```
#!/bin/bash
# =============================================================================
# Developed by: Darren E. Holland, Bryan V. Egner, Julie V. Logan
# Modified by Darren Holland 2020-11-02
# =============================================================================
# This wrapper creates the geometry (CreateMat.py and CreateGeo.inp), compiles
# the Geant evaluation code, runs Geant for each measurement location, and
# calls the analysis script (Analyze.py).
# =============================================================================
# Inputs to ./GeantWrapper.sh $1 $2 $3
# $1 is the folder name
# $2 is the geometry increment number/identifier
# $3 is the results file to return to Dakota

#*******************************************************************************
***************
# Settings
#*******************************************************************************
***************

#Job
threads=2               # Total number of processors
numproc=2               # Number of processors to use per group (On Bridgman use
16 aka all of a node)
# G4MPI (Geant parallel processing) directory
MPIDIR=/app/afit/geant4/geant4.10.03.p02-install-vis/G4MPI/lib64/G4mpi-10.3.2

# Output spectrum binning
nbins=664               # Number of spectrum bins
lbins=0                       # Lowest bin energy (MeV)
# Output spectrum energy cut-offs for detector response curve creation (Use full
energy peak only)
ubins=0.663             # Highest bin energy (MeV)
lcut=0.661              # Ignore all spectrum values below this energy (MeV)
ucut=1000000            # Ignore all spectrum values above this energy (MeV) NOT
CURRENTLY IMPLEMENTED

# Source
StartSource=1           # Start source at voxel edge for voxel at this angle
(degrees)
                    # Example: 25 with deltaphi = 12 starts source at
int(25/12)=2 * 12=24 degrees
s_subdiv=2              # Number of theta sampling subdivisions (Nyquist
criterion)
s_dist=100.             # Source distance (cm)
SourceEnergyType=none         # Source energy spectrum, none = monoenergetic
source
SEnergy=0.662           # Source energy (MeV)
SPart=gamma             # Particle type (gamma or neutron)
nParts2Run=50000    # Number of particles to run for each angle

# Geometry (note origin is at detector's geometric center):
StartGeo=0              # Phi position for starting geometry (voxel edge)
MaskMaterial=PMMA # Mask Material, Options: PMMA
MaskMinThick=0.81 # Mask's minimum thickness must be greater than zero to
connect mask elements (cm)
DH=1.27                       # Detector HALF height (cm)
DR=1.27                       # Detector radius (cm)
SOR=2.69875             # Sleeve outer radius (cm)
SH=2.69875              # Sleeve height above origin (cm)
SB=55                   # Sleeve bottom (cm)
RSMTet=0                # (0) Use Tesselated Solid elements

# Code Directories
```

```
Dakotadir=$(pwd)  # Dakota working directory
cd ../../
homedir=$(pwd)          # Location of Neutron Mask Optimization Code (aka this
code)
Method="MOGA"


#*******************************************************************************
***************
# The following variables are set in the Dakota call (to calculate number of
variables)
#*******************************************************************************
***************
#READ DAKOTA DISCRITIZATION
finthick=$(echo "$(grep ' finthick' "$Dakotadir/$1.MOGApara")" | grep -Eo "[-+]?
[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?.")
wallthick=$(echo "$(grep ' wallthick' "$Dakotadir/$1.MOGApara")" | grep -Eo "[-
+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?.")
deltatheta=$(echo "$(grep ' deltatheta' "$Dakotadir/$1.MOGApara")" | grep -Eo
"[-+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?.")
deltaphi=$(echo "$(grep ' deltaphi' "$Dakotadir/$1.MOGApara")" | grep -Eo "[-+]?
[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?.")
finwidth=$(echo "$(grep ' finwidth' "$Dakotadir/$1.MOGApara")" | grep -Eo "[-+]?
[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?.")
wallwidth=$(echo "$(grep ' wallwidth' "$Dakotadir/$1.MOGApara")" | grep -Eo "[-
+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?.")

