The Status of the Muon Propagation Code

D. Chirkin[1], W. Rhode[2]

chirkin@physics.berkeley.edu

rhode@wpos7.physik.uni-wuppertal.de

[1] *Physics Department, University of California at Berkeley, CA 94720, USA*

[2] *Physics Department, University of Wuppertal, D 42097, Germany*

Motivations

1. **Raising the accuracy of the calculation per interaction**
   (a) before: 1e-2 (insufficient for multiple scattering)
   (b) goal: 1e-4, getting rid of parameterization tables
2. **Overcoming computing limitations - using new technologies**
   (a) faster computers than were available before (obvious)
   (b) connecting more computers in parallel

Physics of the calculation

1. **Ionization**

   (a) Bethe-Bloch formula:

   $$-\frac{dE}{dx} = f(E)$$

   (b) Special role of ionization

$$(1 - dP(E(x_i))) \cdot ... \cdot (1 - dP(E(x_f)))$$

$$= \exp(-\sigma(E(x_i)) \cdot dx) \cdot ... \cdot \exp(-\sigma(E(x_f)) \cdot dx)$$

$$= \exp(-\int \sigma(E(x))) \cdot dx$$

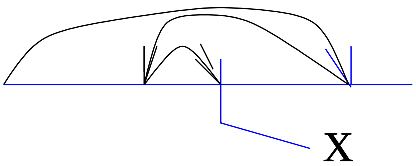
$$= \exp(-\int \frac{\sigma(E)}{-f(E)} \cdot dE) = \xi$$

Physics of the calculation

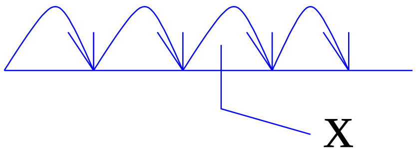
1. **Ionization**
   (a) knocked on electrons - higher energy ionization electrons
2. **Bremsstrahlung**
   (a) elastic (on a nucleus)
   (b) inelastic
      i. *on a nucleus with internal structure*
      ii. *on electrons (with recoil)*
3. **Pair production**
4. **Photo-nuclear interactions**

Math of the calculation

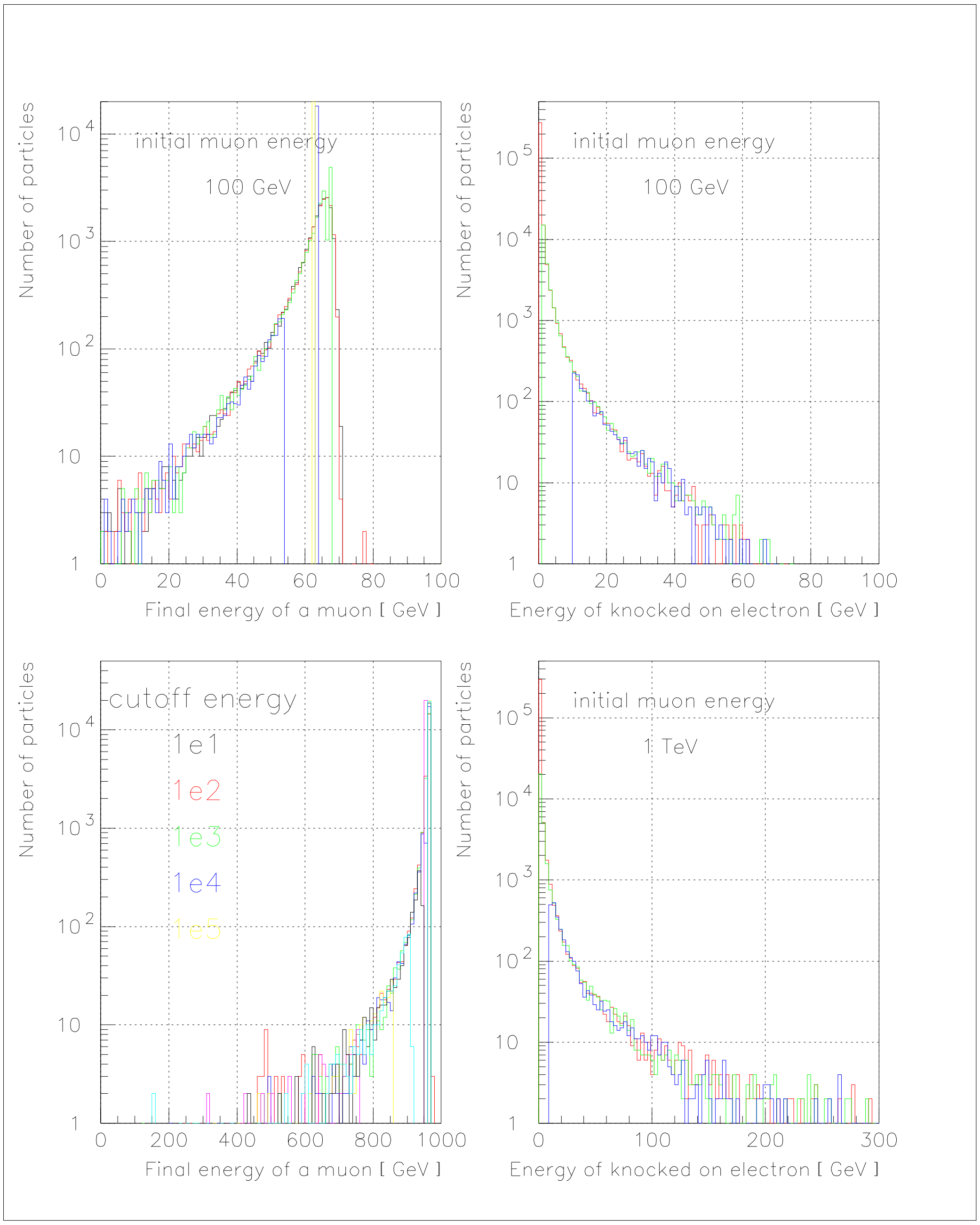
1. **Integral evaluation**
   (a) Romberg integration (trapezoid + interpolation)
2. **Root finding**
   (a) Newton-Raphson method + bisection bracketing
3. **Finding of an upper limit of an integral equal to a random number**
   (a) in the particle tracking - locating the interaction point - smooth more or less monotomical function

