



Московский Государственный Университет им. М.В. Ломоносова  
факультет Вычислительной математики и кибернетики  
кафедра Суперкомпьютеров и Квантовой информатики



# Курсовая работа

**Возможности пакета TaiChi для математического моделирования на современных вычислительных архитектурах**

Выполнила:  
студентка группы 538, Ши Хуэй

Руководитель:  
доцент кафедры СКИ, к.т.н., Русол А.В.

# ОБЪЕКТ ИССЛЕДОВАНИЯ

---

Ткань является легко деформируемым, достаточно прочным материалом, со сложной структурой.

Динамическое поведение которого определяется множеством факторов:

- небольшие упругие деформации при растяжении нитей
- большие деформации при изгибе нитей
- эффекты демпфирования
- граничные условия
- отклик на столкновения

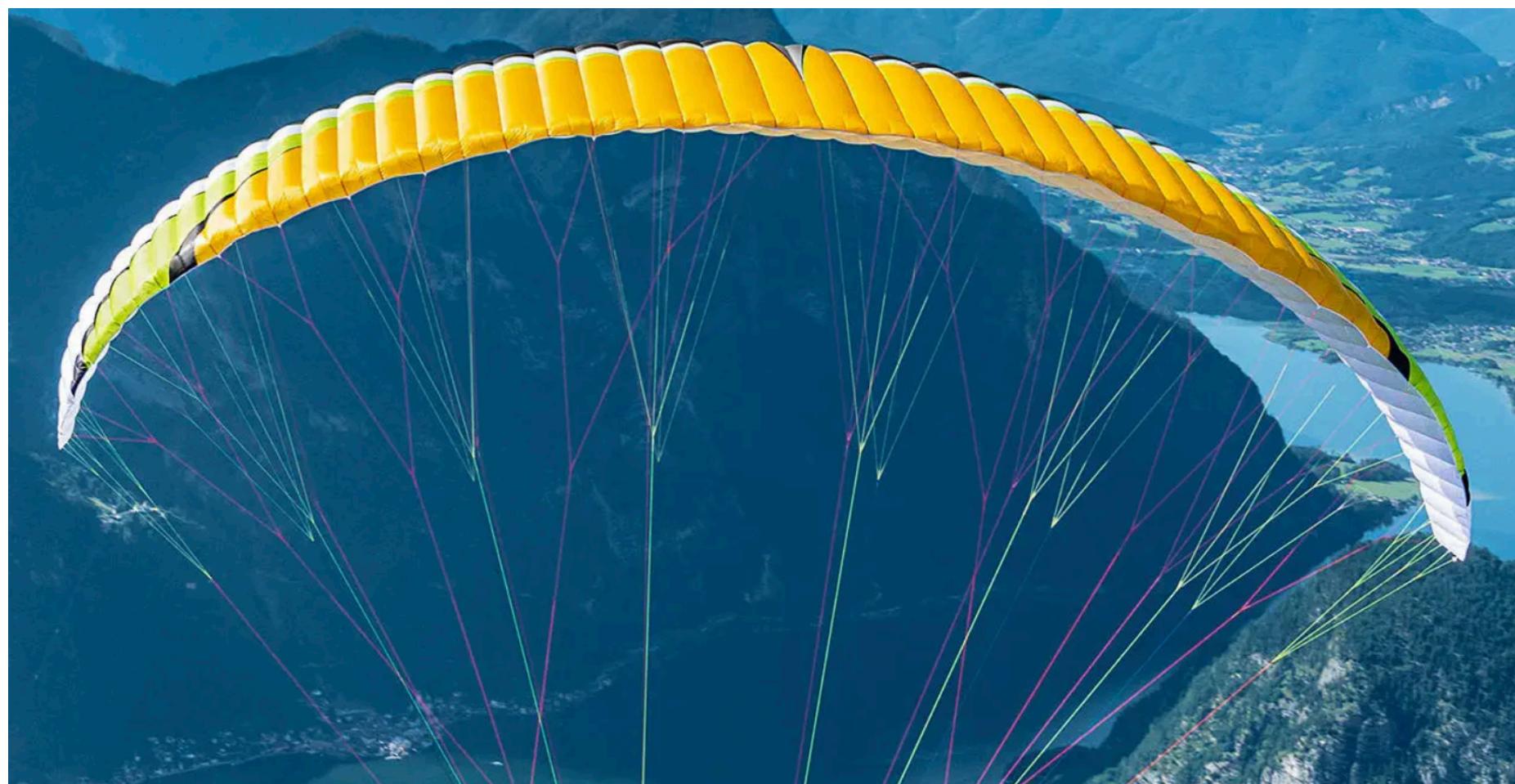
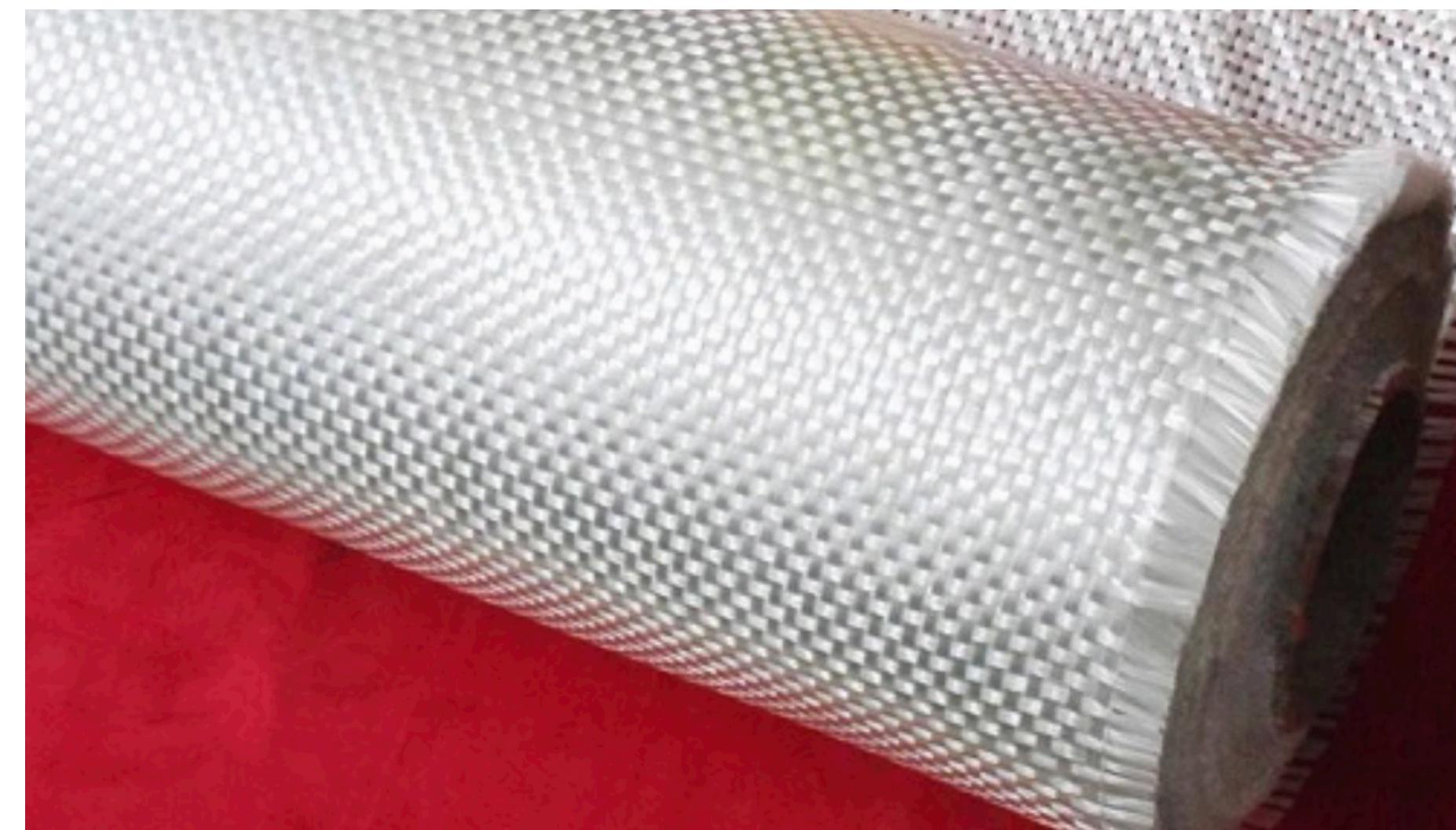
Основные технические применения:

