USER'S GUIDE
GEANT4 simulation of the DANCE array

Marian Jandel
C-INC, Los Alamos National Laboratory
mjandel@lanl.gov

June 5, 2006

Contents

The version 4.8.0 of GEANT was used for the development of the DANCE simulation application. In order to run the application, GEANT4.8 and CLHEP libraries are required to be installed. Check with the installation procedures on the GEANT4 webpage how to properly set environment variables. The ROOT file is produced as an output of the simulation. Therefore the ROOT has to be installed in the system as well.

"make clean", "make" will compile the application.

1 Run modes of DANCE-GEANT4

After the succesfull compilation of DANCE-GEANT4 application the executable is place in your GEXE directory. Run the application with:

$GEXE/DANCE **inputfile**

Depending on weather you provide the **inputfile** or not, you may run the application in two modes:

- Interactive Mode - no **inputfile**
- Batch Mode - **inputfile** = file you chose to be the MasterInput File

Both modes will require the MasterInputFile. For the interactive mode, the MasterInputFile is set by default to be MasterInputFile.txt. The structure of the input file is described in 2.

1.1 Interactive Mode

If an interactive session is chosen, a user can interactively adopt the simulation options. As a default MasterInput.txt is expected to be present in the DANCE distribution directory as well as vis.mac macro file. *The option of an external input file for generating primary events is not available in this mode.* The interactive session means, that all the commands of GEANT4 User Interface (UI) can be used in the command line after the start of the application. A user can change sources, for example:

\gps\source ion
\gps\energy 0 keV
\gps\ion 39 88 0 0

If the appropriate environmental variables of GEANT are set one can visualize the detector using OPENGL or DAWN (this requires an existing OPENGL and DAWN in your system, and also the compilation of GEANT has to be performed with the OPENGL and DAWN flags switch on. The user is refered to the GEANT4 documentation for more details.)

1.2 Batch Mode

This mode is probably the most convenient and is recommended. The user has a choice to run the application more times for different settings and different sources and in addition, the choice of external input file for generating the primary event is available.

1.2.1 Single Run

To execute the DANCE-GEANT4 application in batch mode the **inputfile** has to be provided in the execute command, for example:

$GEXE/DANCE MasterInput_Run1.txt

where the MasterInput_Run1.txt file has to be present in the directory where the DANCE-GEANT4 is installed.

1.2.2 Multiple Runs

The user can perform multiple DANCE-GEANT4 simulations in a row, by writing a simple bash shell script, for example:

```
# file: multiple_runs.sh

$GEXE/DANCE MasterInputFile_run1.txt
$GEXE/DANCE MasterInputFile_run2.txt
$GEXE/DANCE MasterInputFile_run3.txt
$GEXE/DANCE MasterInputFile_run4.txt
.
.
.
```

Of course, corresponding input files have to exist and the output file name defined in the respective input files should have unique names for each run, in order to prevent an overwriting (ouput files are created with RECREATE option, so if there exist the file with the same name it will be overwritten).

2 MasterInputFile contents

A set of parameters in the MasterInputFile has to be defined in order to succesfully run the application. The use of MasterInputFile.txt included in the distribution is recommended. Parameters in this file were obtained by many iterations where experimental and simulated spectra were compared for a set of calibration γ sources. Conditions may change from experiment to experiment. This file provides the flexibility in changing geometry settings and other variables in order to achieve desirable agreement with the experiment, without the changes in the code itself. Here is an example of the MasterInputFile.txt:

```
# file: MasterInput.txt

# Set the following variables to customize the Geant4 output
# The name of the variables should not be changed

BinaryCrystalFile        false
BinaryClusterFile        false
AddPrimaryGammas         false
AddTimeOfGammas          false
RootAutoSave             5000
RootFile                 true
RootFileName             RootOut3.root
ClusterMGatedHists       true
CrystalMGatedHists       true

SingleGateEcrystal_1     511 keV     0.1 MeV
SingleGateEcrystal_2     1274 keV    0.1 MeV
SingleGateEcrystal_3     0 MeV       0.1 MeV

SingleGateEcluster_1     0 MeV       0.1 MeV
SingleGateEcluster_2     0 MeV       0.1 keV
SingleGateEcluster_3     0 MeV       0.1 MeV

LiHShell                 true        true
BeamPipe                 true        true
SupportingSphere         true        false
Holders                  true        true
PMLense                  true        true
PVCShell                 false
ShowCrystals             0 1000

BeamPipeVacuum           false
CrystalDistance          19.0 cm
LiHDensity               0.7288
LiHInnerRadius           10.5 cm
LiHOuterRadius           16.5 cm
E_threshold              0.3 MeV


MacroInput               true
MacroInputFile           Y88.mac
ExternalInput            false
ExternalInputFile        input.dat
```

