



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



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.

ti.field

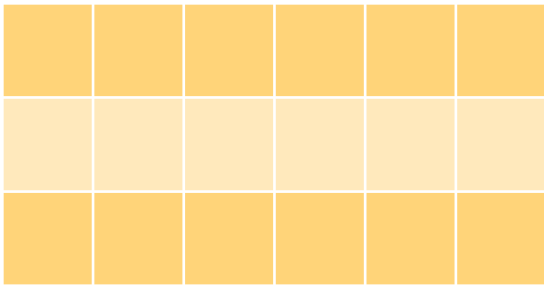
ti.field(dtype, shape)

dtype: ti.i32/ti.f32

shape = n



shape = (3, 6)



Matrix

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

f = 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)

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

ti.Struct.field(dtype=ti.f32, shape=(10, 20, 30))

SNode and ti.field

```
x = ti.field(ti.f32)  
ti.root.place(x)
```



```
x = ti.field(ti.f32, shape=())
```

```
x = ti.field(ti.f32)  
ti.root.dense(ti.i, 3).place(x)
```



```
x = ti.field(ti.f32, shape=3)
```

```
x = ti.field(ti.f32)  
ti.root.dense(ti.ij, (3,4)).place(x)
```



```
x = ti.field(ti.f32, shape=(3,4))
```

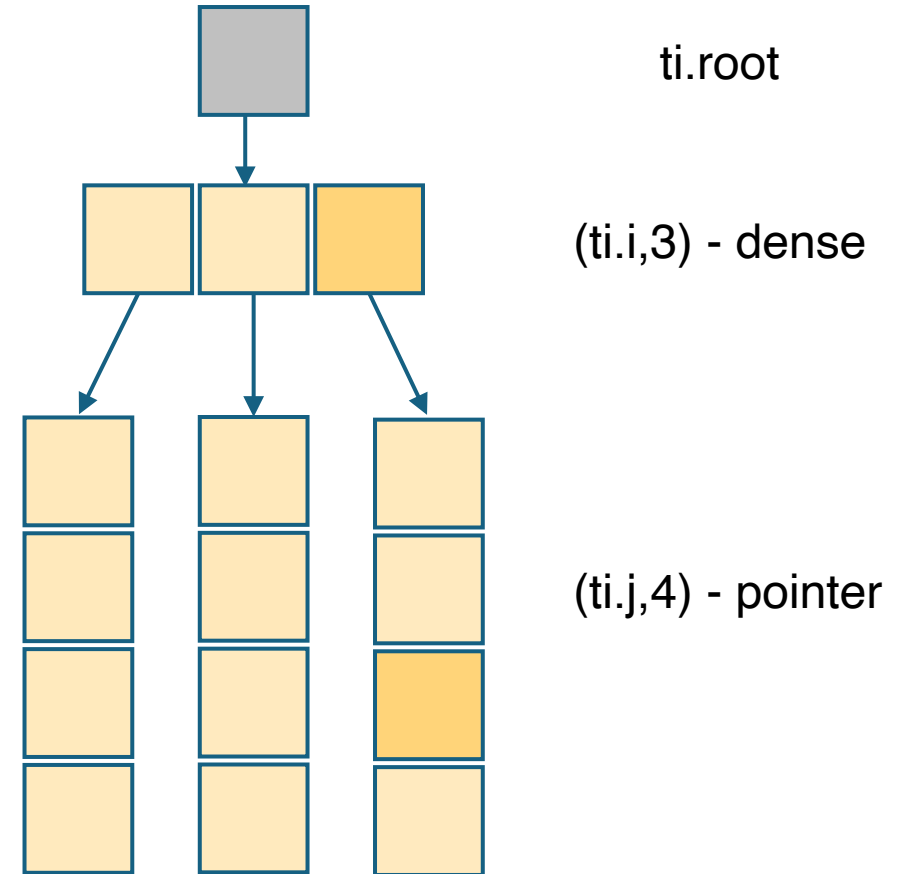
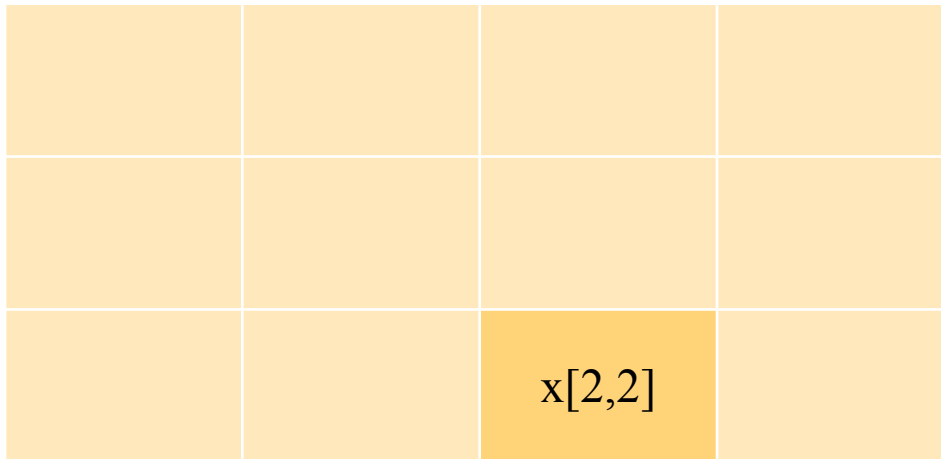
$x[i,j]$

```
x = ti.field(ti.f32)  
ti.root.dense(ti.i, 3).dense(ti.j, 4).place(x)
```

