


Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Кафедра высокопроизводительных вычислений и системного программирования

Лаборатория ITLab

Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики
Кафедра высокопроизводительных вычислений и системного программирования
Лаборатория ITLab

Введение в анализ производительности и оптимизацию программ

Ускорение решения уравнений Максвелла методом FDTD с использованием вычислений в смешанной точности

Арисова А.Н., Волокитин В.Д., Мееров И.Б.

LOBACHEVSKY UNIVERSITY

Содержание

- Цель лекции
- Постановка задачи
- Метод конечных разностей во временной области
- Алгоритм
- Результаты вычислительных экспериментов
- Выводы

Цель лекции

- Продемонстрировать возможности использования вычислений в смешанной точности для ускорения расчетов в вычислительных науках

Что вы знаете про вычисления в смешанной точности?

Уравнения Максвелла

- **Уравнения Максвелла** — система уравнений в дифференциальной форме, описывающих электромагнитное поле, его связь с электрическими зарядами и токами в вакууме и в сплошных средах.

$$\frac{\partial \boldsymbol{D}}{\partial t} = \nabla \times \boldsymbol{H} - 4\pi \boldsymbol{j} \quad (1)$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E} \quad (2)$$

$$\nabla \cdot \boldsymbol{D} = 4\pi\rho \quad (3)$$

$$\nabla \cdot \boldsymbol{B} = 0 \quad (4)$$

+

Уравнение непрерывности

$$\frac{\partial\rho}{\partial t} + \nabla \cdot \mathbf{j} = 0$$

+

Материальные уравнения

(Среда моделирования-вакуум)

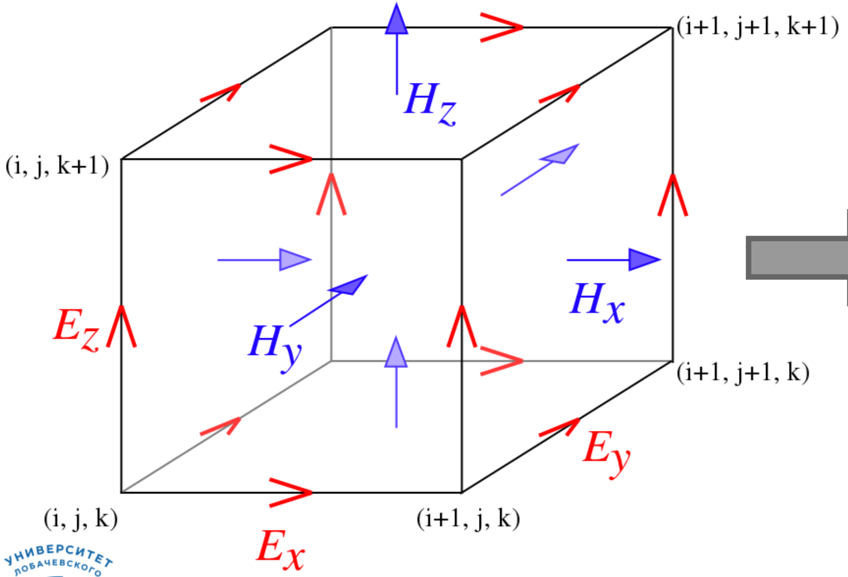
$$\mathbf{D} = \mathbf{E} \text{ и } \mathbf{B} = \mathbf{H}$$

$$\begin{cases} \dfrac{\partial \boldsymbol{E}}{\partial t} = \nabla \times \boldsymbol{B} - 4\pi \boldsymbol{j} \\ \dfrac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E} \end{cases}$$

Метод конечных разностей во временной области (FDTD)

Метод FDTD основан на **дискретизации уравнений Максвелла** с помощью разностных **аппроксимаций** пространственных и временных **частных производных**

Сетка **Yee**



$$\left.\frac{\partial E_z{}^n}{\partial y}\right|_{i,j,k} = \frac{E_z{}^n|_{i,j+1,k} - E_z{}^n|_{i,j,k}}{dy}$$

$$\left.\frac{\partial E_z{}^n}{\partial t}\right|_{i,j,k} = \frac{E_z{}^{n+1}|_{i,j,k} - E_z{}^n|_{i,j,k}}{dt}$$

Поглощающие граничные условия (PML)

Для моделирования свободного пространства при решении неограниченных электромагнитных задач методом конечных разностей во временной области должны использоваться особые поглощающие граничные условия.

$$\begin{cases} \nabla \times \boldsymbol{B} = 4\pi * \boldsymbol{J} + \dfrac{\partial \boldsymbol{E}}{\partial t} \\ \nabla \times \boldsymbol{E} = -\dfrac{\partial \boldsymbol{B}}{\partial t} \end{cases}$$

