SINGLE PHOTON DOUBLE SLIT SIMULATION

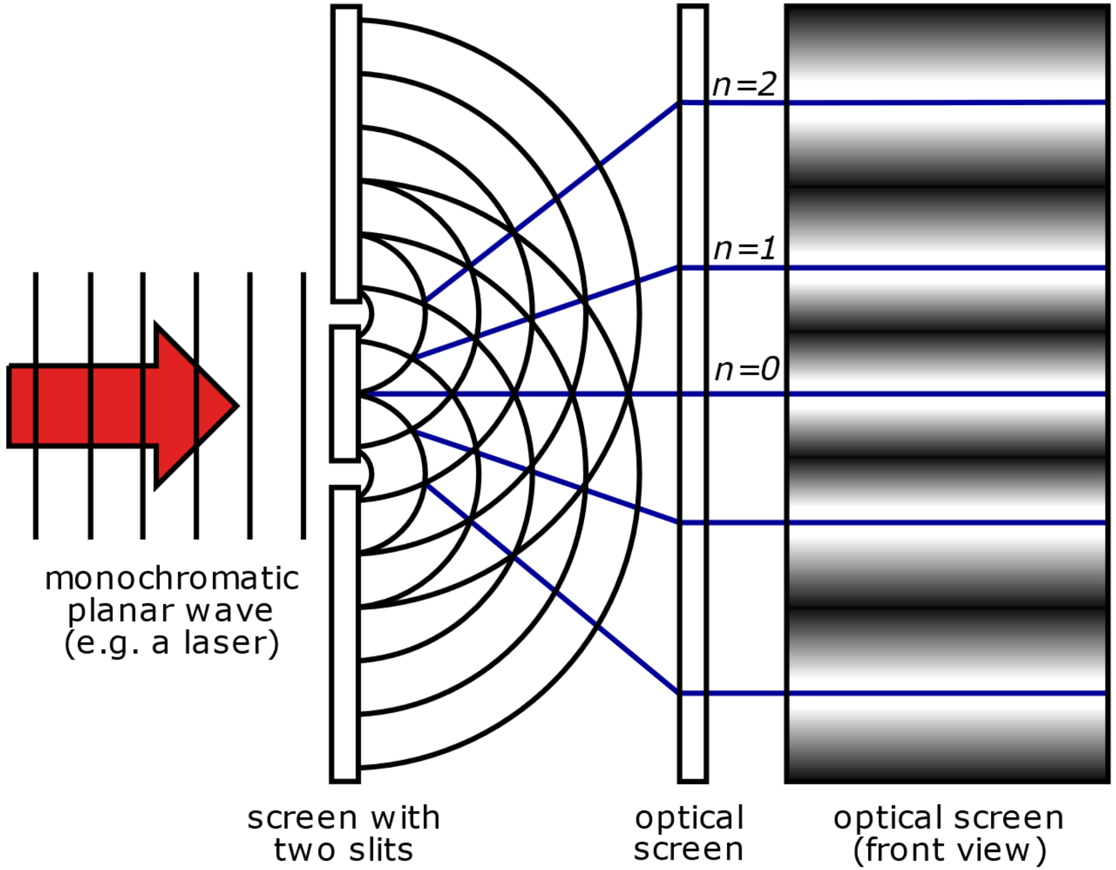
Seminar Talk:

Parallelisation of Physics Calculations on GPUs with Cuda

by **Xianrui Yin**

CONTENTS

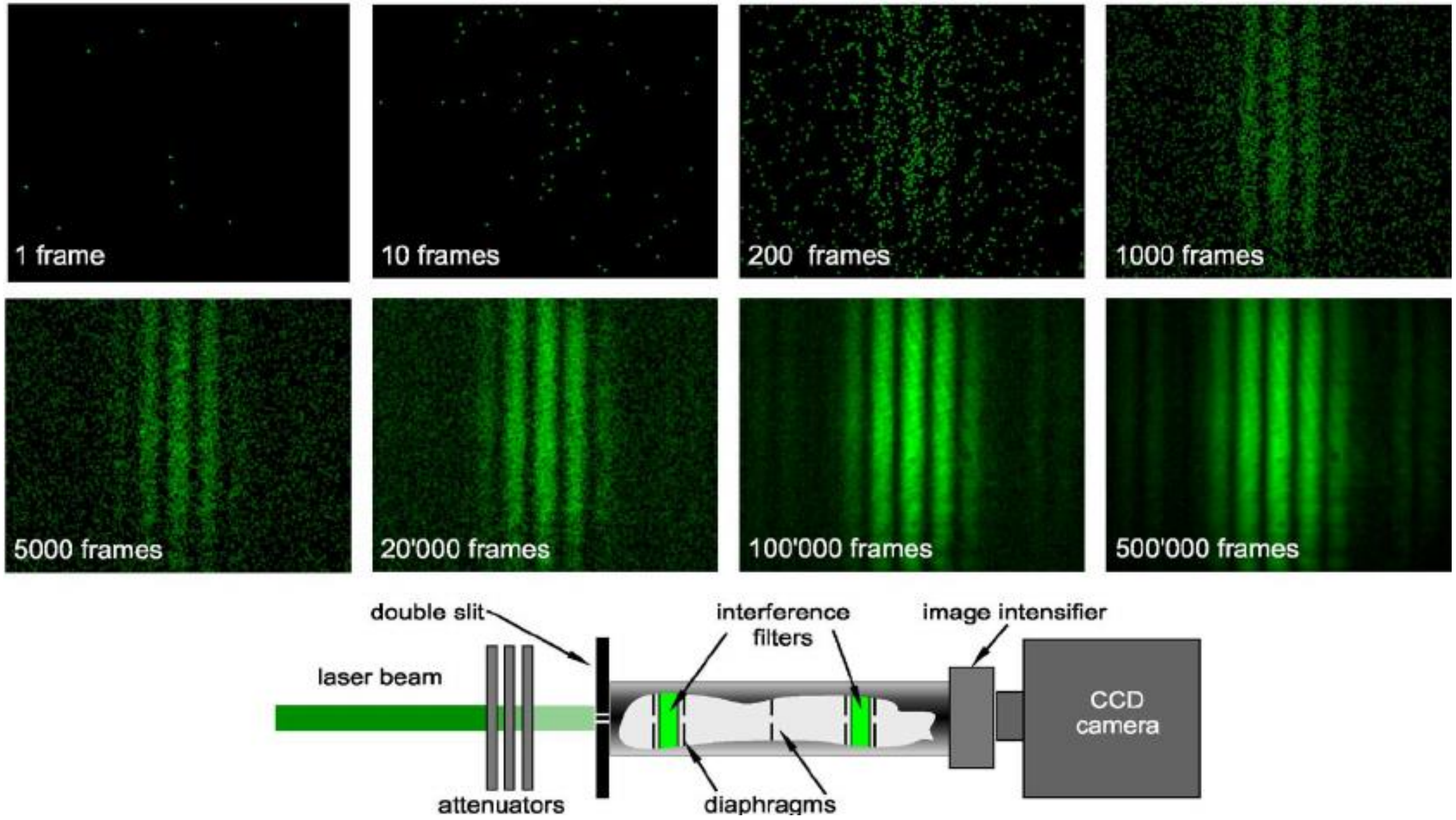
- Double slit experiment (*wave v.s. particle*)
- Simulation based on physics argument
- Parallelisation on GPU
- Why it works?
- What else could be done?



