Parallel Computations for Large-Scale Problems

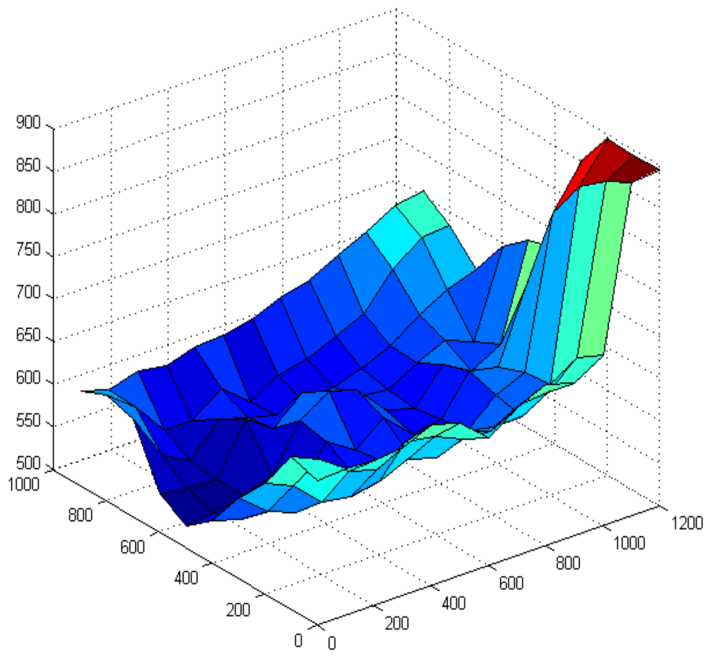
Fractal Terrain Generation

Rémi Domingues

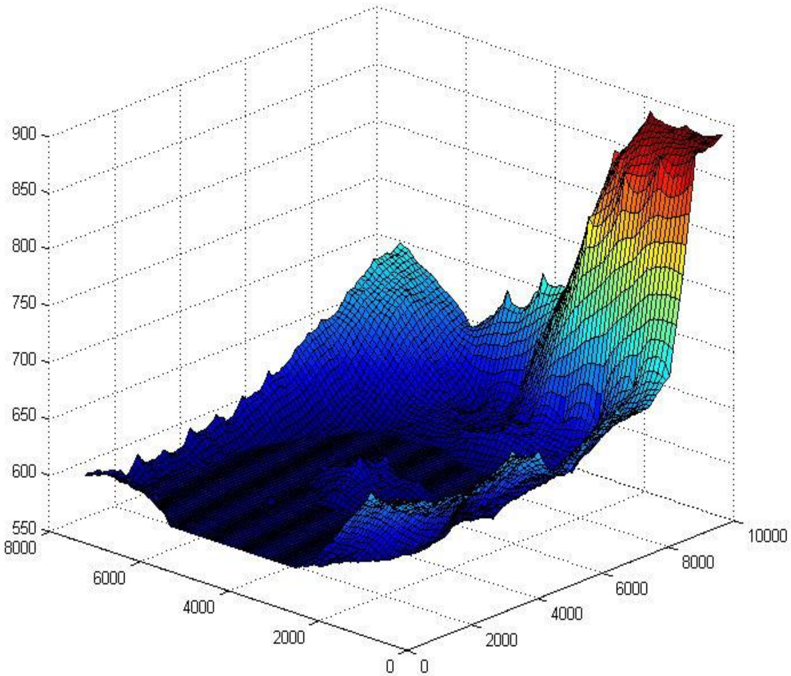
Giacomo Giudice

Diamond-Square

Generating a realistic 2D height map by increasing resolution at each iteration

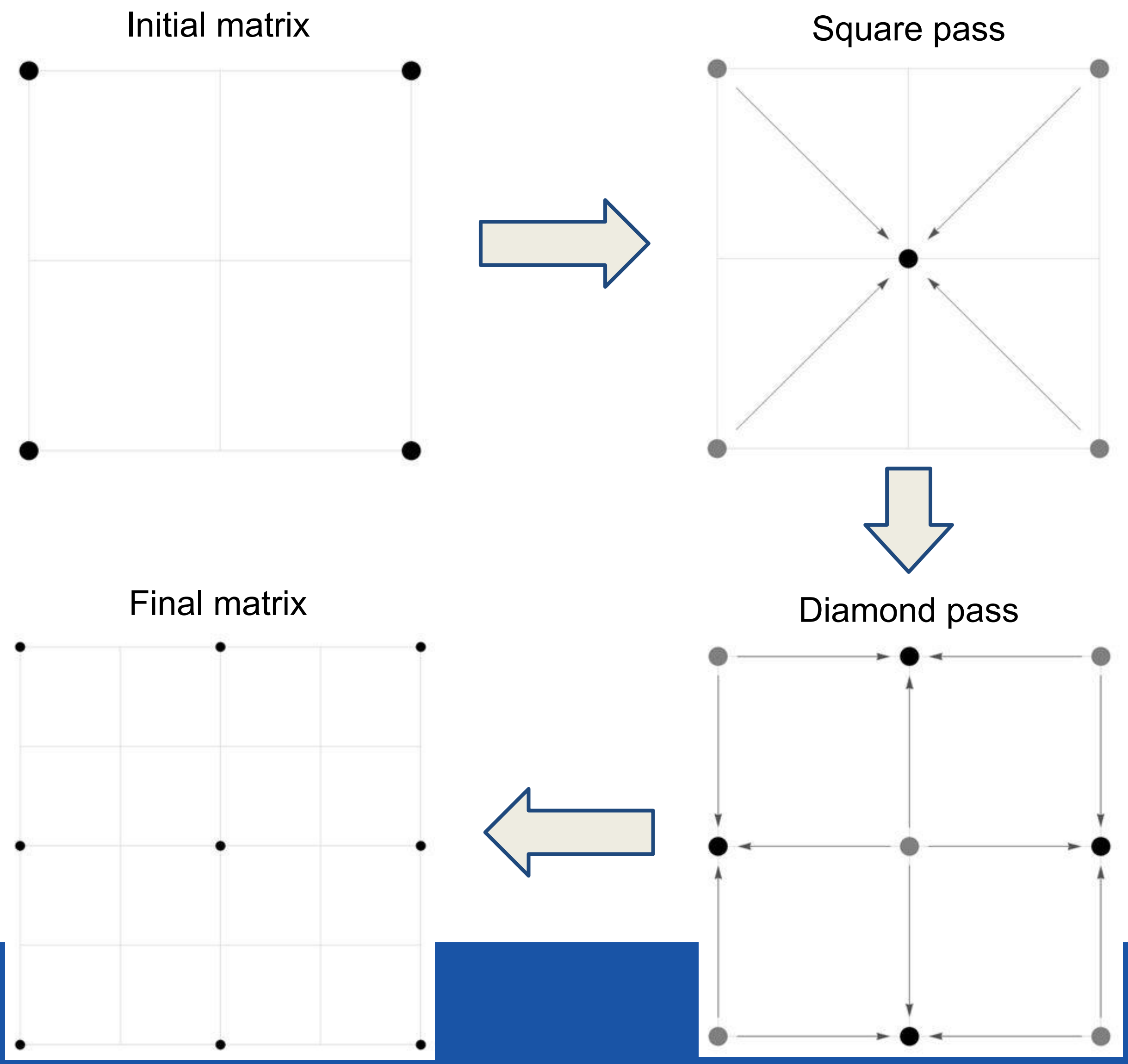


Initial terrain (13x10)



Final terrain - 2 iterations (49x37)

Diamond-Square - Sequential algorithm



Diamond-Square - Tricks

- Use only one matrix of final size at iteration K

$$(width_K, height_K) = 2^K((width_0, height_0) - 1) + 1$$

- Use virtual coordinates, according to the transformation

$$(i_K, j_K) = 2^{K-k}(i_k, j_k)$$

- Randomness boundaries

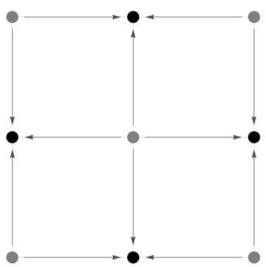
$$0 \leq r \leq scale_k \qquad scale_{k+1} = \frac{scale_k}{2}$$

Diamond-Square - Parallelization

- 2D processes topology (N = PxQ)
- Scatter data with overlaps (L = width = height)

$$L = \frac{L + \sqrt{N} - 1}{\sqrt{N}} + \mathbb{1}_{rank<(L+\sqrt{N}-1)} \mod \sqrt{N}$$