$$+$$

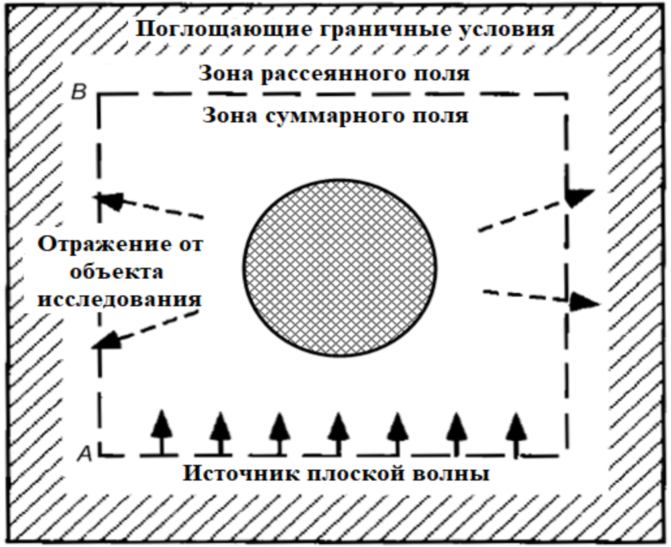
$$\boldsymbol{J} = \tilde{\sigma} * \boldsymbol{E}$$

$$\Downarrow$$

$$\begin{cases} \nabla \times \boldsymbol{B} = \sigma * \boldsymbol{E} + \dfrac{\partial \boldsymbol{E}}{\partial t} \\ -\nabla \times \boldsymbol{E} = \sigma^* * \boldsymbol{B} + \dfrac{\partial \boldsymbol{B}}{\partial t} \end{cases}$$

Расщепление компонент полей на две составляющие:
$E_{xy}, E_{xz}, E_{yx}, E_{yz}, E_{zx}, E_{zy}$
$B_{xy}, B_{xz}, B_{yx}, B_{yz}, B_{zx}, B_{zy}$

Экспоненциальный шаг по времени
$$\Delta t \to \frac{1 - e^{-\sigma \Delta t}}{\sigma}$$



Адаптировано из Berenger, J. *Computational Physics*, 1994, pp. 185-200.

Идея

- **Основная идея** – выполнять **в половинной точности** наиболее **вычислительно-трудоемкие расчеты**, а **одинарную или двойную точность** использовать для **подсчета** и **уменьшения ошибок вычислений**, добившись ускорения вычислений при сохранении приемлемой точности

- **Вопросы**:
  - Какие этапы вычислений нужно выполнять в половинной точности?
  - Какие возникают проблемы и как их решать?
  - Какой точности удастся достигнуть?
  - На какое ускорение рассчитывать, при условии что в современном оборудовании происходит расширение поддержки типов данных пониженной точности?

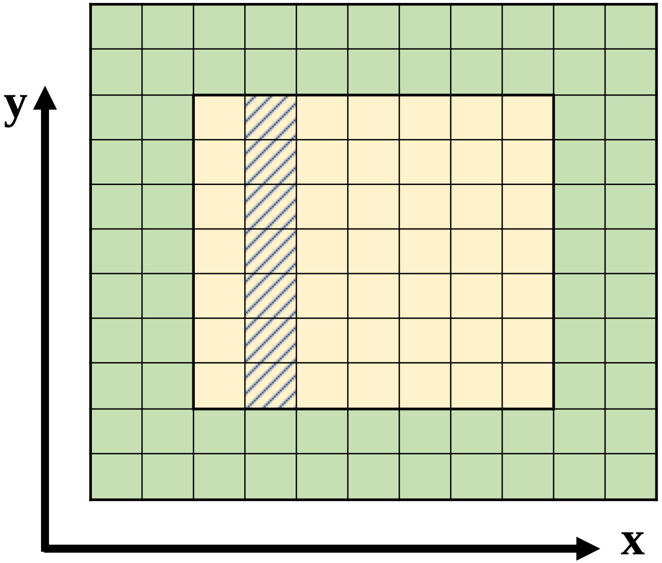
Реализации метода FDTD с PML

- Реализаций метода FDTD существует довольно много, но вопрос об использовании смешанной точности остается открытым.
- Среди них есть *бесплатные пакеты* программного обеспечения с открытым исходным кодом, например:
  - **Meep** – высокоэффективный пакет FDTD, разработанный в Массачусетском технологическом институте;
  - **gprMax** – программное обеспечение из Эдинбургского университета;;
  - **OpenEMS** – бесплатный решатель, разработанный в Университете Дуйсбург-Эссен, использует расширенный интерфейс Matlab (или Octave) для определения опций FDTD.

Реализации метода FDTD с PML

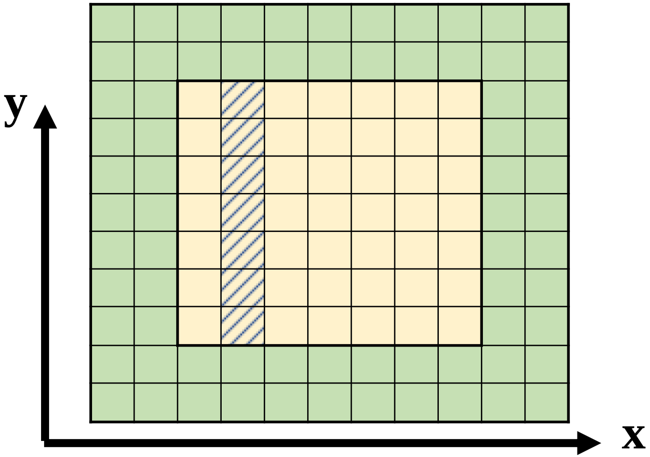
- Также разработаны *коммерческие пакеты* программного обеспечения для моделирования метода FDTD:
  - **Ansys Lumerical FDTD** – продукт компании Ansys для моделирования нанофотонных устройств, процессов и материалов с помощью точно настроенной реализации метода FDTD;
  - **XFdtd® 3D EM Simulation Software [14]** – инновационный полнофункциональный решатель для электромагнитного моделирования компании Remcom.
- Разработки ННГУ и ИПФ РАН для суперкомпьютерного моделирования лазерной плазмы методом частиц в ячейках:
  – PICADOR (C++)
  – Hi-Chi (Python + C++)
- В данной лекции рассматривается отдельный код – реализация FDTD и PML

