

# Beam Simulation and Radiation Dose Calculation at the Advanced Photon Source with *shower*, an Interface Program to the EGS4 Code System\*

L. Emery  
Argonne National Laboratory  
9700 So. Cass Ave., Argonne, IL 60439

## Abstract

The interface program *shower* to the **EGS** [1] Monte Carlo electromagnetic cascade shower simulation code system was written to facilitate the definition of complicated target and shielding geometries and to simplify the handling of input and output of data. The geometry is defined by a series of namelist commands in an input file. The input and output beam data files follow the SDDS [2] (self-describing data set) protocol, which makes the files compatible with other physics codes that follow the same protocol. For instance, one can use the results of the cascade shower simulation as the input data for an accelerator tracking code. The *shower* code has also been used to calculate the bremsstrahlung component of radiation doses for possible beam loss scenarios at the Advanced Photon Source (APS) at Argonne National Laboratory.

## I. Introduction

The APS consists of a linear accelerator, a positron accumulator ring (PAR), a booster, and a storage ring (SR) which provide synchrotron radiation for users. These accelerators are (mostly) above ground and require heavy shielding to protect personnel from radiation produced by electron or positron beam loss. The shielding has been designed [3] in a conservative way using empirical formulas [4] of bremsstrahlung and neutron production for various loss scenarios. However, from a physicist's point of view, it is desirable to obtain a more accurate value of radiation dose, say from running a Monte Carlo electromagnetic cascade shower code such as **EGS** [1]. Only when comparing an actual radiation dose measurement with a realistic calculation can one conclude whether the shielding performs as expected.

Another important calculation is the simulation of beams through foils, targets, or any other medium where the beam emerging on the other side requires further tracking. An accelerator code tracks particles before reaching the medium, then a program like **EGS** tracks particles within the medium, and the accelerator code tracks the emerging particles afterwards. In general, a program is written as a stand-alone unit, with no thought of its output being used by another program. Fortunately, the structure of **EGS** allows users to write their own interface to input and output files. I wrote the *shower* interface program for **EGS** with a newly defined protocol for data files [2] in mind, which makes *shower* data files compatible with the accelerator tracking code *elegent* [5] which uses the same protocol.

The paper will present the *shower* interface to the **EGS** system of code, and some applications to APS.

## II. Interface to **EGS**

The code system **EGS** is a tried-and-true Monte Carlo shower code used by the health physics community and the high energy

physics detector community. **EGS** simulates electromagnetic showers by tracking individual particles with random discrete scattering events through regions filled with material. By summing up the energy losses in regions simulating human tissue, one can calculate the absorbed dose for humans.

In **EGS**, the user must specify one subroutine which defines the geometry of the problem and another for handling the output of data. The main routine is also user-provided and controls the calling of the main **EGS** routine. **EGS** users have traditionally written these subroutines in **MORTRAN** which requires a preprocessor to convert the code into **FORTRAN**. As I am dissatisfied with **MORTRAN** and **FORTRAN**, I decided to write these subroutines in **C** in order to use essential features of **C** such as structures and pointers. These subroutines and other supporting **C** subroutines form the interface. The **FORTRAN** physics core of **EGS** is left unchanged and linked to the **C** interface with no difficulty.

*shower*'s command line requires two input files: a geometry file describing regions of material, and a particle coordinate file for the incident beam. The command line requires the naming of a summary file into which energy deposition and radiation dose information of each region is written. Other command line options control the random number generator seed, multiple sampling of the input beam, trajectory recording, input or output particle type filtering, particle splitting across boundaries, actions taken when particles go out of bounds or enter an undefined region, etc.

The geometry file contains namelist-type commands that define material regions shaped as blocks or cylinders with an optional output file specification for writing coordinates of particles crossing the material boundary. For example, the following lines define a 7-mm-thick tungsten target and a region where particle coordinates are recorded, possibly for further tracking.

```
&cylinder label="W target" material="W"  
  r'lower'cm=0 r'upper'cm=2  
  z'lower'cm=0 z'upper'cm=0.7 &end  
&cylinder label="Output volume"  
  material="Vacuum"  
  outputfile="target-products"  
  r'lower'cm=0 r'upper'cm=1  
  z'lower'cm=0.7 z'upper'cm=1.0 &end
```

With variables such as *x*'slices, one can generate multiple regions divided equally along one or more spatial coordinates. This is useful for calculating radiation dose in a material as a function of one or more spatial coordinates. If a region definition overlaps a previously defined region, then the volume in common keeps the previous material definition. Thus, complicated geometries can be generated.