- гибкие покрытия
- парашюты, парапланы, оболочки летательных аппаратов
- основа композиционных материалов



# АКТУАЛЬНОСТЬ

---

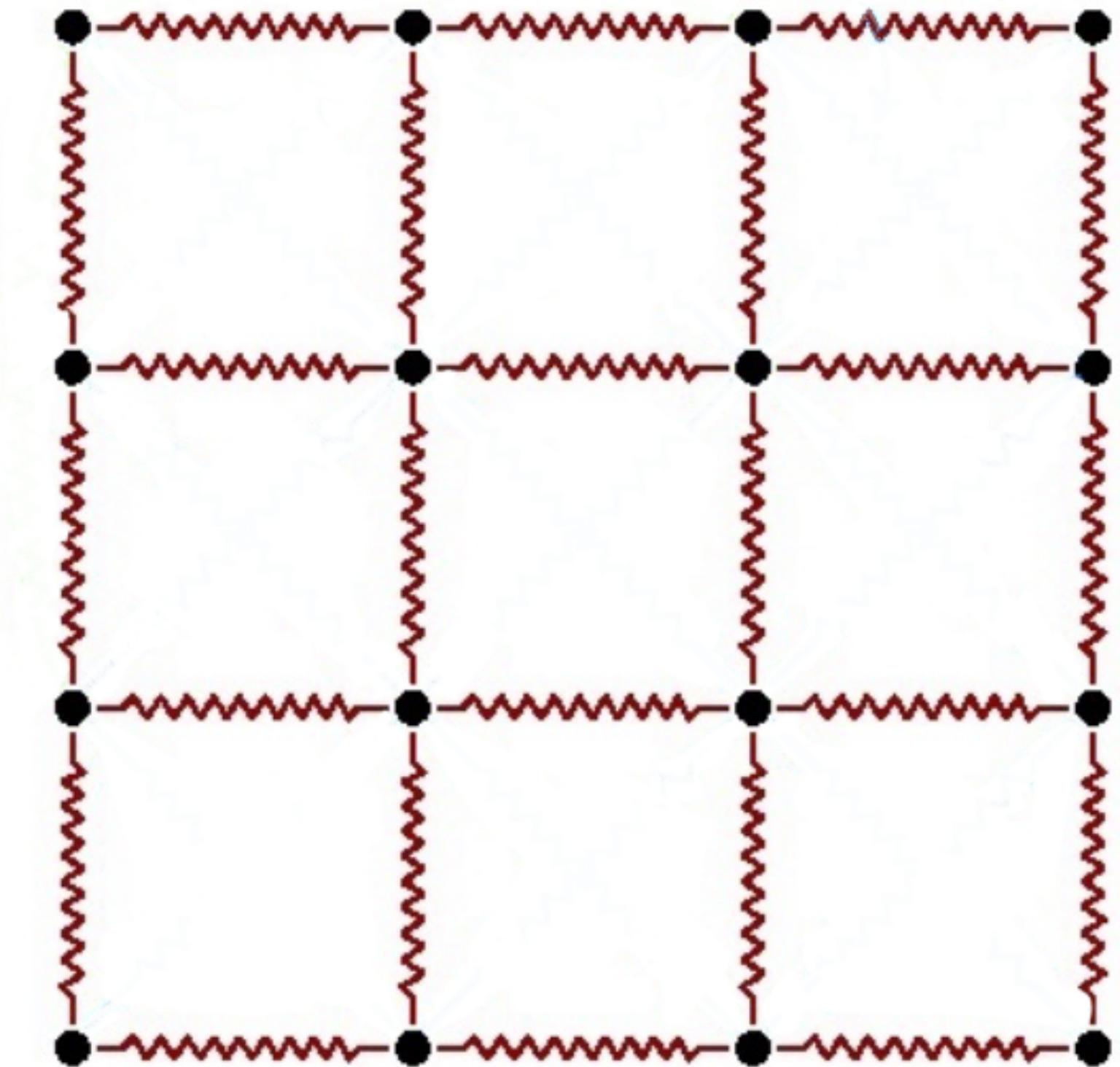


# ПОСТАНОВКА ЗАДАЧИ

## Математическое моделирование ткани

В данной работе ткань моделируется системой соединенных друг с другом масс. Массы распределены равномерно, одним слоем, связи существуют только между непосредственными соседями.

Необходимо построить математическую модель и реализовать программное решение, позволяющее проводить быстрое и эффективное численное моделирование деформации и движения ткани под действием внешних сил, с учетом граничных условий и взаимодействий с препятствиями

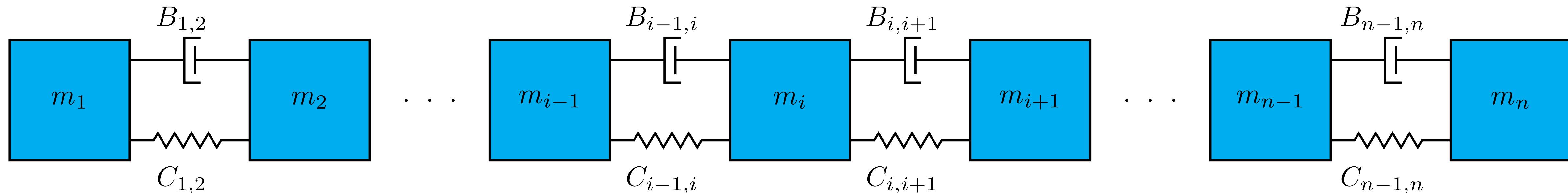


## ЦЕЛЬ

---

- I) построение математической модели динамики ткани, основанной на массо-пружинной системе (модели частиц и пружин);
- II) решение уравнений динамики ткани с использованием численных методов интегрирования;
- III) реализация высокопроизводительных вычислений на платформе **Taichi**;
- IV) экспериментальный анализ и визуализация полученных результатов.

# МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ФИЗИЧЕСКИХ СВОЙСТВ ТКАНИ



**С<sub>i,j</sub>** - Упругости пружин; **С<sub>i,j</sub> \* X<sub>i,j</sub>** - Сила упругости;

**В<sub>i,j</sub>** - Демпфирования; **В<sub>i,j</sub> \* v<sub>i,j</sub>** - Сила демпфирования;

Система уравнений динамики цепочки осцилляторов :

$$m_1 \cdot \frac{d^2x_1}{dt^2} = -C_1 \cdot (x_2 - x_1) - B_1 \cdot (v_2 - v_1)$$

$$m_i \cdot \frac{d^2x_i}{dt^2} = -C_i \cdot (x_i - x_{i-1}) - B_i \cdot (v_i - v_{i-1}) - C_{i+1} \cdot (x_{i+1} - x_i) - B_{i+1} \cdot (v_{i+1} - v_i)$$

$$m_n \cdot \frac{d^2x_n}{dt^2} = -C_n \cdot (x_n - x_{n-1}) - B_n \cdot (v_n - v_{n-1})$$

# МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ФИЗИЧЕСКИХ СВОЙСТВ ТКАНИ

Сила упругости  $F_{spring} = -C_s \cdot \left( |x_{ij}| - l_0 \right) \cdot \frac{x_{ij}}{|x_{ij}|}$

