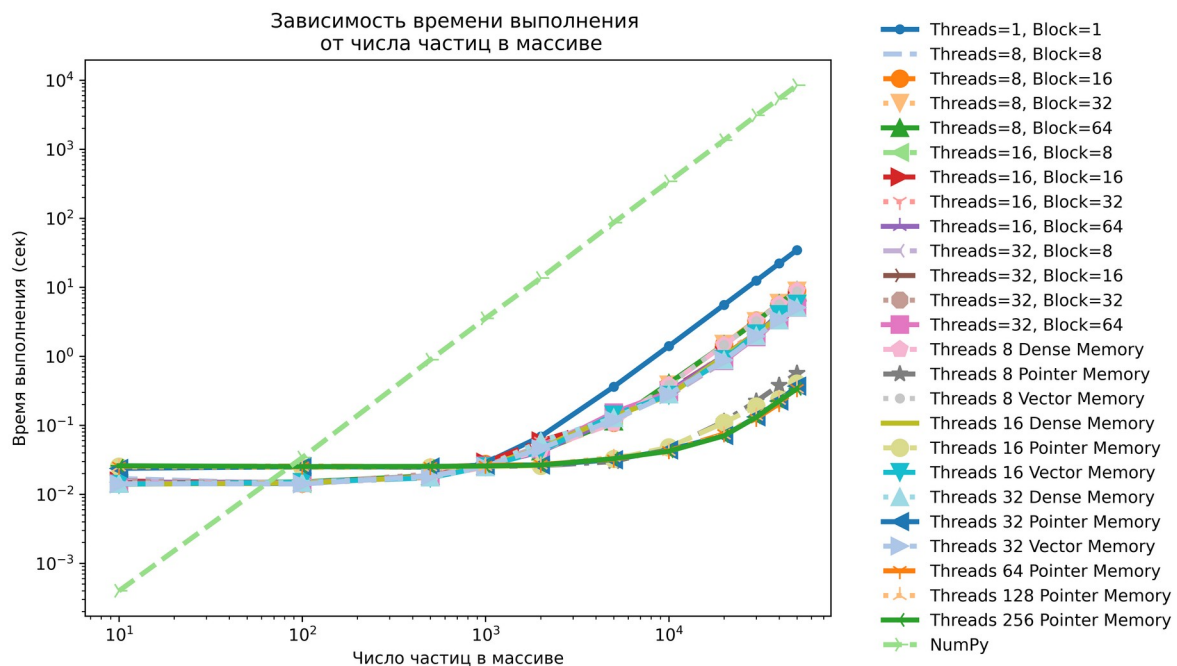
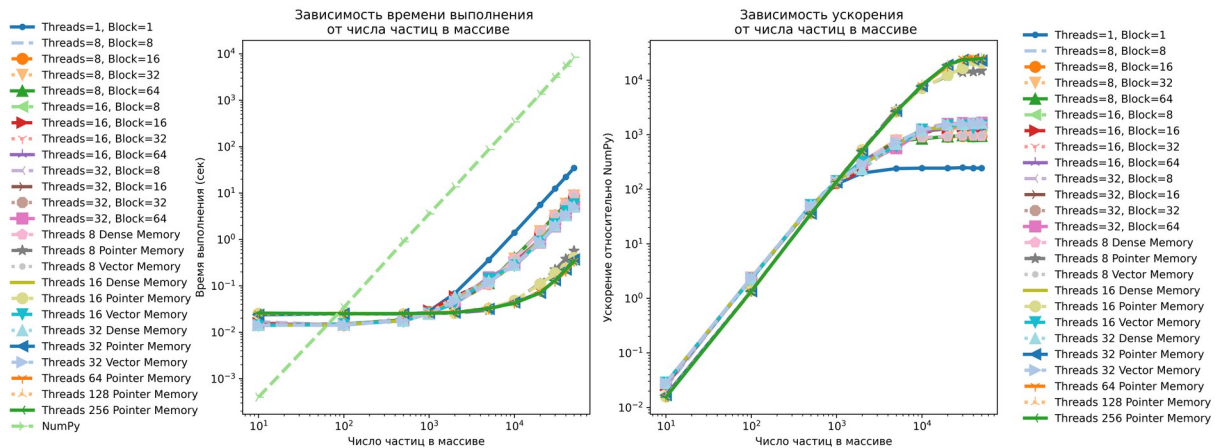