ti.field and SNode

```
x = ti.field(ti.f32)  
S = ti.root.dense(ti.i, 3).pointer(ti.j, 4).place(x)
```

$x[2,2] = 3.14$



Названия	Taichi Field (x)	SNode дерево (dense)
Назначение	Единый интерфейс для работы с пользователем.	Оптимизация структуры памяти и способа хранения данных.
Роль	Доступ к конкретным значениям через индексы.	Описание многослойной структуры данных, оптимизация доступа к памяти и вычислений.
Представление	Логическая глобальная структура, поддерживающая полный доступ по индексам.	Физическая многослойная структура памяти (такие как dense, pointer и т.д.).
Связь	Связывается с SNode через place.	Обеспечивает базовое хранение, недоступно напрямую для пользователя и абстрагируется в индексы Field.

Taichi Field	NumPy ndarray	PyTorch Tensor
Плотные и разреженные сетки	Только плотные сетки	Плотные и разреженные сетки
Поддержка CPU, GPU, Metal, Vulkan, CUDA	Поддержка только CPU	Поддержка CPU и GPU (CUDA/OpenCL)
Поддерживает автоматическую параллелизацию и оптимизацию для разных платформ	Зависит от CPU, автоматическая параллелизация не поддерживается	Поддержка GPU-ускорения (но производительность может быть ниже, чем у Taichi)
Подходит для физических симуляций, вычислений с разреженными матрицами и других задач высокой производительности. Не поддерживает сложную линейную алгебру.	Поддерживает множество операций линейной алгебры, широковеatching и срезы, функционал обширный.	Поддерживает автоматическое дифференцирование и сложные операции с тензорами, идеально для глубокого обучения.
Используется для объявления символических переменных, значения которых вычисляются динамически в Kernel-функциях. Обновление данных возможно только внутри Kernel	Значения изменяемы, можно напрямую модифицировать через индексы. Обновление происходит в Python-среде.	Значения изменяемы, поддерживаются индексное присваивание и вычисление градиентов. Обновление возможно как в Python, так и на GPU.
Высокопроизводительные физические симуляции, обработка разреженных данных, приложения с необходимостью кросс-платформенного GPU-ускорения.	Общая обработка численных данных, предварительная обработка данных, задачи без необходимости в GPU-ускорении.	Приложения глубокого обучения, автоматическое дифференцирование, задачи с эффективным GPU-ускорением.