MLS-MPM(Moving-Least Squares Material Point Method) by Taichi

9 Feb. 2026

2018 年に Hu らによって提案された粒子とグリッドを両方用いるハイブリッドな手法．具体的には，粒子からグリッドへと情報を受け渡す Particle to Grid（P2G）ステップと，グリッドから粒子へと情報を書き戻す Grid to Particle (G2P)ステップを繰り返して，粒子の座標を更新.

MLS-MPM:
https://docs.taichi-lang.org/docs/quant#mls-mpm

ここから，以下の github のページに飛ばされ，このファイルがサンプルコード
MLS-MPM github:
https://github.com/taichi-dev/taichi_elements/blob/master/demo/demo_quantized_simulation_letters.py

で，これを動かすおおもとは，これ.
taici_elements:
https://github.com/taichi-dev/taichi_elements
以下をダウンロードして，展開する.
taichi_elements-master.zip

以下のディレクトリが作られる.
taichi_element-master/
readme.md に以下が書かれている.

README MIT license

Taichi Elements Build Status codecov 36%

Taichi elements is a high-performance multi-material continuum physics engine (work in progress).

Features:

- Cross-platform: Windows, Linux, and OS X
- Supports multi-threaded CPUs and massively parallel GPUs
- Supports multiple materials, including water, elastic objects, snow, and sand
- Supports (virtually) infinitely large simulation domains
- Supports sparse grids
- Highly efficient and scalable, especially on GPUs

Using `taichi_elements` in Python

Run demo with Python

- Install taichi with `pip`: `python3 -m pip install taichi`
- Execute `python3 download_ply.py` to download model files used by the demos
- Execute `python3 demo/demo_2d.py` and you will see

実行してみる．taichi はインストールされているはずなので，以下は要らない．

> Install taichi with pip: python3 -m pip install taichi

アクティベイトはする．

> source taichi_env/bin/activate

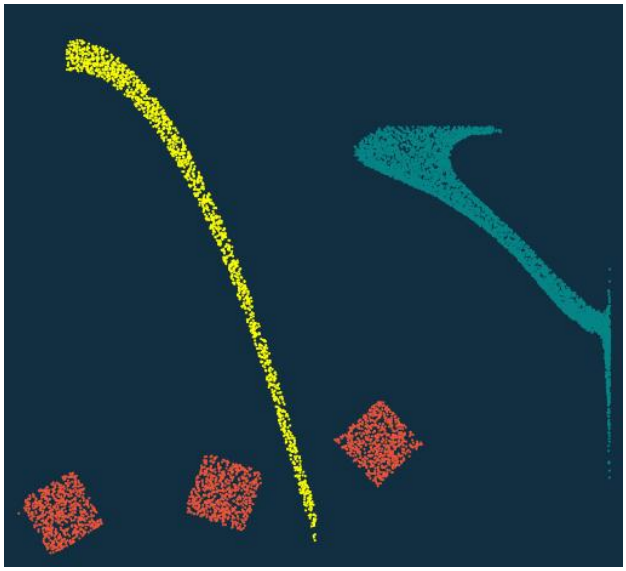
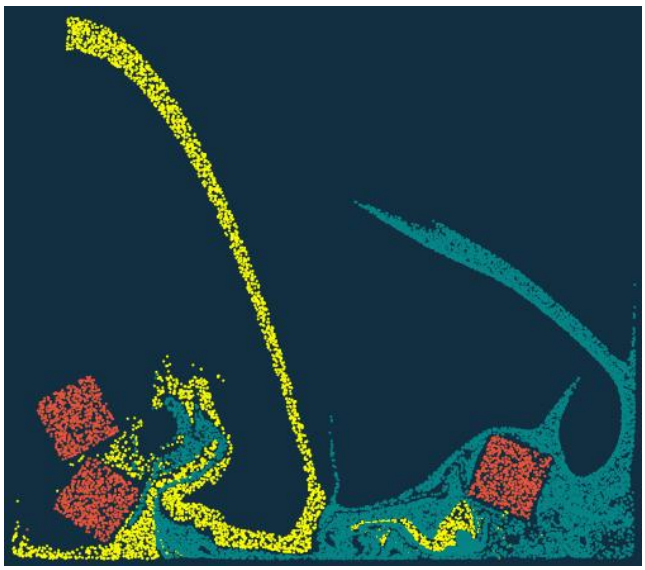
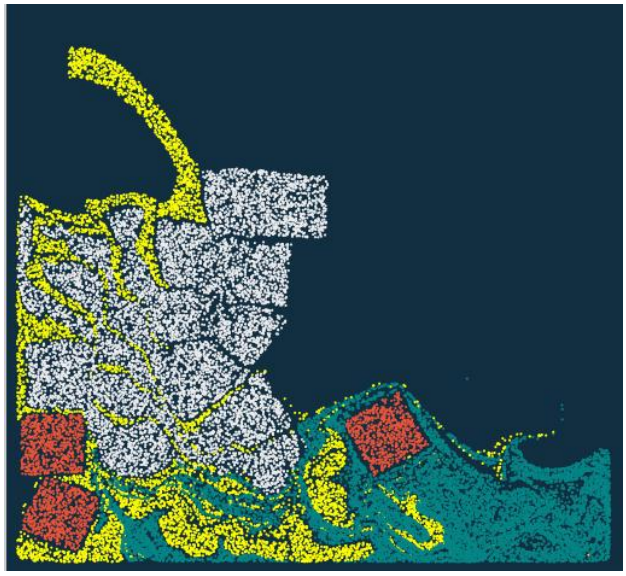
モデルファイルをダウンロードする．taichi_element-master のディレクトリで，`download_ply.py` があることを確認し，そこから以下を実行する．

