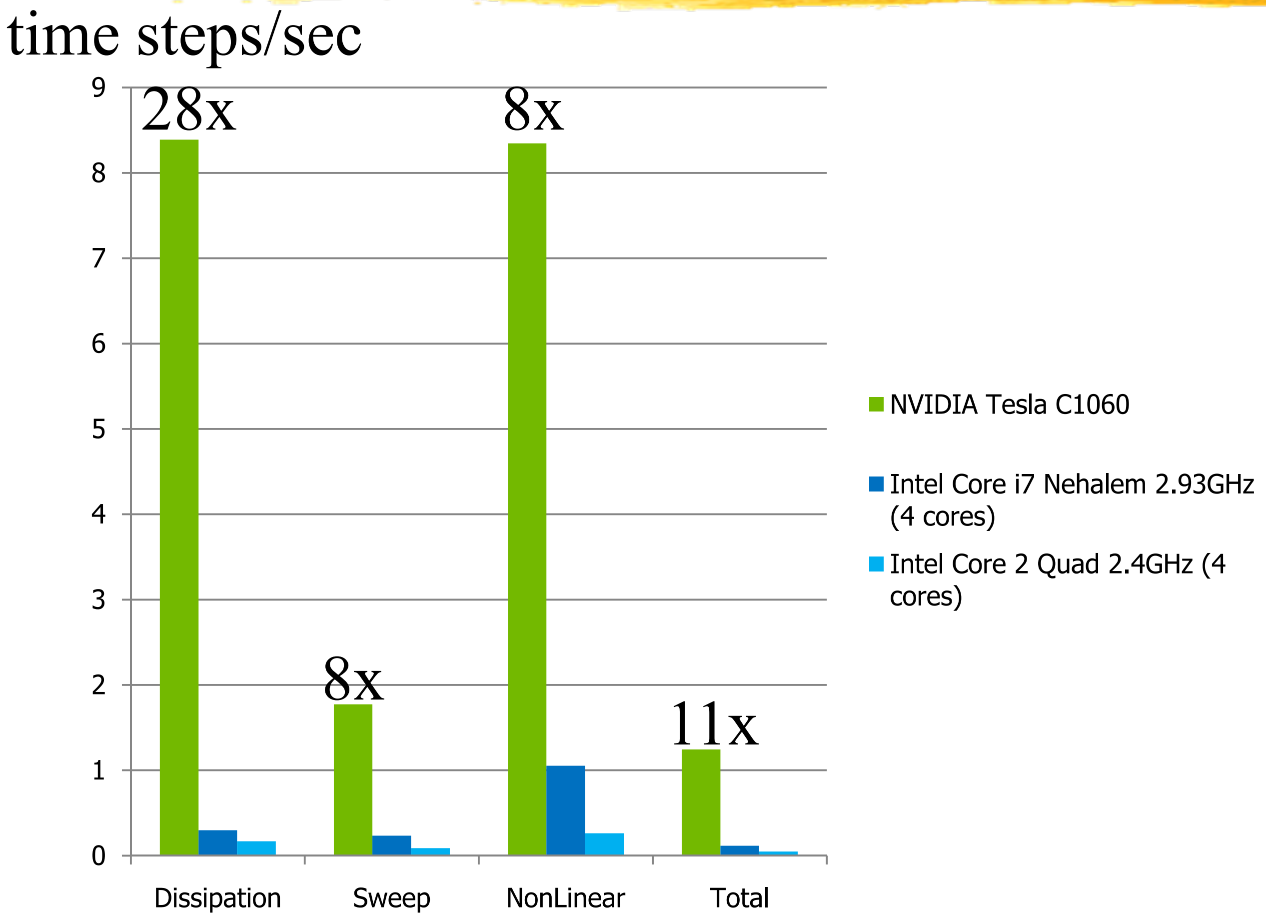
Тест – 128 - float



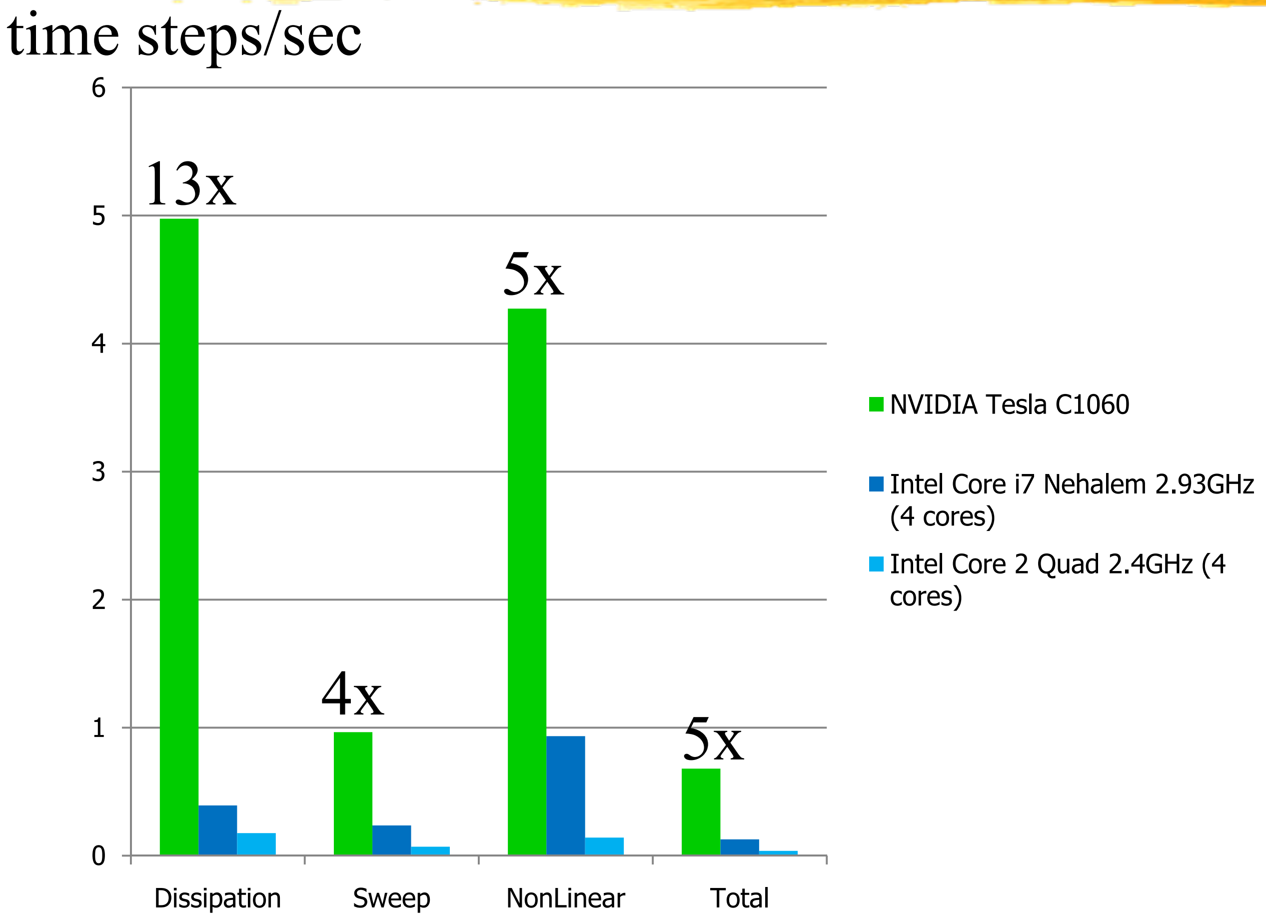
Тест – 128 - double



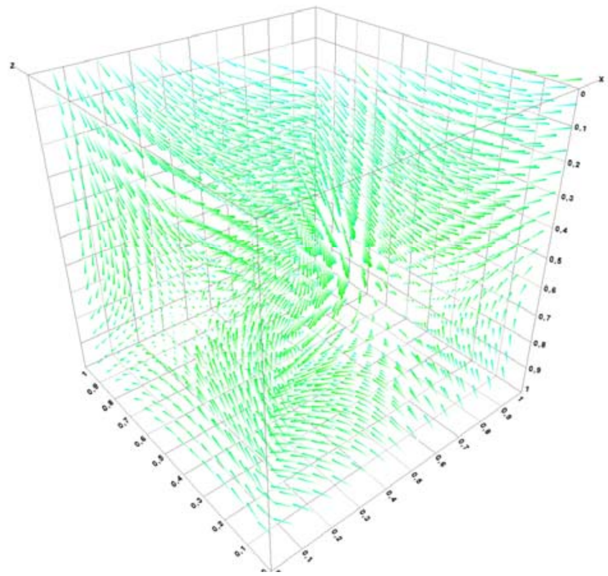
Тест – 192 - float



Тест – 192 - double

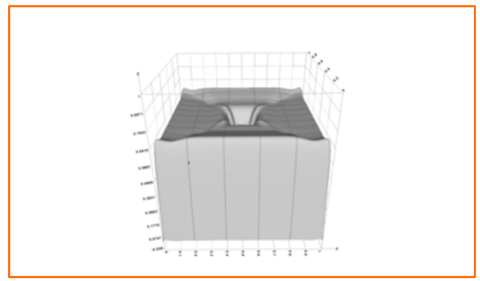


Визуализация

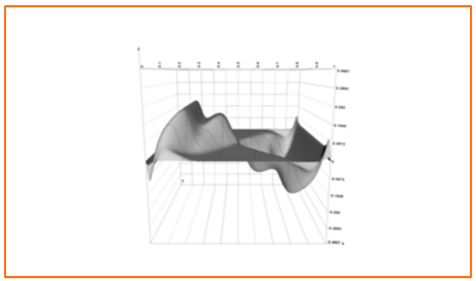


Векторное поле скоростей

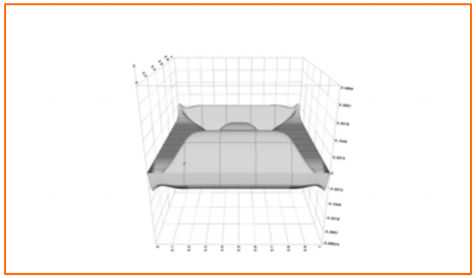
u



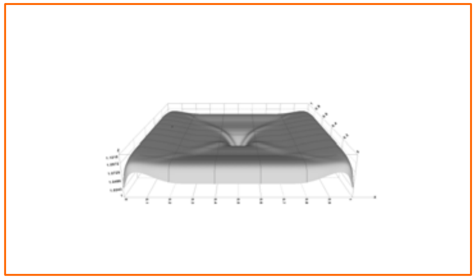
v



w



T



Срез вдоль X

Выводы

- Высокая эффективность Tesla в задачах аэро-гидродинамики
- Программная модель CUDA – удобное средство утилизации ресурсов GPU
- Применение GPU открывает новые возможности для исследования

Вопросы