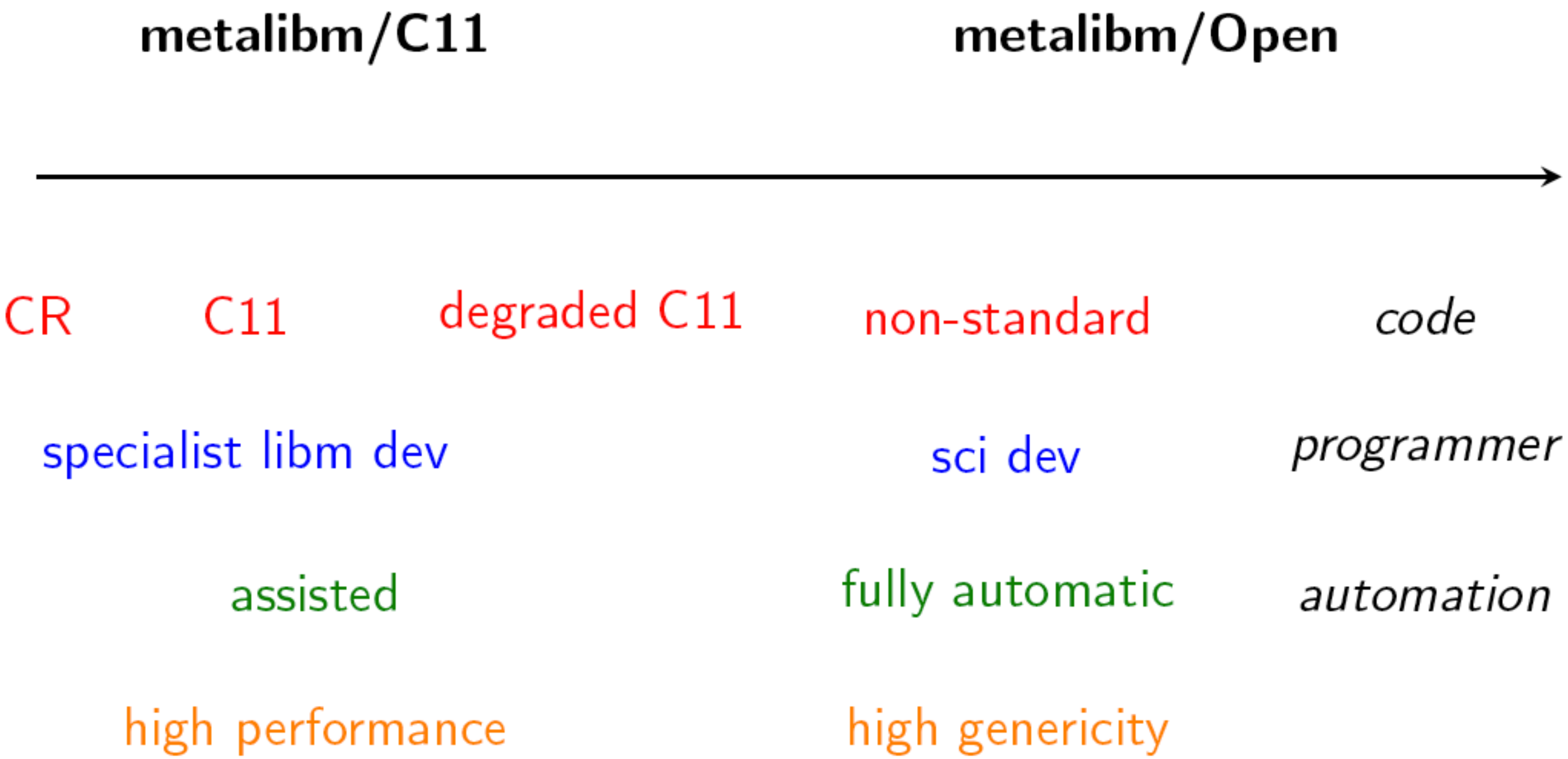
# CERN for MetaLibm

Vincenzo Innocente, Danilo Piparo
CERN/PH-SFT

# Project MetaLibm

- Four year project founded by ANR (France)
  - Involves major French Laboratories (and individuals) active in the field
- Main deliverables (all open source)
  - A  tool to generate mathematical functions
  - A library of specific generators
  - A modern implementation of “C11 libm”
  - An extension of “libm”
  - A unified tool covering also “Filters”
  - A high level interface for scientists

