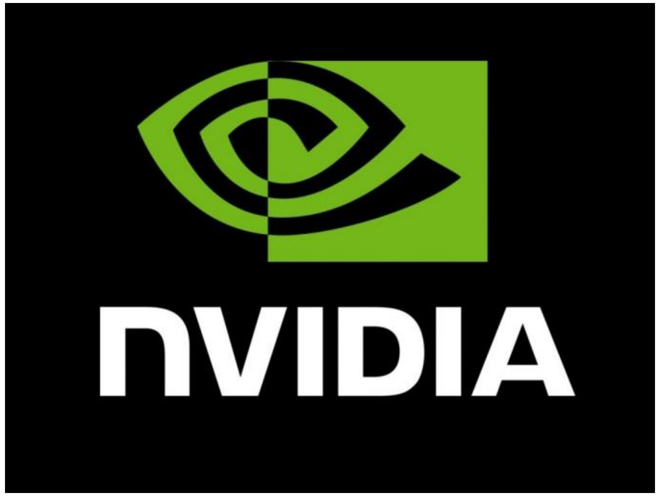
HPC: Implementing and benchmarking CPU algorithms for Nvidia GPUs

Author: **Daniel Álvarez Conde**
Supervisors: **Jonas Rembser / Lorenzo Moneta**

15 August, 2023

Introduction

Framework for data processing, focused on High-Energy Physics



CUDA
(Compute Unified Device Arquitecture)

Negative Log Likelihood (NLL)

