Update of PhysicsVectors and global functions

V.Ivanchenko, CERN & G4AI

18th Geant4 Collaboration Workshop
23-27 September 2013
Sevilla, Spain

Introduction

- Outline
  - Fast mathematical functions
  - Data vectors
- Motivation from CPU profiling:
  - significant contribution of standard library (log, exp, sincos .. – decenting order)
  - Significant contribution of data access methods for cross section computations
- MT migration required design change of data classes

G4Pow

- Initially this class was created in order to provide super fast values of frequently used functions of integer arguments Z and A
  - Z, A may vary from 1 to 512
- With time number of G4Pow methods increased
  - Additions were needed for PreCompound/DeExcitation models and cross scetions
  - Fast computations with double arguments were also added, in particular, for fast computing of cross sections

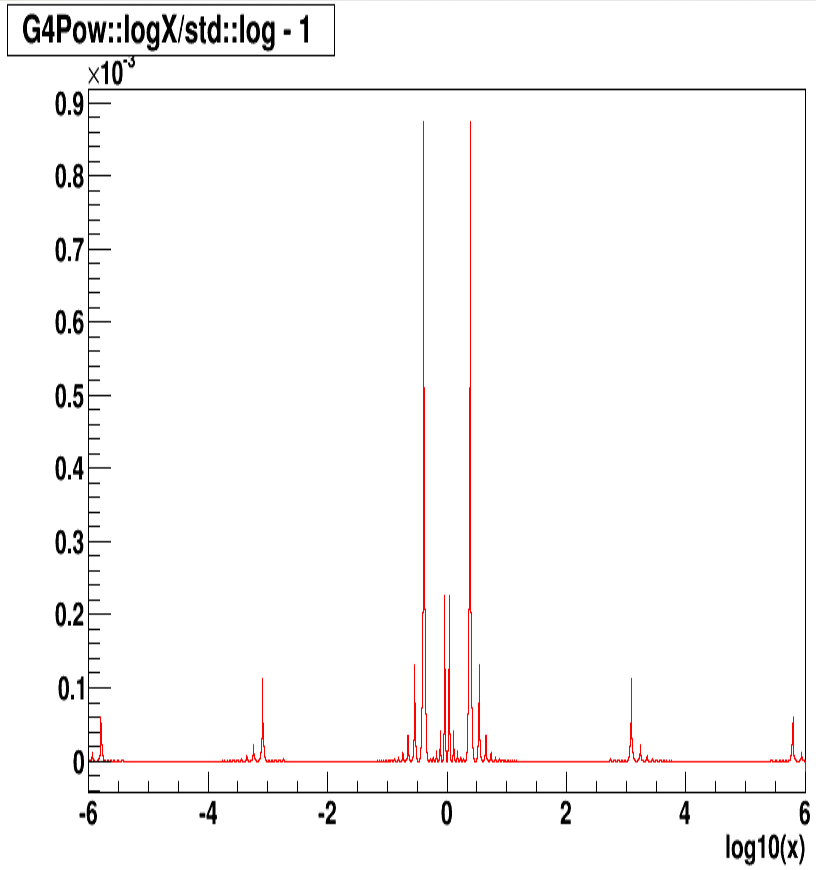
G4Pow

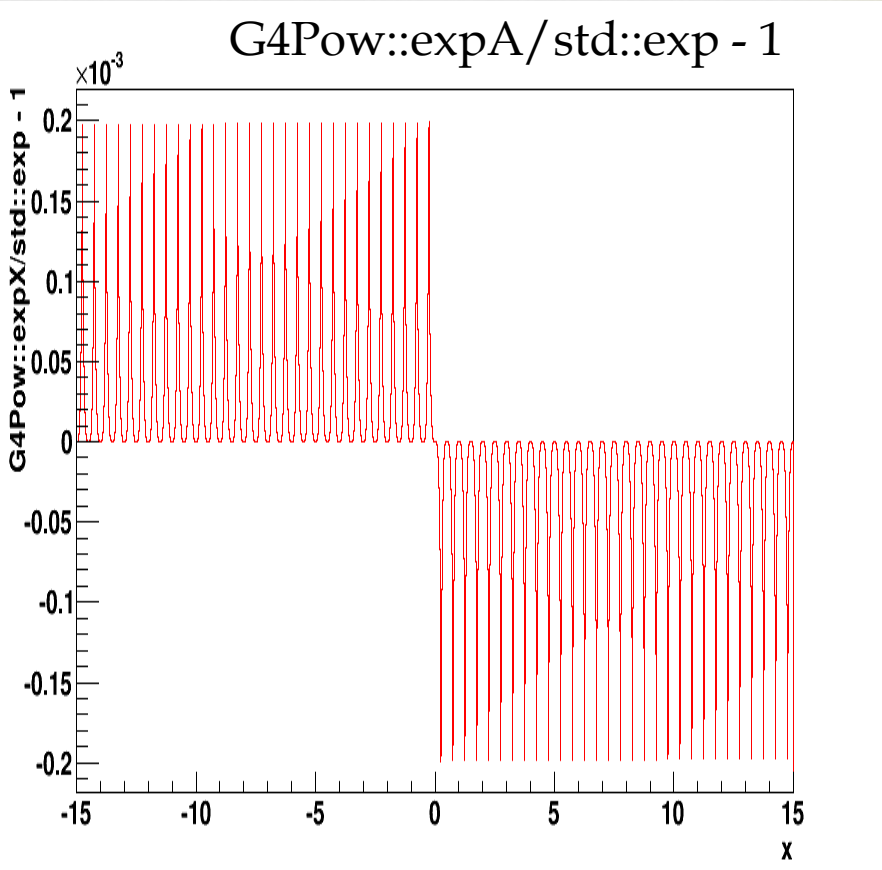
- Currently it is G4ThreadLOcal singleton
  - I would propose to make it global singleton
- Current methods with integer arguments:

  Z13(G4int)
  Z23(G4int)
  logZ(G4int)
  log10Z(G4int)
  powZ(G4int, G4double)
  factorial(G4int)
  Logfactorial(G4int)
- Current methods with double arguments:

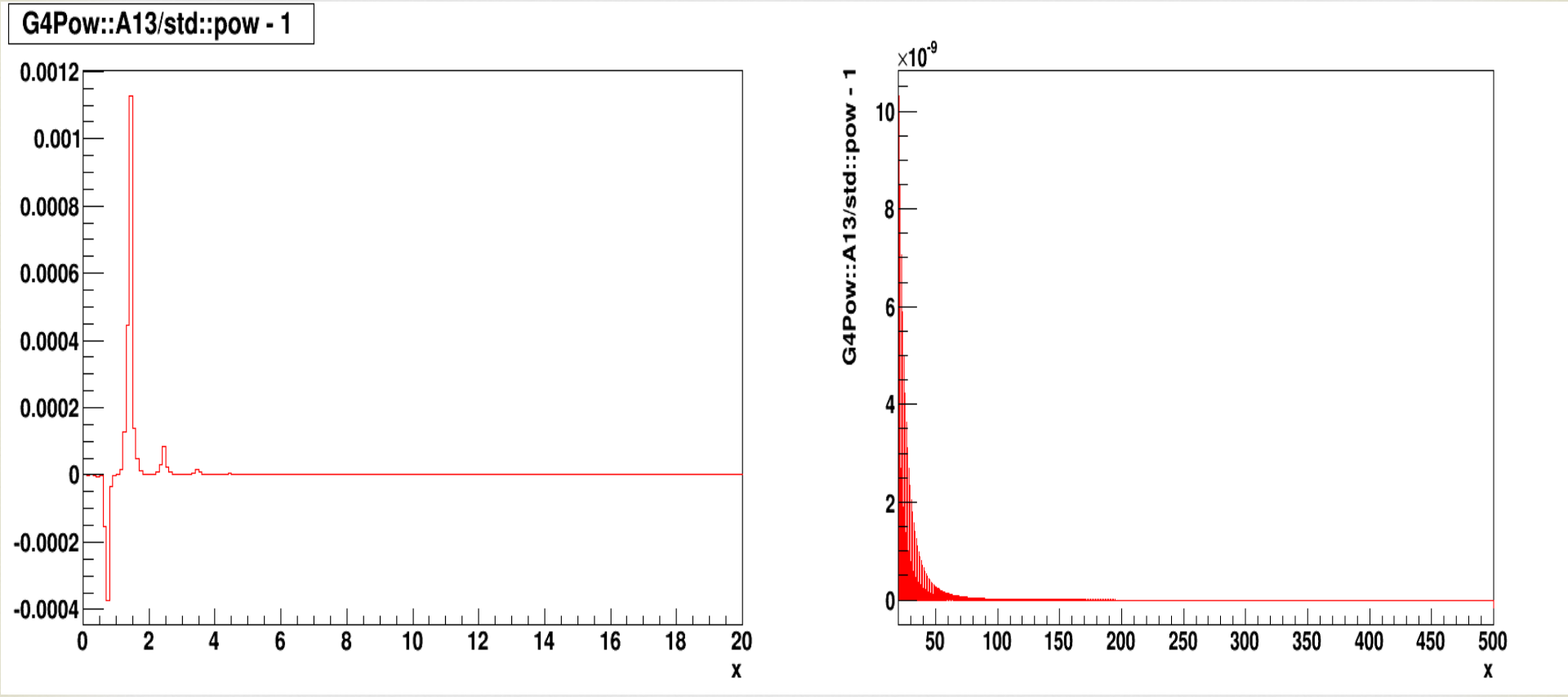
  A13(G4double)
  A23(G4double)
  logA(G4double)
  logX(G4double)
  log10A(G4double)
  expA(G4double)