Сила демпфирования  $F_{damping} = -B \cdot (V_i - V_j)$

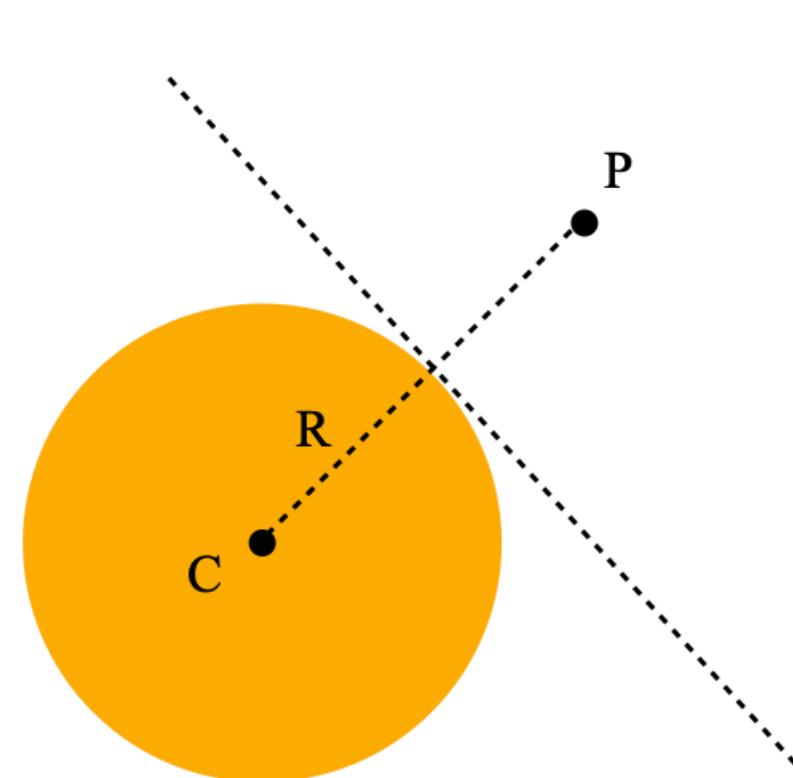
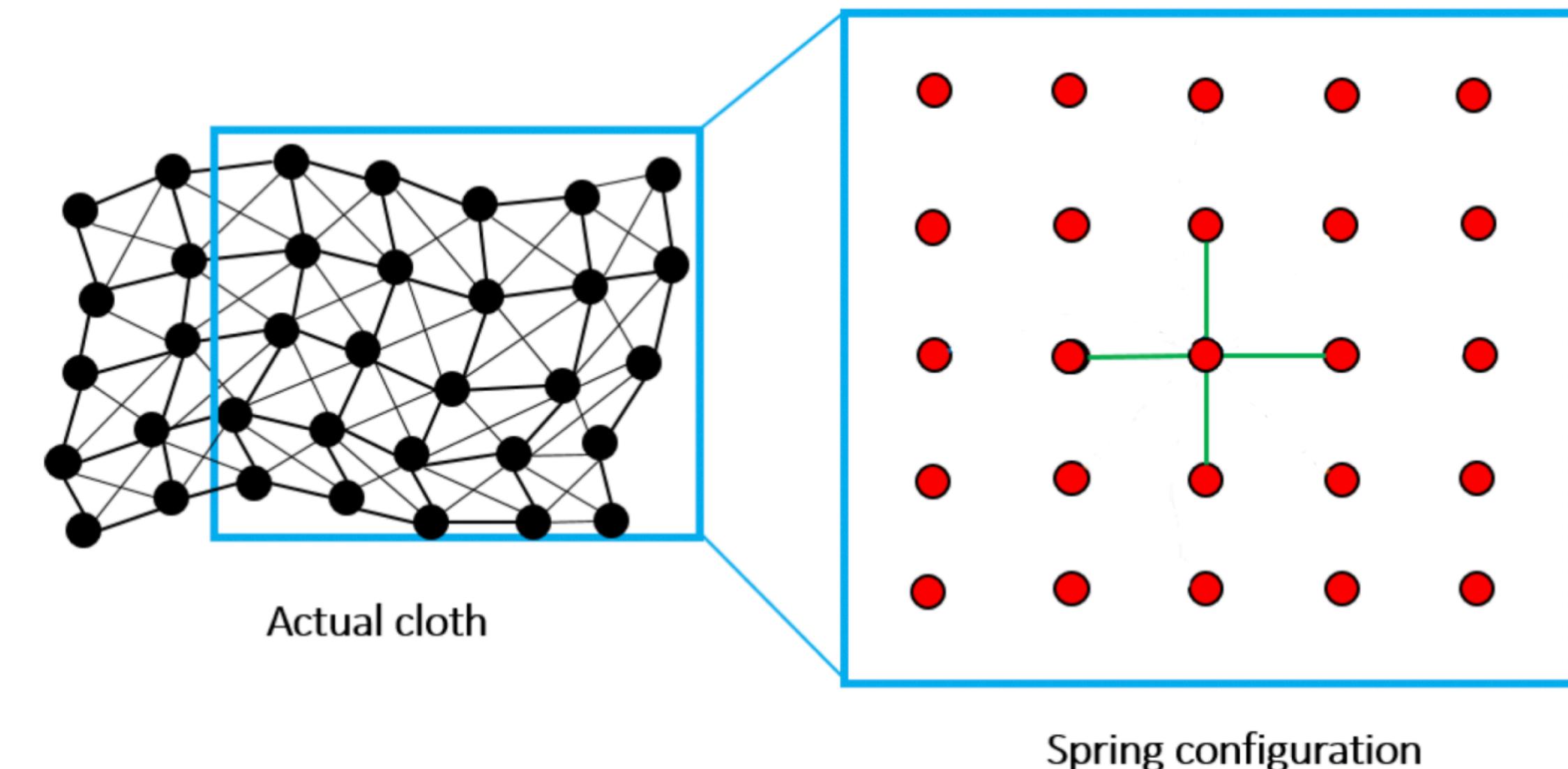
Гравитация  $F_{gravity} = m \cdot g$

Силы контакта и столкновения

$$F_{ij}^{sphere} = \begin{cases} C_s \left( R - \|x_{ij}\| \right) \frac{x_{ij}}{\|x_{ij}\|} + B_s \left( v_{ij} \cdot \frac{x_{ij}}{\|x_{ij}\|} \right) \frac{x_{ij}}{\|x_{ij}\|}, & \|x_{ij} - c\| < R \\ 0, & \text{else} \end{cases}$$

**C** - Центр сферы; **P** - Частица; **R** - Радиус сферы

$$F_{ij}^{total} = F_{ij}^{spring} + F_{ij}^{damping} + F_{ij}^{gravity} + F_{ij}^{sphere}$$



# ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

---

Метод Эйлера (**Euler Method**)

$$a_{ij}(t) = \frac{F_{ij}^{total}(t)}{m}$$

$$v_{ij}(t + \Delta t) = v_{ij}(t) + a_{ij}(t) \cdot \Delta t$$

$$r_{ij}(t + \Delta t) = r_{ij} + v_{ij}(t + \Delta t) \cdot \Delta t$$

Метод Верле (**Verlet Method**)

$$a_{ij}(t) = \frac{F_{ij}^{total}(t)}{m}$$

$$v_{ij}(t) = \frac{r_{ij}(t + \Delta t) - r_{ij}(t - \Delta t)}{2 \cdot \Delta t}$$

$$r_{ij}(t + \Delta t) = 2 \cdot r_{ij}(t) - r_{ij}(t - \Delta t) + a_{ij} \cdot \Delta t^2$$



**Taichi Lang** - это императивный язык параллельного программирования с открытым исходным кодом для высокопроизводительных численных вычислений. Он встроен в **Python** и использует фреймворки компилятора **JIT (just-in-time)**, например **LLVM**, для передачи кода **Python**, требующего больших вычислений, в собственные команды **GPU** или центрального процессора.

Почему **Taichi Lang** ?