$$\sum_{i=1}^{n} -\log(P_i)$$

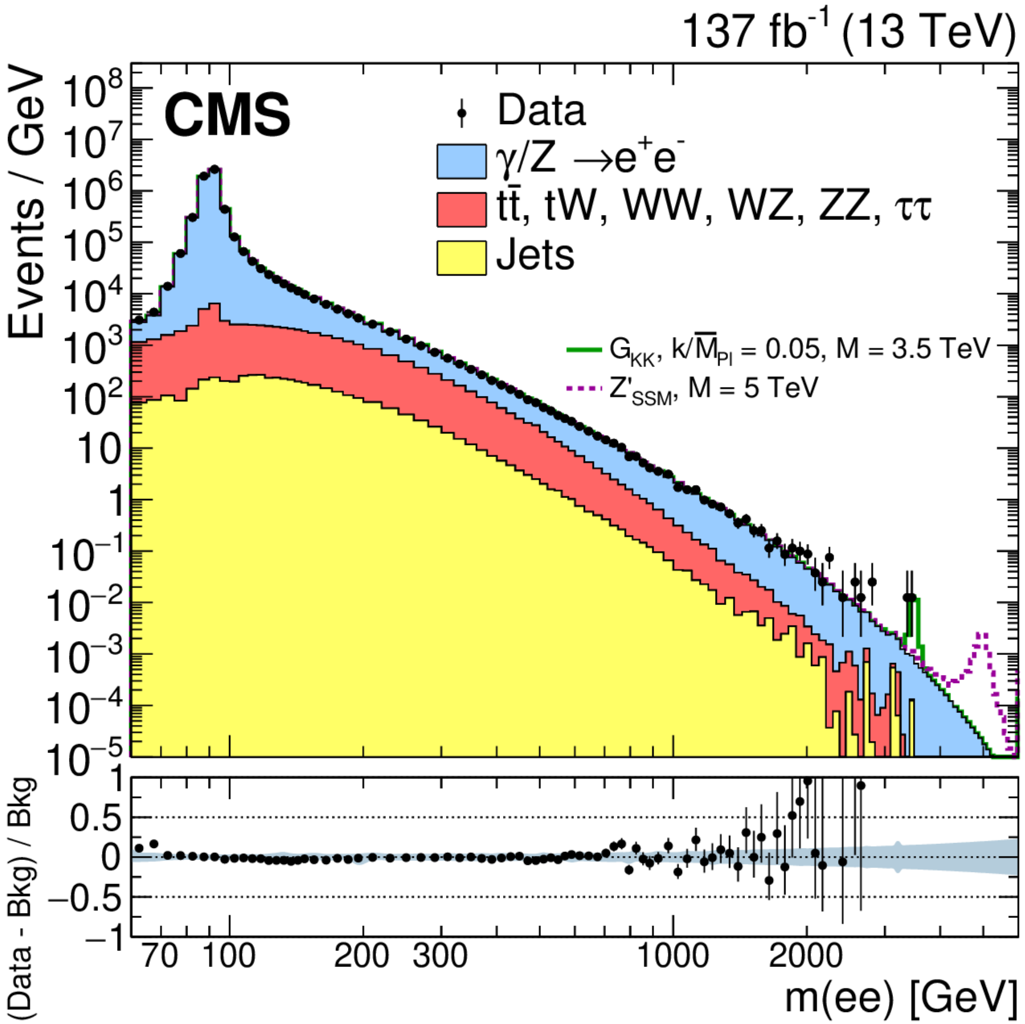
Unbinned Fits

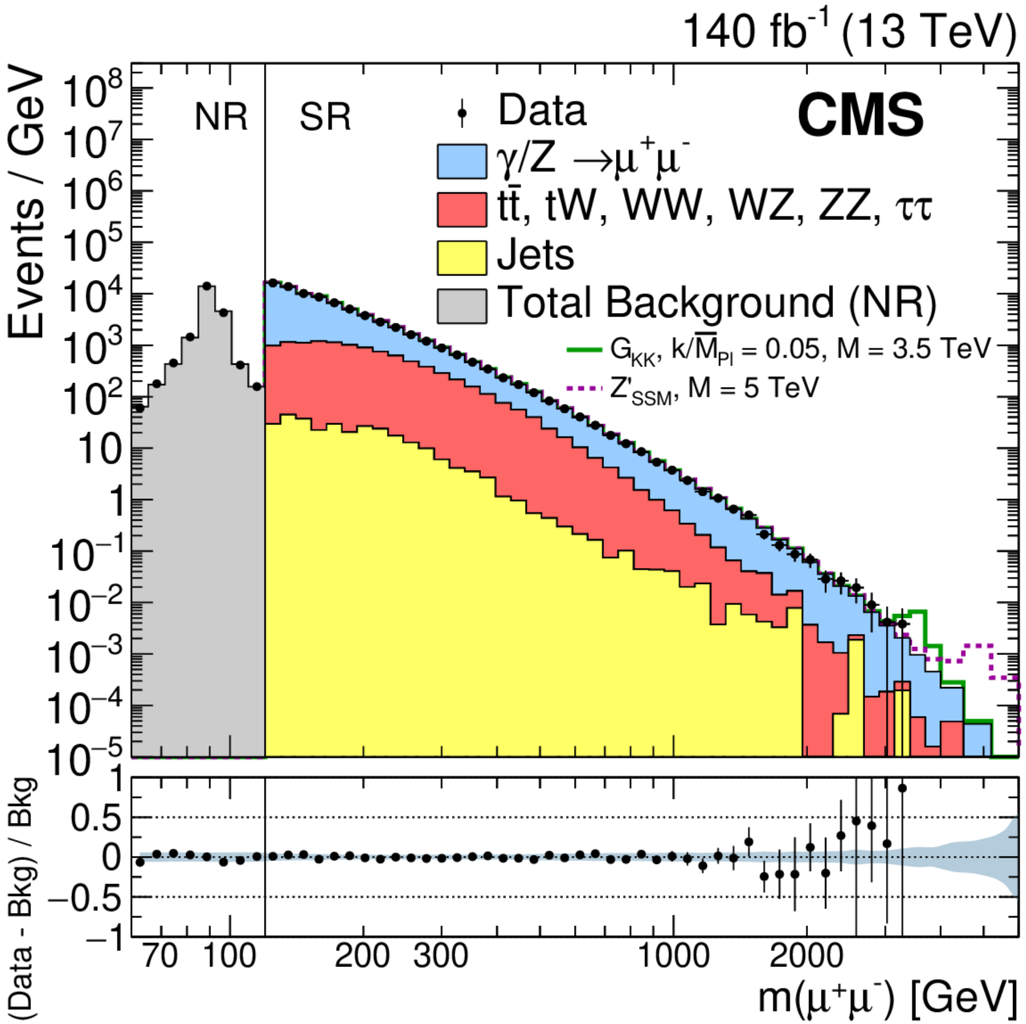
$$\sum_{i=1}^{n} -W_i * \log(P_i)$$

Binned Fits

- **n** -> the size of the array
- **W** -> Weight. Number of events of each sample i
- **P** -> Probability for each sample i