Выбор режима точности (типов данных)



- Выделим три области
- Введем два шаблонных параметра:
- **ftype** – тип данных для вычислений в основной расчетной области (**желтые ячейки**)
- **ftypePML** – тип данных для вычислений в области гран. условий PML (**зеленые ячейки**)
- **ftypeGenerator** – тип данных для генератора полей, *всегда в двойной точности* (серые ячейки). **ftypeGenerator = double**
- Хотелось бы все считать в половинной точности, но не получится добиться приемлемых результатов
- Нужно экспериментировать!

Выбор режима точности (типов данных)



- **ftype** – тип данных для вычислений в основной расчетной области (**желтые ячейки**)
- **ftypePML** – тип данных для вычислений в области гран. условий PML (**зеленые ячейки**)
- **ftypeGenerator** – тип данных для генератора полей, *всегда в двойной точности* (серые ячейки). **ftypeGenerator = double**

- Было изучено, какие вычисления можно проводить в пониженной точности, определено, что оказывает существенное влияние на точность результатов
- Выяснилось, что порядок операций критически важен при расчетах в пониженной точности
- Наибольшие потери при суммировании: **алгоритм Кэхэна** может помочь

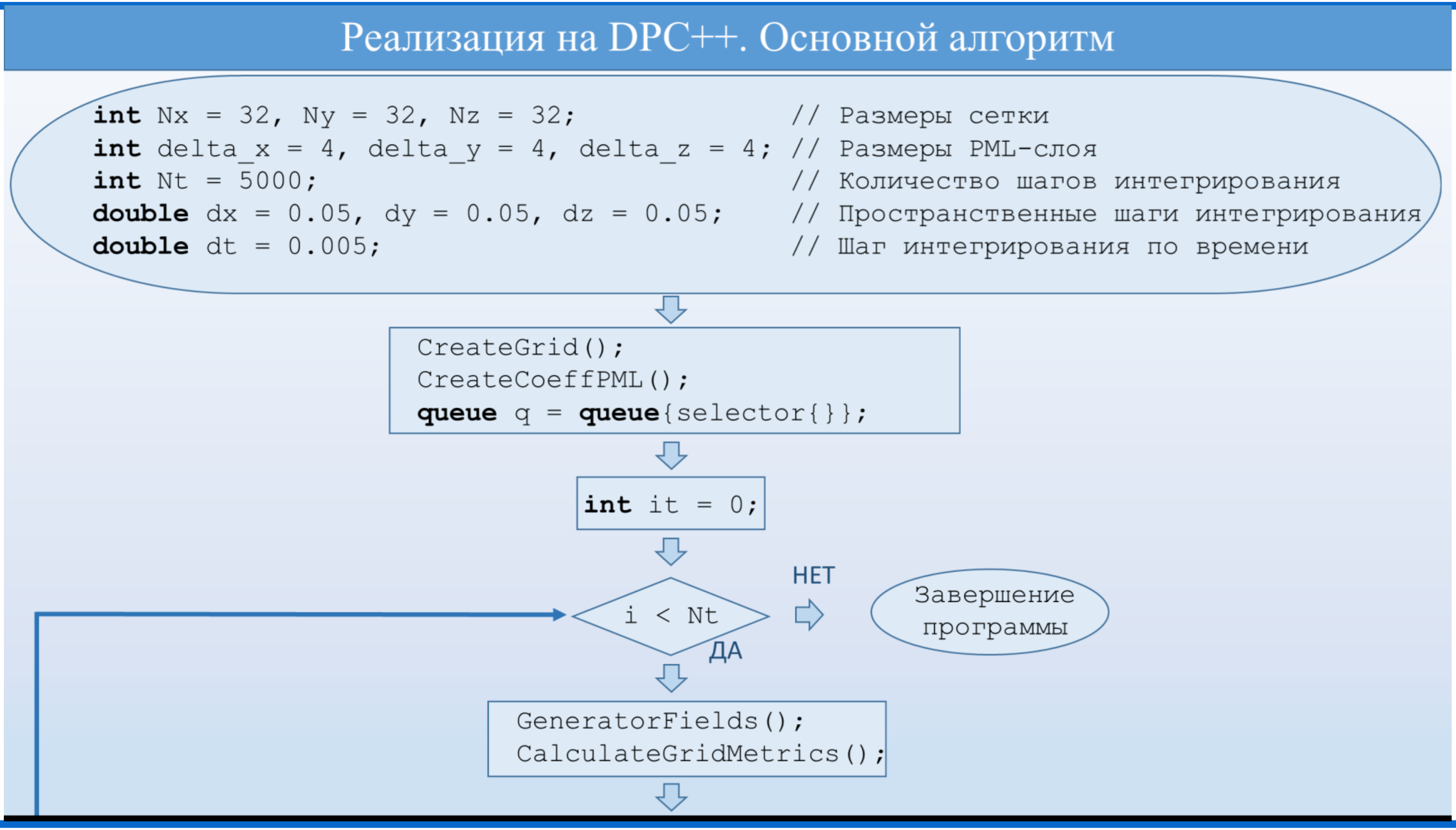
Суммирование с компенсацией (алгоритм Кэхэна)