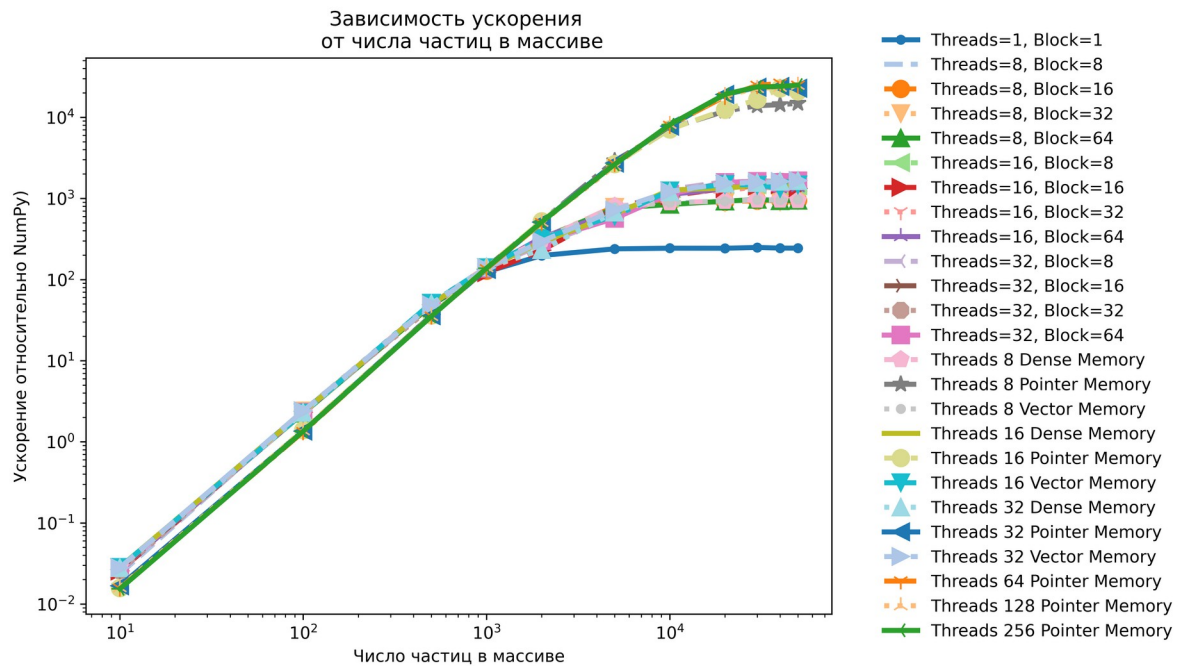


