Accelerated Python Programming

Yazeed Balasmeh
Arpan Ghosal

Universität Siegen

CPPS Center for Particle Physics Siegen

Accelerated Python Programming

Beyond Python…. C++ In Python !

How do we do that?

Accelerated Python Programming

➔ Ctypes

A **Foreign Function Interface (FFI)** library for Python, enabling:

- **C-compatible data types** in Python
- Direct calls to functions from **dynamic link libraries (DLLs)** or **shared libraries**
- Simple integration of **C libraries** into Python for enhanced functionality and performance

Boost Python's Power by leveraging C for high-performance operations in your applications.

Accelerated Python Programming

➔ Ctypes

Check the Google collab notebook.

Accelerated Python Programming

➔ Cython

A Superset of the Python Programming Language

- Features Python-like syntax combined with C/C++-style static typing.
  a. How much of your code is C-like and what parts have pythonic interactions
  b. Utilises static typing
  c. Supports structs, unions, enums
  d. C-style pointers
- Includes an optimizing static compiler for improved performance.

Accelerated Python Programming

➔ Cython

Check the Google collab notebook.

Accelerated Python Programming

➔ Numba

- Numba enables pure Python functions to be **JIT-compiled** into native machine instructions.
- It delivers performance comparable to C, C++, and Fortran.
- The `@jit` decorator incurs a one-time compilation overhead during the function's first execution.
- Once JIT compilation is complete and the function is cached, subsequent calls are executed rapidly.

Accelerated Python Programming

- ➔ Numba

Check the Google collab notebook.