- Уильям (Вольф) Кэхэн – лауреат премии Тьюринга, один из авторов концепции представления чисел с плавающей запятой в стандарте IEEE-754
- **Алгоритм суммирования с компенсацией:**

```
ftype KahanSum(ftype input[], size_t n) {
    ftype sum, c, y, t; sum = c = 0.0;
    for (size_t i = 0; i < n; i++) {
        y = input[i] - c;
        t = sum + y;
        c = (t - sum) - y;
        sum = t;
    }
  return sum;
}
```

Фото из Wikipedia

Программная реализация на Data Parallel C++ (SYCL)



Программная реализация на Data Parallel C++ (SYCL)

Реализация на DPC++. Основной алгоритм

```
q.submit([&](handler& h) {
   h.parallel_for(sycl::range<3>(Nx+2delta_x,Ny+2delta_y,Nz+2delta_z), [=](auto index){
         int i = index[0] + 1;
         int j = index[1] + 1;
         int k = index[2] + 1;
         if (i>delta_x&&i<Nx+delta_x && j>delta_y&&j<Ny+delta_y&&k>delta_z&&k<Nz+delta_z)
              UpdateElectric();
         else
              UpdateElectricPML();
 });}).wait_and_throw();
```

```
q.submit([&](handler& h) {
   h.parallel_for(sycl::range<3>(Nx+2delta_x,Ny+2delta_y,Nz+2delta_z), [=](auto index){
         int i = index[0] + 1;
         int j = index[1] + 1;
         int k = index[2] + 1;
         if (i>delta_x&&i<Nx+delta_x && j>delta_y&&j<Ny+delta_y &&k>delta_z&&k<Nz+delta_z)
              UpdateMagnetic();
         else
              UpdateMagneticPML();
 });}).wait_and_throw();
```

```
it++;
```

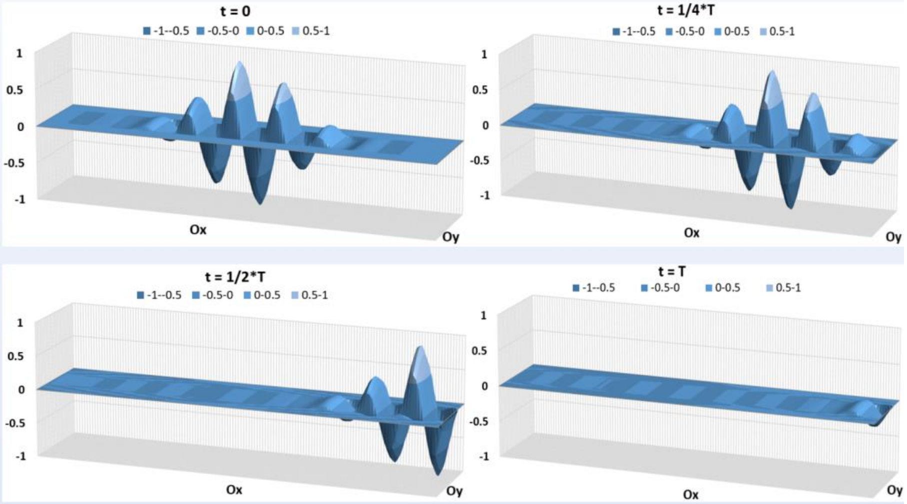
Модельная задача

Тестовая задача с заданием **источников** в виде **граничных условий**

- **Начальные условия**
  $\boldsymbol{E} = (0,0,0)^T, \quad \boldsymbol{B} = (0,0,0)^T$
- Расчетная область $[0, 4\pi] \times [0, 16\pi] \times [0, 16\pi]$
- $N_x = 256, N_y = N_z = 128, \Delta t = 1/512$
- Граничные условия PML, толщина слоя $\delta = 8, \sigma = 46.5, n = 4$
- **Задание источников**

$$E_y^n(i^*, j, k) \mathrel{+}= 4\pi\Delta t * J_y^{electric}(i^*, j, k, n)$$

$$B_z^n(i^*, j, k) \mathrel{+}= 4\pi\Delta t * J_z^{magnetic}(i^*, j, k, n)$$



$$J_y^{electric}(i^*, j, k, n) = exp\left(-\frac{(n\Delta t - t_0)^2}{t_p^2}\right) exp\left(-\frac{(y_j - y^0)^2}{a_t^2}\right) exp\left(-\frac{(z_j - z^0)^2}{a_t^2}\right) \cdot \sin\big(\omega(n\Delta t - t_0) - k(x_{i^*} + \bar{x})\big)$$

$$J_z^{magnetic}(i^*, j, k, n) = exp\left(-\frac{\left((n\Delta t + \bar{t}) - t_0\right)^2}{t_p^2}\right) exp\left(-\frac{\left(y_j - y^0\right)^2}{a_t^2}\right) exp\left(-\frac{\left(z_j - z^0\right)^2}{a_t^2}\right) \cdot \sin\big(\omega(n\Delta t + \bar{t} - t_0) - kx_{i^*}\big)$$

Методика оценки корректности

- **Вопрос**: как оценить корректность, что использовать в качестве метрики?
- Что принять за эталон?
  - *Очевидный ответ*: можно использовать результаты, полученные при расчетах в двойной точности
- Какую метрику использовать?
  - Абсолютная ошибка
  - Относительная ошибка
- Каковы недостатки этих метрик в данном случае?
- В какой точке считать?