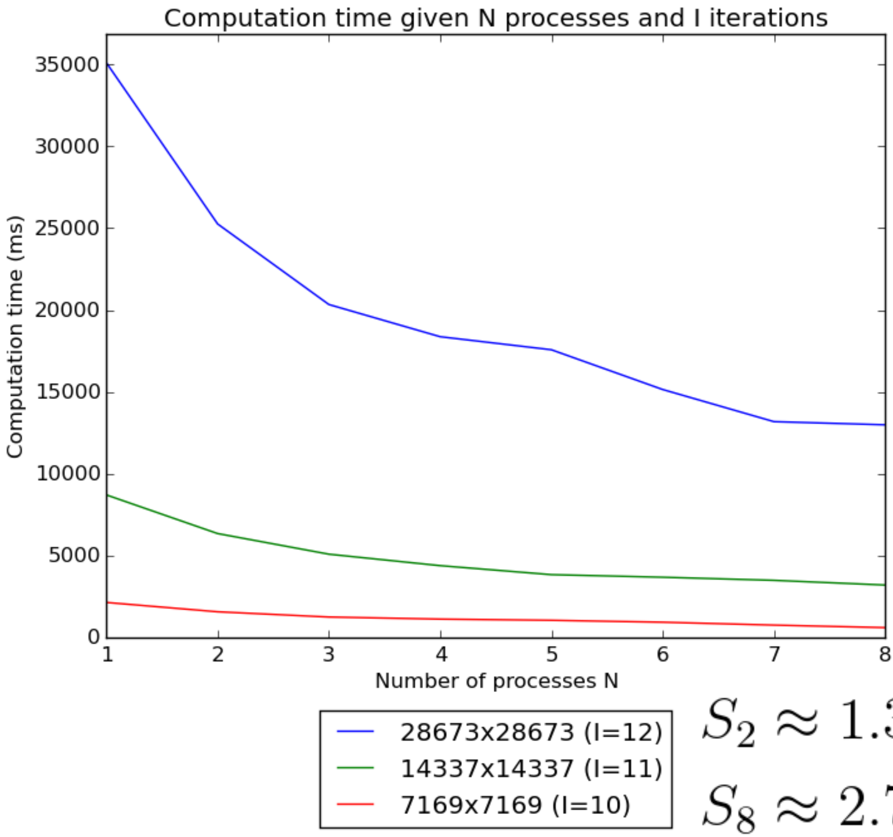
- Diamond pass
    - Ghost cells from the square pass: $\frac{L_k - 1}{2}$
    - Red-Black communications



