

Разработка кроссплатформенной библиотеки для моделирования процессов термодинамики средствами Metal

В.С. Тарасов

В.А. Ушаков, e-mail: aurel.vu@ya.ru

Воронежский государственный университет

***Аннотация.** На основе метода конечных элементов разработана программа для GPU с использованием фреймворка Metal.*

***Ключевые слова:** GPU, термодинамика, метод конечных элементов, C++, HPC, MacOS.*

Введение

Решение задач термодинамики является крайне актуальным на сегодняшний день и затрагивает большое количество дисциплин: физиология, химия, биология, физика. Постоянные новшества и изменения требуют постоянного развития, например, в области анализа химических и физических процессов, а также проектирования различных двигателей автомобильной и авиационной техники. Сегодня применение термодинамики к необратимым процессам является главной задачей, и имеются большие успехи в построении теории, по объему и важности сравнимой с термодинамикой обратимых процессов.

Большое количество профессионалов на сегодняшний день используют в своей повседневной работе MacBook с операционной системой MacOS. В 2021 году появились версии с увеличенным количеством графических ядер, а наличие объединенной памяти до 64 Гб потенциально делает эти устройства крайне перспективными для моделирования физических процессов и обучения моделей глубоких нейронных сетей. Планшеты Apple iPad комплектуются процессорами, аналогичными младшим ноутбукам в линейке MacBook, имея при этом малый вес, возможность отсоединения клавиатуры и сенсорный экран, что делает из них прекрасные мобильные станции, которыми могут пользоваться большое количество инженеров и других профессионалов, которым требуется дополнительный софт для расчетов. Однако на данных моделях устройств отсутствует возможность применения CUDA, вместо нее предлагается использование проприетарного фреймворка Metal.

Данная работа посвящена проблеме разработки программного обеспечения, позволяющего производить моделирование процессов термодинамики на устройствах Mac с применением графических ускорителей. В работе рассматривается моделирование процесса,

реализация программы с применением фреймворка Metal и замер его производительности.

1. Разработка моделей

В качестве задачи для моделирования взята задача термодинамики, в которой у нас имеет конечное пространство с теплопроводящими стенками, нижняя часть которого непрерывно нагревается. Вид функции f определяется решением дифференциального уравнения теплопроводности Фурье, которое для однородного изотропного тела имеет вид:

$$\partial T / \partial t = \alpha \Delta T + qV / c\rho, \quad (1)$$

где qV — количество теплоты, выделяемое внутренним источником теплоты в единице объема тела за единицу времени, c — удельная теплоемкость тела, ρ — плотность тела, $\Delta = \partial^2 / (\partial x^2) + \partial^2 / (\partial y^2) + \partial^2 / (\partial z^2)$ — оператор Лапласа, α — коэффициент теплопроводности (характеризует скорость выравнивания температуры в неравномерно нагретом теле). Данное уравнение является уравнением параболического типа.

Для стационарной теплопроводности ($\partial T / \partial t = 0$) уравнение (1) примет вид:

$$\alpha \Delta T + qV / c\rho = 0 \quad (2)$$

В отсутствии внутренних источников теплоты ($qV = 0$):

$$\Delta T = 0 \quad (3)$$

В одномерном стационарном случае уравнение Фурье можно использовать в следующем виде:

$$dQ = -\alpha dT / dx dS dt, \quad (4)$$

где dQ — количество теплоты, которое переносится через площадку dS в направлении нормали x к этой площадке в сторону убывания температуры за время dt , K — коэффициент теплопроводности.

Метод конечных объемов (МКО) — это численный метод решения дифференциальных уравнений с частными производными и интегральных уравнений, возникающих при решении задач прикладной физики. МКО широко используется для решения задач механики деформируемого твёрдого тела, гидродинамики, теплообмена, электродинамики и топологической оптимизации.

Для производной по времени используется метод Эйлера. Это явная схема по времени, где берутся искомые с предыдущего шага. На первом шаге алгоритма задаются как начальные условия.

$$dT / dt = (T_c - T_c^0) / \text{delt}, \quad (5)$$

где delt – численное значение шага по времени, индекс C означает значение T в текущем элементе C , индекс 0 означает значение T на предыдущем шаге (то есть уже известное)

Четыре грани составляют периметр элемента и по ним производится интегрирование. Для каждого элемента в результате получается дискретный аналог исходного дифференциального уравнения.