# The big Picture



**metalibm/C11**

**metalibm/Open**

CR C11 degraded C11 non-standard *code*

specialist libm dev sci dev *programmer*

assisted fully automatic *automation*

high performance high genericity

# Many Flavours (total: 12K functions)

| Precision | Ulp Error | Decimal Digits | Needed Accuracy | Verdict |
|---|---|---|---|---|
| Single | 1 ulp | 7 | $2^{-24}$ | fast |
| Single (CR) | 0.5 ulps | 7 | $2^{-50}$ | slow |
| Double | 1 ulp | 16 | $2^{-53.5}$ | fast |
| Double (CR) | 0.5 ulps | 16 | $2^{-150}$ | slow |

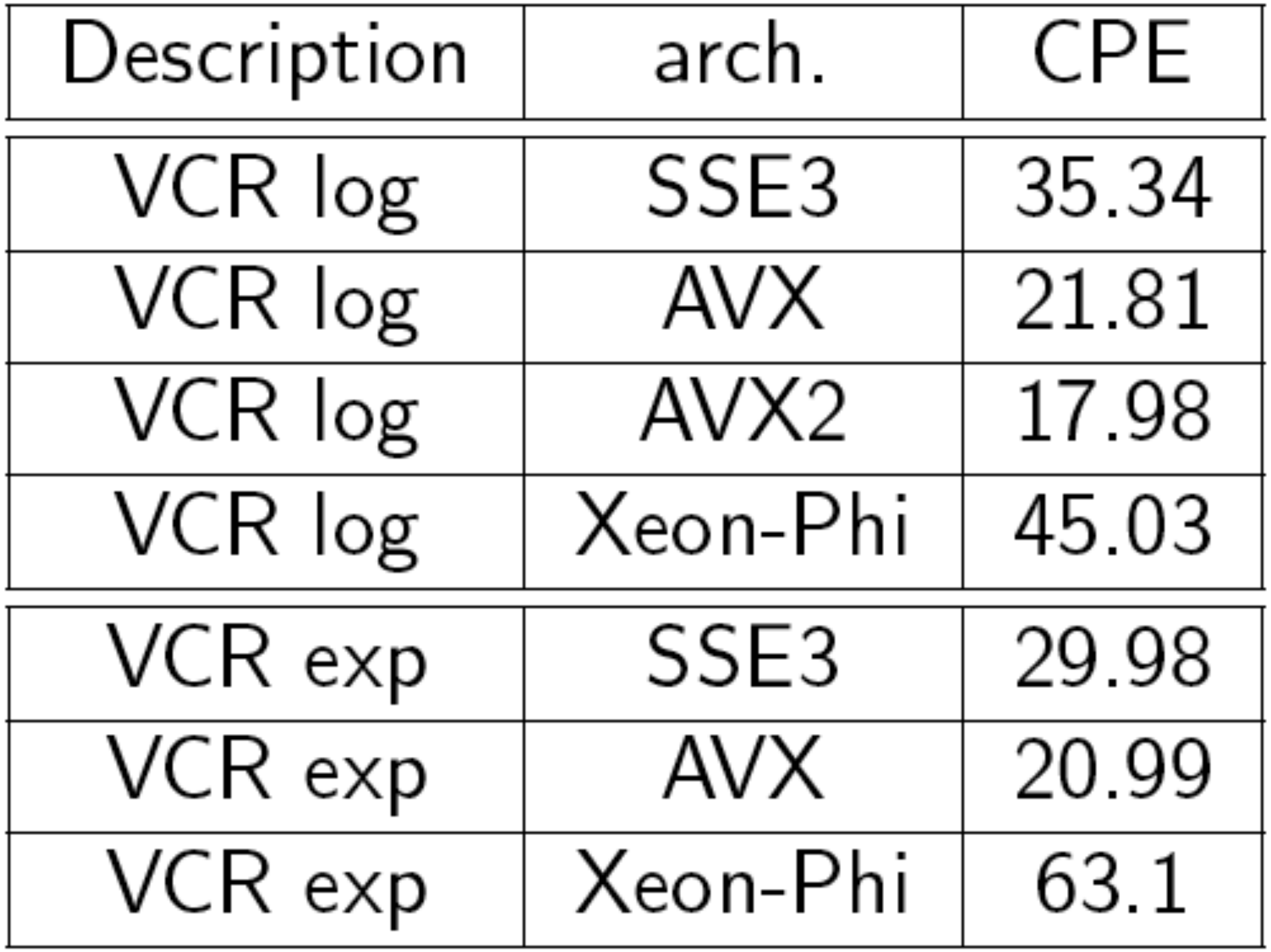
Not fast enough? Still more flavors

| | | | | |
|---|---|---|---|---|
| Single | $2^5$ ulps | 5 | $2^{-17}$ | faster |
| Single | $2^{11}$ ulps | 3 | $2^{-11}$ | the fastest |
| Double | $2^{11}$ ulps | 12 | $2^{-40}$ | faster |
| Double | $2^{42}$ ulps | 3 | $2^{-10}$ | the fastest |