Построен на основе **Python**

Производительность: **CPU, GPU**

Гибкость: **SNode**

Портативность: **Vulkan, CUDA, ...**

**ti.field(dtype, shape)**

**shape = 9**



**ti.Matrix.field(n=3, m=2, dtype=ti.f32, shape=(3, 3))**

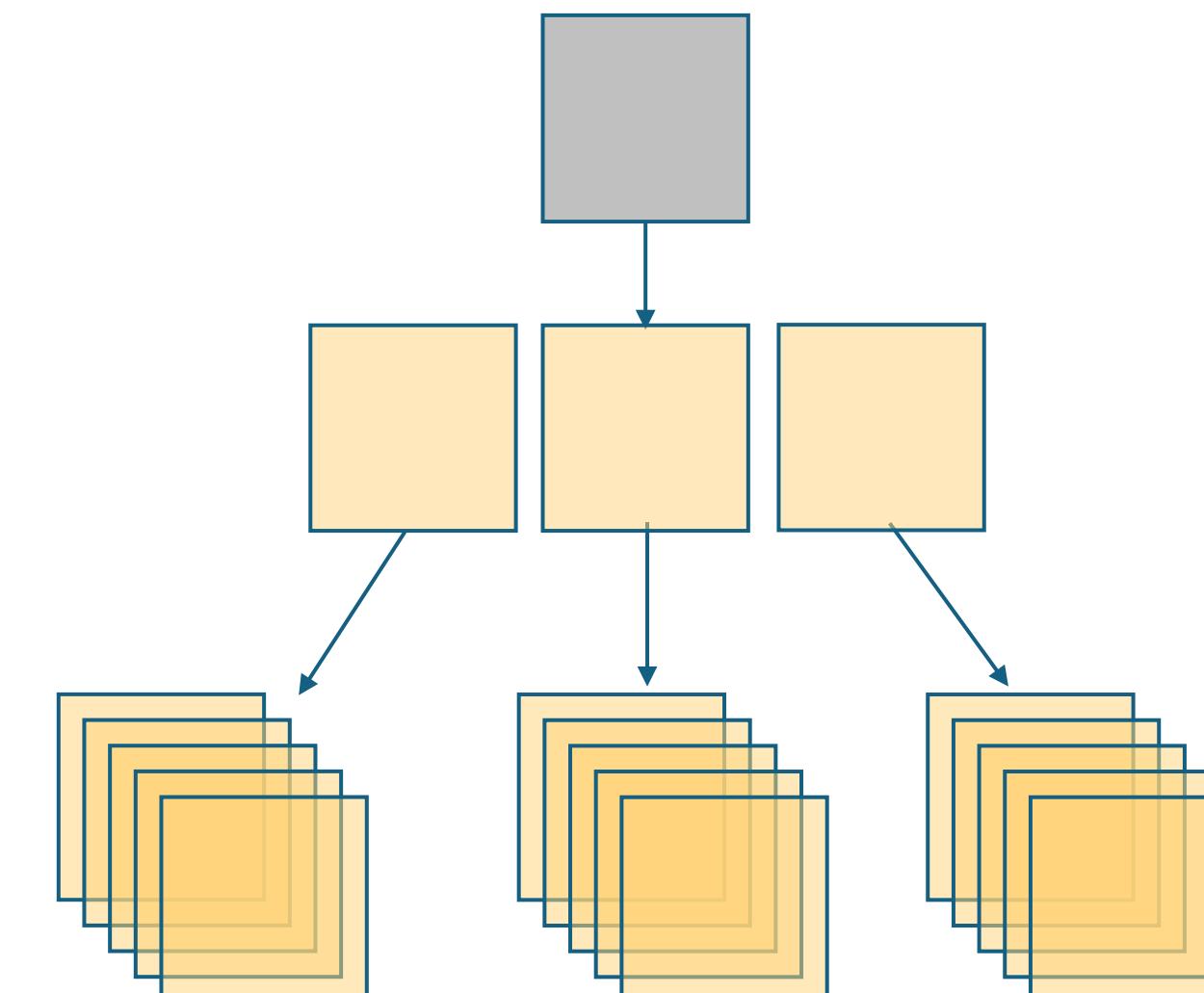
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32
ti.f32	ti.f32	ti.f32	ti.f32	ti.f32	ti.f32

