Generation of Cosmic Rays in Geant4 with MUSUN

A.N. Villano

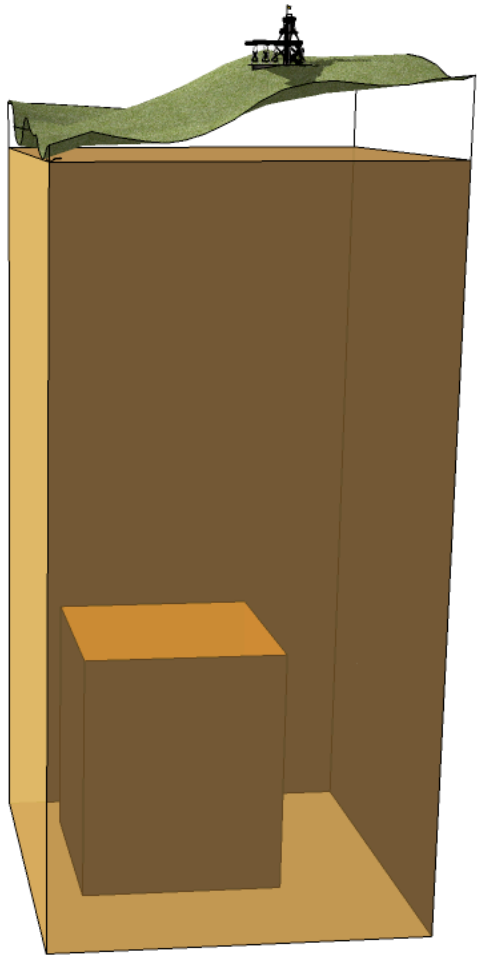
University of Minnesota

Muons Deep Underground

- MUSIC and MUSUN developed in Fortran77

- Create slant paths then quickly generate muons at depth

- Many billions of muons can be generated on-the-fly in a pseudorandom fashion very quickly



Stand-Alone Fortran77 Code

- Text file output
- Gives data as event number, PID, energy (GeV), x, y, z, p_x, p_y, p_z
- PID is -13 for muons and 13 for anti-muons
- First line is random seed(s) for generator

```
565701196          0          0
     1  13     388.5   -97.2   -24.4   200.0  0.3661  0.5604 -0.7429
     2 -13     890.3   163.0   329.1   200.0  0.3350 -0.4195 -0.8437
     3  13     107.5   200.0  -363.4   -89.1 -0.4123 -0.3565 -0.8384
     4 -13      37.3  -200.0   -42.0    36.1  0.2941 -0.3184 -0.9012
     5 -13     118.2   162.0   -10.9   200.0 -0.0925  0.4821 -0.8712
     6 -13      16.1   156.0  -400.0   -36.5  0.4199  0.7616 -0.4936
     7 -13      41.9   200.0  -283.7   174.1 -0.6560 -0.2071 -0.7258
     8 -13      43.5   124.4  -399.9   200.0 -0.3998 -0.2402 -0.8846
     9  13      18.2  -200.0   183.6  -169.8  0.4217  0.1939 -0.8858
    10 -13      19.7   -30.5  -141.1   200.0 -0.6465 -0.2356 -0.7257
    11 -13     684.2    49.0   256.0   200.0  0.3858  0.3083 -0.8695
    12  13      20.2   116.4   281.7   200.0 -0.4453  0.1051 -0.8892
    13 -13     210.4    91.8  -206.1   200.0 -0.0972  0.5541 -0.8268
    14  13     994.2   -81.4  -347.4   200.0  0.4947  0.4890 -0.7185
    15  13     109.2    -1.4   280.5   200.0  0.4105  0.4541 -0.7908
    16  13      63.6  -177.5   -75.1   200.0 -0.0728 -0.2181 -0.9732
    17 -13     422.1   154.5    83.8   200.0  0.4593  0.3304 -0.8246
    18  13      17.1   200.0   297.2   -90.2 -0.2054  0.3744 -0.9042
    19 -13      36.7  -200.0  -171.8   170.7  0.4515 -0.1646 -0.8770
    20 -13      28.5  -200.0   -94.4  -118.7  0.4358  0.0111 -0.9000
    21 -13     712.9   -87.8    33.0   200.0  0.2746  0.1261 -0.9533
    22 -13      19.4   167.2   -12.6   200.0 -0.3126  0.0805 -0.9465
    23 -13      54.8  -200.0  -104.3    -2.7  0.7692  0.0198 -0.6387
    24  13     112.7  -113.3   107.6   200.0  0.0903  0.1305 -0.9873
    25 -13     302.4  -104.7  -122.6   200.0 -0.2096 -0.3633 -0.9078
```

Using with Geant4: Porting to C++

- Current use with Geant4
  - make many such files
  - use a class to externally read them and throw them with standard Geant4 sources (G4ParticleGun)
- This is awkward
  - if you run out of files, your simulation is over until generate more files
  - a lot of disk I/O which can be costly in a grid environment
- If port to C++ can easily build into Geant4 particle source
  - no more disk I/O (can load and store slant-paths)
  - generation on-the-fly so that you can run as many muons as the pseudorandom generator allows without repeats (a lot – for Mersenne Twister 2^{19937})
  - ref: http://en.wikipedia.org/wiki/Mersenne_twister

Alternatively…

- Keep development in Fortran77 but build functions into static libraries, and develop C/C++ wrappers
- Link to these libraries from Geant4 or FLUKA (or any other code)
- Code does not become part of the Geant4 source but can have Geant4 examples based on it
- Some disadvantages here
  - requires independent support for libraries
  - intoduce dependencies into codes that use them, Geant4 won't want this, so they'd have to be "optional" in some capacity
- Some advantages here
  - sure that all codes linking these libraries have the same generators (it's the same code)