Directions for running st_mona/geant4/mona_analysis

Compiling and preparing the codes:

1) Compile st_mona code: go into ./st_mona/src and type "make"
2) Compile geant4 code: go into ./n2_geant and type "make"
3) Compile codes used to convert st_mona output to geant4 output: go into ./n2_geant/control and type "./Makefile".
4) Setup directory for where your Sweeper maps are: go into ./st_mona/bin and open mk_maps_icc.sh. Modify all the directories to the directory that you keep your maps.

Running the code:

The entire program is run using a single script: MoNASim_Script_24O_HopeSchiller.sh.

Modifying the script (starting at the top of script):
I often have three lines shown (with 2 commented out) assuming you will want to simulate three different cases: (1) 24O beam with no reaction, (2) 24O(-n) to 23O(g.s.), and (3) 24O(-n) to 23O(excited state)->22O+n.

1) *FileName* is the prefix that will define all the files output from the simulation

```
#FileName="024_022+n"
#FileName="024_023"
FileName="024_024"
```

2) *Arguments passed to st_mona*. This describes how the reaction is simulated. The "exp" input tells the code what experiment and the defining aspects of the experiment are then in st_mona.cc.
The "-geant" input implements the use of geant4. This should be used regardless of neutron emission since the code has been setup to expect this.
The "-reac" input defines the type of reaction/decay. "Glaub" is using glauber kick and then 1n decay. "Glaub_noDecay" is glauber kick and no neutron decay.
The "-e" input provides the energy distribution for the decay. As shown below, this is only used in the cases where we have a neutron decay. It is not used with "Glaub_noDecay".
The "-n" input is the number of events to simulate
The "-strag" and "-glaub" define the strength of the straggling and glauber kick, respectively.
The "-nn" defines the number of neutrons to be stripped during the reaction. (Note that the last line shown below has –nn 0, so this simulated O24 beam with no reaction).

```
#arguments="-exp 09028_024 -geant -reac glaub -e asymbw 0.3 0.1 2 0  -n 2000 -strag 1 -glaub 1.0 -nn 1" #-n number of events.
#arguments="-exp 09028_024  -geant -reac glaub_noDecay  -n 2000 -strag 1 -glaub 1.0 -nn 1" #-n number of events.
arguments="-exp 09028_024 -geant -reac glaub_noDecay  -n 2000 -strag 1 -glaub 1.0 -nn 0" #-n number of events.
```

3) *FRAGname* tells the mona_analysis.cc code what fragment to use and then has some predefined information about the fragment/reaction. Set it according to the chosen reaction (defined above).

```
#fragment for mona_analysis.cc
#FRAGname="022_09028"
#FRAGname="023_09028"
FRAGname="024_09028"
```

4) *MoNaDetFile* defines the file used to input the positioning of MoNA. These files are kept in ./n2_geant/ folder.

```
#MoNA Detector Config File
MoNADetFile="detector_config_ArtemisBe14"
```

5) The MENATEG4XS variable needs to be set either in the script or in your .bashrc to define where to find the associated cross-sections. You should use the folder included in the .zip file as it is the most up-to-date. You will also need to change the 4 paths below to define where each code is run from and lastly, where to put the output files.

```
#export MENATEG4XS="/projects/mona-sim/hope_geant/MENATE_R_CrossSections/"

#Paths to access the executables, and to save the data files.
analysispath="/projects/mona-sim/kohley"
stmonapath="/projects/mona-sim/kohley/st_mona"
n2geantpath="/projects/mona-sim/kohley/n2_geant"
DataPath="/mnt/simulations/MoNA_10He"
```

6) *Outputfiles* are given the suffix defined by the UniqueID. So, if you run one of the reactions with 5 different iterations than you can change the UniqueID for each iteration.
You do not need to change the last 4 lines. But they do define the naming convention I use for the output files. You will get the following 3 output files
a) *.root: is the output from st_mona, which contains the information about the reaction and charged fragment.
b) *-geant.root: is the output after geant4 has run. This contains information on the reaction, charged fragment and neutrons.
c) *-geant-anal.root: is the output after the final mona_analysis has completed. This contains calculated decay energies and Jacobi coordinates.

```
#used to allow multiple geant runs simultaneously
UniqueID="_test"
#Make file names based on the file name given.
PreGeant=${DataPath}/${FileName}"${UniqueID}.root"
PostGeant=${DataPath}/${FileName}"-geant${UniqueID}.root"
FinalName=${DataPath}/${FileName}"-geant-anal${UniqueID}.root"
HistDumpName=${DataPath}/${FileName}"-hist${UniqueID}.root"
```

7) The following three sections (shown below) run the three different codes: (1) st_mona, (2) n2_geant, and (3) mona_analysis

```
#st_mona simulation
cd ${stmonapath}
./bin/st_mona $arguments -f ${PreGeant}

#geant afterburner
cd ${n2geantpath}/control
#control.py -i ${PreGeant} -o ${PostGeant} -t 0.15 ${controlargs} -u ${UniqueID} -d ${DataPath} -c ${MoNADetFile}
control.py -i ${PreGeant} -o ${PostGeant} -t 0.38 ${controlargs} -u ${UniqueID} -d ${DataPath} -c ${MoNADetFile}

#Analysis
#Needed for reconstructed target parameters
cd ${stmonapath}
./bin/mona_analysis -frag ${FRAGname} -if ${PostGeant} -of ${FinalName}
```