[1]

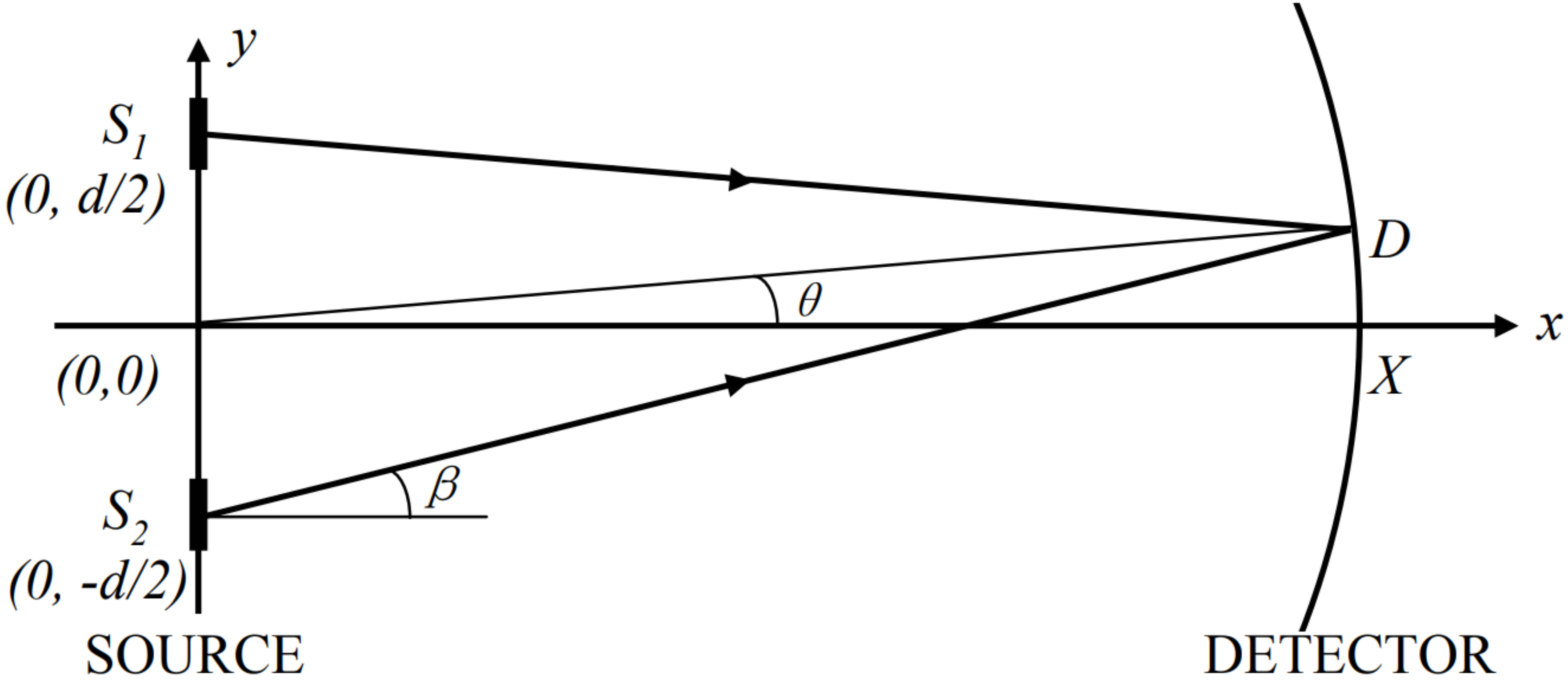
DOUBLE SLIT EXPERIMENT

SINGLE PHOTON



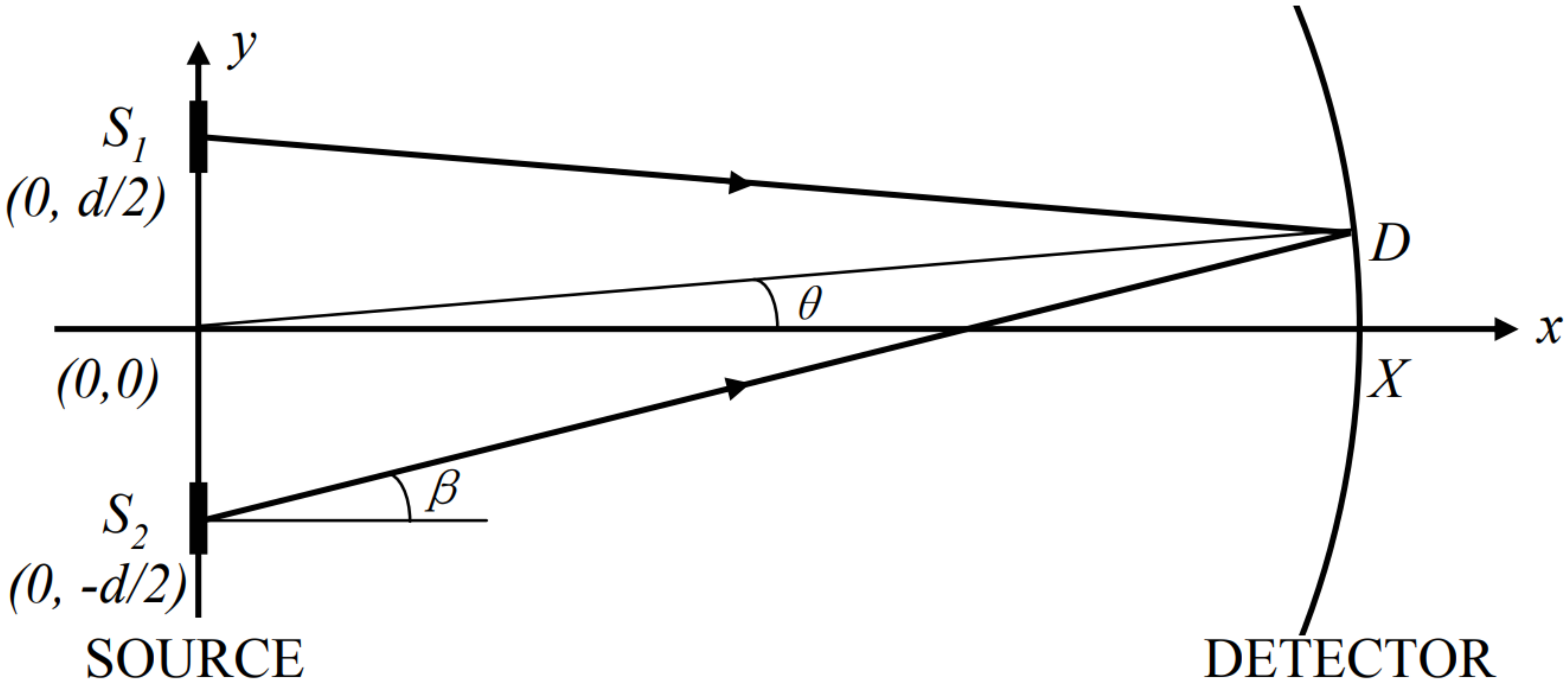
[2]

SET UP THE LAB



[3]

The photon follows a straight line path
and hit one of the detectors on the screen



2M
..
..
..
..
..
..
..
2
1
0

[3]

- Photons are oscillating

$$e = [cos\varphi,\ sin\varphi]$$

- Each detector has a polarization vector

$$p = [p_x,\ p_y]$$

- When photon lands on a detector, it changes its polarization

$$p_{i+1} = \ \gamma p_i + (1-\gamma)e_i$$

- A photon is detected if the threshold is met

$$\Theta(p^2 - r) > 0$$

E-S Correspondence

Experiment	Simulation
line source	choose y and beta randomly