# Set phi limits for geometry (geostart to geofinal) and measurements (phistart
to phifinal)
phistart=$(awk -v dd=$StartSource -v ee=$deltaphi 'BEGIN {printf "%.4f\
n",int(dd/ee)*ee}')
geostart=$(awk -v dd=$StartGeo -v ee=$deltaphi 'BEGIN {printf
"%.4f\n",int(dd/ee)*ee}')
phifinal=155                  # Phi final limit (uncomment to set manually)
geofinal=$(awk -v dd=$geostart -v ee=170 -v ff=$deltaphi 'BEGIN {printf "%.4f\
n",int((ee-dd)/ff)*ff+dd}')

#RSMmaxsize is depreciated for this code (set to one so no effect)
RSMmaxsize=1                  # Maximum RSM radius (cm)

# Copy source code into project and working directories
filedir="$homedir/$1/GeantRuns"
workdir="$Dakotadir"
export PYTHONPATH=$PYTHONPATH:$filedir
mkdir "$workdir/bld"
cp "$homedir/SpartOpt.cc" "$workdir/bld"
cp "$homedir/CMakeLists.txt" "$workdir/bld"
if [[ ! -d "$filedir" ]]
then
      # Create working dirctory and copy files
    mkdir "$filedir"
    mkdir "$filedir/res"
      cp -R "$homedir/src/" "$filedir"
      cp -R "$homedir/include/" "$filedir"

      cp "$homedir/Analyze.py" "$filedir"
      cp "$homedir/CreateGeo.py" "$filedir"
      cp "$homedir/CreateMat.py" "$filedir"
      cp "$homedir/intersectLineCylinder.py" "$filedir"
      cp "$homedir/intersectLinePlane.py" "$filedir"
      cp "$homedir/linePosition3d.py" "$filedir"
      cp "$homedir/submitGeant.pbs" "$filedir"
fi
cp "$homedir/Settings.cc" "$workdir/bld/Settingstemp.txt"
cp "$homedir/Geoinp.py" "$workdir/bld/Geoinptemp.py"

# Set contant geometry inputs
sed -i -e  "s?RSMmaxsize=20?RSMmaxsize=$RSMmaxsize?g"
"$workdir/bld/Geoinptemp.py"
sed -i -e  "s?MaskMinThick=0.1?MaskMinThick=$MaskMinThick?g"
"$workdir/bld/Geoinptemp.py"
sed -i -e  "s?sleeve_bottom=55?sleeve_bottom=$SB?g" "$workdir/bld/Geoinptemp.py"

sed -i -e  "s?sleeve_height=5.0?sleeve_height=$SH?g"
"$workdir/bld/Geoinptemp.py"
sed -i -e  "s?sleeve_outer_rad=4.1275?sleeve_outer_rad=$SOR?g"
"$workdir/bld/Geoinptemp.py"
sed -i -e  "s?sleeve_inner_rad=3.81?sleeve_inner_rad=$DR?g"
"$workdir/bld/Geoinptemp.py"
sed -i -e  "s?det_rad=3.81?det_rad=$DR?g" "$workdir/bld/Geoinptemp.py"

sed -i -e  "s?det_height=7.62?det_height=$DH?g" "$workdir/bld/Geoinptemp.py"
sed -i -e  "s?Settings.cc?$workdir/bld/Settingstemp.txt?g"
"$workdir/bld/Geoinptemp.py"

#*******************************************************************************
***************
# Implement Geant settings
#*******************************************************************************
***************
# Set number of threads for parallel Geant runs
sed -i -e  "s?NumThreads = 64?NumThreads = $threads?g"
"$filedir/include/B4aEventAction.hh"
# Set spectrum binning
lbinsmev=$(awk -v dd=$lbins 'BEGIN {printf "%.4f\n",dd/1000}')
ubinsmev=$(awk -v dd=$ubins 'BEGIN {printf "%.4f\n",dd/1000}')
sed -i -e  "s?nofBins(10)?nofBins($nbins)?g" "$workdir/bld/Settingstemp.txt"
            # Number of bins
# Source info
sed -i -e  "s?AmBe.mac?$SourceEnergyType?g" "$workdir/bld/Settingstemp.txt"
              # Source spectrum
sed -i -e  "s?SourceDiv(2)?SourceDiv($s_subdiv)?g"
"$workdir/bld/Settingstemp.txt"           # Number of measurement substeps
sed -i -e  "s?nParts2Run(50000)?nParts2Run($nParts2Run)?g"
"$workdir/bld/Settingstemp.txt"           # Number of source particles
sed -i -e  "s?gamma?$SPart?g" "$workdir/bld/Settingstemp.txt"
                    # Source particle
sed -i -e  "s?SourceDist(86.36)?SourceDist($s_dist)?g"
"$workdir/bld/Settingstemp.txt"     # Source distance (cm)
sed -i -e  "s?numEnergies(1)?numEnergies(1)?g" "$workdir/bld/Settingstemp.txt"
                  # Number of source energies (currently commented out)
sed -i -e  "s?energiesMeV(0.662)?energiesMeV($SEnergy)?g"
