# Mandelbrot Area Challenge

The almond bread tilers / Jose, Julia, Florian, Niclas, Saurabh, David

## Organization among the Team

split into smaller teams working on subproblems

- improving the algorithm
- implementing it on the GPU
- combining results (see later!)

- use git to collaborate on code (repository)

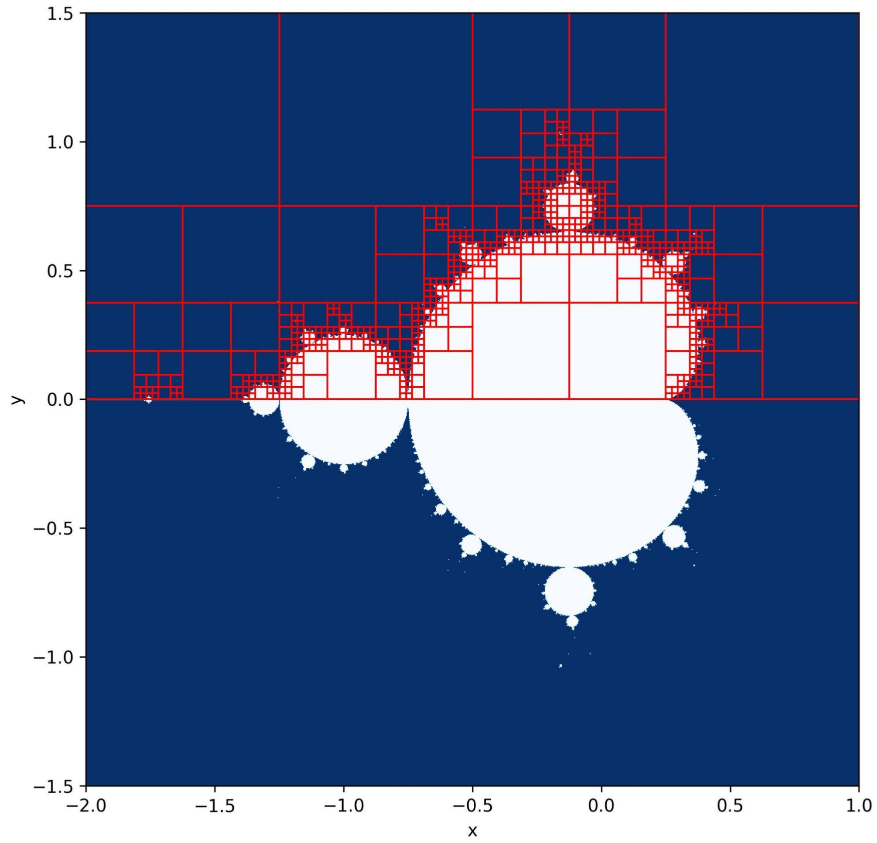
## Ansatz

1. Optimization of the algorithm
2. Implementation on GPU

ERUM DATA HUB

## 1. Optimization of the algorithm

- **exploit symmetry**
- **optimized tiling**: smaller tiles at the border of the mandelbrot set
  - idea: tiles completely inside or outside of the Mandelbrot set will converge quickly
  - start with coarse tiling
  - run count_mandelbrot with **small sample size on CPU**; if there are both convergent and divergent points, **recursively split the tile** until a certain depth
  - run count_mandelbrot with **large sample size on the GPU** on the found tiles
  - calculate area + uncertainty *per tile*
  - combine results