photon phase change φ	$e = [cos\varphi,\ sin\varphi]$
polarization of the detector material	$p = [p_x,\ p_y]$
Interaction between photon & detector	$p_{i+1} = \ \gamma p_i + (1-\gamma)e_i$
sensitivity	$p^2 > r, r\ is\ random$

CPU CODE REVIEW

LET’S MOVE TO GPU!

HOW TO PARALLELIZE?

Can we do everything all together?

Sadly not…

The polarization of the detectors must be updated sequentially.

SPLIT THE PROCEDURE INTO 2+1 PARTS!

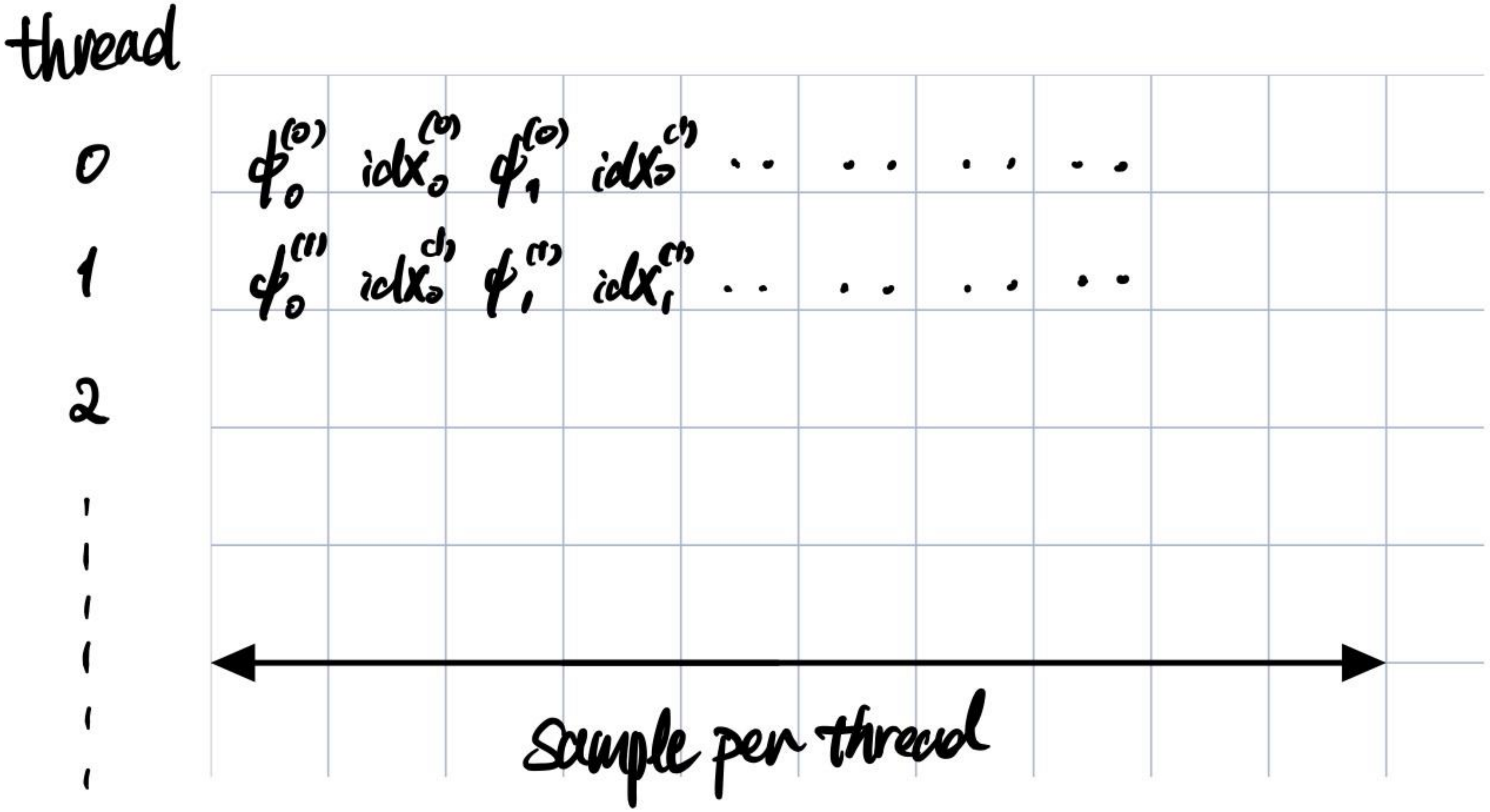
- __global__ void sample(…)
    - Sampling can be done concurrently
    - Let each thread generate some random numbers
    - Record the target detector index and the phase of the photon
- ……