All beam data files and the summary file comply with the SDDS self-describing data file protocol [2] developed at APS. Once such a protocol is adopted, one can immediately draw on the SDDS toolkit for graphics and data postprocessing. SDDS-protocol files are organized in "pages" consisting primarily of

\*Work supported by U.S. Department of Energy, Office of Basic Energy Sciences under Contract No. W-31-109-ENG-38.

tables of data. Each page may have non-tabular values, such as parameters, associated with it. The particularly useful feature of the protocol is the attribution of names, units, data type, and descriptions to the data by way of a file header. Thus, columns in the data tables and parameters can be operated upon by listing their names as part of an option of an SDDS toolkit command line. For instance, suppose that we want to filter data, histogram the filtered data, and plot the results. We assume that the data file `target-products` exists and has at least a data column named `Energy` and a parameter called `Type` giving the particle type as a string. Then one issues the following three self-explanatory SDDS commands:

```
sddsprocess target-products target.positrons
-match=parameter,Type=positrons,
sddshist target.positrons
target.positrons.hist -dataColumn=Energy,
and
sddsplot -column=Energy,frequency
target.positrons.hist
where target.positrons and
target.positrons.hist are SDDS files.
```

The columns defined in the particle coordinate files are `Energy`, `x`, `y`, `z`, `u`, `v`, `w`, where the last three are direction cosines. Each data page has a string parameter `Type` assigned one of `electron`, `positron`, or `photon`. Therefore, if a **shower** run produces the three types of particles, then at least three data pages are created in the output file, one for each type. **shower** does not presently support neutron creation or transport. However, if neutrons were included in the simulation, one would simply introduce the particle type `neutron`. The structure of the data file is unaffected.

The summary file contains data related to each region defined, such as the coordinates of the center of the region, the volume, the energy deposited from each type of particles, and absorbed dose. In the case of a region defined with human tissue, the absorbed dose is taken to be the dose equivalent for biological damage.

The availability of a toolkit permits the design of a lean interface program. One does not have to build in graphics or data processing into the code itself. The beam data can be written to a file for later processing, and one doesn't have to anticipate or implement all possible ways of presenting data in a physics code. Referring to the examples above, the tool **sddshist** can histogram any variable in the beam data files, a preferable method than imbedding histogram routines inside **EGS** or **shower**. Another tool in frequent use is **sddsprocess** which uses the data to perform mathematical calculations and other general processing functions such as filtering.

### III. Applications

#### A. Beam Loss at Undulator Chamber

The APS ring will have Al undulator chambers with small vertical aperture to accommodate narrow gap undulators. A tapering Cu transition connecting the undulator chamber to the regular storage ring extruded chamber will be the most probable location of beam loss. Copper is a relatively heavy target that may generate a shower in a short distance, which tends to concentrate the dose. Most of the radiation will be directed forward into the curving storage ring high-density concrete tunnel at a small incident angle, thus giving a large effective depth. However the portion of radiation directed sideways, though relatively weaker, travels through only the 0.56 m of concrete. The **shower** calculation for this situation can confirm the amounts

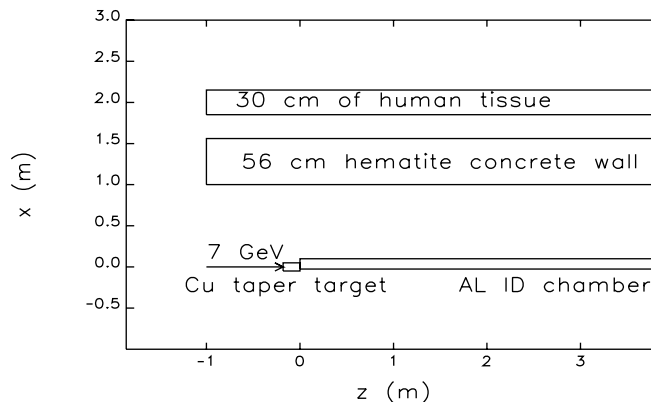


Figure 1: Top view of the material layout for beam loss at the undulator chamber

of extra shielding required.