ERUM DATA HUB

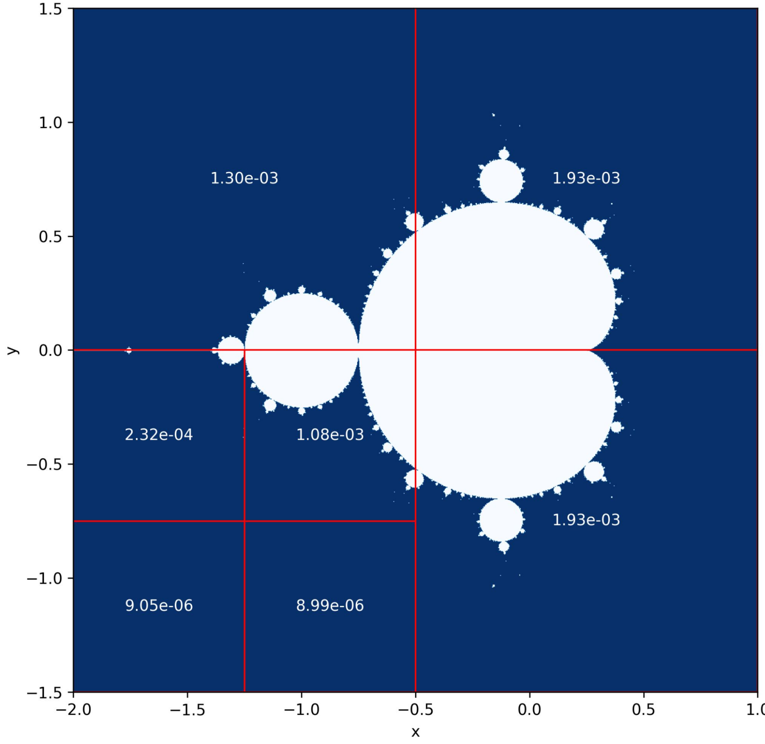
## 1.1 Uncertainties

$$\sigma = \sqrt{\sum \sigma_i^2}$$

Checked wald uncertainties:

- uncertainty for full area
  3.36e-3
- total uncertainty of different sized tiles
  3.21e-3

for same sample (N = 1e6)



ERUM DATA HUB

## 2. GPU Implementation

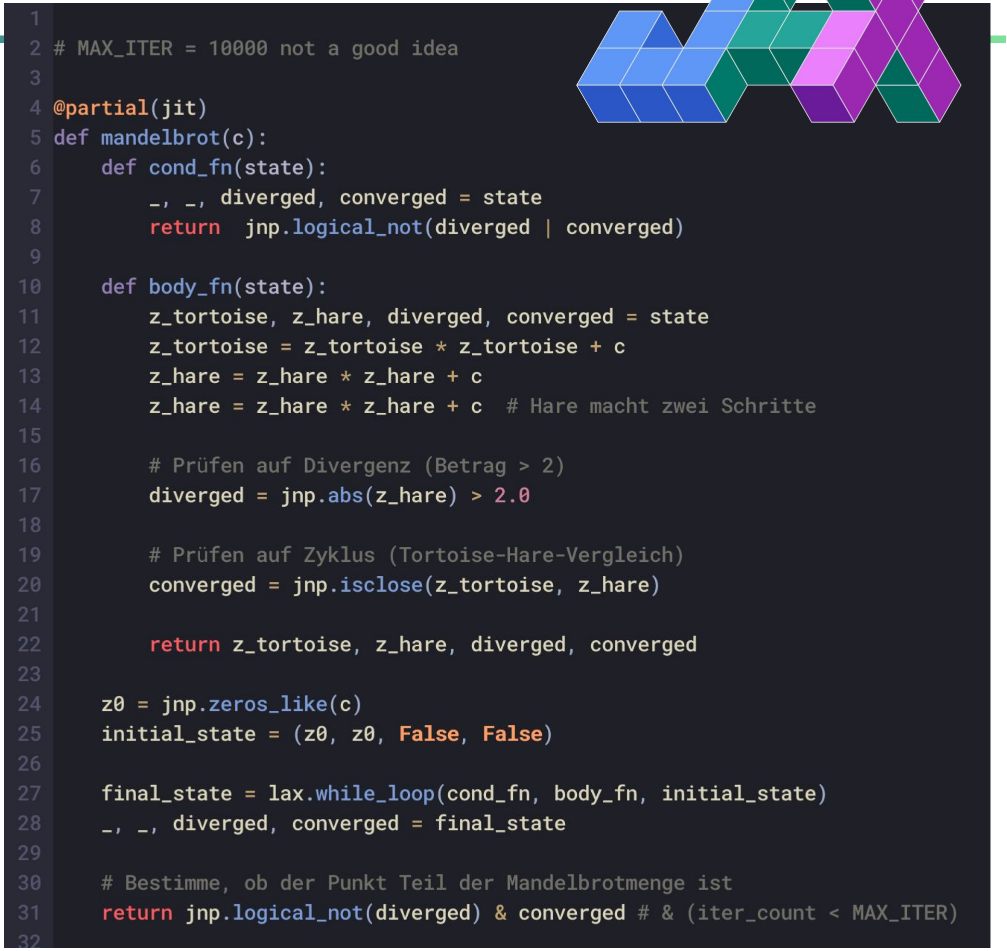
- condition-function
    - check if diverged/converged