- set floating-point flags or not 2 possibilities
- support all rounding modes or not: 4 possibilities

# Many platforms

# Prototype results for vector CR



| Description | arch. | CPE |
|---|---|---|
| VCR log | SSE3 | 35.34 |
| VCR log | AVX | 21.81 |
| VCR log | AVX2 | 17.98 |
| VCR log | Xeon-Phi | 45.03 |
| VCR exp | SSE3 | 29.98 |
| VCR exp | AVX | 20.99 |
| VCR exp | Xeon-Phi | 63.1 |

# Programme for the first year

| Title | Type | Date |
|---|---|---|
| Kick-off meeting | M | M1 |
| Website + open-source repositories | W | M3 |
| Consortium agreement | M | M6 |
| Survey on existing bibliography | R | M6 |
| Specification meeting at CERN | M | M6 |
| Experiments around exp and log in fixed point | S | M6 |
| Meeting to define range reduction scheme interface | M | M9 |
| Meeting for filters and functions specification | M | M9 |
| Prototype tool-chain for filter approximation and evaluation | S | M10 |
| Yearly workshop | M | M12 |
| Prelim. results on the unified formalism of arithmetic resources | R | M12 |
| Generator for exponential, logarithm, and trigs | S | M12 |
| Yearly progress report | R | M13 |

Type: M=Meeting, R=Report, S=Software, W=Website

# Goals for CERN

- Main Goal for CERN in MetaLibm is a “Toolkit for physicists”
- Tool available to scientists for generating an optimized version of a specified function
  - In a given range
  - For a given target accuracy
  - For a given set of hardware architectures
  - Implemented in a specified language (compiler)

# Examples: ranges

```
G4double
G4HadronCrossSections::GetCaptureCrossSection(c
onst G4DynamicParticle* aParticle, G4int ZZ)
 { […]
    G4double ek = aParticle->GetKineticEnergy()/GeV;
  […]
   G4double ekx = std::max(ek, 1.e-9);
   if (ekx != lastEkx) {
     lastEkx = ekx;
     lastEkxPower = std::pow(ekx*1.e6, 0.577); }
[…]
```

Argument of pow
is at most 1e-3.

```
G4NucleiModel::generateNucleonMomentum(G4int type, G4int zone) const {
   G4double pmod = getFermiMomentum(type, zone) * G4cbrt(inuclRndm());
   […]
```

Flat rndm
between 0 and 1.