It is commonplace to use empirical formulas to estimate dose rates for situations such as this one. In this case, the empirical formulas give dose rates that are about 10 times that calculated by **shower**. This factor is due mostly to the treatment of the attenuation of photons in the shielding. Part of the factor is the sometime erroneous assumption that the electromagnetic shower expends itself in a short distance compared to the distance from the source to the dose point.

Figure 1 shows the layout of the radiation dose calculation. The human tissue region used in the calculation is split into smaller regions along the `z` direction so that the average dose in each section can be obtained as a function of `z`.

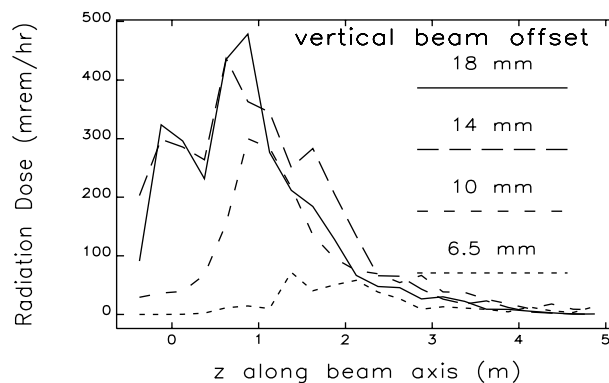


Figure 2: Calculated radiation dose profile outside the shielding wall along the beam direction for a 308-W injected beam loss with concrete shielding only.

Figure 2 shows the dose profile outside the shielding wall with no extra shielding for various injected beam vertical steering errors. The variation between profiles is due to the varying amount of Cu material through which the 7-GeV beam travels.

#### B. Beam Loss at Booster

In a possible worst-case radiation exposure incident in the booster, the fully accelerated 7-GeV positron beam hits the vacuum chamber inside a booster magnet below a 1-m-thick concrete ceiling in the extraction area. A fraction of the shower

products will spray upwards through the ceiling, and into an area that may be occupied by people.

For the beam loss location, I selected a focusing quadrupole where both a loss and the largest dose outside shielding is likely to occur. Though it would seem that any focusing quadrupole is as good as any other, I took one followed by a substantial accelerator component, like an rf cavity, so that the increase in dose outside the shielding due to additional scattering may be taken into account.

In the modeling of the geometry, all the material shapes have been generated by adding blocks of materials and vacuum. For instance, the elliptical vacuum chamber is modeled as a 1-mm-thick rectangular chamber of the same outer dimensions as the elliptical one. Similar changes to the quadrupole poles were made. The rf cavities were approximated by copper boxes of the same outer dimensions as the real ones. The calculation does not include any additional shielding that may eventually be installed to reduce the dose rate. The dose rate profile of the worst case of incident angle and position is shown in Figure 3.

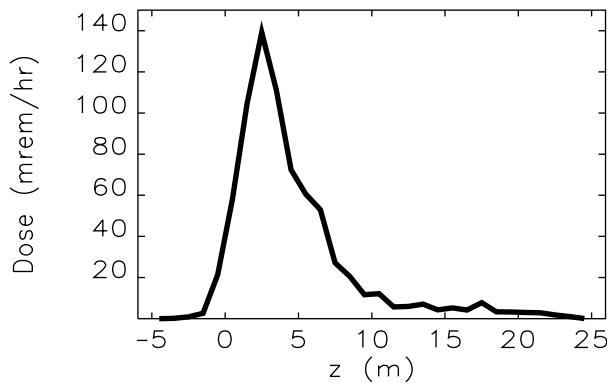


Figure 3: Calculated radiation dose profile above the booster ceiling along the beam direction for a 308-W booster beam loss with concrete shielding only.

