DEM simulation: Taichi vs Warp

This document presents a comparative analysis of the performance of Taichi and Warp in the context of Discrete Element Method (DEM) simulations.

Scenario and Parameters

The simulation involves a grid of 64 x 64 particles, each with randomly assigned sizes, falling from a height. These particles are confined by the left, bottom, and right boundaries. The computation of contact forces is triggered by the distance between particles or the distance from a particle to a boundary. When the distance, minus the particle radius, is negative (indicating contact), the contact force is computed using the following equations:

$$F_{normal} = err * k_n + min(v_n, 0) * k_d$$
$$F_{tangent} = min(v_t * k_f, F_{normal} * k_\mu)$$
$$F_{change} = - F_{normal} * n - F_{tangent} * t$$

The err denotes the overlapping distance. The n and t are the unit vectors in the normal and tangential directions, and the k_n and k_d represent stiffness and damping. k_f, and k_μ denote the friction coefficient for tangential velocity and normal force, respectively. The tangential force follows the Coulomb friction law, where the magnitude of the frictional force is capped by the product of the normal force and its frictional coefficient.

Table 1 outlines the parameters used for the simulation in both Taichi and Warp. We didn't use the same domain size because I can't find a way to adjust domain size in Taichi. Thus, we scale the particle size and domain size in Taichi to fit Warp. Other parameters such as simulation time frame and coefficients for contact force are the same in Taichi and Warp.

Table 1. Simulation parameters in Taichi and Warp

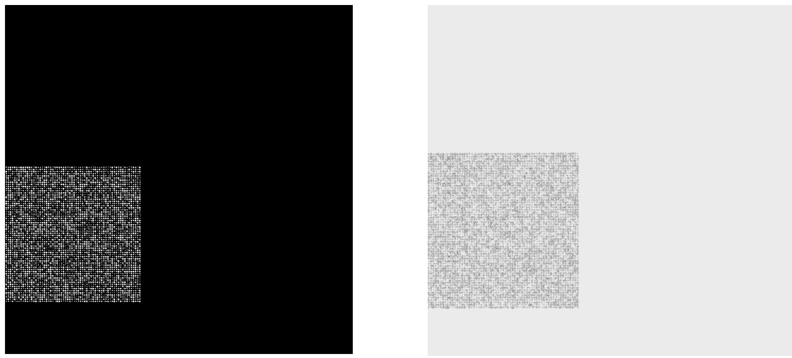
Parameter	Taichi	Warp
k_n	8000	
k_d	2.0	
k_f	1.0 (100)*	
$k_\mu(\phi)$	0.5 (26.6°)	
m	1/64	
dt	1/3840	
Substeps per frame	64	
Total frames	400	
Simulation time	6.67 s	
No. Particle	4096	
Particle size range	[0.0015, 0.003]	[0.05, 0.1]
Domain size	[0, 1]	[0, 33.3]**

Number of grids	128 x 128	128 x 128*128**

*$k_f = 1.0$ for particle collision; $k_f = 100$ for boundary contact

**In Warp, the simulation is in 3D because of the built-in hash grid function, but we ignore the z–direction by resetting everything to zero in z-direction.

Figure 1 shows the initial state of the simulation in Taichi and Warp. A grid of 64 by 64 particles is generated with random radius. We put noises in the particle position, so the particles will not bounce in vertical direction only.



(a) Taichi (b) Warp

Figure 1. The initial state of simulation

Results

A notable difference emerges when analyzing the outcomes of simulations conducted with Taichi versus those done with Warp, even after applying a scale factor on size parameters in Taichi to match Warp. The Taichi simulation (Figure 2a) shows particles distributed evenly with a gentle slope. In contrast, the Warp simulation (Figure 2b) depicts most particles clustered in the left corner, forming a steep slope, with only a limited number migrating right.

Interestingly, when Warp's particle size and domain dimensions are aligned with Taichi's (0.0015 to 0.003 particle size and 0 to 1 domain), Warp's particle distribution resembles Taichi's more closely (Figure 2c). This suggests particle and domain size scaling alone may not suffice. We might need to quadratically scale mass to maintain consistent particle density between the two. Further investigation into mass scaling's impact is warranted to better understand the observed differences.

(a) Taichi

(b) Warp

(c) Warp (same particle size as the size in Taichi)

Figure 2. The simulation results after 400 frames

Besides the particle distribution, we also compare simulation efficiency by measuring the average time for 100 simulations. All simulations run in Colab with a Tesla T4 GPU. The results in Table 2 show Warp is twice as efficient as Taichi. One reason could be that Warp's built-in hash grid function enables very efficient collision detection, compared to Taichi's collision detection.

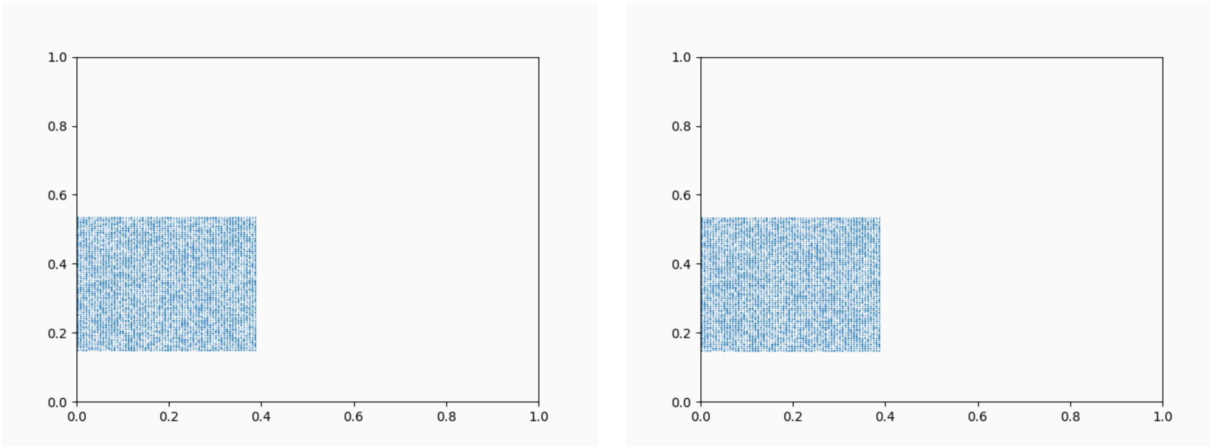
Table 2. Simulation efficiency

	Taichi	Warp
100 iterations	10 min 08 sec	5 min 22 sec
Average	6.08 sec/iter	3.22 sec/iter

Note that Warp is actually running in 3D scenario, all vectors of particle positions, velocity, and force are in 3D.

Figure 3 plots the particle trajectories in Taichi (3a) and Warp (3b), both with matched particle and domain sizes. Different migration behaviors emerge. First, in Taichi, a lump of high horizontal velocity particles slides right upon colliding with the bottom and left boundaries. In contrast, Warp shows only a few high velocity particles. Second, most Taichi particles bounce back into the air while traveling right, but Warp's remain in a lump on the ground. Airborne particles only experience gravity, enabling longer rightward travel. Grounded particles undergo tangential friction, slowing their horizontal movement. This likely explains Taichi's wider distribution - more particles go airborne.

The collision behaviors differ despite using the same contact force parameters. In Taichi, particles rebound after hitting boundaries, sustaining horizontal velocity. Warp particles seem to lose momentum upon collision. The source of this discrepancy remains unclear and merits further analysis into the distinct collision handlings.



(a) Taichi (b) Warp

Figure 3. Trajectories of simulation