Accuracy of G4Pow::logX and G4Pow::expA





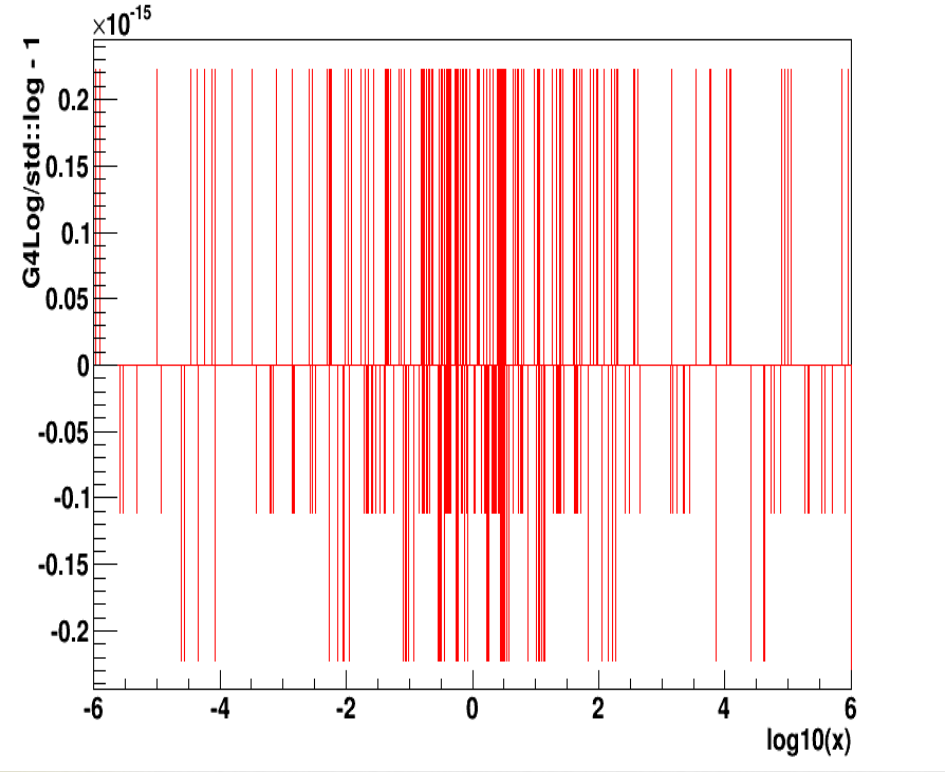
Accuracy of G4Pow::A13

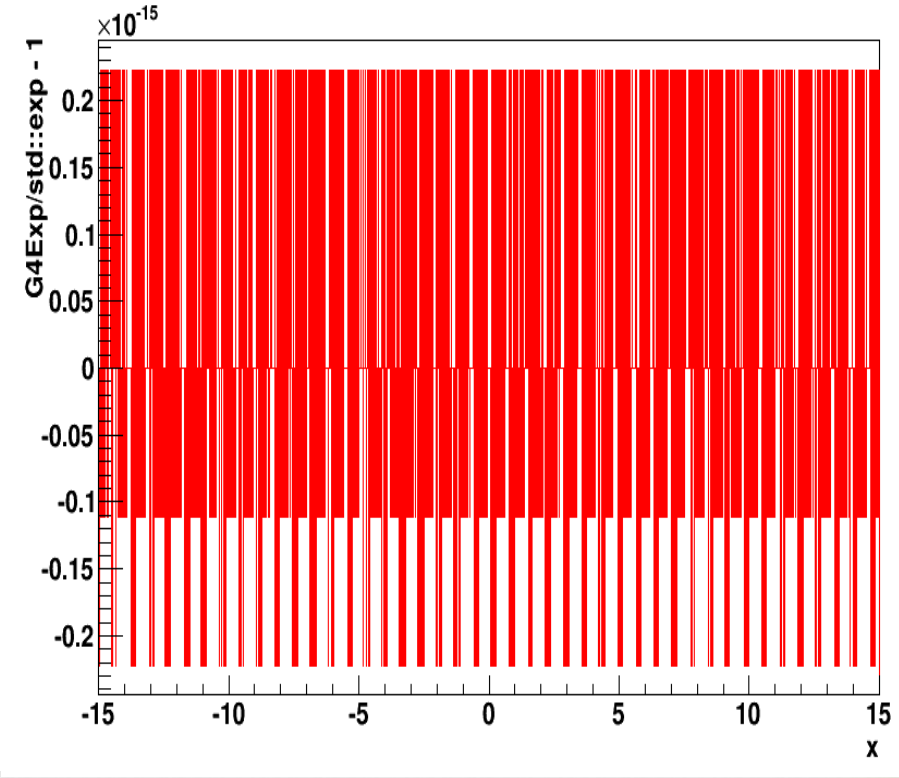


G4Log and G4Exp

- Danilo Piparo, Thomas Hauth, Vincenzo Innocente developed VDT mathematical library
  - Alternative to cmath
  - May be preloaded and substitute standad library
  - This is not very convinient from management point of view
- In Geant4 9.6ref09 two new classes are added G4Log and G4Exp
  - These classes are extracted by me from VDT
  - Were adapted to Geant4 platforms by Gabriele Cosmo

Accuracy of G4Log and G4Exp





CPU performance

	Std	G4 VDT	G4Pow
Log	8.97	4.91	5.19
Exp	13.93	1.95	1.34
$A^{1/3}$	20.46	7.03	0.77
$Z^{1/3}$	-	-	0.01

Random number generator was used to select arguments of math functions «background time» was subtracted

Data Classes Evolution

- Main data classes:
  - G4PhysicsVector – was there from the beginning
  - G4Physics2DVector – introduced for SeltzerBerger model for g4 9.5
    - Now also used by muon models
  - G4ElementData introduced for Geant4 9.6 to keep atomic data
    - Livermore photo-electric
    - Muon models
    - NeutronXS
- For MT prototype these classes were developed as «splitted» – cache was thread local, mail class shared
- For 10.0beta and just after a design iteration was done
  - Splitted classes provide too many problems for MT
  - Design was not elegant
  - Now these classes are read-only in run time, no cache
  - No CPU penalty when cache sub-classes were removed
    - User code if needed may have tread local cache implementation