- Overall complexity: $\mathcal{O}\left(\frac{L^2}{N}\right)$

Diamond-Square - Benchmarks

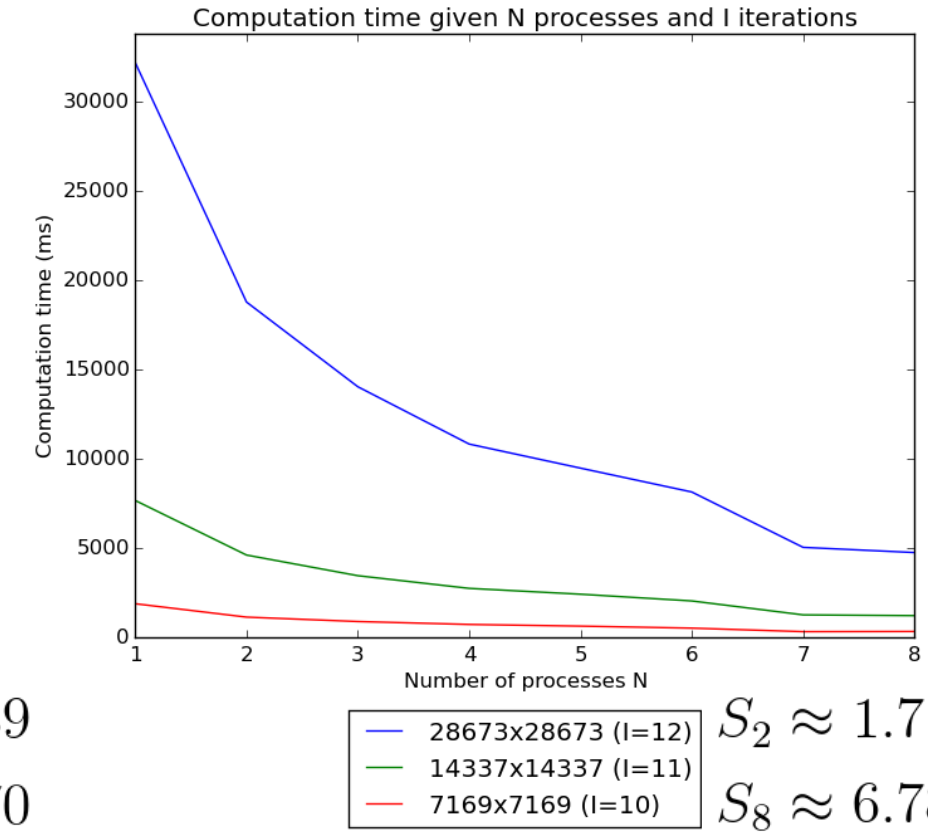
Complete generation



$$S_2 \approx 1.39$$

$$S_8 \approx 2.70$$

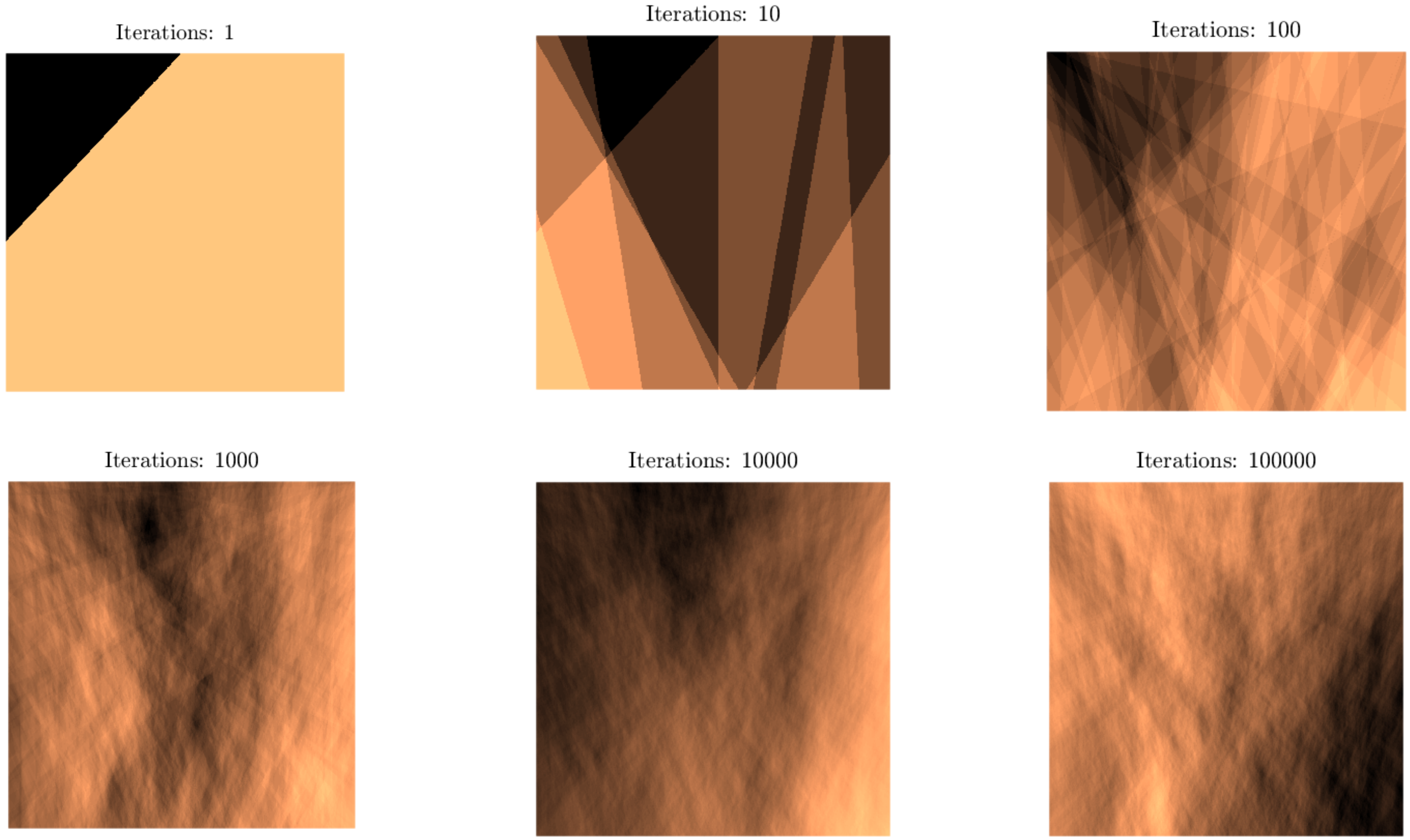