The routine that reads the input file is searching for the keywords. If the keyword is found the following string is read in and used to set the variable that corresponds to the keyword. The keywords may be given in any order, although **all of them** have to be present in the inputfile and set appropriately. For clarity, in the code variables are name exactly as the keywords, apart from some exceptions. The user may write the comments directly in the input file although the he/she has to refrain from using the keywords as it would result in a corrupted input read out. Let us go through all the variables and their meaning, although most of them are self explanatory.

- **BinaryCrystalFile, BinaryClusterFile, AddPrimaryGammas,AddTimeOfGammas:** These are not yet implemented and setting them false or true do not influence the simulation. However, information on the primary event is available, though only if the external input file for generating primary events is chosen **ExternalInput**=true.

- **RootAutoSave ——————— int Nevents:** Sets the automatic save of all the root histograms each **Nevents** into **RootFileName**.

- **RootFile ——————— false/true:** Sets the RootFile output flag. The user should true all the time, as for now, no other output is produced.

- **RootFileName ——————— string:** Sets the name of the root output file.

- **ClusterMGatedHists ——————— false/true:** If **true** spectra of the total gamma-ray energy in the event Etot are produced for cluster multiplicity M=2,3,4,5,6,7 (exclusively) and M>=8.

- **CrystalMGatedHists ——————— false/true:** If **true** spectra of the total gamma-ray energy in the event Etot are produced for crystal multiplicity M=2,3,4,5,6,7 (exclusively) and M>=8.

- **SingleGateCrystal_i ——————— double E MeV/keV— double dE MeV/keV:** Up to three gated spectra are available - i=1,2,3. The user has the option to gate on the energy of the gamma-ray in the crystal. If the energy deposited in any crystal is in the interval E±dE, Etot spectrum is produced omitting the energy of the gamma-ray that fell in the gate interval. If E=0 the spectrum is not produced. MeV and keV units are recognized in the input.

- **SingleGateCrystal_i ——————— double E MeV/keV— double dE MeV/keV:** The same as **SingleGateCrystal_i** except the gate is applied on the cluster gamma-ray energy.

- **LiHShell,BeamPipe,SupportingSphere,Holders,PMLense – true/false – true/false:** A first flag determines the presence of the part of DANCE in simulation. The second flag is just for visualization purposes and does not affect the results of the simulation.

- **PVCShell – true/false:** The PVC shell can be added to DANCE set-up,to mimic the plastic plates that are put on the front face of the crystals. It is recommended to set it **false** for now.

- **ShowCrystals – int a – int b :** Flag to visualize the crystals from No. a - No. b. However, this does not work properly yet. The variable does not influence results of the simulation.

- **BeamPipeVacuum – true/false:**
  If **true** the beam pipe will be pumped down to the vacuum (recommended when simulating cascades or whenever the experimental spectra were taken with the pumped down beamline.) For sources simulation, choose **false** as usually, the calibration spectra are taken in air. In set-up, the beampipe will be filled with air also.

- **CrystalDistance — x cm**
  The distance of the front face of the crystals from the center of the ball in cm.

- **LiHDensity —0.7288:** in g/cm^3.