- body-function
    - does the tortoise/hare technique
    - actually computes the condition

- use jax.lax.while_loop

- tried to implement max_iter but that would have given wrong results
    - need to wait for some points for a long time

Tile 90: 1M samples ~ 5s
Tile 100: 1M samples > 5 minutes



```python

# MAX_ITER = 10000 not a good idea

@partial(jit)
def mandelbrot(c):
    def cond_fn(state):
        _, _, diverged, converged = state
        return  jnp.logical_not(diverged | converged)

    def body_fn(state):
        z_tortoise, z_hare, diverged, converged = state
        z_tortoise = z_tortoise * z_tortoise + c
        z_hare = z_hare * z_hare + c
        z_hare = z_hare * z_hare + c  # Hare macht zwei Schritte

        # Prüfen auf Divergenz (Betrag > 2)
        diverged = jnp.abs(z_hare) > 2.0

        # Prüfen auf Zyklus (Tortoise-Hare-Vergleich)
        converged = jnp.isclose(z_tortoise, z_hare)

        return z_tortoise, z_hare, diverged, converged

    z0 = jnp.zeros_like(c)
    initial_state = (z0, z0, False, False)

    final_state = lax.while_loop(cond_fn, body_fn, initial_state)
    _, _, diverged, converged = final_state

    # Bestimme, ob der Punkt Teil der Mandelbrotmenge ist
    return jnp.logical_not(diverged) & converged # & (iter_count < MAX_ITER)
```

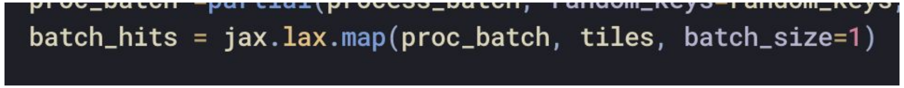
## Results

ERUM DATA HUB

## 3. Scaling Out

Plan:

- Run multiple jobs on different tiles
- scale-out with jax.vmap, jax.lax.map, jax.scan….

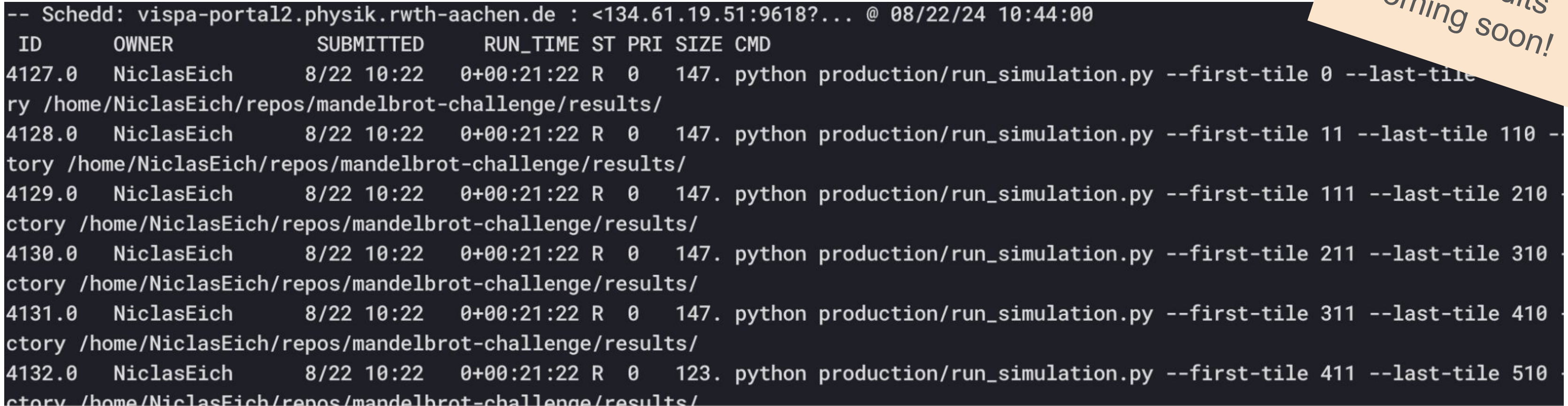


```
proc_batch =partial(process_batch, random_keys=random_keys,
batch_hits = jax.lax.map(proc_batch, tiles, batch_size=1)
```