Without gathering data

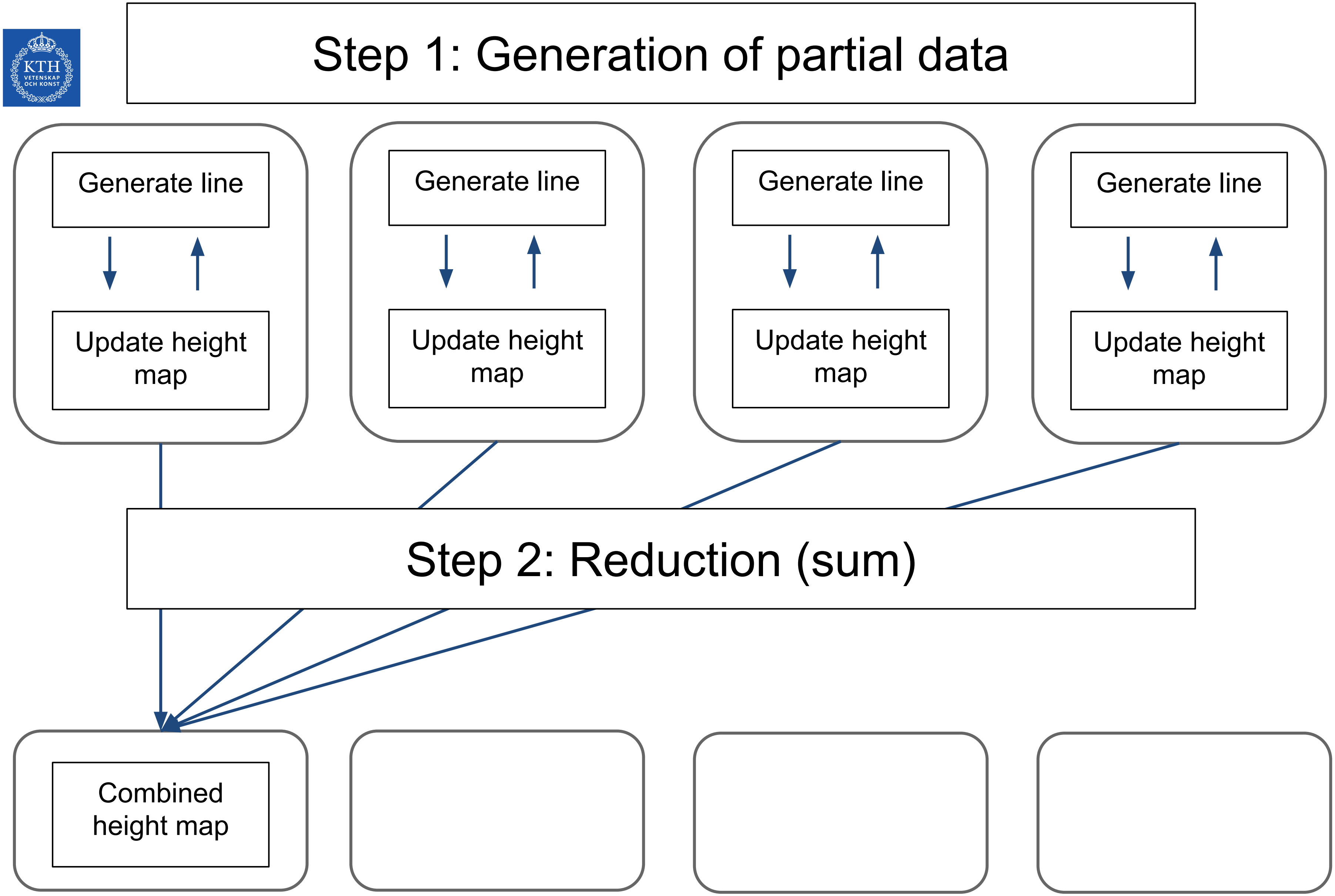


$$S_2 \approx 1.71$$

$$S_8 \approx 6.78$$

Linear Displacement

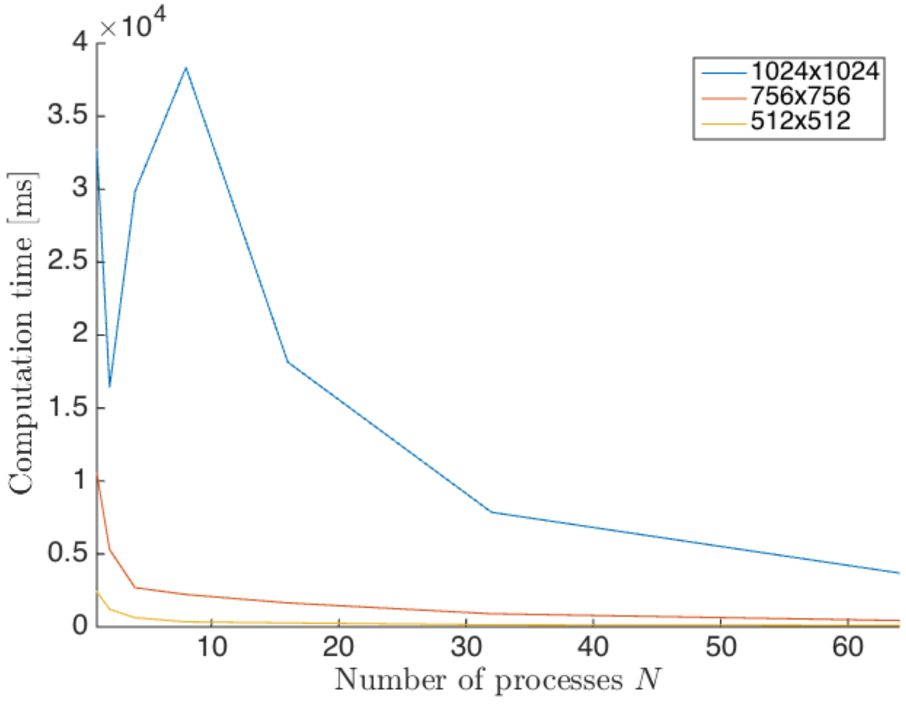


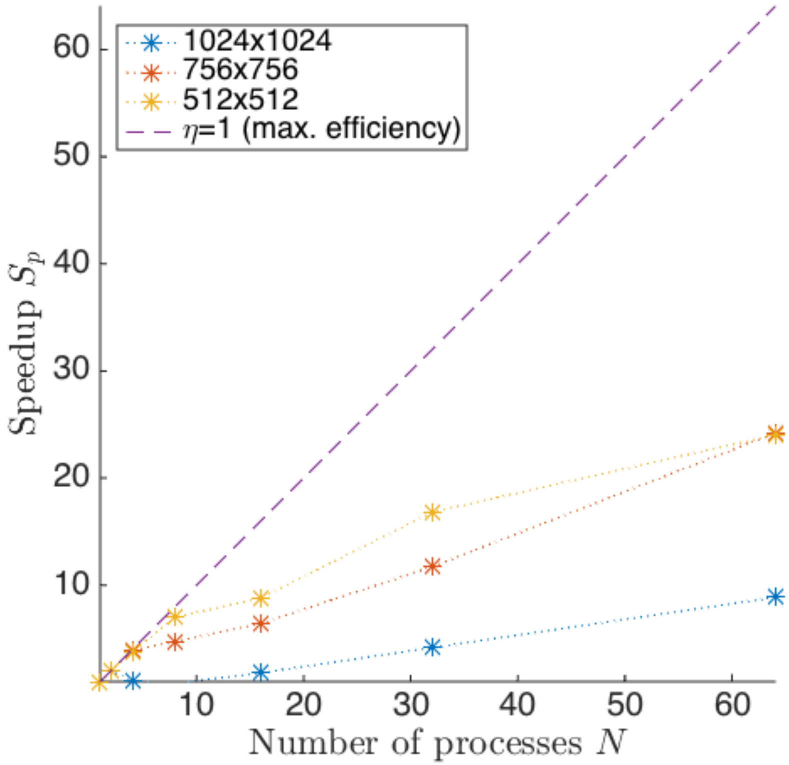


Step 1: Generation of partial data
Generate line
Update height map
Generate line
Update height map
Generate line
Update height map
Generate line
Update height map
Step 2: Reduction (sum)
Combined height map