Произведем подстановку в уравнение:

$$(T_c - T_c^0)/\text{delt} + \text{fluxE} * T_E - \text{fluxE} * T_c - \text{fluxW} * T_c + \text{fluxW} * T_w + \text{fluxN} * T_N - \text{fluxN} * T_c - \text{fluxS} * T_c + \text{fluxS} * T_S = 0 \quad (6)$$

$$(T_c - T_c^0)/\text{delt} + \text{fluxE} * T_E + \text{fluxW} * T_w + \text{fluxN} * T_N + \text{fluxS} * T_S - (\text{fluxE} - \text{fluxW} - \text{fluxN} - \text{fluxS}) * T_c = 0 \quad (7)$$

$$T_c^{NEW} = T_c + \text{delt} * (\text{fluxC} * T_c - (\text{fluxE} * T_E + \text{fluxW} * T_w + \text{fluxN} * T_N + \text{fluxS} * T_S)), \quad (8)$$

где $\text{fluxC} = (\text{fluxE} + \text{fluxW} + \text{fluxN} + \text{fluxS})$

При написании программы было принято решение использовать подход с обменом указателей на буферы. Это позволяет хранить в системе только два буфера, не производя выделение памяти под каждый и не инициализируя каждый с нуля после каждого шага по времени. На рисунке 1 визуальное приведено использование различных переменных для задания исходных условий. Из них следует отметить следующие:

- n – размер сетки по горизонтали,
- m – размер сетки по вертикали,
- h – шаг сетки,
- T_b – температура на нижней грани,
- T_0 – температура на боковой грани,
- k_L – коэффициент теплопроводности нижней грани,
- k_R – коэффициент теплопроводности верхней грани,
- k_U – коэффициент теплопроводности левой грани,
- k_D – коэффициент теплопроводности правой грани,
- k – коэффициент теплопроводности материала.

2. Фреймворк Metal

Metal предоставляет возможность произведения неграфических вычислений с помощью графических процессоров. Архитектурно он близок к иным инструментам для написания программ для SIMD устройств, таким как CUDA и OpenCL[2].

Изначально доступ к Metal производится при помощи ObjectiveC, однако компания Apple представила 8 июня 2022 года обертку на C++

под названием metal-cpp, что значительно облегчило работу с данным фреймворком.

Использование Metal близко к использованию Cuda. Если вычисление индекса потока в Cuda происходит следующим образом

Листинг 1

Вычисление индекса потока в сетке в CUDA

```
int i = blockIdx.x * blockDim.x + threadIdx.x;
```

то в Metal это происходит за счет иных констант:

Листинг 2

Вычисление индекса потока в сетке в Metal

```
int i = threadgroup_position_in_grid[0] *  
threads_per_threadgroup[0] +  
thread_position_in_threadgroup[0];
```

Притом данные константы передаются в конце в аргументах функции ядра

Листинг 3

Передача индексов в аргументах в Metal

```
uint2 threadgroup_position_in_grid [[  
threadgroup_position_in_grid ]],  
uint2 thread_position_in_threadgroup [[  
thread_position_in_threadgroup ]],  
uint2 threads_per_threadgroup      [[  
threads_per_threadgroup ]])
```

Одним из главных неудобств является необходимость передачи всех аргументов исключительно через буферы с указателями на них, что приводит к небезопасной работе с памятью и необходимости приведения значений от указателя к его значениям

Листинг 4

Передача целочисленных аргументов в Metal

```
kernel void solver_kernel(  
    device const float* T,  
    device float* T_new,  
    device const int* n_pointer,  
    device const int* m_pointer,  
    ...  
)  
{  
    int n = *n_pointer;  
    int m = *m_pointer;  
    ...  
}
```

Компания Apple продвигает разработку программного обеспечения при помощи их IDE XCode и их системой сборки, которые значительно уступает по возможностям остальным IDE. Для унификации использования с другими проектами был написан CMakeList.txt, обеспечивающий компиляцию шейдеров при сборке и подключение библиотеки Metal:

Листинг 5

Настройка системы сборки CMake для автоматического компилирования шейдеров и подключение Metal

```
...
add_custom_command(
    TARGET METAL_CPP
    COMMAND xcrun -sdk macosx metal -c
    ${CMAKE_CURRENT_SOURCE_DIR}/kernel.metal -o kernel.air
)

add_custom_command(
    TARGET METAL_CPP
    COMMAND xcrun -sdk macosx metallib kernel.air -o
    default.metallib
)

...
# Metal cpp library (linker)
target_link_libraries(METAL_CPP
    "-framework Metal"
    "-framework Foundation"
    "-framework QuartzCore")
```

3. Анализ результатов

Для моделирования системы были взяты следующие начальные условия: температура $T=240$ К для нижней грани и $K=0$ для левой, правой, верхней граней и для всех исходных точек. На рисунках 2 и 3 приведен пример распространения тепла при слабоизолированных боковых гранях, а именно для левой и правой граней были установлены значения коэффициента теплопроводности = 10. В результате мы можем наблюдать то, как тепло постепенно распространяется внутрь пространства, а также форма распространения имеет изгибы по краям, что говорит о корректном расчете потерь тепла через боковые грани.

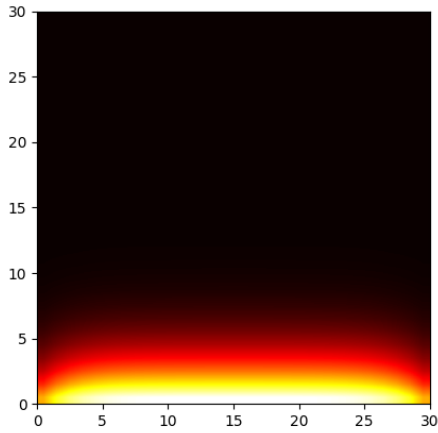


Рис. 1. Нагрев материала с коэффициентами теплопроводности боковых сторон 10 и момент времени $t=20$

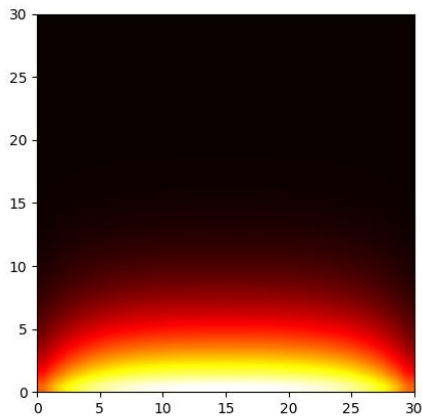


Рис. 2. Нагрев материала с коэффициентами теплопроводности боковых сторон 10 и момент времени $t=500$

Также была измерена скорость выполнения программы. Для последовательной версии данной программы моделирование

пространства 210 на 210 элементов заняло 15 с. Для процессора Intel i5 9400f, для программы с использованием Metal на процессора M1 Pro с 16 Gb объединенной памяти – 163 мс, что дает прирост в скорости в 92 раза. Однако аналогичная программа на CUDA с использованием Nvidia Geforce 1660 Ti дает скорость в 1 мс, что говорит о значительном отставании как технологии Apple, так и графической части их продукции от полноценных видеокарт.

Заключение

В результате проведенной работы была успешно разработана программа с использованием фреймворка Metal для неграфических вычислений с использованием GPU. Программа позволяет ускорить моделирование в 92 раза для компьютеров с операционной системой MacOS и планшетов iPad, не поддерживающих разработку на CUDA и OpenCL, что позволяет развить инженерное программное обеспечение на данных устройствах.

Литература

1. Патанкар, С. В. Численное решение задач теплопроводности и конвективного теплообмена при течении в каналах / С.В. Патанкар. – М.: МЭИ, 2019. – 312 с.

2. Performing Calculations on a GPU: Use Metal to find GPUs and perform calculations on them. [сайт]: электронный документ. – Режим доступа:

https://developer.apple.com/documentation/metal/performing_calculations_on_a_gpu