"$workdir/bld/Settingstemp.txt"     # Source particle energy
# Geometry discretization
sed -i -e  "s?deltatheta(10)?deltatheta($deltatheta)?g"
"$workdir/bld/Settingstemp.txt"     # Theta increments
sed -i -e  "s?deltaphi(10)?deltaphi($deltaphi)?g"
"$workdir/bld/Settingstemp.txt"           # Phi increments
sed -i -e  "s?phifinal(170)?phifinal($phifinal)?g"
"$workdir/bld/Settingstemp.txt"           # Final phi position
# Geometry (in cm)
sed -i -e  "s?RSMTet(0)?RSMTet($RSMTet)?g" "$workdir/bld/Settingstemp.txt"
                # TesselatedSolids (newer designs)
sed -i -e  "s?DetRad(3.81)?DetRad($DR)?g" "$workdir/bld/Settingstemp.txt"
                # Detector radius
sed -i -e  "s?DetHeight(3.81)?DetHeight($DH)?g" "$workdir/bld/Settingstemp.txt"
                  # Detector HALF height
sed -i -e  "s?SleeveOuterRad(4.1275)?SleeveOuterRad($SOR)?g"
```

```
"$workdir/bld/Settingstemp.txt"     # Sleeve radius
sed -i -e  "s?SleeveHeight(5)?SleeveHeight($SH)?g"
"$workdir/bld/Settingstemp.txt"           # Top of sleeve (extends past
detector)
sed -i -e  "s?SleeveBottom(55)?SleeveBottom($SB)?g"
"$workdir/bld/Settingstemp.txt"           # Total length of sleeve


# Set design number
cd "$workdir/bld"

if [[ $2 -lt 10 ]]
then
      geonum=""0000$2""
fi
if [[ $2 -lt 100 ]] && [[ $2 -ge 10 ]]
then
      geonum=""000$2""
fi
if [[ $2 -lt 1000 ]] && [[ $2 -ge 100 ]]
then
      geonum=""00$2""
fi
if [[ $2 -lt 10000 ]] && [[ $2 -ge 1000 ]]
then
      geonum=""0$2""
fi
if [[ $2 -ge 10000 ]]
then
      geonum=""$2""
fi

#*****************************************************************************
***************
# Create Geometry
#*****************************************************************************
***************
# Overwrite template values and filenames
cp "$workdir/bld/Geoinptemp.py" "$workdir/bld/Geoinp.py"
sed -i -e  "s?EigMat.txt?DesignMat$Method$2.txt?g" "$workdir/bld/Geoinp.py"

Geofile="$workdir/bld/DesignMat$Method$2.txt"
inodes="$workdir/"$1"Nodes"$Method$geonum".inp"
ielements="$workdir/"$1"Ele"$Method$geonum".inp"
sed -i -e  "s?s_dist=86.36?s_dist=$s_dist?g" "$workdir/bld/Geoinp.py"
# Node and element filenames
sed -i -e  "s?Nodesfile='/nodes.inp'?Nodesfile='$inodes'?g"
"$workdir/bld/Geoinp.py"
sed -i -e  "s?Elemfile='/elem.inp'?Elemfile='$ielements'?g"
"$workdir/bld/Geoinp.py"

# Create nodes and elements
Bad_Geo=$(python3 $workdir/bld/Geoinp.py $finthick $wallthick $deltatheta
$deltaphi $finwidth $wallwidth $geostart $geofinal $DR $DH)
# If invalid geometry skip evaluation
if [[ $Bad_Geo == 0 ]]
then
    echo "Bad_Geo 0" > $Dakotadir/$3
      # Calculate variance reduction using cone angle created in geometry file
      coneangle=$(grep 'coneangle(' "$workdir/bld/Settingstemp.txt" | cut -c
32-)
      coneangle="${coneangle%)*}"
      VRval=$(awk -v dd=$coneangle 'BEGIN {printf "%.6e\n",(1 - cos(dd*atan2(0,-
1)/180)) / 2}')
```

```
      # Read nodes and elements into Geant
      cp "$workdir/bld/Settingstemp.txt" "$workdir/bld/Settings.cc"
      sed -i -e  "s?RSM_nodes.inp?$inodes?g" "$workdir/bld/Settings.cc"
      # Nodes
      sed -i -e  "s?RSM_elements.inp?$ielements?g" "$workdir/bld/Settings.cc"
      # Elements
      sed -i -e  "s?Ofile?"$1$Method$geonum"?g" "$workdir/bld/Settings.cc"   #
Geometry output file

      # Create copy of settings file for records
      cp "$workdir/bld/Settings.cc" "$workdir/bld/Settings"$Method$geonum".txt"


#*******************************************************************************
***************
    # Implement Geant measurement settings

#*******************************************************************************