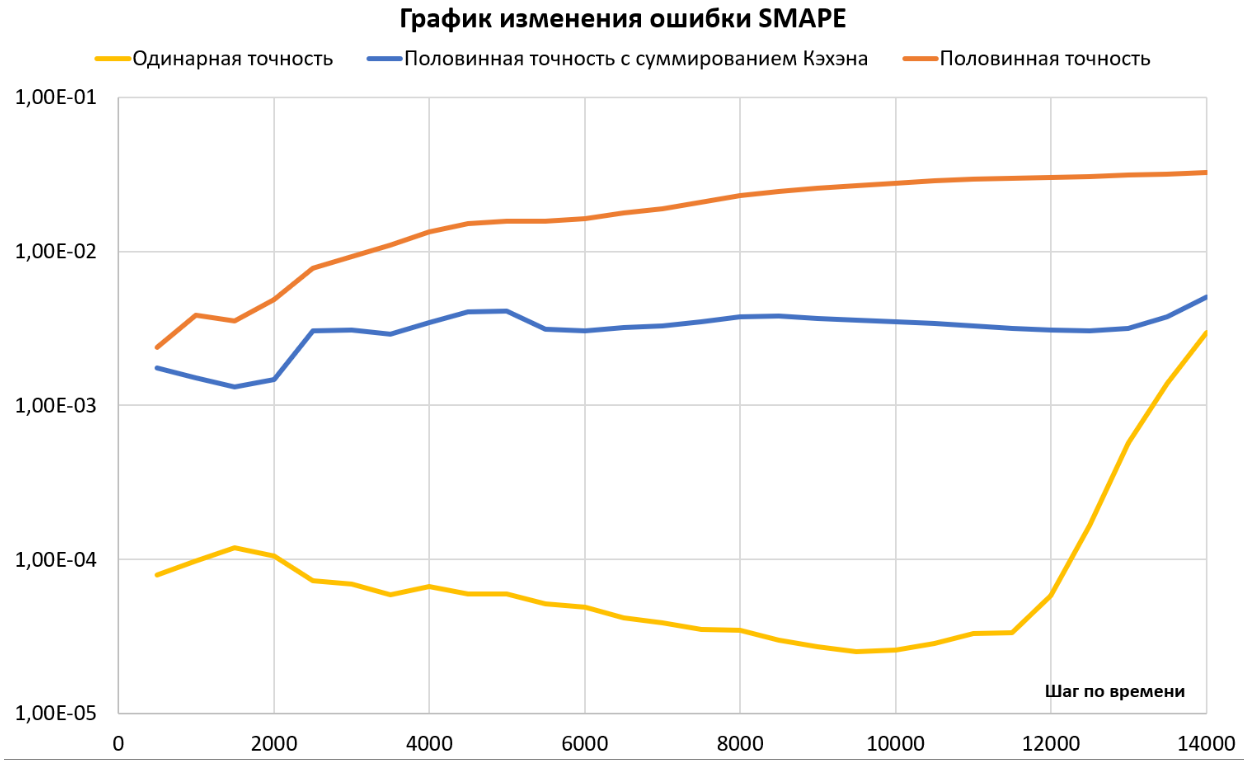
Методика оценки корректности

- **Используем следующую метрику (SMAPE):**

$$Error_{\mathrm{SMAPE}} = \frac{1}{6 \cdot N_x \cdot N_y \cdot N_z} * \sum_{i,j,k} \sum_{x \in \{E_x, E_y, E_z, B_x, B_y, B_z\}} \frac{2 \cdot |x^* - x|}{|x^*| + |x|}$$

