# Coulomb barrier in the Geant4 Bertini cascade code

Aatos Heikkinen

aatos.heikkinen@cern.ch

Helsinki Institute of Physics, www.hip.fi

P.O.Box 64, 00014 University of Helsinki, Finland

11th Geant4 Collaboration Workshop

LIP, Lisbon, Portugal, October 9-14, 2006

# Current Bertini implementation of Coulomb barrier

- Currently Bertini cascade has an effective Coulomb barrier implemented:
  - In the interfaces possibility for elastic / inelastic interaction is checked using Coulomb barrier

- Coulomb implementation is done:
  - Using phenomenological formulation
  - Also used to set cut off energy for stopping intra nuclear cascade

# Implementation example - G4EquilibriumEvaporator.cc

```
G4double coul_coeff;

 coul_coeff = 1.4;

 std::pair<std::vector<G4double>, std::vector<G4double> > parms =
paraMaker(Z);

std::vector<G4double> AK = parms.first;

// potential

V[i] = coul_coeff * Z * Q[i] * AK[i] / (1.0 + EEXS / E0) /
(std::pow(A1[i], one_third) + std::pow(AN[i], one_third));


 G4InuclSpecialFunctions::paraMaker(G4double Z)

  // Calculates the coefficients for the phenomenological formulas for

  // Coulumb barier, c.s. etc needed for evaporators
```

Trunctuated version of this formulation paraMakerTruncated(Z) is used by pre-equilibrium evaprator.

# Conclusion - Improving Bertini Coulomb barrier

- Current Bertini, with its effective Coulomb barrier implementation, is tuned to provide:
  - The correct total cross-section, fragment production and satisfy energy conservation
  - So some of the problems reported might not be Caused by Coulomb barrier:
    - I'm currently investigating this option
- Yet, there seems to be cases where Coulomb barrier implementation is not working properly:
  - I'll do additional validation using all options available in the code, to clarify this
- When we change Coulomb treatment, we need carefully indentify the cases, where improvement can be made without affecting overall performance:
  - Careful tuning of parameters should be favoured instead of remodelling