## Сравнение производительности NumPy и TaiChi Lang

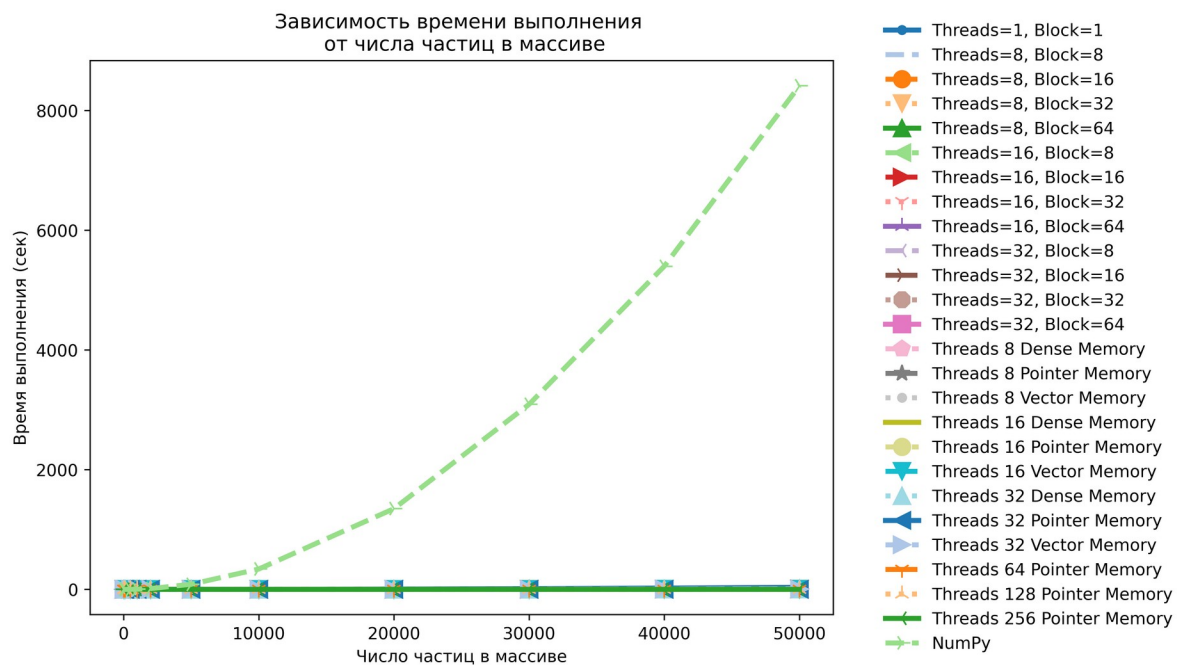
Для сравнения производительности использована задача вычисления силы в системе N тел. Сравнения производилось для систем тел, содержащих от 10 частиц до 50000 частиц. Вычисления проводились на компьютере с CPU: Intel i9-10900 (10 ядер, 20 потоков) частота 5.200GHz.

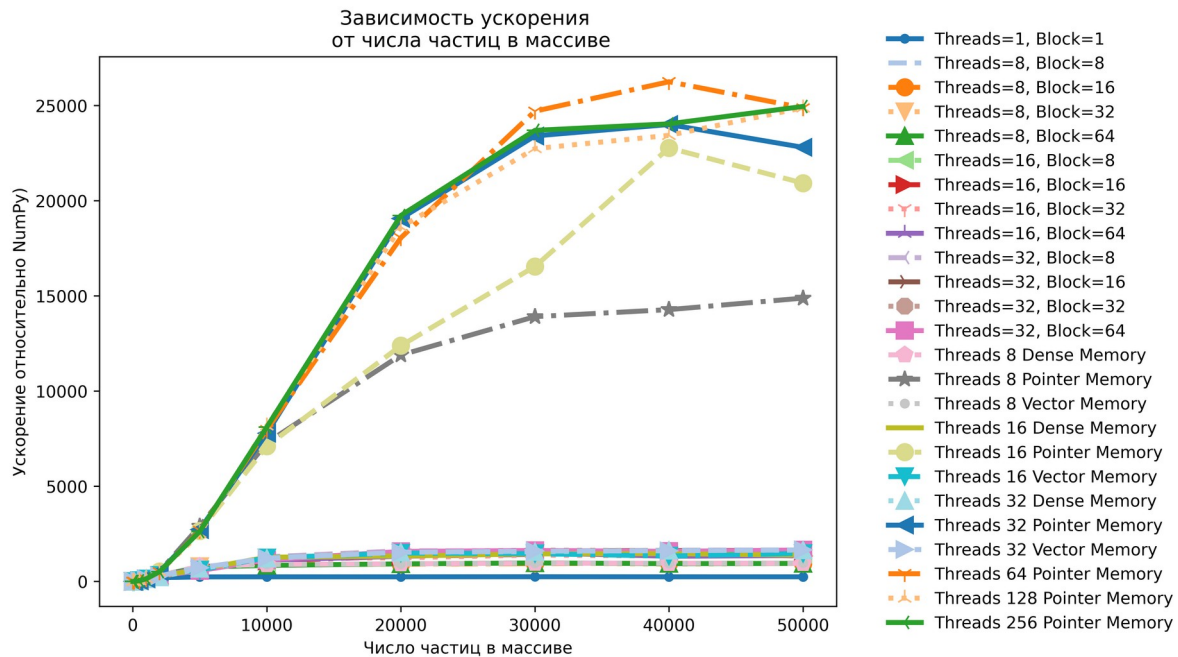
### Графики зависимости времени выполнения и ускорения относительно NumPy в логарифмических шкалах