- __global__ void detect(…)
    - All the detectors are independent!
    - Send the pre-generated photons one-by-one

SPLIT THE PROCEDURE INTO 2+1 PARTS!

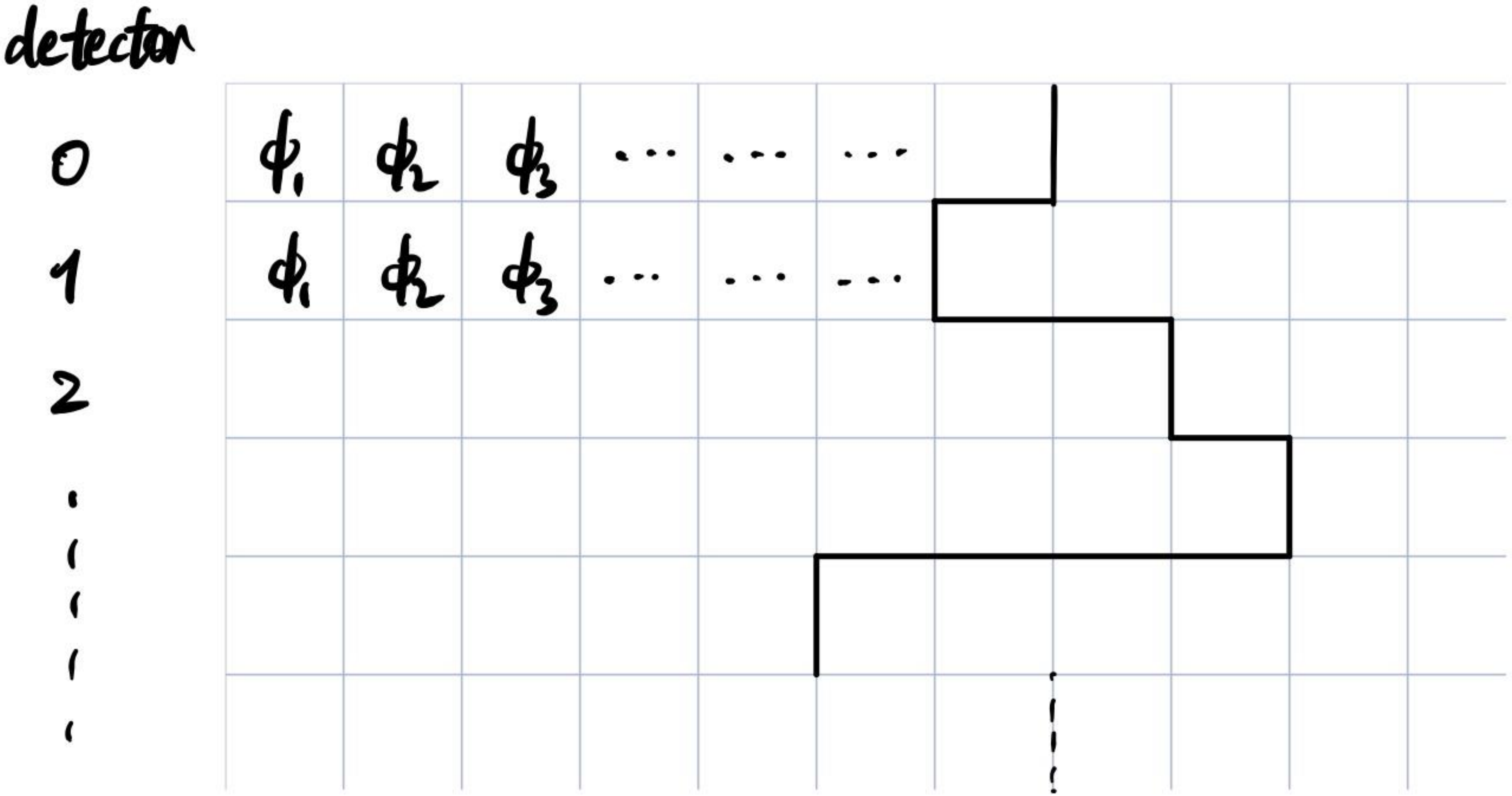
- __global__ void sample(…)
  - Sampling can be done concurrently
  - Let each thread generate some random numbers
  - Record the target detector index and the phase of the photon
- ……how to communicate?
- __global__ void detect(…)
  - All the detectors are independent!
  - Send the pre-generated photons one-by-one