Negative Log Likelihood: Binned Fits





Optimizing time execution

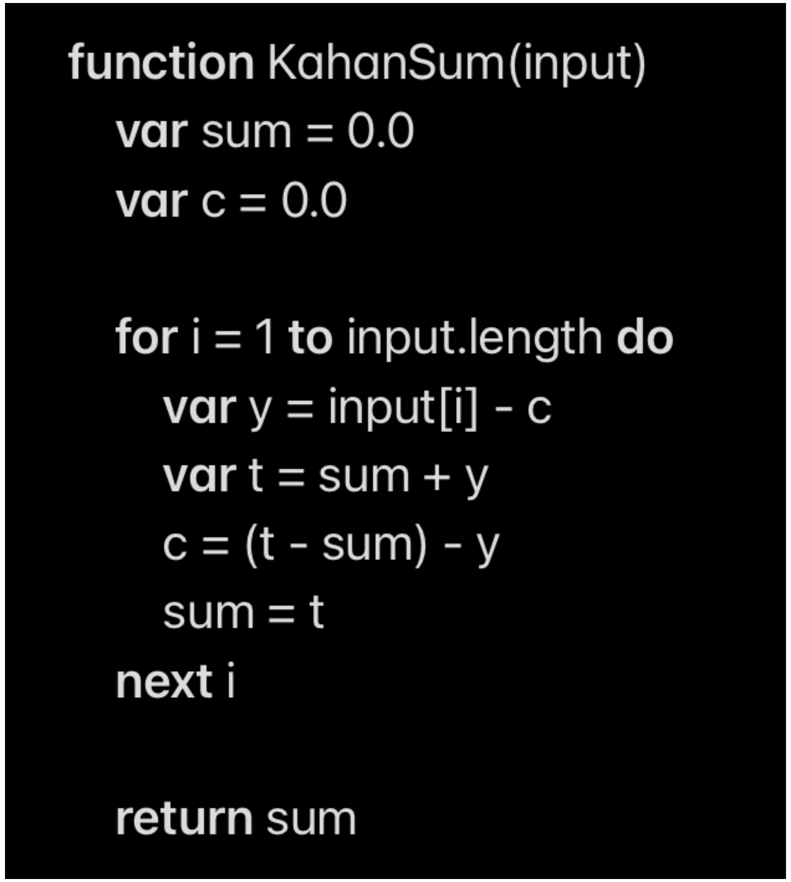
- **Initial situation**: The single and double precision variables can store either **very large** numbers or **very small** numbers.

- **Problem**: When you try to add (or subtract) them, the information of the decimal part will be lost:

```
PS> $a = 1; $b = 0.0000000000000000000000001
PS> $a + $b
1  -> RESULT!?
```