      - This is not needed in many Geant4 cases

Recent Updates

- G4PhysicsVector::Value(e) is one of the most frequently called methods seen in the top of all profing results
  - It calls bin location method
    - In EM usually logarithmic vector is used
    - Std::log was substituted by G4Pow::logX()
  - A new method G4PhysicsVector::Value(e, idx) was introduced as an alternative
    - This method allows user code to have cache avoiding call to bin location
  - A new method G4PhysicsVector::SampleLinearX() was introduced for cumulative functions providing linear sampling of argument according to the distribution
    - This aloows to remove doulicated code in different Geant4 classes
- The same updates were introduced in G4Physics2DVector

Migration to MT

- Having read-only physics vectors is very convinient for MT mode
  - Before this migration there were difficulties in debugging of Geant4 in MT mode
- All classes based its internal data on G4ElementData structures were easely migrated to MT providing sharing these data between threads

Summary and «to do»

- CPU effect of VDT and G4Pow math functions is very significant,so for 10.0 we need:
  - migrate all sensitive code (cross sections first of all) to
    - G4Log, G4Exp, G4Pow::A13, G4Pow::A23
  - If there are math functions with integer argumetrs then migrate to
    - G4Pow::Z13, G4Pow::Z23, G4Pow::logZ,G4Pow::factorial
- Data structure classes redesign was an essential part of MT migration
  - The next step is to make G4PhysicsVector be free of virtual methods