Unfortunately

- JAX steals 75% of VRAM as default…
- VISPA job-killing mechanism currently off

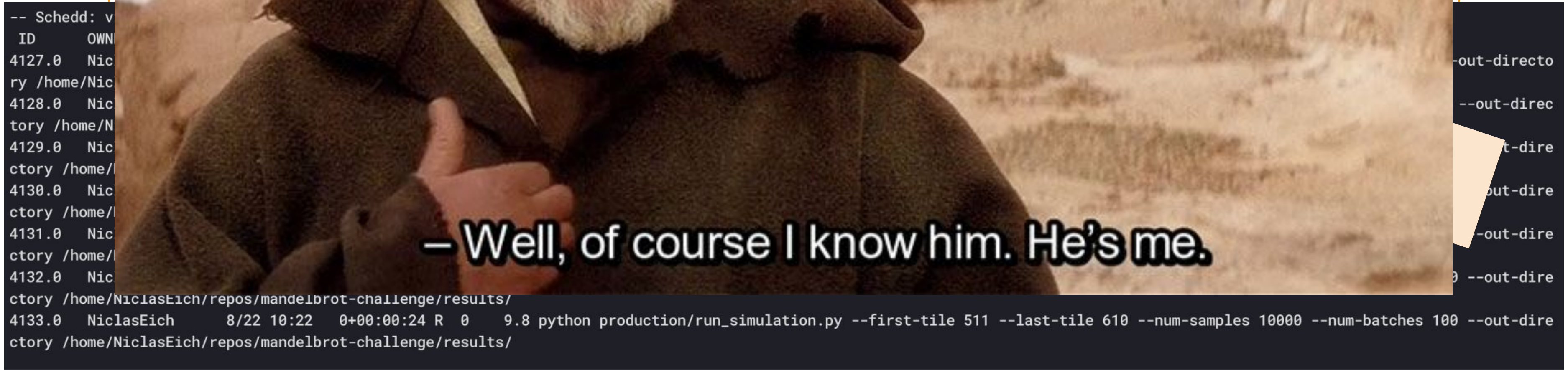
**XLA_PYTHON_CLIENT_PREALLOCATE=false**



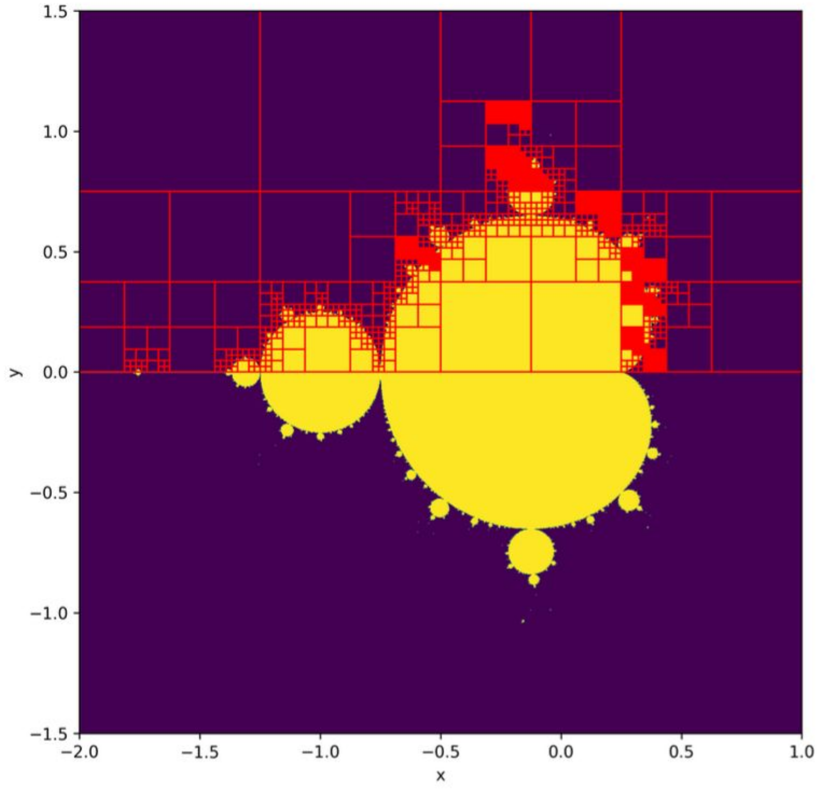
```
-- Schedd: vispa-portal2.physik.rwth-aachen.de : <134.61.19.51:9618?... @ 08/22/24 10:44:00
 ID       OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
4127.0    NiclasEich      8/22 10:22   0+00:21:22 R  0    147. python production/run_simulation.py --first-tile 0 --last-tile
ry /home/NiclasEich/repos/mandelbrot-challenge/results/
4128.0    NiclasEich      8/22 10:22   0+00:21:22 R  0    147. python production/run_simulation.py --first-tile 11 --last-tile 110 -
tory /home/NiclasEich/repos/mandelbrot-challenge/results/
4129.0    NiclasEich      8/22 10:22   0+00:21:22 R  0    147. python production/run_simulation.py --first-tile 111 --last-tile 210
ctory /home/NiclasEich/repos/mandelbrot-challenge/results/
4130.0    NiclasEich      8/22 10:22   0+00:21:22 R  0    147. python production/run_simulation.py --first-tile 211 --last-tile 310
ctory /home/NiclasEich/repos/mandelbrot-challenge/results/
4131.0    NiclasEich      8/22 10:22   0+00:21:22 R  0    147. python production/run_simulation.py --first-tile 311 --last-tile 410
ctory /home/NiclasEich/repos/mandelbrot-challenge/results/
4132.0    NiclasEich      8/22 10:22   0+00:21:22 R  0    123. python production/run_simulation.py --first-tile 411 --last-tile 510
ctory /home/NiclasEich/repos/mandelbrot-challenge/results/
```

BIG results coming soon!