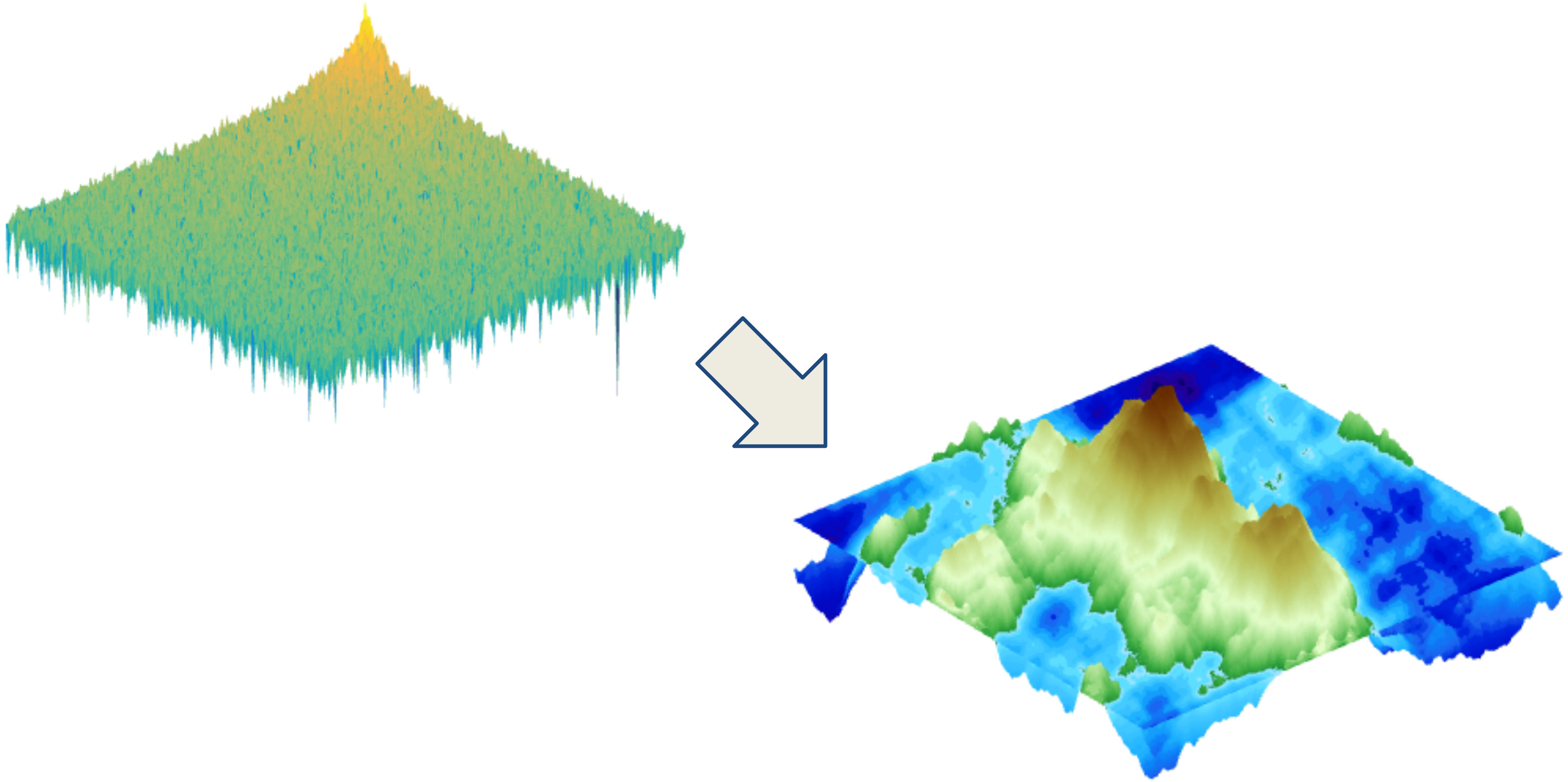
Linear Displacement - Performance

Complexity $$\underbrace{\mathcal{O}(qL^2/N)\mathcal{O}(L^2/2)}_{\text{iterations}} + \underbrace{\mathcal{O}(L^2 \log N)}_{\text{reduce}} = \underbrace{\mathcal{O}(L^4/N)}_{\text{overall}}$$