### C. Gas Bremsstrahlung Photon Scattering

The collisions between 7-GeV positrons of the APS stored beam and residual gas atoms in the ring 15-m straight sections produce x-ray photons ( $E > 1$  MeV) which propagate down the x-ray beam pipe in the same direction as those of insertion device (ID) device x-rays ( $E < 10$  keV). These photons are separated thirty meters downstream by a 1.3-m copper mirror which externally reflects the ID photons by 0.3 degree (5.2 mrad). About half of the bremsstrahlung photons hit the mirror and get absorbed or scattered. The other half hit an absorber or an absorbing wall behind the mirror. A very small fraction of high-energy photons scatters from the mirror and into beam the small solid angle acceptance of the ID photon collimators. These high-energy photons get scattered further downstream at an x-ray stopper inside a shielded beam pipe located in the experiment hall. Since the shielding surrounding these scatterers is designed for ID photons, and since these photons have energies of the order of MeVs (a thousand times the ID photons), one must make sure that this shielding is sufficient.

The geometry of the problem is somewhat complicated, as there are five scatterers, some widely spaced: a 15-m length of STP N<sub>2</sub> which simulates, with proper pressure scaling, the effect of few nT-pressure residual gas ( $z = 0$  m); a Cu mirror ( $z = 30$

m); a Pb collimator ( $z = 37.5$  m); a x-ray W stopper ( $z = 41$  m); and a lead-sandwich shielding beam pipe ( $z = 41$  m).

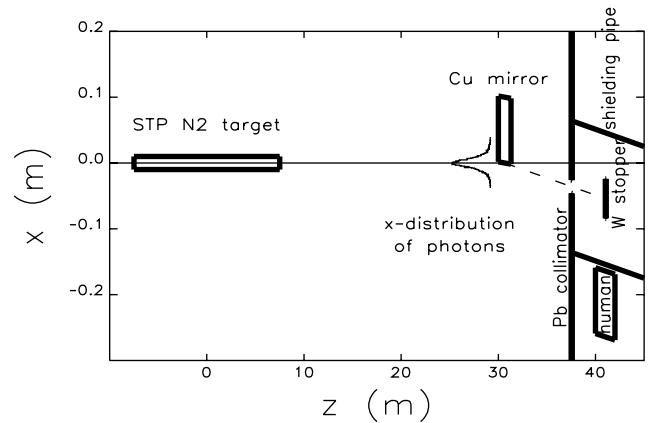


Figure 4: Top view of the material layout for bremsstrahlung photon propagation

Figure 4 lays out the geometry of the problem with an expanded horizontal scale.

The problem is split into four **shower** runs, about one per scatterer. Particle data are available after every run. Thus, particle fluxes and intensity distributions can be analyzed or plotted for selected points along the  $z$  axis. One such distribution appears in Figure 4.

For this situation, the radiation dose near the x-ray beamline from gas bremsstrahlung is given in mrem/hr by  $2 \times 10^{-3} (E[\text{GeV}]/7) I[\text{A}] P[\text{nT}]$  where  $E$  is the energy of the stored beam (normally 7 GeV),  $I$  is the stored current (0.1 A or 0.3 A), and  $P$  is the residual gas pressure (a few nT). The typical radiation dose is a few microrrem/hr, a very small value.

## IV. Acknowledgement

I thank H. Moe for guidance on the subject of radiation production. I thank M. Borland for advice on data and program organization.

## V. References

- [1] W. R. Nelson, H. Hirayama, and D. W. O. Rogers, "The EGS4 Code System," SLAC 265, SLAC, December 1985.
- [2] M. Borland, "A Self-Describing File Protocol for Simulation Integration and Shared Postprocessors," these proceedings.
- [3] H. J. Moe, "Advanced Photon Source: Radiological Design Considerations," APS LS-141, revised, ANL, July 1991.
- [4] W. P. Swanson, "Radiological Safety Aspects of the Operation of Electron Linear Accelerators," Technical Report Series N 188, IAEA, 1979.
- [5] M. Borland, "User's Manual for **elegant**," APS LS-231, ANL, May 1993.