

## 2. Evaluation of autocorrelation function with CUDA kernel

Write the CUDA kernel for calculation of autocorrelation function using Python language and Numba package.

Choose an arbitrary function  $f(x)$  where  $\{x_i\}_{1\dots N}$  are  $N$  function points and calculate autocorrelation function ACF as

$$\text{ACF}(\text{lag}) = \sum_{\text{lag}=0}^{\frac{N}{2}} \sum_{i=1}^{N-\text{lag}} f(x_i) \cdot f(x_{i+\text{lag}}) / \sum_{i=1}^N f(x_i) \cdot f(x_{i+0})$$

Compare the calculation with parallel implementation in Python and with serial version (can be that one implemented in Numpy package).

Deliver these graphical outputs:

- Arbitrary function and its ACF.
- Comparison of ACFs from serial, parallel and CUDA. Must be the same!
- Comparison of calculation times for CUDA, parallel and serial version as a function of  $N$  variable.

Discuss the effectivity of implementation using CUDA kernel and determine the borderline where the CUDA approach is beneficial.