- **LiHInnerRadius,LiHOuterRadius – x cm :** The inner and outer radius of LiH shell in cm

- **E_threshold — x MeV:**
  The lower threshold for the crystal gamma-ray energy. Hits with the lower energy are rejected from the event. The same threshold is used for all crystals.

- **MacroInput,ExternalInput — true/false:**
  When one is set to **true**, the other one should be set to **false**. However, if External input is set to **true** macro should not be loaded (not checked yet).

- **MacroInputName,ExternalInputName — string:**
  Corresponding names of the input files.

3 External input file

The user can supply for the DANCE-GEANT4 simulation the primary event information. The input file format is very simple:
MultGamma
GammaEnergy[0] GammaEnergy[1] ... GammaEnergy[MultGamma]
.
.
Gamma-rays are isotropically emitted from the center of the ball. Simulation stops when the end of input file is reached. At this stage, the primary particles are required to be gamma-rays. If external input file option is chosen several histograms for the characterization of the primary events will be produced.

4 ROOT output histograms

As mentioned earlier, the Root histogram classes are used to produce the output. For now, this is the only option for the output, however I think it is quite sufficient. Here is a list of available Root histograms that the user can find in the root output file after the successful simulation run.

- Total gamma-ray energy of an event
  TH1F *f1=new TH1F("Etot","Etot",1000,0,16)

- Crystal Egamma spectrum. All crystals fills this one
  TH1F *f2,=new TH1F("Egamma","Egamma",1000,0,16)

- Crystal multiplicity of an event
  TH1F *f3=new TH1F(”MultCrystal”,”MultCrystal”,170,-0.5,169.5)
- Cluster Egamma spectrum. All Clusters fills this one
  TH1F *f4=new TH1F(”EgammaCluster”,”EgammaCluster”,1000,0,16),
- Time spectrum for Crystal Hits
  TH1F *f5=new TH1F(”TimeSpectrumCrystal”,”TimeSpectrum_Crystal”,1000,0,200);
- CLuster Multiplicity
  TH1F *f6=new TH1F(”MultCluster”,”MultCluster”,170,-0.5,169.5);
- theta and phi dist. for clusters
  TH1F *f7=new TH1F(”ThetaCluster”,”ThetaCluster”,360,0.,180.)
  TH1F *f8=new TH1F(”PhiCluster”,”PhiCluster”,720,0.,360.)
- theta and phi dist. for crystals
  TH1F *f9=new TH1F(”ThetaCrystal”,”ThetaCrystal”,360,0.,180.)
  TH1F *f10=new TH1F(”PhiCrystal”,”PhiCrystal”,720,0.,360.)
- Time spectrum for Cluster Hits
  TH1F *f11=new TH1F(”TimeSpectrumCluster”,”TimeSpectrumCluster”,1000,0,200);
- Crystal ID’s
  TH1F *f12=new TH1F(”Crystal_ID”,”Crystal_ID”,170,-0.5,169.5);
- Energy of primary gamma rays
  TH1F *f13=new TH1F(”PrimaryEgamma”,”PrimaryEgamma”,1000,0,16)
- Two dimensional histograms of primary gamma-rays multiplicity vs crystal and cluster multiplicities
  TH2F *f14=new TH2F(”Mult_PrimVsCrystal”,”Mult_PrimVsCrystal”,50,-0.5,49.5,50,-0.5,49.5);
  TH2F *f15=new TH2F(”Mult_PrimVsCluster”,”Mult_PrimVsCluster”,50,-0.5,49.5,50,-0.5,49.5);
- primary gamma multiplicity
  f16=new TH1F(”PrimaryMult”,”PrimaryMult”,170,-0.5,169.5);
- Total energy of en event for Cluster multiplicities = 1,2,3,...
  TH1F *f1_cl1=new TH1F(”Etot_cl1”,”Etot_cl1”,1000,0,16)
  TH1F *f1_cl2=new TH1F(”Etot_cl2”,”Etot_cl2”,1000,0,16)
  TH1F *f1_cl3=new TH1F(”Etot_cl3”,”Etot_cl3”,1000,0,16)
  TH1F *f1_cl4=new TH1F(”Etot_cl4”,”Etot_cl4”,1000,0,16)
  TH1F *f1_cl5=new TH1F(”Etot_cl5”,”Etot_cl5”,1000,0,16)
  TH1F *f1_cl6=new TH1F(”Etot_cl6”,”Etot_cl6”,1000,0,16)
  TH1F *f1_cl7=new TH1F(”Etot_cl7”,”Etot_cl7”,1000,0,16)
  TH1F *f1_cl8=new TH1F(”Etot_cl8”,”Etot_cl8”,1000,0,16)