- **Каковы преимущества такого подхода?**

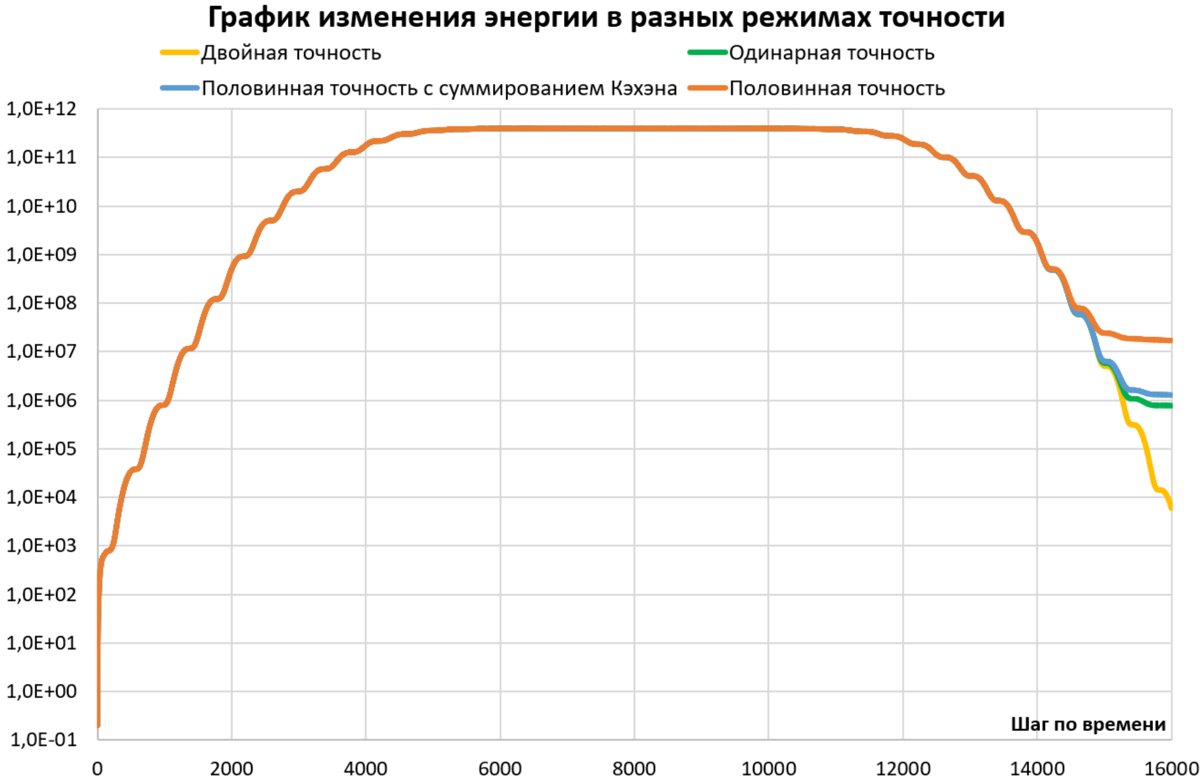
Результаты (сравнение с эталоном)



- **Половинный режим точности с суммированием Кэхэна** стабильно работает с ошибкой $\mathbf{3 \cdot 10^{-3}}$

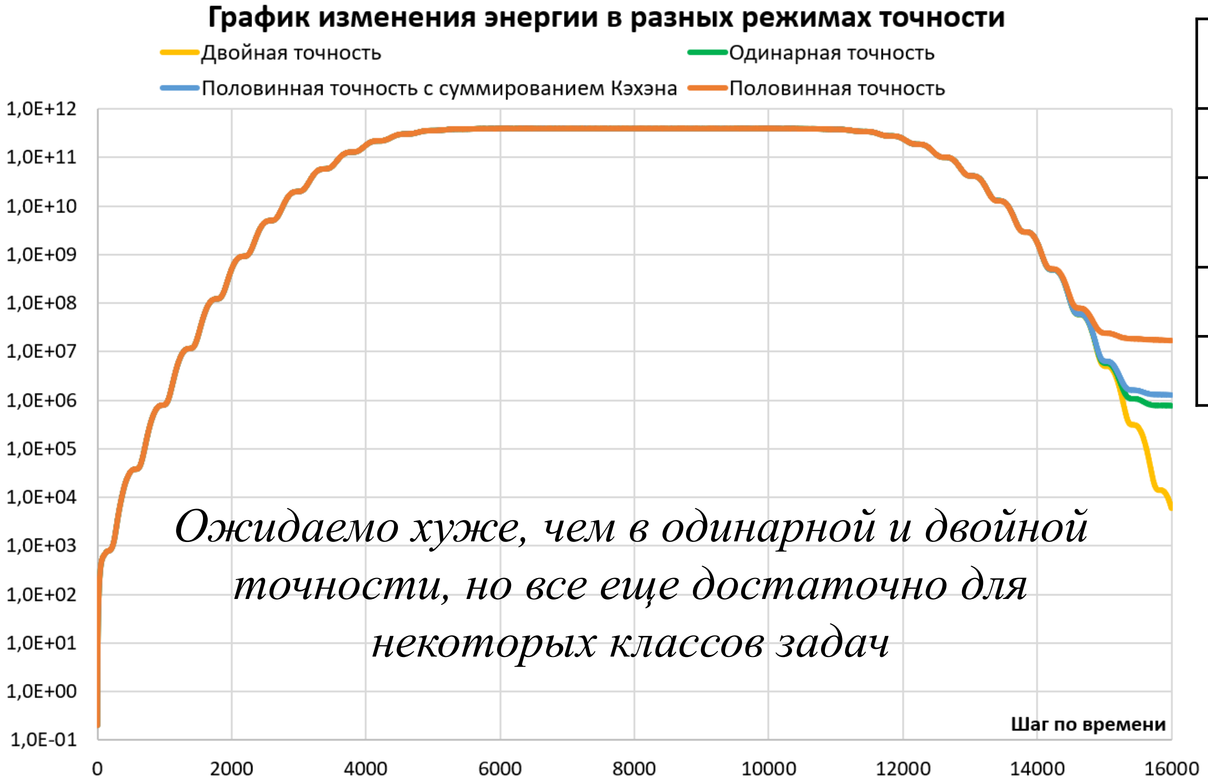
- В случае без суммирования Кэхэна ошибка на порядок хуже.