Event array



detector array

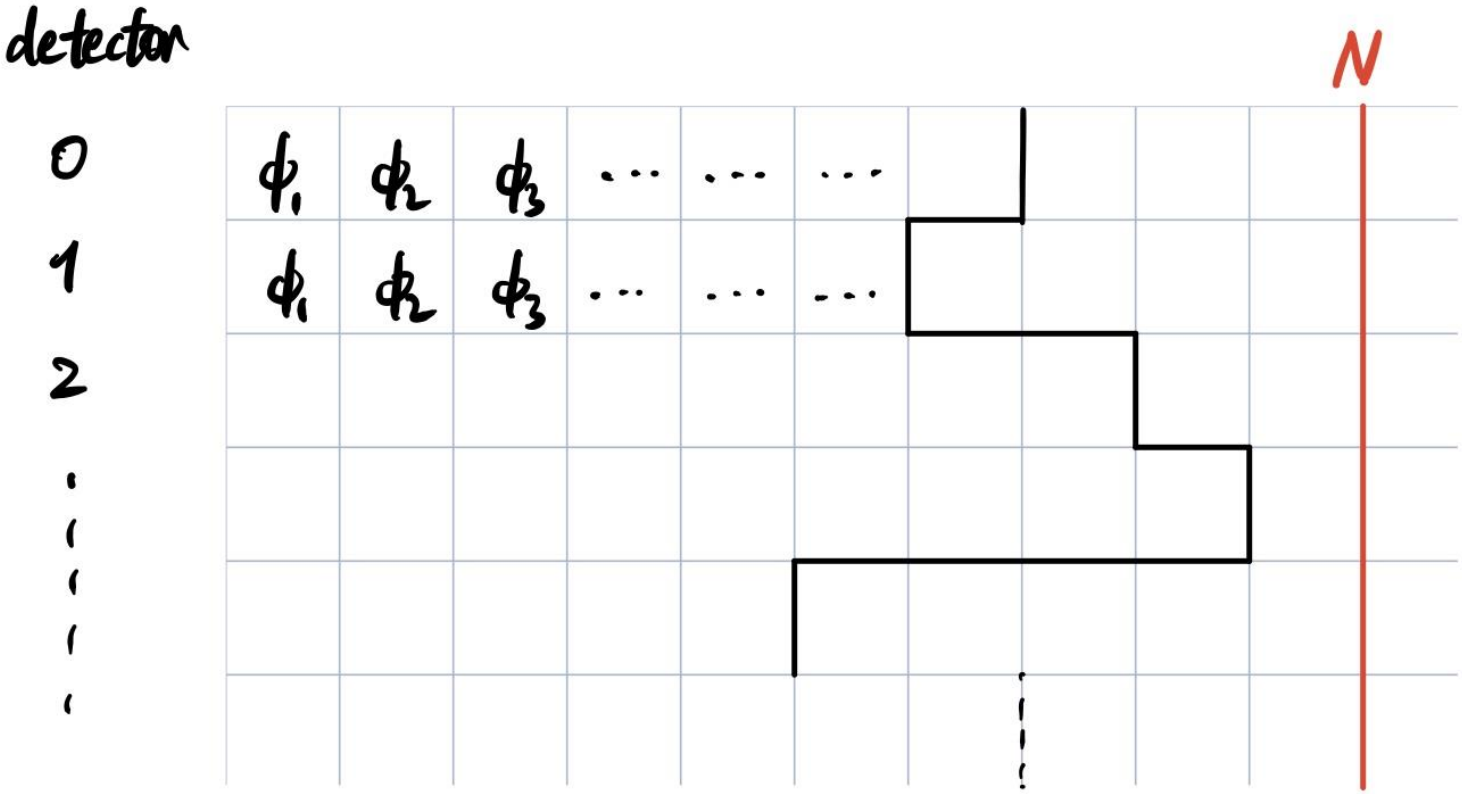
we don't know the length of each row



- Sort<<<1, 1>>>(event, det, len, …)