- Total energy of en event for Crystal multiplicities = 1,2,3,...
  TH1F *f1_cr1=new TH1F("Etot_cr1","Etot_cr1",1000,0,16)
  TH1F *f1_cr2=new TH1F("Etot_cr2","Etot_cr2",1000,0,16)
  TH1F *f1_cr3=new TH1F("Etot_cr3","Etot_cr3",1000,0,16)
  TH1F *f1_cr4=new TH1F("Etot_cr4","Etot_cr4",1000,0,16)
  TH1F *f1_cr5=new TH1F("Etot_cr5","Etot_cr5",1000,0,16)
  TH1F *f1_cr6=new TH1F("Etot_cr6","Etot_cr6",1000,0,16)
  TH1F *f1_cr7=new TH1F("Etot_cr7","Etot_cr7",1000,0,16)
  TH1F *f1_cr8=new TH1F("Etot_cr8","Etot_cr8",1000,0,16)

- Crystal energy gated histograms of Etot

  TH1F *f2_crgate1=new TH1F("EgammaCr_gate1",title,1000,0,16)
  TH1F *f2_crgate2=new TH1F("EgammaCr_gate2",title,1000,0,16)
  TH1F *f2_crgate3=new TH1F("EgammaCr_gate3",title,1000,0,16)

  A "title" will be deduced according to the energy interval of the gate

- Cluster energy gated histograms of Etot
  TH1F *f2_clgate1=new TH1F("EgammaCl_gate1",title,1000,0,16)
  TH1F *f2_clgate2=new TH1F("EgammaCl_gate2",title,1000,0,16)
  TH1F *f2_clgate3=new TH1F("EgammaCl_gate3",title,1000,0,16)

5 How2's

5.1 How to add a ROOT histogram

If the user wants to add more histograms, these steps should be followed:

1. Reserve a pointer for the new histogram of your choice in **include/DANCEEventAction.hh**
   For example, TH3D *newhist;
   If you want to use let's say TH3D, check if in the beginning of DANCEEventAction.hh file, TH3D class is included and if not add #include < TH3D.h >,

2. Define the dimensions, name, and title in the **src/DANCEEventAction.cc** file in the constructor of this class DANCEEventAction::DANCEEventAction().

3. Fill the histogram in the **src/DANCEEventAction.cc** in the method of this class void EndOfEventAction(const G4Event* evt). Be careful what you filling in. In this method, an event is being analyzed and several variables are being calculated in given order. Read the code carefully and its comments, they should help in detrmining where the given variable is already calculated and may be used for the output.

4. Add your historam to **AutoSave** method of the class DANCEEventAction again in the **src/DANCEEventAction.cc** file:
   newhist->Write();

Now the new histogram is added to the simulation output.

5.2 How to simulate radioactive source

It is recommended to use RADIOACTIVE_DECAY class of GEANT4. The branching ratios for every source is taken from the databases and simulation of the primary event is very precise and easy. I am not really sure if the angular distributions are being reproduced also, however this may not be of a great concern. One should use the MacroInput and /gps/source commands. Plethora of setting are provided to set the geometry of the source. Examples of Na22 and Y88 and other sources are provided with the distribution.

However, one has to set the RADIOACTIVE_DECAY path variable for GEANT4 otherwise it won't fine the datafiles. If one follows instructions for GEANT4 installation, this variable should be properly set. Make sure, that you downloaded all the datafiles required and that thay exist in corresponding directory.