   (b) in the cross section evaluations - unpredictable function

Preliminary results



Work in progress

1. Not all cross sections are in the code yet
2. Interpolation formulas for some cross sections (photo-nuclear) will have to be improved to fit the best measurements currently available
3. Optimization, optimization and once more - optimization (never ending work in progress)
4. integration in the AMANDA mass production monte carlo

SYMPHONY Java vs. other approaches

1. **fast and easy connectivity of clients and job submission**
   (a) virtually unlimited number of clients not willing to install any special software
2. **the Java language is well suited for scientific calculations**
   (a) programming is more straightforward and less cumbersome than in C++
   (b) compiled program is faster than C++ or Fortran

Class Hierarchy

- class java.lang.Object
  - class MainTest (implements DFunctionOfx)
  - class MathModel
    - class FindRoot
    - class Integral
    - class TrackIntegral (implements DFunctionOfx)
    - class TrackIntegral2 (implements DFunctionOfx)
  - class New
  - class PhysicsModel
    - class CrossSections
      - class IonizationLoss (implements FunctionOfx)
      - class KnockOn (implements DFunctionOfx)
    - class Medium
    - class Particle
    - class Propagate (implements FunctionOfx)

Class Integral

```
java.lang.Object
   |
   +----MathModel
           |
           +----Integral
```

---

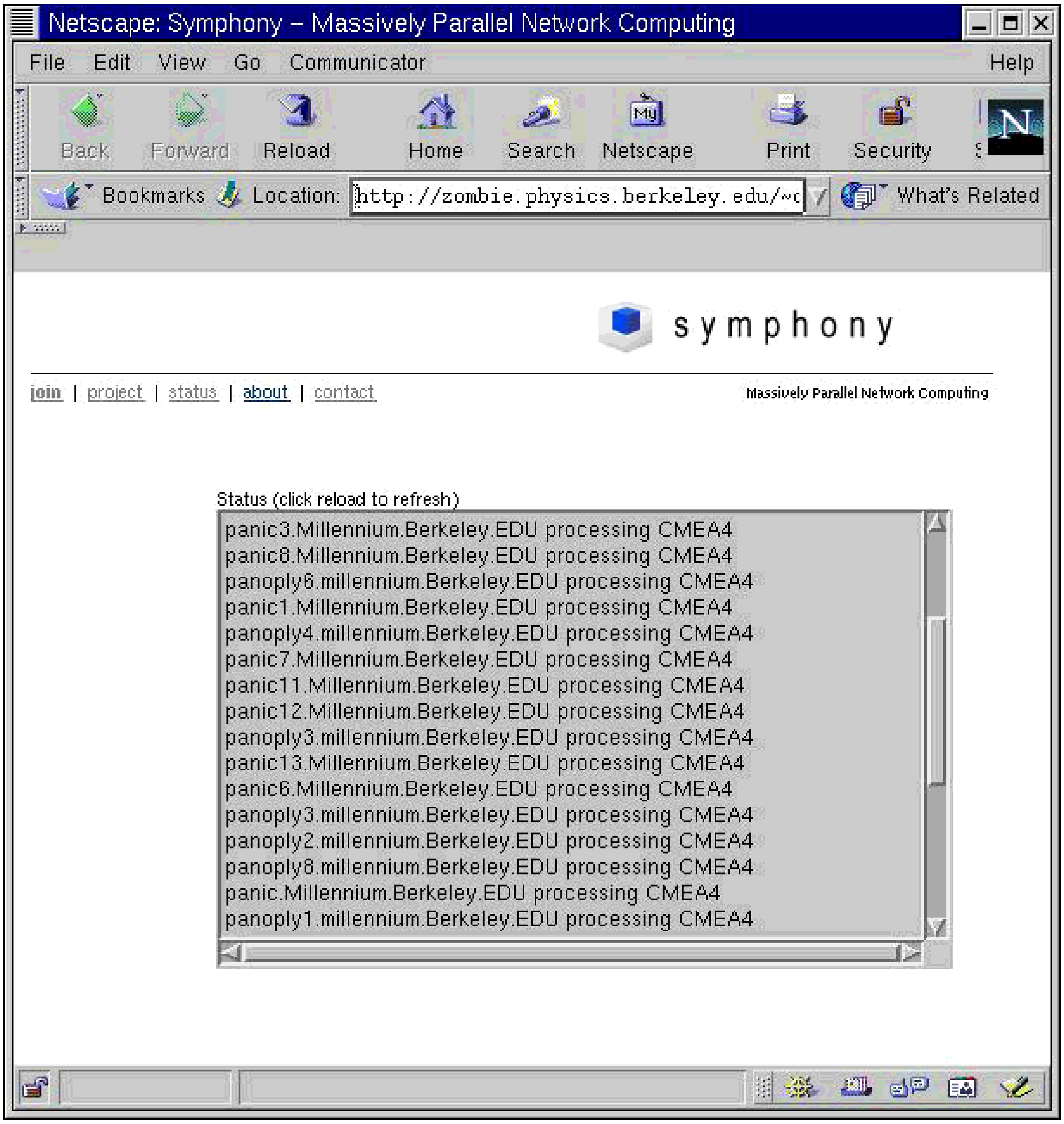
public class **Integral**
extends MathModel