# Examples: accuracy

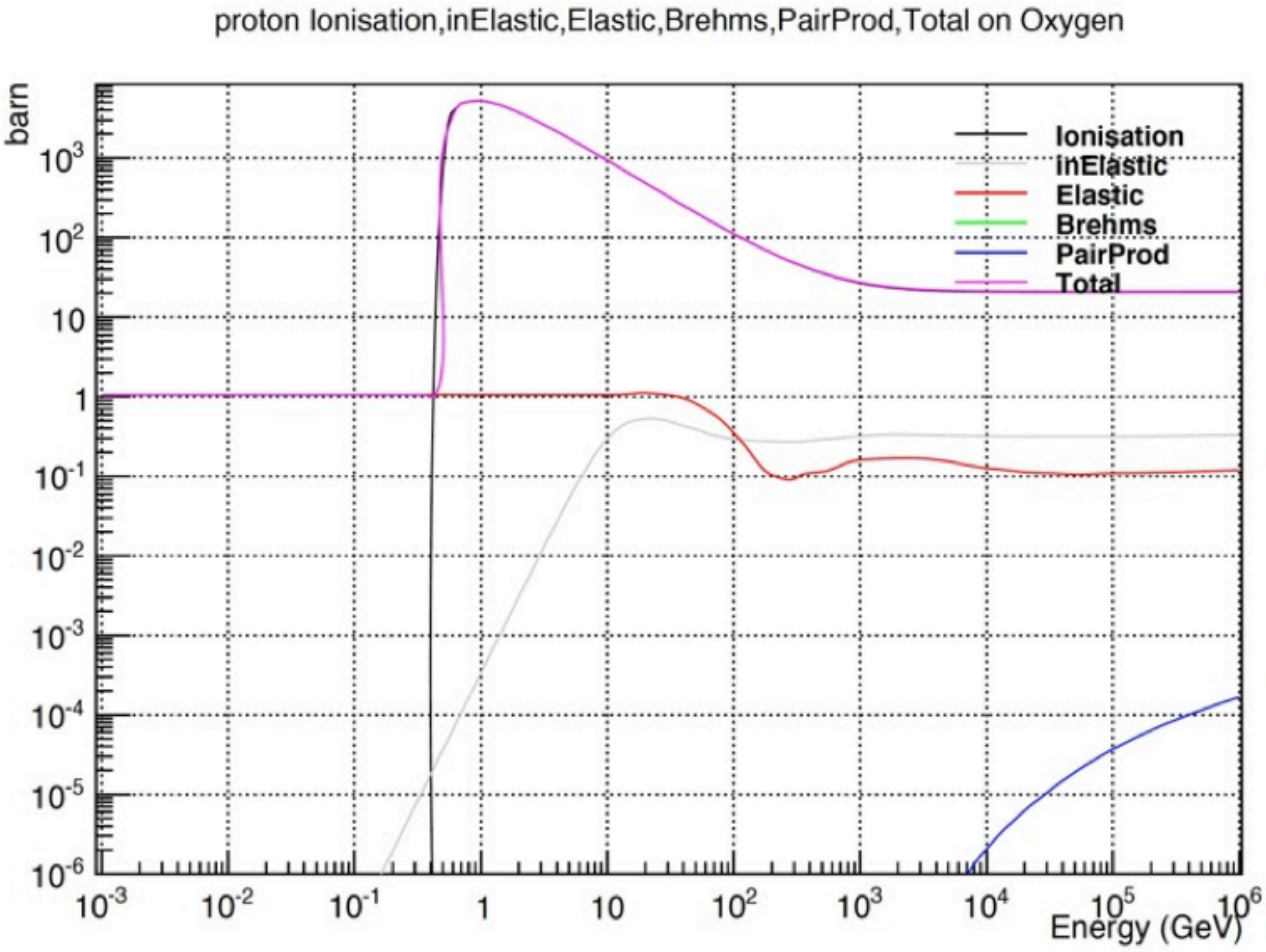
```
G4double
G4HadronCrossSections::GetCaptureCrossSection(c
onst G4DynamicParticle* aParticle, G4int ZZ)
 { […]
   G4double ekx = std::max(ek, 1.e-9);
   if (ekx != lastEkx) {
     lastEkx = ekx;
     lastEkxPower = std::pow(ekx*1.e6, 0.577); }

   G4int izno = ZZ;
   if (izno > 100) izno = 100;      // Not in GHESIG
  izno = izno - 1;      // For array indexing
  G4double sigcap = 11.12*cscap[izno]/lastEkxPower;

sigcap = sigcap*millibarn;
   return sigcap; }
```

Probably double precision is not needed.

# Examples: “Black box”



Geant4 simulation toolkit: cross sections of proton in Oxygen

F. Carminati

# Past Activities at CERN/SFT

- Collection/development of math functions
  - Last CERN mathematician retired in the `90
  - Ported to C++ in early 2000. mostly now in Root
    - Origin: CERNLIB, Cephes, GSL, Ab-St….
- VDT project
  - Recode Cephes routine to allow vectorization
- Produce "reduced precision" versions for specific "target accuracy" use cases
  - Use of Sollya, Maxima, copy/paste from Ab-St

# First activities

- Review current literature and implementations
- Produce an updated catalog of “functions of interest in HEP”
  - Emphasis on those not in “libm” (e.g. Gamma)
- Compile use-cases for “limited range” and specific target accuracy requirements
- Brainstorm on a possible “specification interface”

# Near term Activities

- Early test of new (cr)libm in HEP applications (all 12000 functions!)
- Collaborate to other deliverables
  - Integer algorithms
  - Vector APIs (incl. C++ standard/extensions)
  - Auto-vectorization on various hardware architectures
  - Implementations other than polynomials
- Host students to work on the above