Another caveat sits in the lifetime of some radioactive sources. If the lifetime is too long, GEANT4 will refuse to decay it in the lifetime of the simulation. One way is to go directly to the datafile for the corresponding isotope and change the lifetime to very small number. Not 0, then it would become stable and GEANT4 would leave it be. I encountered this issue with 137Cs, and there should be more elegant way around it and force the decay regardless its lifetime. I will implement it when I resolve this issue.

Number of event to be simulated is controled by the command:
/run/beamOn Nevents
either in macro or from interactive mode command line.

6 DANCE geometry

The DANCE array consists from 160 BaF_2 crystals of four different types: A - regular pentagon B - irregular pentagon, C - irregular hexagon and D - regular hexagon (see Fig. ??). The dimensions of each crystal type is the same as of the real crystal. Following materials and parts are included in the geometry of the DANCE detector in simulations:

1. 1mm of the wrapping material from PVC around each crystal
2. 6 inches of aluminium holders for each crystal (exact shape of the holders is reproduced)
3. 1mm thick quartz at the back face of the crystal the aluminium photomultiplier holder
4. a supporting aluminium sphere, 2 inches thick with the holes for the appropriate holders
5. ^{6}LiH moderator sphere inside the DANCE array with 10.5 cm and 16.5 cm of the inner and outer radius
6. aluminium beam line
7. radioactive target holder (RTH)

All parts of the DANCE array were simulated as close as possible to the reality.

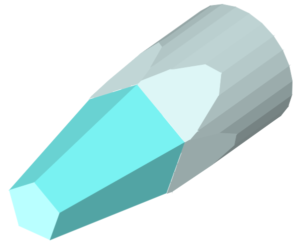

Figure 1: Crystal A with the Al holder

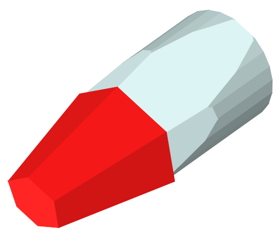

Figure 2: Crystal B with the Al holder

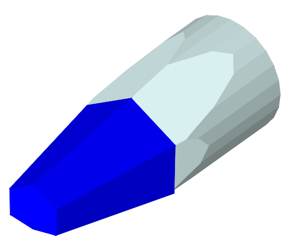

Figure 3: Crystal C with the Al holder

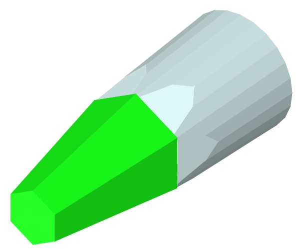

Figure 4: Crystal D with the Al holder

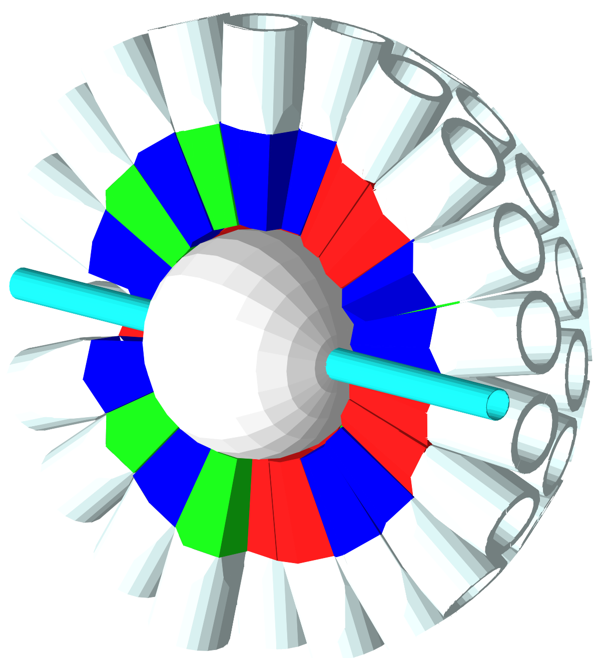

Figure 5: LiH shell, beampipe and 80 crystals of DANCE with Al holders