> `python3 download_ply.py`

```
(taichi_env) kikut@kikut3:~/taichi/taichi_elements-master$ ls
LICENSE    blender      demo  download_ply.py  render_particles.py  setup.py  utils
README.md  codecov.yml  docs  engine           requirements.txt     tests
(taichi_env) kikut@kikut3:~/taichi/taichi_elements-master$ python3 download_ply.py
Downloading bunny_low.ply...
Downloading quantized.ply...
Downloading simulation.ply...
Downloading taichi.ply...
Downloading suzanne.npy...
Download finished!
```

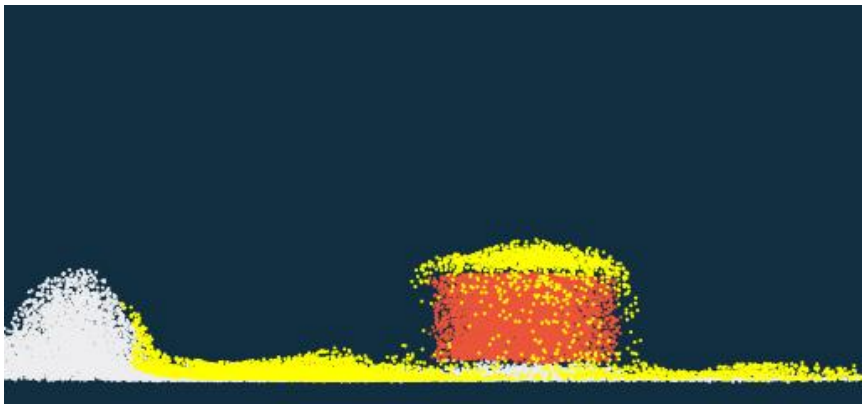
その後，以下を実行すると，2D シミュレーションが実行される．赤が連続体の固体．それ以外は流体．気体は計算しない固気液 3 層流計算．おそらく力学的正確さはほとんどない．

> `python3 demo/demo_2d.py`

3 次元計算も実行してみる.

> python3 ./demo/demo_3d.py

GGUI を使った表示のデモをしようとすると，Vulkan が無いと言ってくる.