Результаты (уровень поглощения)



- Численная характеристика корректности работы метода – **уровень поглощения**, который вычислялся как отношение энергии поля после полного поглощения и энергии исходного поля

Результаты (уровень поглощения)



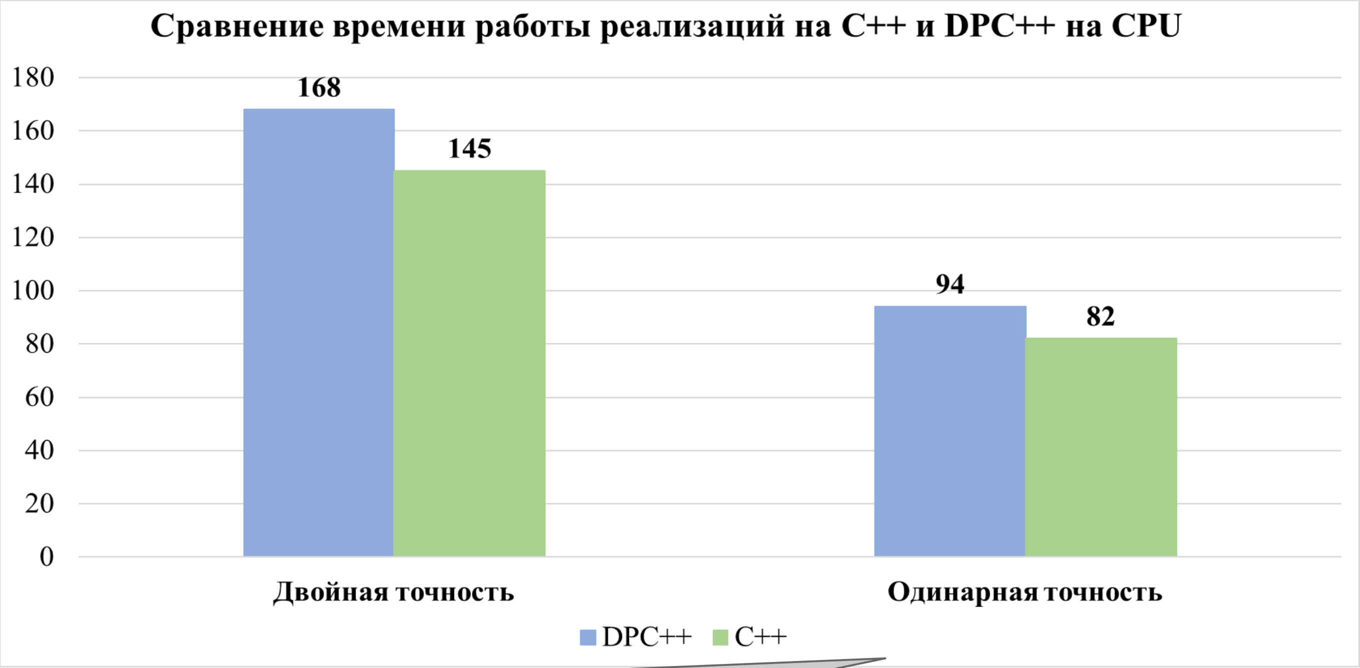
Ожидаемо хуже, чем в одинарной и двойной точности, но все еще достаточно для некоторых классов задач

Режим точности	Уровень поглощения
Половинная точность	2.81138e-05
Половинная точность с суммированием Кэхэна	1.05449e-06
Одинарная точность	3.00066e-08
Двойная точность	4.09413e-10

$$R = \frac{Energy_N}{Energy_0}$$

$$Energy = \sum_{i,j,k} \left(\left|\vec{E}\right|^2 + \left|\vec{B}\right|^2 \right)$$

