

Mathlib: Do It Yourself

toward tool for scientists to generate optimal implementations of mathematical functions adapted to the task at hand

Vincenzo Innocente & Danilo Piparo

CERN, EP/SFT

Motivation

- Scientists and engineers have the need to implement mathematical functions that best fit their needs
 - Precision, validity range
 - Target architecture (different vectorization, etc)
 - Environment and language
- Our goal is to provide a tool for scientists integrated into the analysis eco-system deployed in HEP in particular as provided by CERN

Strategy

- Provide a layered tool that allows the development and implementation of mathematical functions tailored to a specified need
- Integrate it in the CERN standard analysis ecosystem
- Make it available into the “cloud”

Implementation

- With the collaboration of the CERN/SFT “Packages & Releases project” we have integrated Sollya, its dependencies and its python interface into the standard CERN software distribution
 - This makes Sollya and eventually MetaLibm available ubiquitously on the Cloud
- We have developed a set of “jupyter” notebook that provide examples of use including generation of python and C code

Addressing Heterogeneity

- One of the main challenge of Cloud computing is the ability to adapt to any computation environment notably exploiting a large variety of computing architecture
- Our strategy is to use standard compiler technology to build multi-target libraries and to rely on compiler optimization (in particular auto-vectorization) to produce the best code for each target
- We prefer pre-compiled libraries to Jit-technologies as the former can be fully tested and validated before deployment and will not depend on details of the running environment.
- Porting to GPU (cuda) proven to be trivial at least as *device* functions (easy to achieve bitwise compatibility as well)
 - Still not obvious how to build reusable efficient kernels as data copy may easily dominates

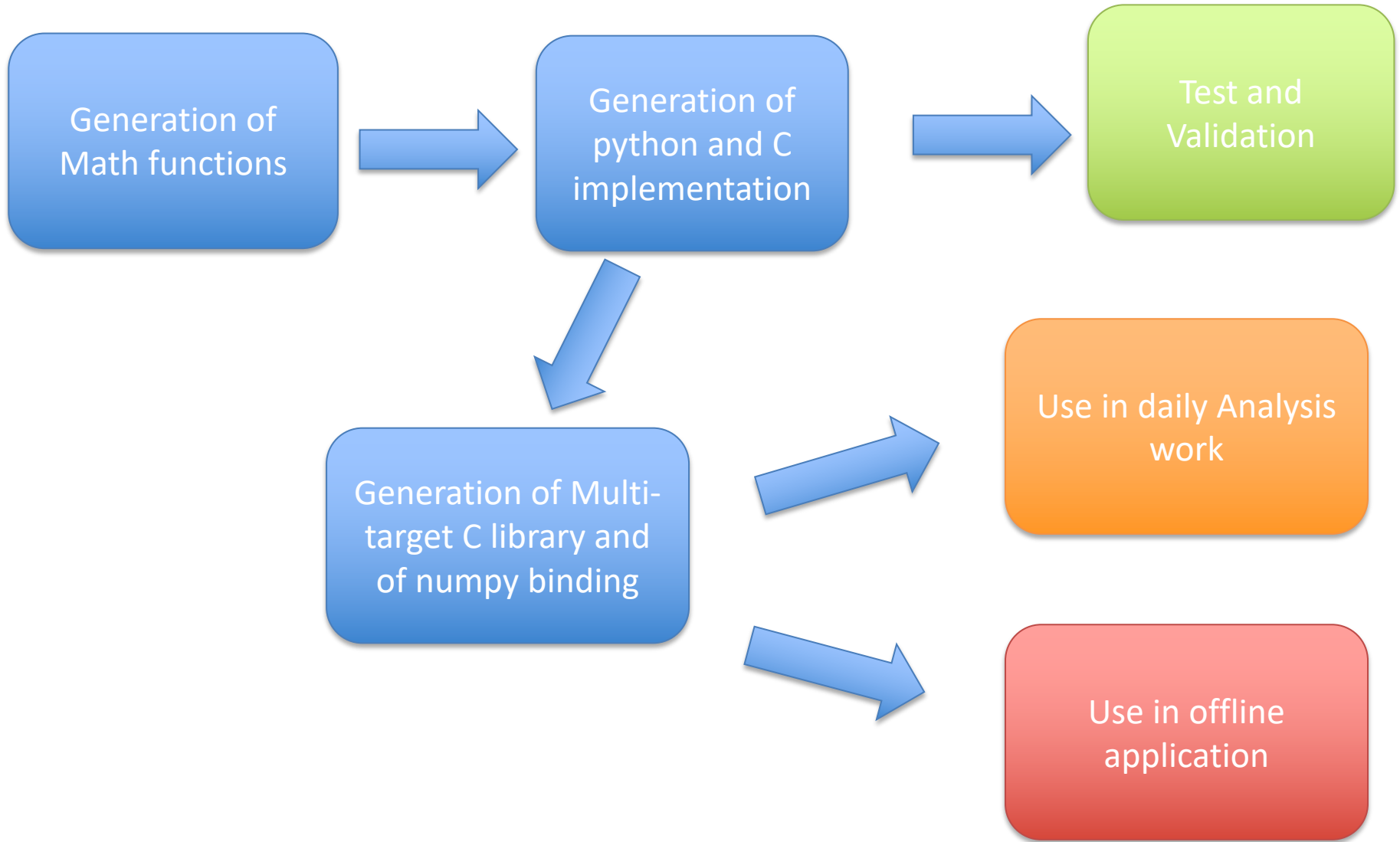
Reproducibility

- No FMA, no problem
 - C/C++ up to O3 guarantee full bitwise reproducibility across IEEE754 compliant architectures (including x86_64, arm, PC, GPUs)
- FMA: no standard re-association rules
 - Only solution to date is to force no re-association and use explicitly **std::fma**
- Need to teach code generators to properly use **std::fma** (or equivalent macro)

Deployment

- CERN is working on next generation analysis eco-system
- Its portal will most probably be “SWAN”: Service for Web-based data Analysis
 - <https://swan.web.cern.ch>
- SWAN is therefore the natural environment where our tool will be run
- It will provide a twofold use
 - Generation of optimized mathematical function using Sollya and MetaLibm
 - Their integration in the analysis eco-system through a direct, optimized numpy binding

Workflow



Status

- Sollya fully integrated in CERN environment
 - Tested under SWAN
- Examples as jupyter notebook produced
 - <http://test-innocent.web.cern.ch/test-innocent/MetaLibm/>
 - Some components may be packaged as standalone python modules
- Multi-target library and full numpy binding released for VDT
 - <https://github.com/dpiparo/vdt/blob/master/progs/numpyBootstrap.csh>