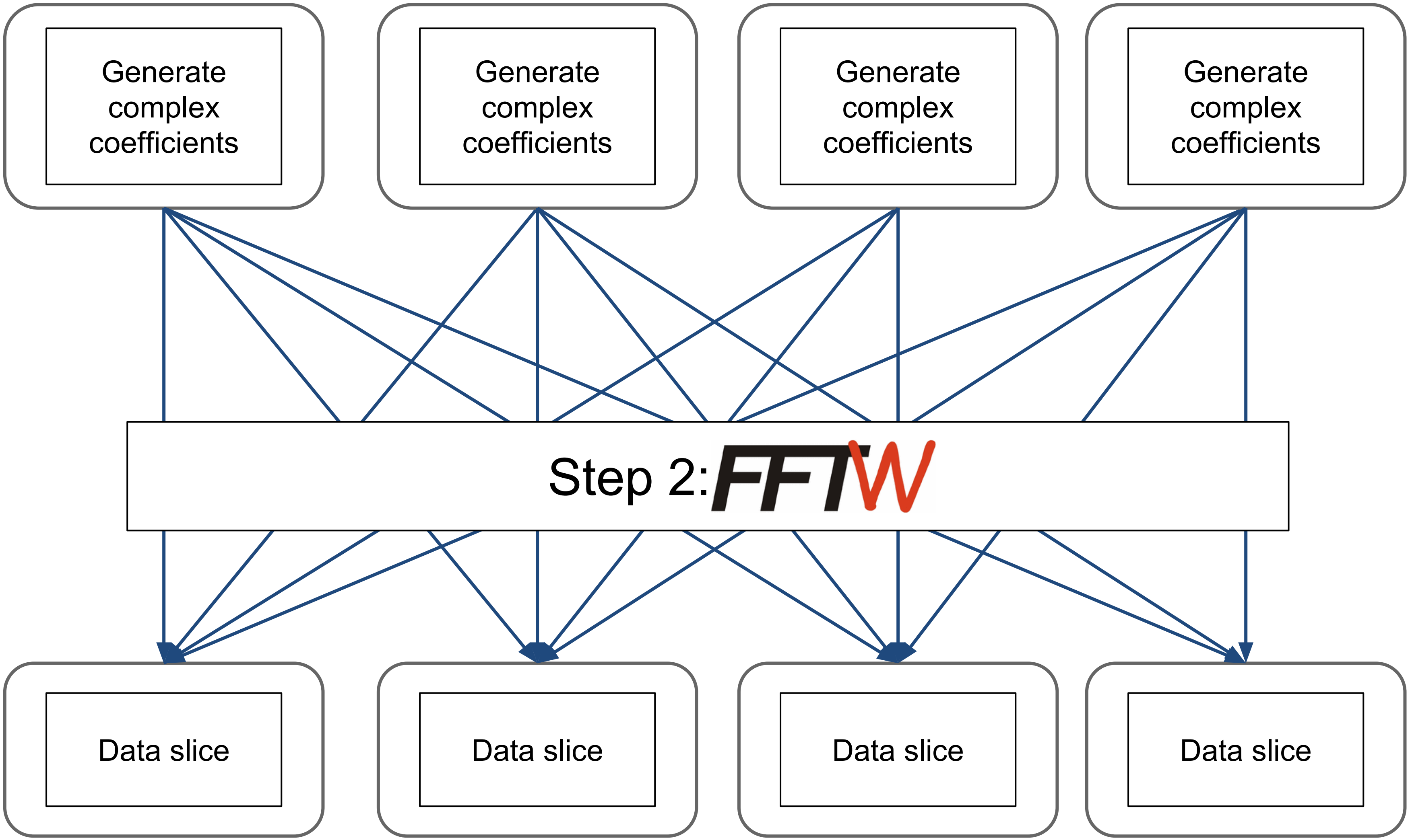




Fast Fourier Transform



Step 1: Generate noise in frequency space



Fast Fourier Transform - Performance

Complexity

$$\underbrace{\mathcal{O}(qL^2/N)\mathcal{O}(L^2/2)}_{\text{iterations}} + \underbrace{\mathcal{O}(L^2 \log N)}_{\text{reduce}} = \underbrace{\mathcal{O}(L^4/N)}_{\text{overall}}$$