> python3 ./demo/demo_3d_ggui.py

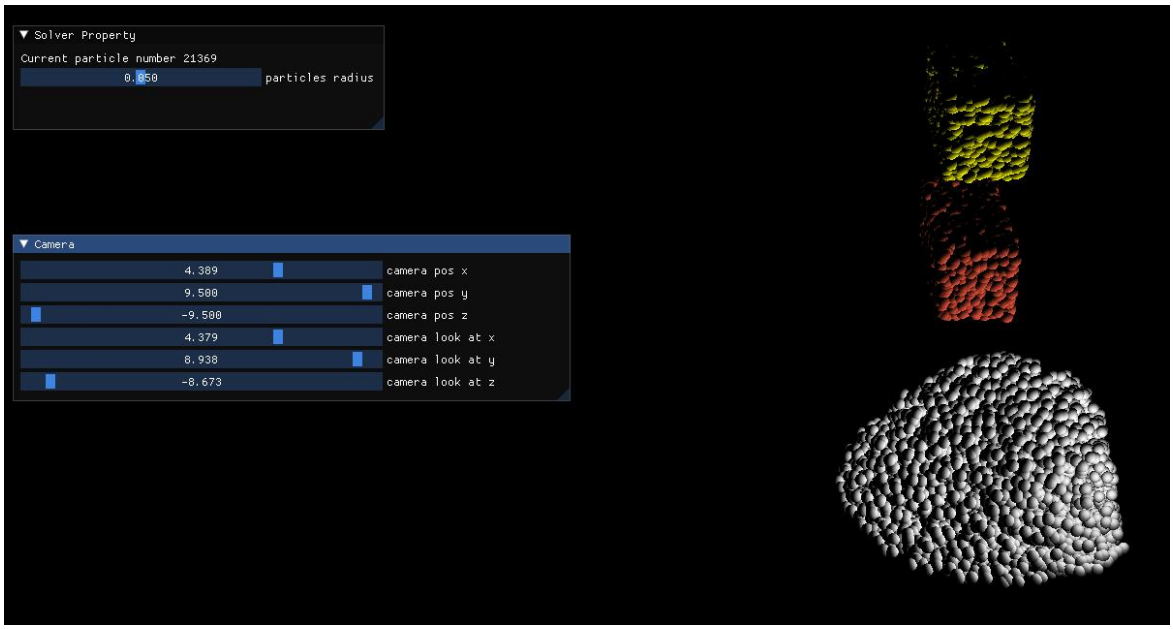
Vulkan: GPU を使って描画や汎用計算を行うための規格（API）．Khronos Group（OpenGL や OpenCL を作った団体）が開発．Taichi は Vulkan をバックエンドの一つとして使い、ネイティブな GPU アクセラレーション（CUDA のない環境でも高速化）を提供している．genesis は gpu から cpu に切り替えるだけで，vulkan を使わず実行してくれるが，taichi は微妙．このコードは変更だけでとりあえず動いた．

`demo_3d_ggui.py` の `gpu` を `cpu` に切り替えるととりあえず動く．おそらく `100` 倍遅いけど...

```
# ti.init(arch=ti.gpu, device_memory_GB=4.0)
↓
ti.init(arch=ti.cpu)
```

その後実行すると，インタラクティブな window で 3D デモが始まる．GUI はかなりしっかりしている．

```
> python3 ./demo/demo_3d_ggui.py
```



点群処理関数が無い場合には以下のパッケージを入れる．

```
> pip install plyfile
```

これ以降の Advance なプログラムは GPU が必要なようだ．CPU 用に書き換える必要性があるのか後日検討．ニーズが多ければ書き換える．とりあえず，私の PC では segmentation fault になる．

```
> python3_qantized_simulation_letters.py
```

はかろうじて動く．．．が，恐ろしく重い．以下は，80G byte のメモリーで動かしているとのこと．粒子は 1G 個．100 万点の流体計算の 1000 倍か．．．．