***************
    # Set start and final phi measurement for Geant (+0.000001 so get last
angle)
      sed -i -e  "s?StartPhi(0)?StartPhi($phistart)?g"
"$workdir/bld/Settings.cc"
    sPhi2=$(awk -v dd=$phistart -v ee=$phifinal -v ff=$deltaphi 'BEGIN {printf
"%.6f\n",int((ee-dd)/ff)*ff+dd+ff-0.000001}')
      sed -i -e  "s?EndPhi(170)?EndPhi($sPhi2)?g" "$workdir/bld/Settings.cc"

    # Calculate total number of measurements
    nphi=$(awk -v dd=$phistart -v ee=$phifinal -v ff=$deltaphi 'BEGIN {printf
"%.6f\n",int((ee-dd)/ff+1)}')
    ntheta=$(awk -v ff=$deltatheta -v ss=$s_subdiv 'BEGIN {printf "%.6f\
n",int(360/ff*ss)}')
      runs=$(awk -v ff=$nphi -v ss=$ntheta 'BEGIN {printf "%.0f\n",ff*ss-1}')
    # Make folder for each measurement
      for cc in $(seq 0 $runs)
        do
          mkdir $cc
        done

#*******************************************************************************
***************
    # Run Geant

#*******************************************************************************
***************
    # Compile Geant design
      cmake -DFILE_DIR=$filedir -DExeName=SpartOpt -DSettings=Settings -
DExeName2="SpartOpt$geonum$ii"  -DG4mpi_DIR=$MPIDIR .
      make "-j$numproc"
    # Run Geant design
      mpiexec -n $threads $workdir/bld/SpartOpt$geonum$ii
    # Check to see if Geant runs are complete
    comp=0
    while [ $comp == 0 ]
      do
          # Check if a thread has finished
          if [[ ! -f  "$workdir/bld/"$1$Method$geonum".comp" ]]
          then
               # If not complete, then check again in 30 sec
               sleep 30s
          else
               # ALL threads have finished
               comp=1
               # Wait 1 sec to ensure files are closed
               sleep 1s
          fi
      done
    # Combine results and delete thread files
      for cc in $(seq 0 $runs)
    do
       RunName=$(head ./$cc/SourcePos.txt)
       cat ./$cc/*.ww > $RunName
       rm -r $cc
    done



#*******************************************************************************
***************
    # Analyze results and return to Dakota

#*******************************************************************************
***************
      # Move files and clean up directory
      mkdir $filedir/res/$Method$geonum
      find "$workdir/bld/" -maxdepth 1 -name "*.o" -exec mv -t
"$filedir/res/$Method$geonum" {} +
      find "$workdir/bld/" -maxdepth 1 -name "WallPos.inp" -exec mv -t
"$filedir/res/$Method$geonum" {} +
      find "$workdir/bld/" -maxdepth 1 -name "FinPos.inp" -exec mv -t "$filedir/
res/$Method$geonum" {} +
      find "$workdir/bld/" -maxdepth 1 -name "*.ascii" -exec mv -t
"$filedir/res/$Method$geonum" {} +
      find "$workdir/bld/" -maxdepth 1 -name "*.root" -exec rm {} +
      find "$workdir/bld/" -maxdepth 1 -name "*.comp" -exec rm {} +

    # Wait to ensure files are moved before analyzing them
    sleep 5s

      # Analyze the results
      echo "python3 ./Analyze.py ./dummy_file ./res/$Method$geonum/Results
$SPart $s_subdiv $deltaphi $phifinal $deltatheta ./res/$Method$geonum $VRval
$nParts2Run $lcut $ucut $lbins $ubins $nbins $wallwidth $finwidth 1 $geostart
$geofinal" > $filedir/AnalyzeCommand$Method$geonum.sh
    chmod u+x $filedir/AnalyzeCommand$Method$geonum.sh
    python3 $filedir/Analyze.py $Dakotadir/$3
$filedir/res/$Method$geonum/Results $SPart $s_subdiv $deltaphi $phifinal
$deltatheta $filedir/res/$Method$geonum $VRval $nParts2Run $lcut $ucut $lbins
$ubins $nbins $wallwidth $finwidth 0 $geostart $geofinal
else
   #echo "Skipping invalid geometry..."
   echo "Bad_Geo "$Bad_Geo >> $Dakotadir/$3
fi
echo $Dakotadir/$3
# Change to initial run directory
cd $Dakotadir
```