**ti.Vector.field(n=2, dtype=ti.f32, shape=(3, 3))**

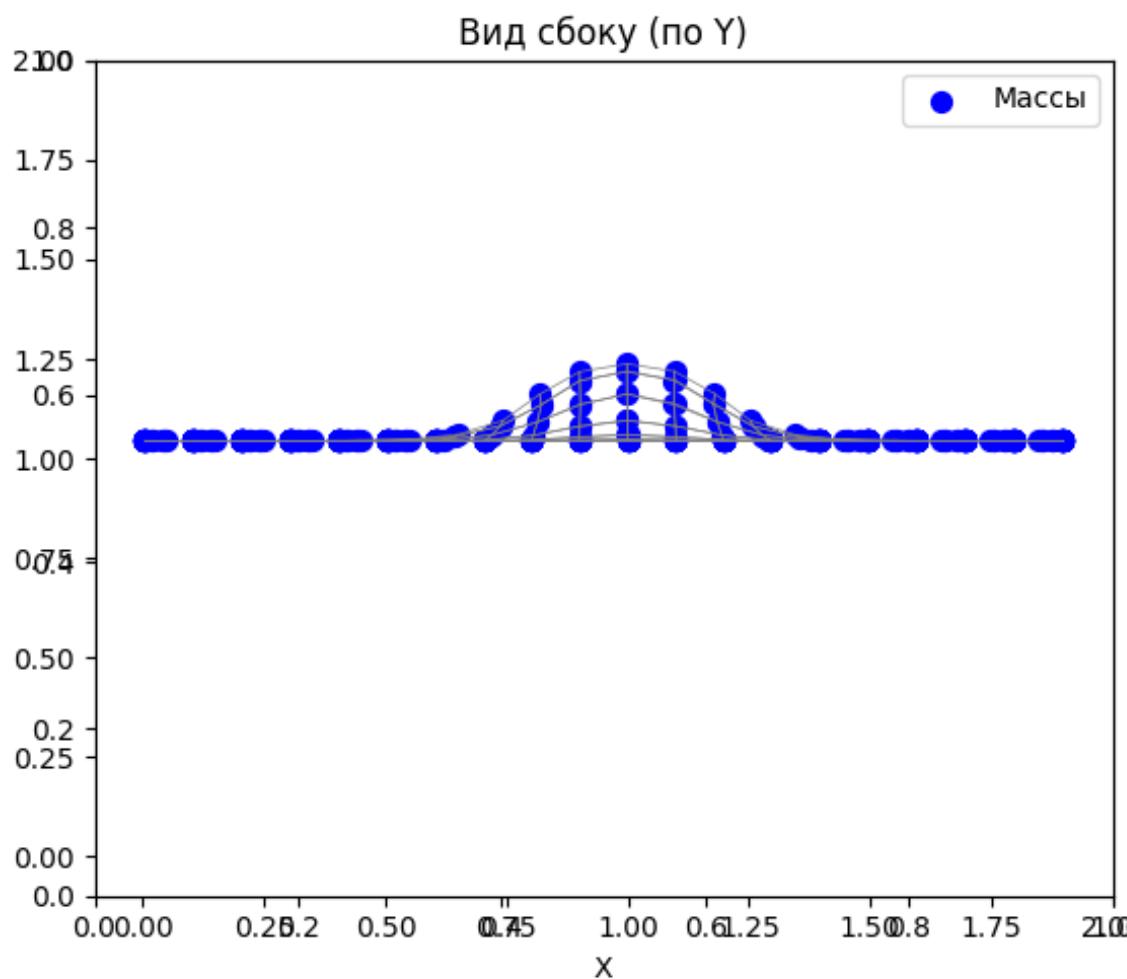
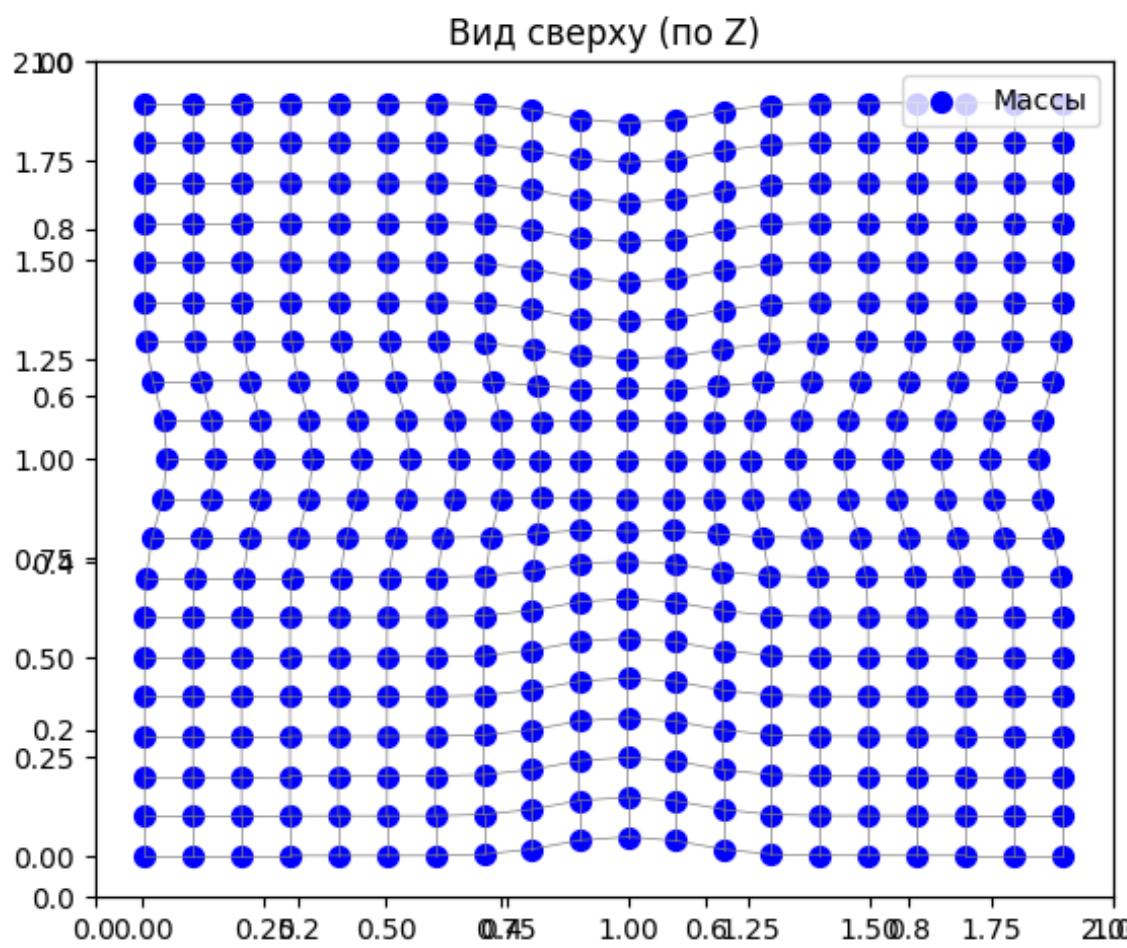
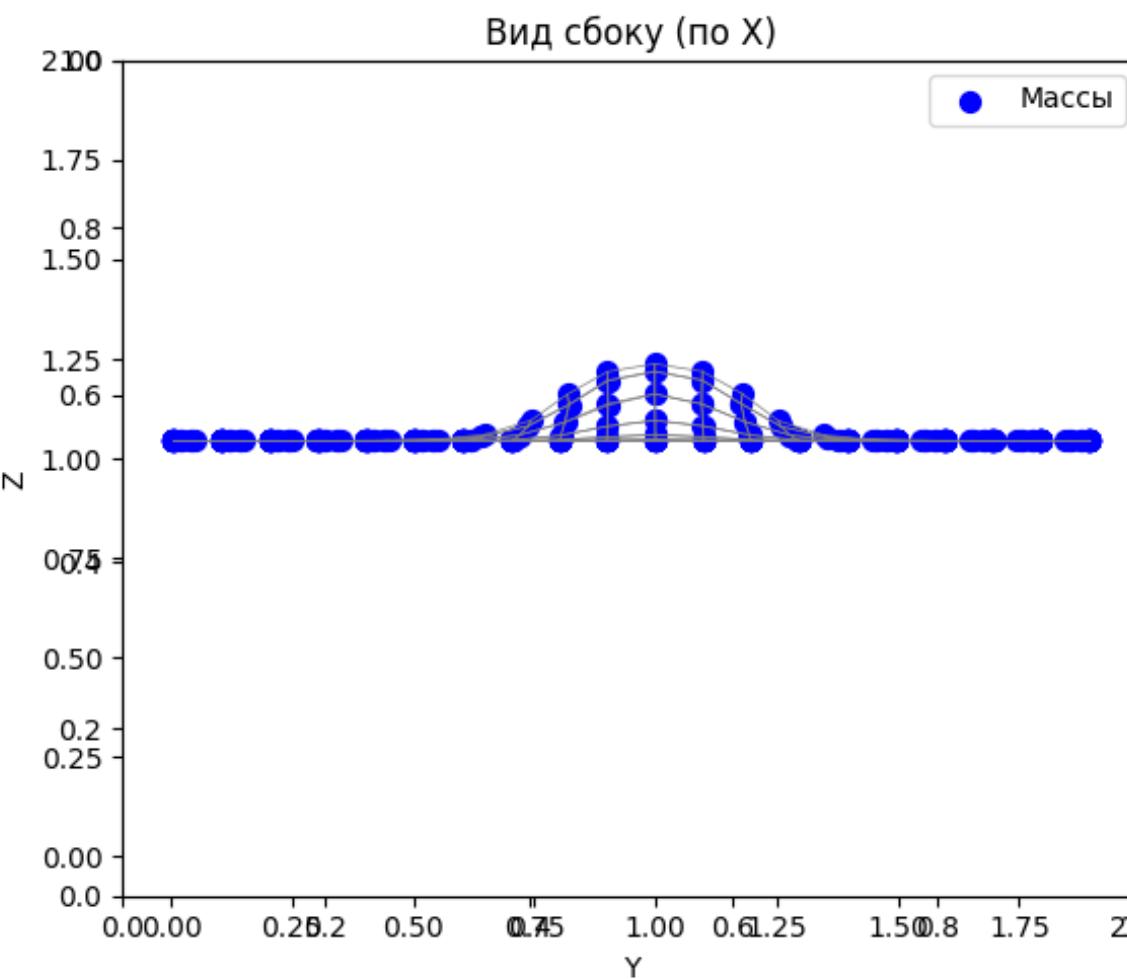