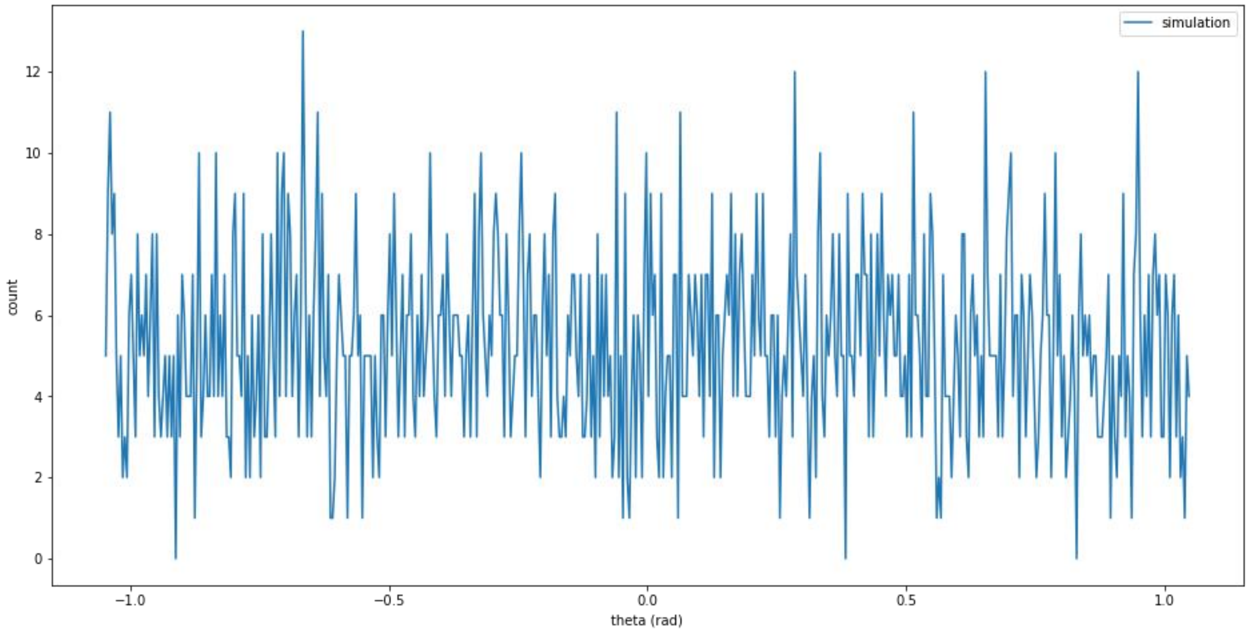
Set a large enough N

Use another 1d array to record the number of photons in each row

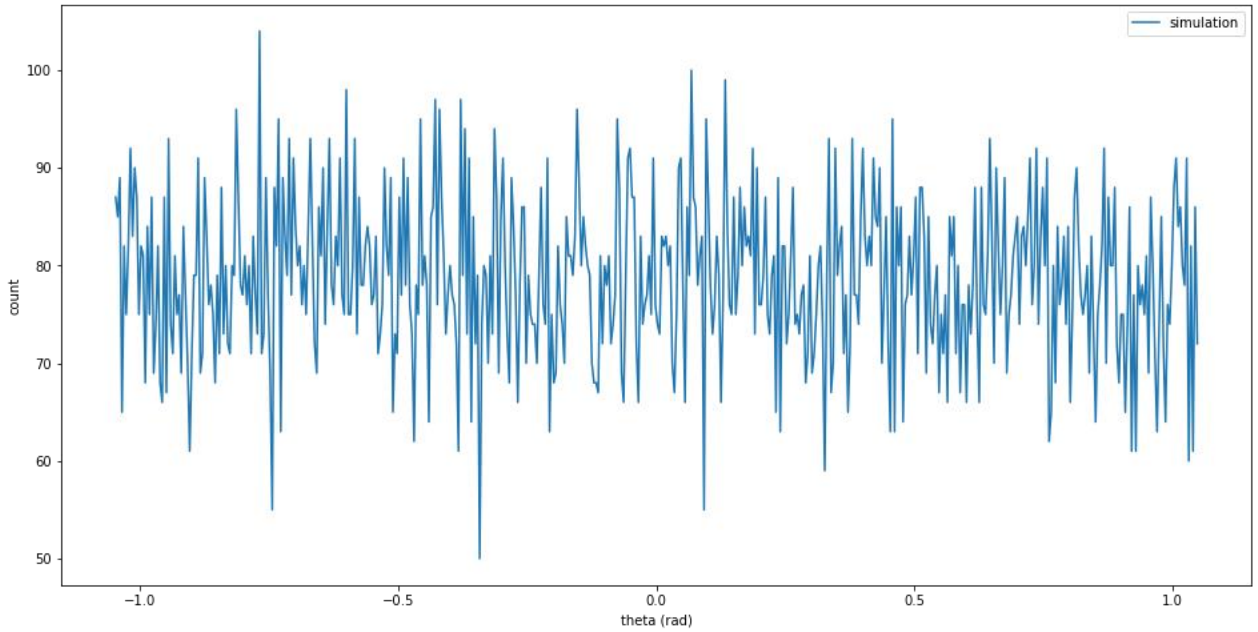


SIMULATION RESULTS

- CPU



Photon aggregation 2^12

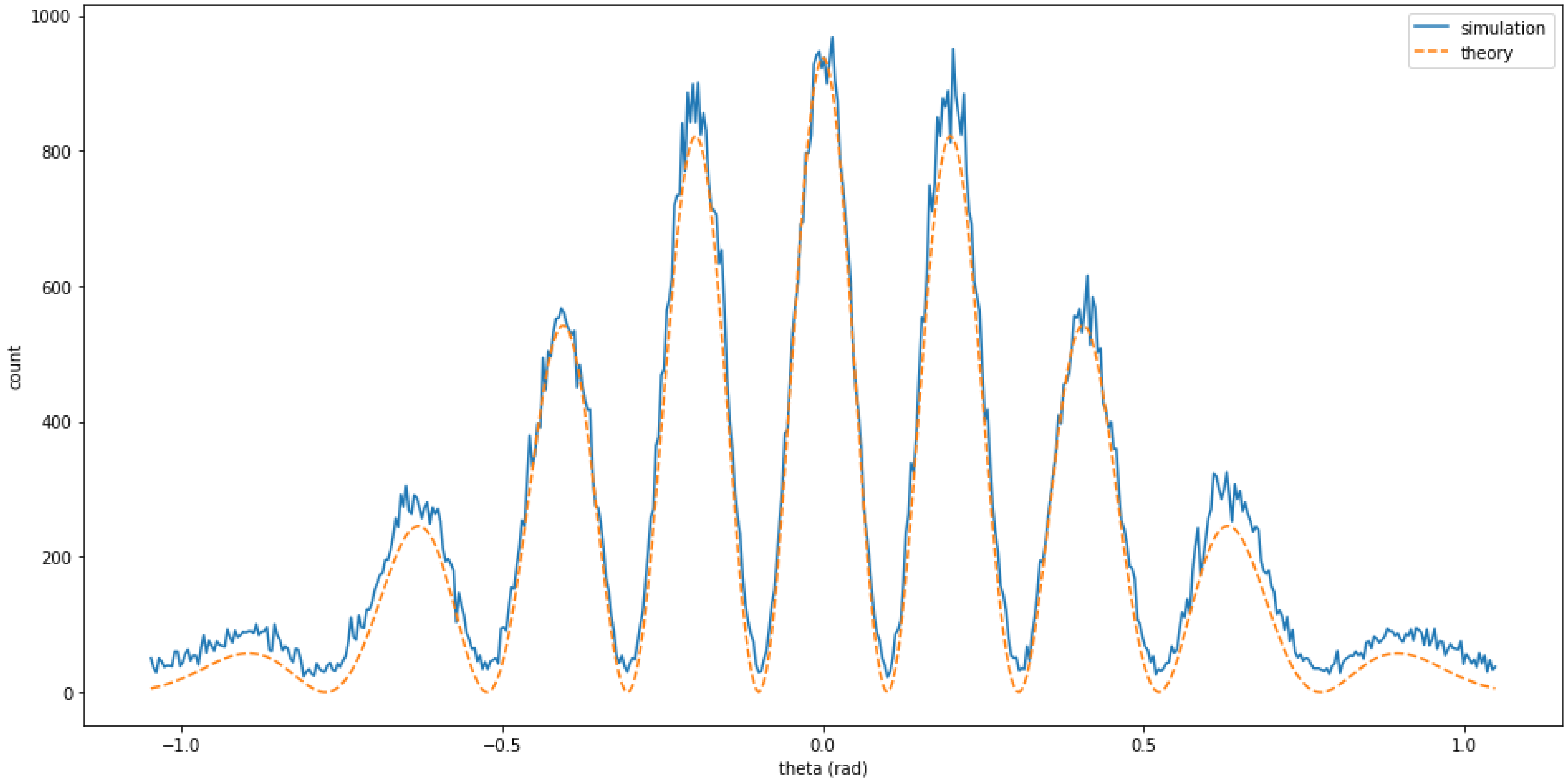


Photon aggregation: 2^16

Photon aggregation **2^21**

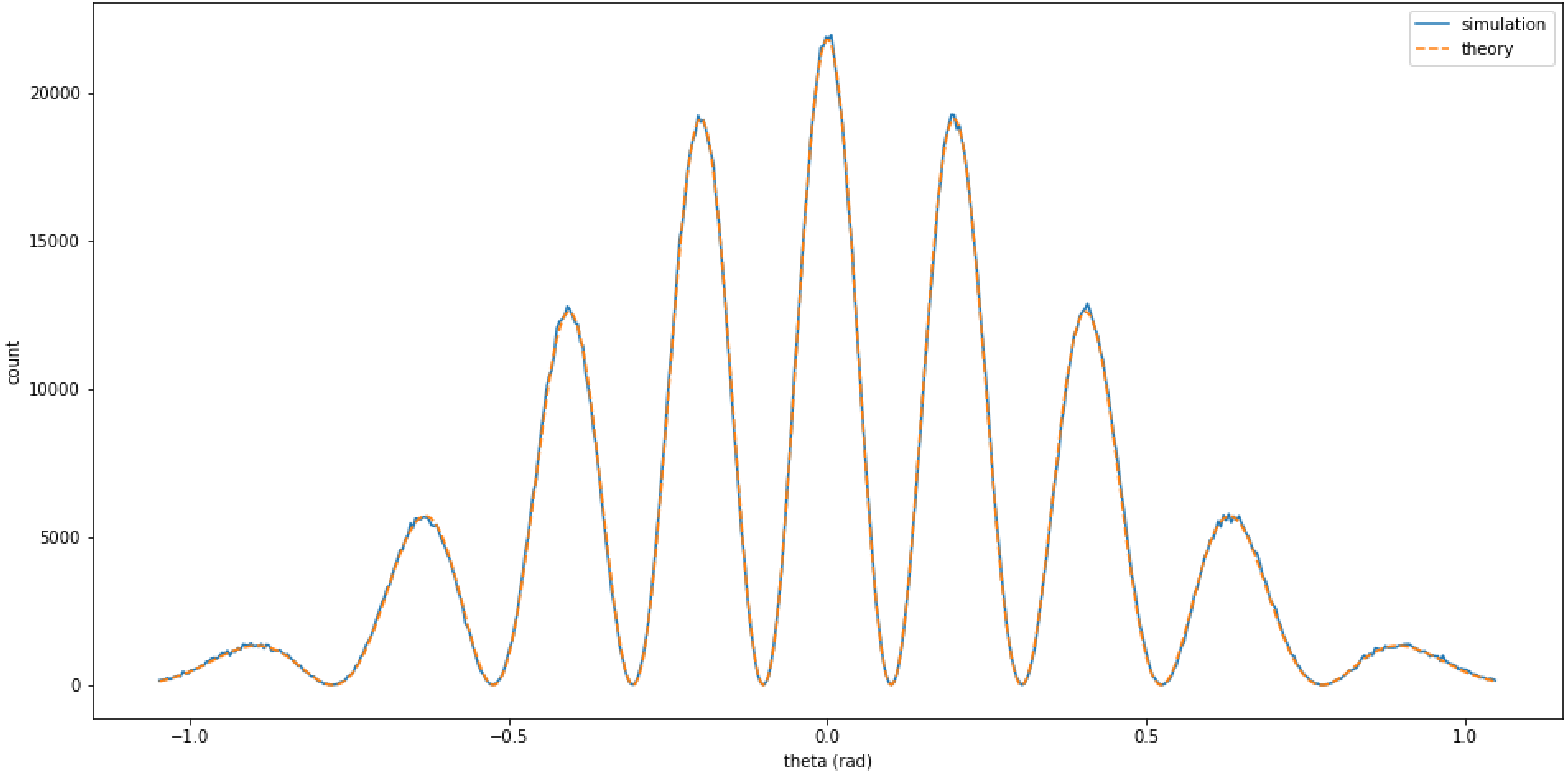
CPU run time: **56.625009059906006**

- As we keep shooting photons

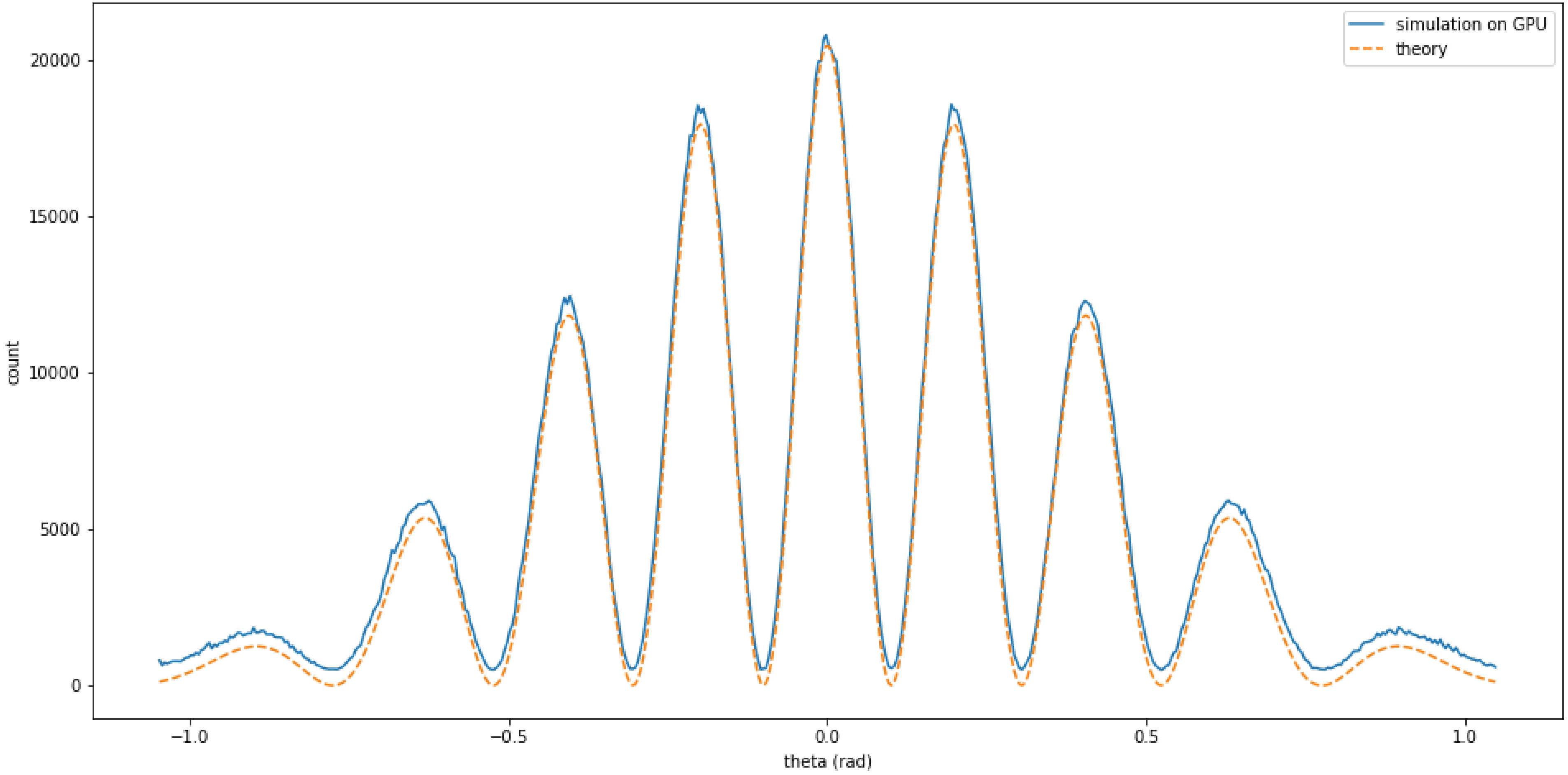


Photon aggregation **2^25**

CPU run time: **793.3088808059692**



Let's do **2^25 photons on GPU!**



THANKS GOOGLE..

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   44C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
==170== NVPROF is profiling process 170, command: ./test60
==170== Profiling application: ./test60
==170== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   91.78%  2.69613s         1  2.69613s  2.69613s  2.69613s  sort(float*, float*, int*, int, int, int)
                    6.10%  179.17ms         1  179.17ms  179.17ms  179.17ms  sample(float*, int, float, float, int)