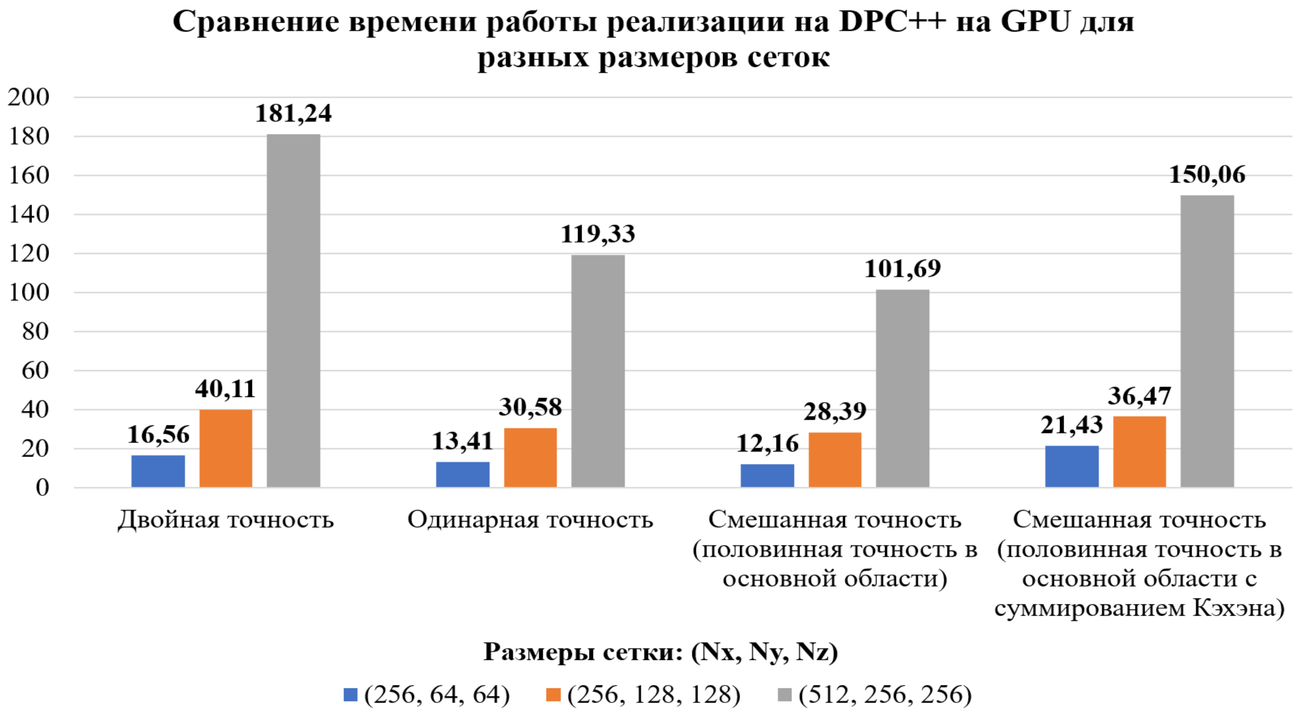
Анализ производительности на CPU



- Размеры сетки:
  $$\boldsymbol{Nx = 256,}$$
  $$\boldsymbol{Ny = 128,}$$
  $$\boldsymbol{Nz = 128}$$
- Время приведено в **секундах**
- Реализация на DPC++ добавляет около **15% накладных расходов**

Запуски проводились на сервисе Intel DevCloud на узле суперкомпьютера Endeavour с 2-мя процессорами **Intel Xeon Platinum 8260L** (*192 ГБ оперативной памяти, Red Hat 4.8.5, компилятором Intel C++ и Intel DPC++).*

Анализ производительности на GPU



Запуски проводились на СК “Лобачевский” на **GPU NVIDIA A100** (*6912 потоковых процессоров, 40 GB HBM2e 1555GB/s*)

- **Чем больше размер сетки, тем больше ускорение**
- **Много времени занимает расчет PML-слоя**. Это приводит к большому число синхронизаций
- Было выявлено что **основной ограничивающий фактор это память**
- **Основные вычислительные ядра** в разных точностях **соотносились** ожидаемо, **с двойным ускорением**

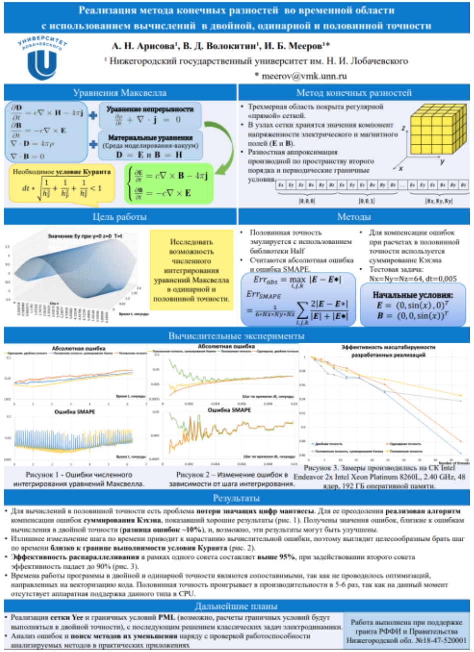
Выводы

- Продемонстрирован подход к организации вычислений в смешанной точности при интегрировании уравнений Максвелла методом конечных разностей во временной области с граничными условиями PML
- Показано, что при должном внимании к минимизации ошибок, связанных с недостаточной разрешающей способностью типа данных half, удается достигать коэффициента отражения от PML порядка 10^{-6}
- Запуски на GPU NVIDIA A100 показали, что использование смешанного режима точности (половинная + одинарная) позволяет получить ускорение ~1.17 раза относительно одинарной точности и ~вдвое экономить память GPU
- Суммирование Кэхэна на порядок повышает точность, но дорого с вычислительной точки зрения
- Возможные пути дальнейшего ускорения – улучшение работы с памятью (аналог блочного алгоритма)

Литература

Подробнее о данной задаче и методе ее решения:

- Арисова А.Н., Волокитин В.Д., Мееров И.Б. Реализация метода конечных разностей во временной области с использованием вычислений в двойной, одинарной и половинной точности // Суперкомпьютерные дни в России. 21–22 сентября 2020 г., Москва
- Арисова А.Н., Волокитин В.Д., Ефименко Е.С., Мееров И.Б. Реализация метода FDTD с граничными условиями PML с использованием вычислений в смешанной точности// Суперкомпьютерные дни в России. 27–28 сентября 2021 г., Москва
- Арисова А.Н., Волокитин В.Д., Ефименко Е.С., Мееров И.Б. Ускорение решения уравнений Максвелла методом FDTD с использованием вычислений в смешанной точности//Параллельные вычислительные технологии. Санкт-Петербург, 28–30 марта 2023 г.



Контакты

Нижегородский государственный университет

http://www.unn.ru

Институт информационных технологий, математики и механики

http://www.itmm.unn.ru

Кафедра высокопроизводительных вычислений и системного программирования