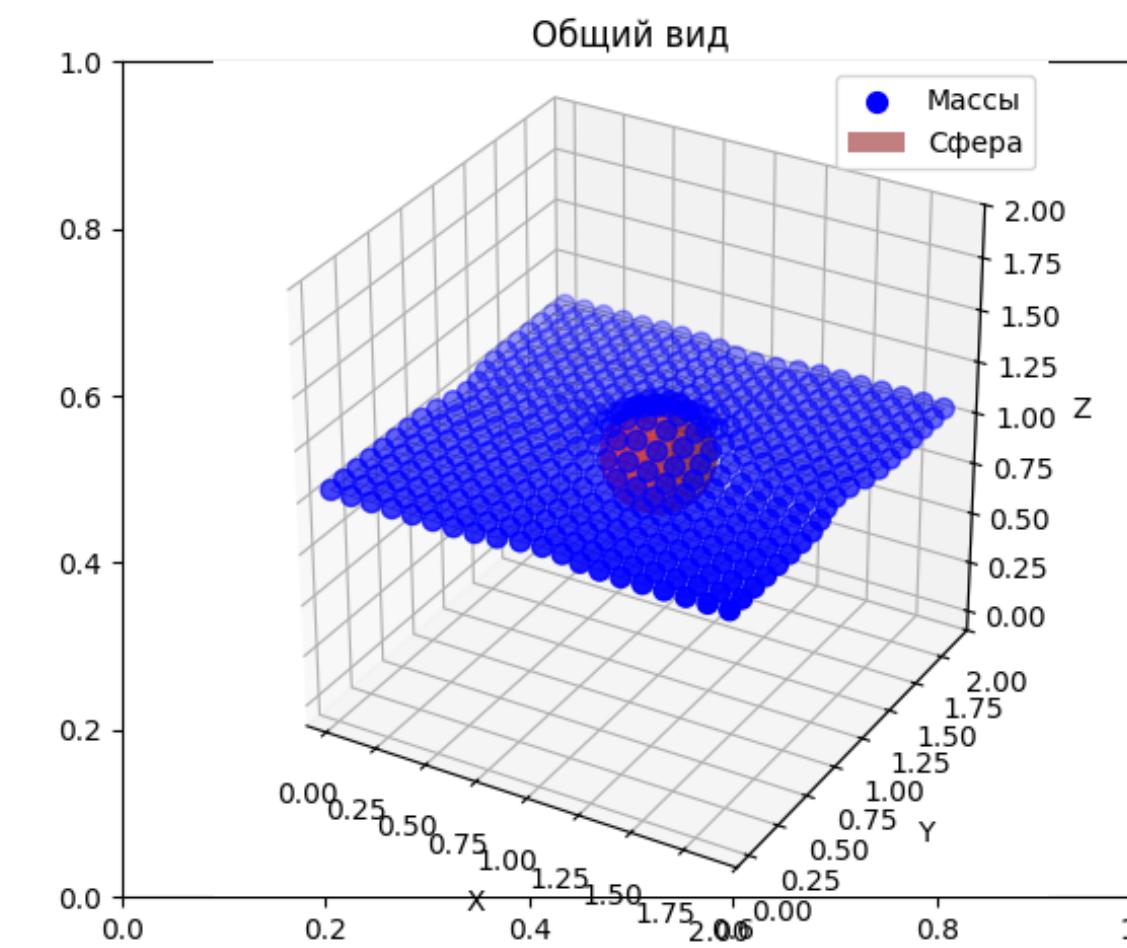
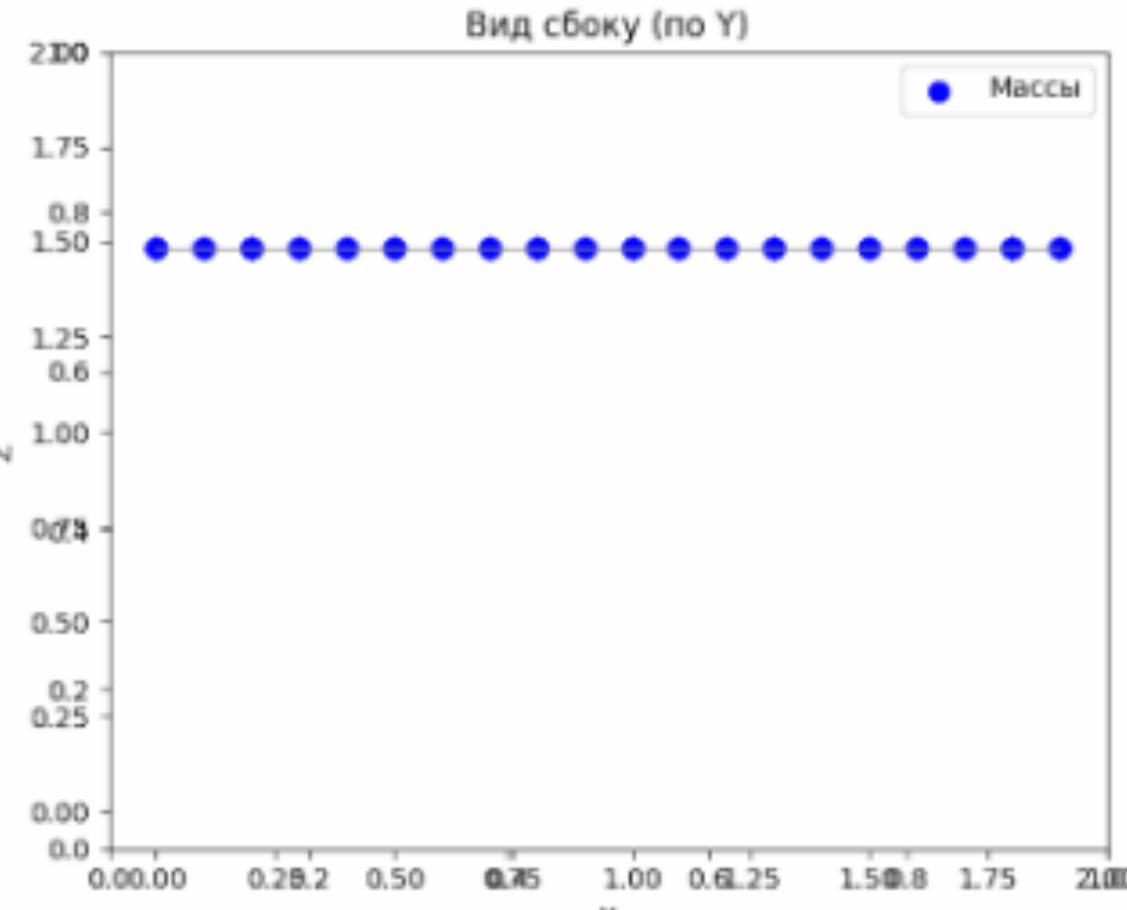
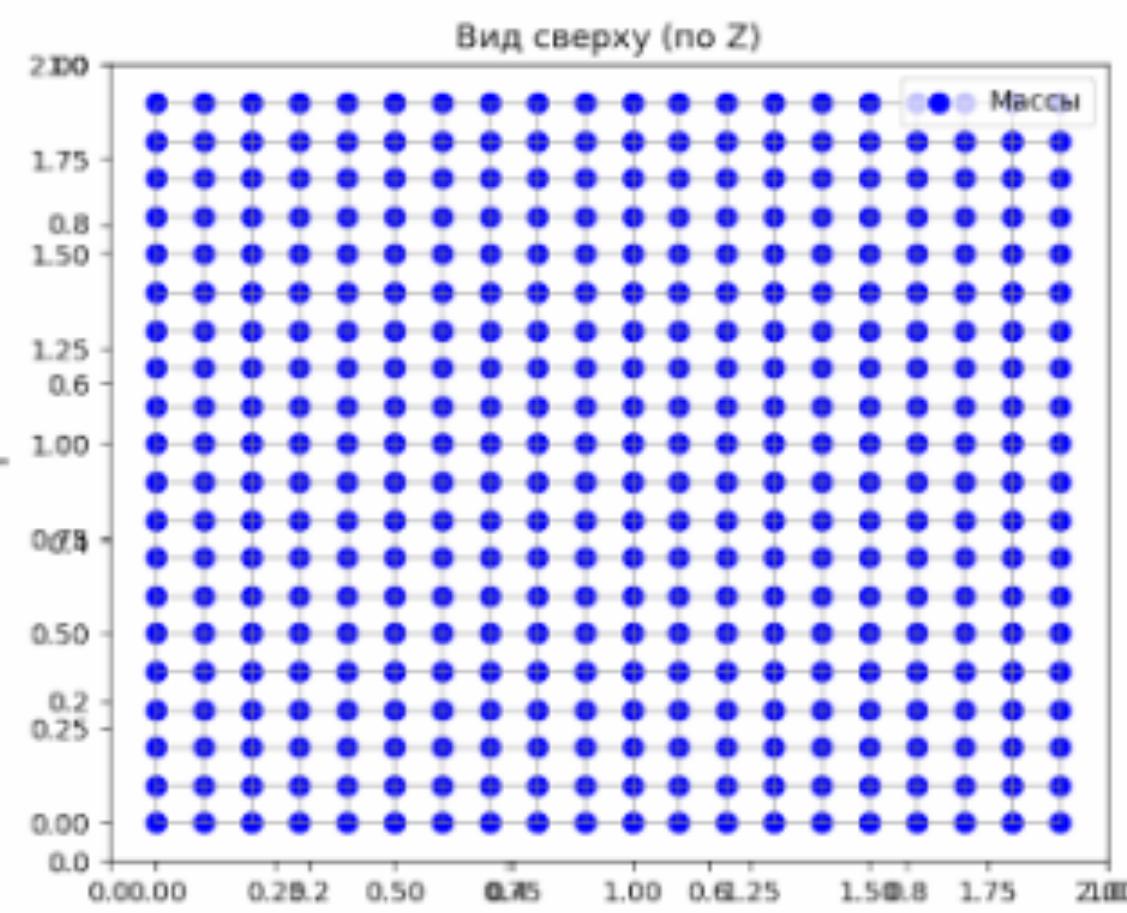
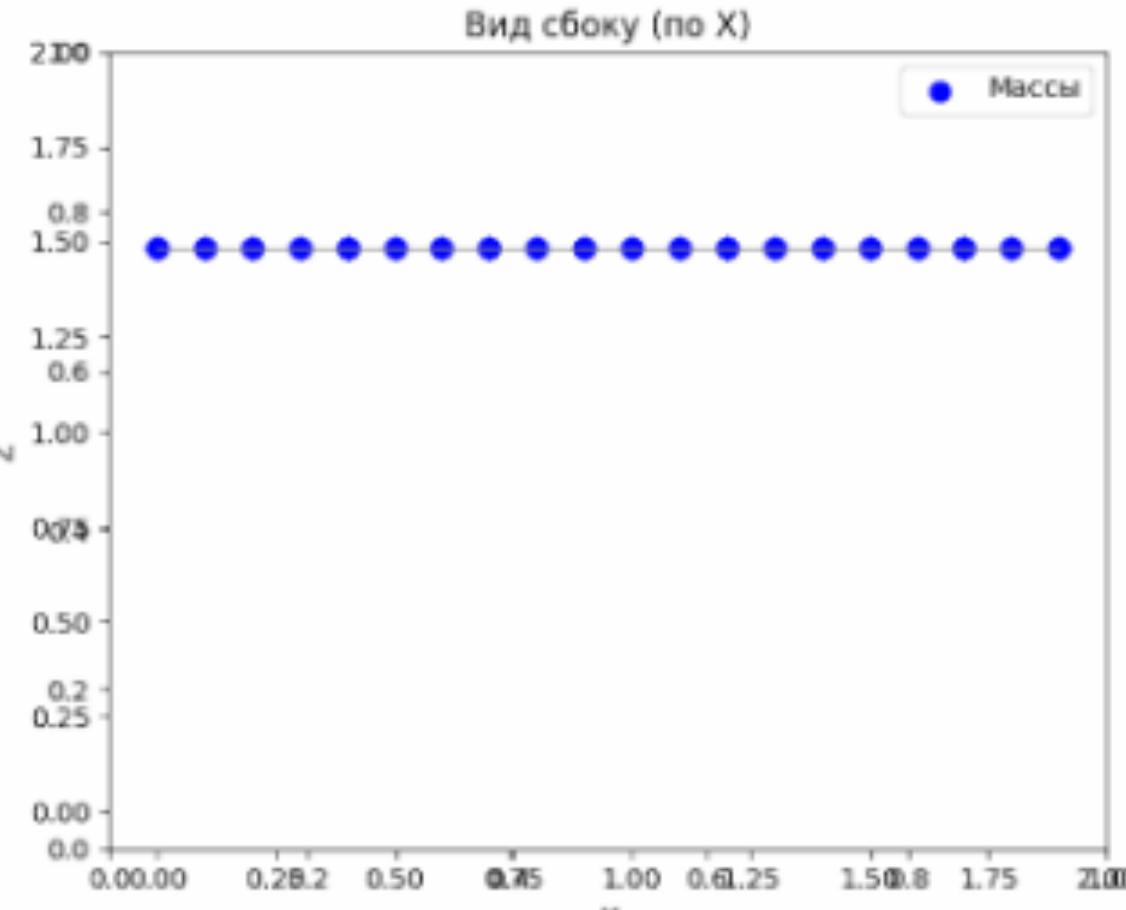