                    2.12%  62.273ms         1  62.273ms  62.273ms  62.273ms  detect(float*, float*, int*, int*, int, int)
                    0.00%  2.6560us         1  2.6560us  2.6560us  2.6560us  [CUDA memcpy DtoH]
      API calls:   83.62%  2.93757s         1  2.93757s  2.93757s  2.93757s  cudaDeviceSynchronize
                   16.26%  571.13ms         5  114.23ms  2.0700us  570.81ms  cudaMalloc
                    0.08%  2.6445ms         4  661.13us  3.1760us  2.2373ms  cudaFree
                    0.03%  1.0065ms         1  1.0065ms  1.0065ms  1.0065ms  cuDeviceGetPCIBusId
                    0.01%  348.49us         1  348.49us  348.49us  348.49us  cuDeviceTotalMem
                    0.00%  157.86us       101  1.5620us     142ns  67.217us  cuDeviceGetAttribute
                    0.00%  51.012us         1  51.012us  51.012us  51.012us  cudaMemcpy
                    0.00%  47.733us         3  15.911us  5.4950us  34.765us  cudaLaunchKernel
                    0.00%  29.370us         1  29.370us  29.370us  29.370us  cuDeviceGetName
                    0.00%  1.8070us         3     602ns     209ns  1.1570us  cuDeviceGetCount
                    0.00%  1.3050us         2     652ns     185ns  1.1200us  cuDeviceGet
                    0.00%     280ns         1     280ns     280ns     280ns  cuDeviceGetUuid
```

Let’s look at some code!

WHY IT WORKS?

$$p_{i+1} = \gamma p_i + (1-\gamma)e_i$$

- The sequence will converge if $\gamma \to 1^-$to the mean of e_i if $\gamma \to 1^-$
- constructive interference: $\langle e_i \rangle$ is large
- destructive interference: $\langle e_i \rangle$ is small

$$\Theta(p^2 - r) > 0$$

MORE TO EXPLORE

- Transient behavior
- ….

Thank you for your attention!

- [1] https://commons.wikimedia.org/wiki/File:Two-Slit_Experiment_Light.svg
- [2] Dimitrova, Todorka & Weis, Antoine. (2008). The wave-particle duality of light: A demonstration experiment. American Journal of Physics. 76. 10.1119/1.2815364.
- [3] Jin F. Towards a corpuscular model of optical phenomena. Groningen: s.n., 2011. 114 p.