This class provides routines for function integration using Romberg method. Include the function to be integrated in a class that implements interface FunctionOfx. Methods contained here are based on the Numerical Recepies (W. H. Press et al.) with the following modifications:

- Algorithm of path choice in the interpolate method is different: first the upper boundary of the triangle is reached in the least possible number of steps, and then it is followed.
- Power of Substitution (POS) is now any real number. If it is zero, no substitution is made. Otherwise the substitution is 1/x^POS if POS>0 or 1/(-x)^(-POS), if POS<0. If one of the limits of integration is zero or has opposite sign than that of POS, it is replaced with infinity. If both limits of integration are zero or have opposite sign than that of POS, integral is considered to have equal to each other (and to infinity) limits, and the returned value is zero.
- It is now possible to evaluate the upper limit of the integral x(rand) (keeping the lower limit at its original value xmin) such that the resulting integral from xmin to x(rand) is a fraction rand of the original full integral from xmin to xmax. Set the ratio rand as the last argument of one of the open integration methods (integrateOpened or integrateWithSubstitution), and get x(rand) by the subsequent call to getUpperLimit.

Method Index

- **getUpperLimit**()
  returns the value of the upper limit x(rand) calculated by the previous call to one of the integrate methods
- **integrateClosed**(double, double, FunctionOfx)
  finds integral for closed intervals
- **integrateOpened**(double, double, FunctionOfx)
  finds integral for opened intervals
- **integrateOpened**(double, double, FunctionOfx, double)
  finds integral for opened intervals and computes the exact value for the x(rand)
- **integrateWithSubstitution**(double, double, FunctionOfx, double)
  finds integral for opened intervals using substitution x -> 1/x^(powerOfSubstitution)
- **integrateWithSubstitution**(double, double, FunctionOfx, double, double)
  finds integral for opened intervals using substitution x -> 1/x^(powerOfSubstitution) and computes the exact value for the x(rand)


Netscape: Symphony – Massively Parallel Network Computing
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
Bookmarks Location: http://zombie.physics.berkeley.edu/~c What's Related
symphony
join | project | status | about | contact
Massively Parallel Network Computing
Status (click reload to refresh)
panic3.Millennium.Berkeley.EDU processing CMEA4
panic8.Millennium.Berkeley.EDU processing CMEA4
panoply6.millennium.Berkeley.EDU processing CMEA4
panic1.Millennium.Berkeley.EDU processing CMEA4
panoply4.millennium.Berkeley.EDU processing CMEA4
panic7.Millennium.Berkeley.EDU processing CMEA4
panic11.Millennium.Berkeley.EDU processing CMEA4
panic12.Millennium.Berkeley.EDU processing CMEA4
panoply3.millennium.Berkeley.EDU processing CMEA4
panic13.Millennium.Berkeley.EDU processing CMEA4
panic6.Millennium.Berkeley.EDU processing CMEA4
panoply3.millennium.Berkeley.EDU processing CMEA4
panoply2.millennium.Berkeley.EDU processing CMEA4
panoply8.millennium.Berkeley.EDU processing CMEA4
panic.Millennium.Berkeley.EDU processing CMEA4
panoply1.millennium.Berkeley.EDU processing CMEA4