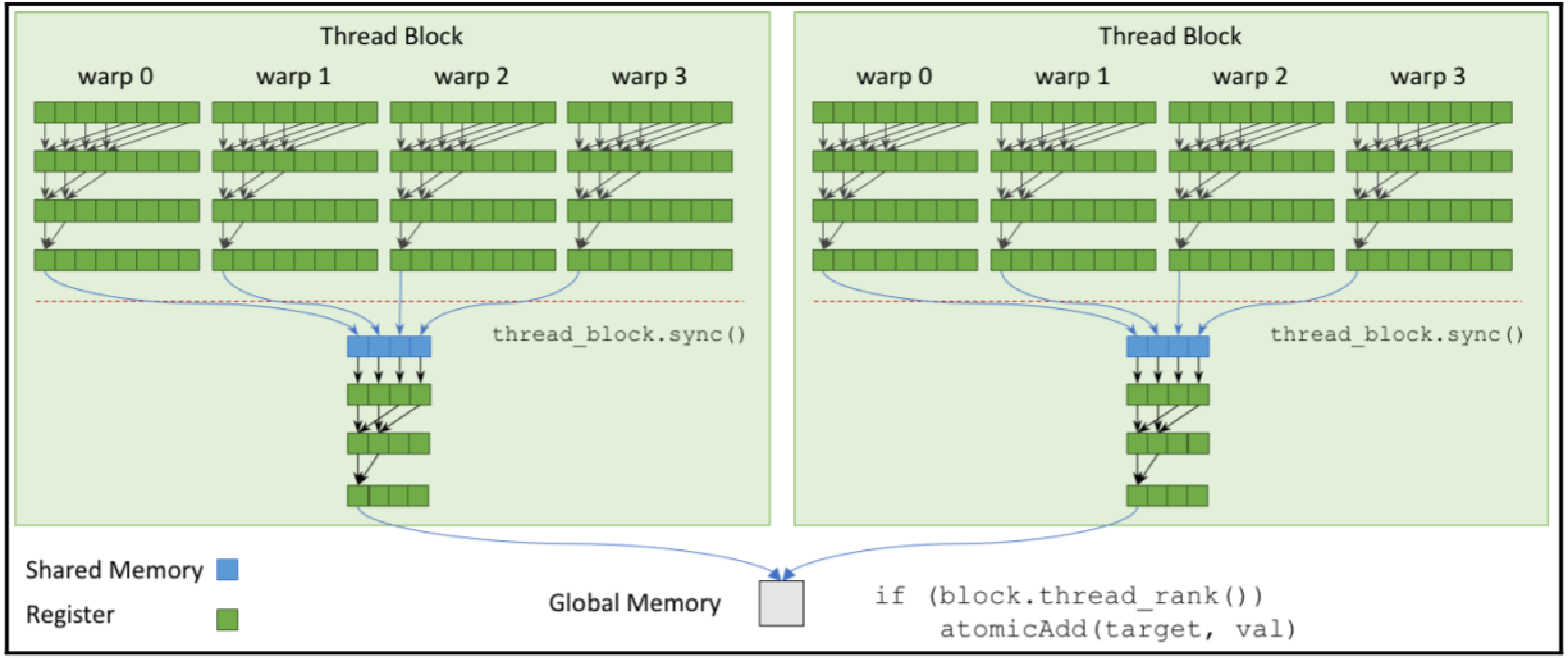
Kahan Summation

Helps to **minimise the error** when adding large data sets, by keeping track of the error made in each addition (variable c):



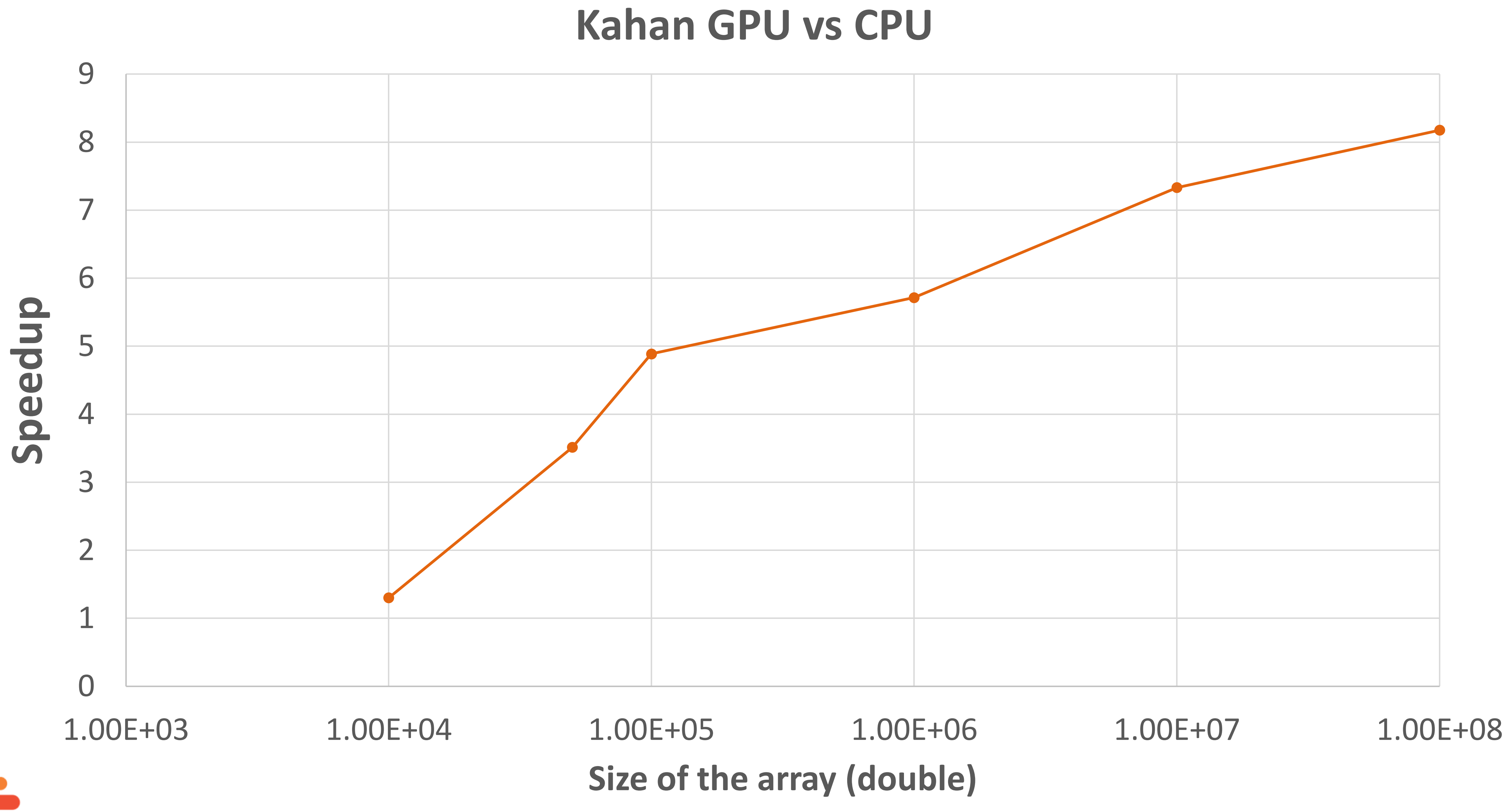
```
function KahanSum(input)
    var sum = 0.0
    var c = 0.0

    for i = 1 to input.length do
        var y = input[i] - c
        var t = sum + y
        c = (t - sum) - y
        sum = t
    next i

    return sum
```