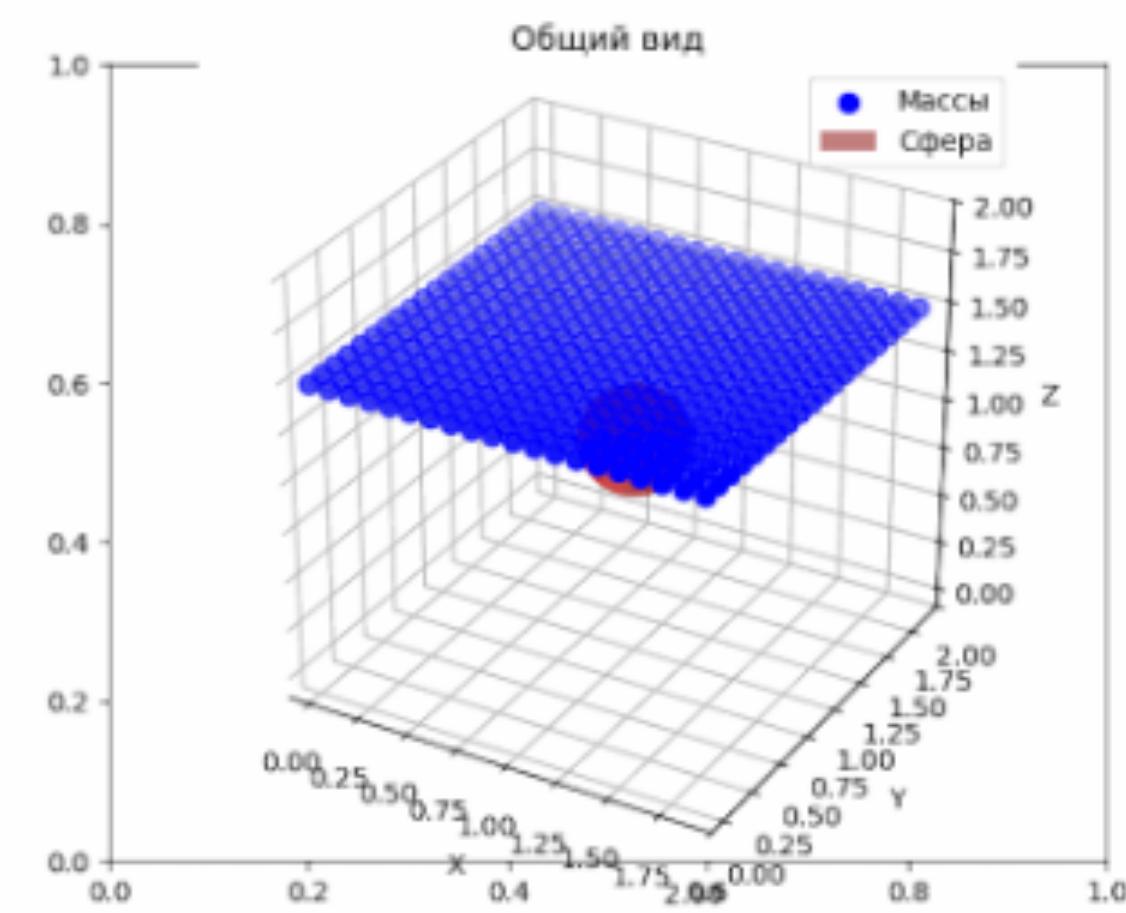
(ti.f32, ti.f32)	(ti.f32, ti.f32)	(ti.f32, ti.f32)
(ti.f32, ti.f32)	(ti.f32, ti.f32)	(ti.f32, ti.f32)
(ti.f32, ti.f32)	(ti.f32, ti.f32)	(ti.f32, ti.f32)