### Графики зависимости времени выполнения и ускорения относительно NumPy в обычных шкалах





Проведенное дополнительное исследование для массивов большего размера показало, что для массивов больших размеров наибольшую производительность обеспечивает использование параметра выполнения **Pointer Memory**.

### Что такое CPU Pointer Memory в TaiChi Lang?

В **TaiChi Lang** параметр **CPU Pointer Memory** определяет, как данные, используемые в ядрах (kernels), взаимодействуют с памятью процессора. Это особенно важно, поскольку **TaiChi Lang** работает с низкоуровневыми структурами данных.

#### Основные принципы:

##### 1. Оптимизация доступа к памяти

- **Pointer Memory** напрямую связан с тем, как данные хранятся и адресуются в памяти. На CPU доступ к памяти часто является узким местом производительности, особенно при работе с большими массивами данных.
- Когда **Pointer Memory** правильно настроен, **TaiChi Lang**:
  - Минимизирует количество операций чтения/записи из памяти.
  - Использует кэши процессора (L1, L2, L3) максимально эффективно.
  - Уменьшает фрагментацию памяти и задержки при обращении к данным.

Это особенно важно для задач, где данные используются повторно (например, в циклах или итеративных алгоритмах).

##### 2. Локальность данных

- CPU работает быстрее, когда данные расположены в памяти последовательно (так называемая "пространственная локальность").
- **Pointer Memory** помогает **TaiChi Lang** организовать данные таким образом, чтобы они были выровнены и доступны блоками, что позволяет использовать SIMD-инструкции (Single Instruction, Multiple Data) процессора.
- SIMD-инструкции позволяют выполнять одну операцию сразу над несколькими элементами данных, что значительно ускоряет вычисления.

### 3. Эффективное использование указателей

- **Pointer Memory** оптимизирует работу с указателями, которые используются для доступа к данным. Это особенно важно для ядер (kernels), которые выполняются на CPU.
- Эффективные указатели позволяют:
  - Быстро перемещаться по массивам данных.
  - Избегать лишних вычислений при обращении к элементам массива.
  - Уменьшить количество инструкций, необходимых для доступа к данным.

### 4. Параллелизм на уровне CPU

- Современные CPU поддерживают многопоточность и векторизацию. **TaiChi Lang** автоматически распараллеливает вычисления, если это возможно.
- **Pointer Memory** помогает **TaiChi Lang** эффективно распределять данные между потоками, минимизируя конфликты при доступе к памяти.
- Это приводит к более равномерной загрузке всех ядер процессора и увеличению общей производительности.

### 5. Автоматическая оптимизация

- **TaiChi Lang** использует компилятор для генерации высоко оптимизированного машинного кода. **Pointer Memory** предоставляет компилятору дополнительную информацию о структуре данных, что позволяет:
  - Лучше предсказывать доступ к памяти.
  - Оптимизировать порядок выполнения инструкций.
  - Снижать количество промахов в кэше.

Для массивов от 10 частиц до 20000 частиц было проведено сравнение полученных значений вычисленной силы, со значениями полученными **NumPy**. Сравнения показало, что все вычисления с помощью **TaiChi Lang** совпадают с **NumPy** с точностью до 2 знака после запятой. Предположительно, это связано с тем, что **TaiChi Lang** использует параметр **fast\_math=True** по умолчанию, т. е. Вычисления проводятся на более низкой точности, из-за использования аппроксимированных функций вместо стандартных.

Исследования влияния параметров настройки поведения **TaiChi Lang** на скорость и точность будут продолжены на задаче о падении ткани на препятствие в форме шара.