Kahan Summation in the GPU



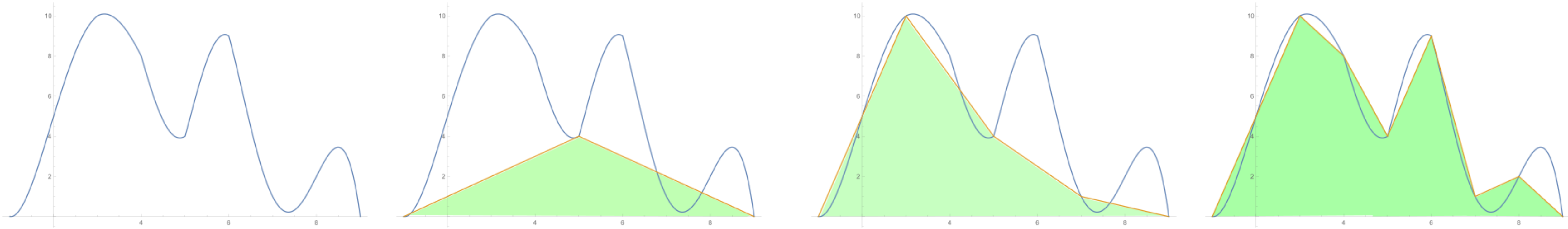
```
if (block.thread_rank())
    atomicAdd(target, val)
```

Ilustration: [PL00, Parallel computing (userdyk-github.github.io)]


Kahan GPU vs CPU
Speedup
9
8
7
6
5
4
3
2
1
0
1.00E+03
1.00E+04
1.00E+05
1.00E+06
1.00E+07
1.00E+08
Size of the array (double)

Compute numeric integrals in the GPU

It is what is limiting most costly realistic uses cases. **Romberg's method** makes a good approximation:



Conclusions

- Don’t use GPUs for **less than 10e7** data.
- Be very careful with the **synchronisation between** threads inside each block.
- Try to have all the cores at **100% work**.
- **Use memory:** 70% of the programs are “compute bound”, meanwhile 30% are “memory bound”.

Thank you for your attention

Questions?

ROOT

Data Analysis Framework

root-project/root

danielalvcon@gmail.com