**SNode**    **x = ti.field(ti.f32)**

**s = ti.root.dense(ti.i, 3).pointer(ti.j, 4).place(x)**



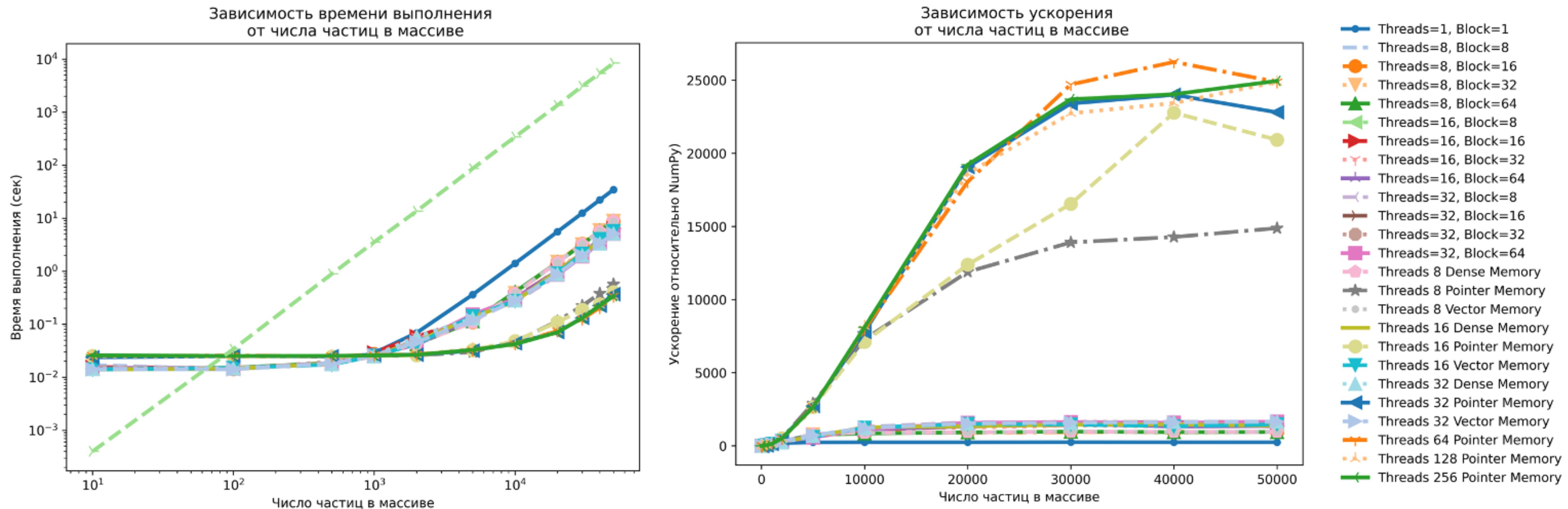
# ВИЗУАЛИЗАЦИЯ



шаг по времени = 0

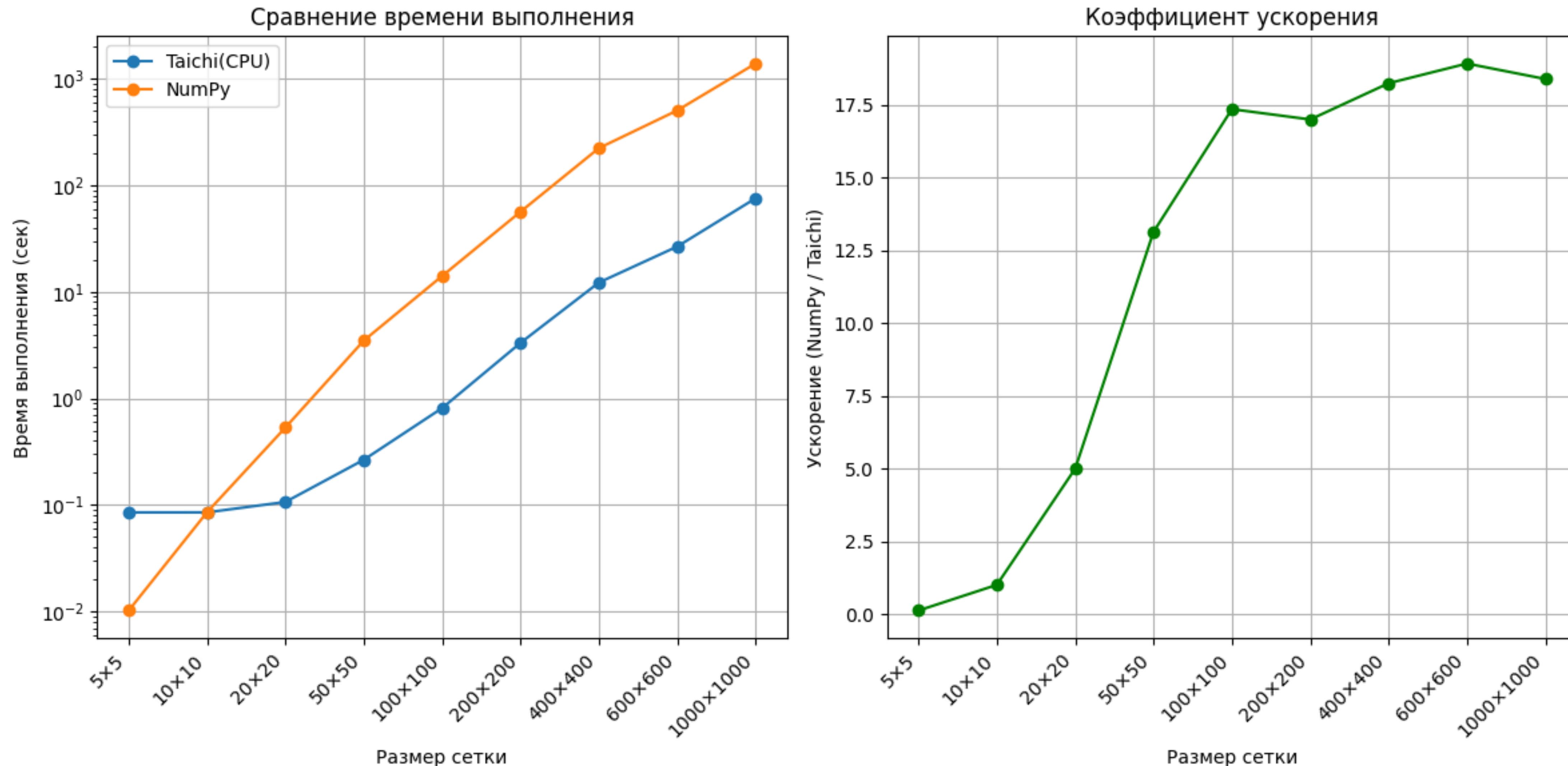
шаг по времени = 50

# АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ



CPU: Intel i9-10900 (10 ядер, 20 потоков) частота 5.200GHz

# АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ



Operating System: Darwin 24.0.0, CPU: Apple M3 Max, Physical Cores: 14, Logical Cores: 14

# ОСНОВНЫЕ РЕЗУЛЬТАТЫ

---

- Рассмотрены применения тканых материалов в различных отраслях
- Изучены основные механические свойства тканых материалов, и их влияние на динамическое поведение
- Изучены способы моделирования тканей на основе систем масс со связями
- Изучены способы применения пакета построения сеток **gmsh** для построения сложных сеточных моделей
- Предложена математическая постановка задачи о динамическом поведении системы связанных масс
- Изучены методы интегрирования для решения динамических задач
- Изучен пакет **TaiChi Lang** для высокопроизводительных вычислений на платформе **Python**
- Реализованы программные решения для математического моделирования ткани на платформе **Python**, основанные на пакетах **NumPy** и **TaiChi**
- Проведено сравнение производительности программных решений на системах масс различной размерности от **5x5** до **1000x1000**
- Сравнение показало, что для малых размерностей системы масс (до **10x10**) целесообразнее применять пакет **NumPy**, т. к. отсутствие накладных расходов делает его более производительным
- Для систем превышающих размерность **10x10** пакет **TaiChi** является существенно более производительным
- Для систем с размерностью более **100x100** пакет **TaiChi** позволяет перейти к решению поставленной задачи на **GPU** простым указание целевой платформы выполнения кода

Спасибо за внимание