Our Jobs killing our jobs like



```
-- Schedd: v
 ID      OWN
4127.0   Nic                                                          -out-directo
ry /home/Nic
4128.0   Nic                                                          --out-direc
tory /home/N
4129.0   Nic                                                          t-dire
ctory /home/
4130.0   Nic                                                          out-dire
ctory /home/
4131.0   Nic                                                          -out-dire
ctory /home/
4132.0   Nic                                                          0 --out-dire
ctory /home/NiclasEich/repos/mandelbrot-challenge/results/
4133.0   NiclasEich      8/22 10:22   0+00:00:24 R  0    9.8 python production/run_simulation.py --first-tile 511 --last-tile 610 --num-samples 10000 --num-batches 100 --out-dire
ctory /home/NiclasEich/repos/mandelbrot-challenge/results/
```



number of samples = 98,863,500

area = 1.415 +- 0.026

110 failed jobs with no sampling!

ERUM DATA HUB

## Different result estimation

Original implementation: $1.50638855 \pm 1.38e-4$, n_samples: 5e8

Numba Cuda implementation: 1.512369 ± 0.00077 , n_samples: 1e6

1.509 ± 7.36e-5, n_samples: 1e7

1.5095 ± 7.26e-6, n_samples: 1e8

1.51028 ± 7.37e-5, n_samples: 1e9

Recursive on CPU: 1.507 +/- 0.007, n_samples: 1522000

## Outlook

- Fix VISPA (You, Niclas!)
- figure out how to calculate uncertainty / confidence intervals correctly
- improve algorithm: uncertainty threshold instead of fixed number of samples