

```
*****
** penEasy_Imaging , version 2010-09-02 **
*****
```

```
@author Andreu Badal
@date 2010/09/02
```

1 Introduction

PenEasy_Imaging is an extension of the general-purpose Monte Carlo particle transport simulation toolkit penEasy that provides new capabilities that allow the simulation of medical imaging systems. Currently, penEasy_Imaging is based on the penEasy version 2008-06-15 and uses the atomic physics modeling subroutines from PENELOPE 2006.

PenEasy_Imaging is being developed at the *U. S. Food and Drug Administration (FDA), Center for Devices and Radiological Health, Office of Science and Engineering Laboratories, Division of Imaging and Applied Mathematics*. Partial funding for the development of this software has been provided by the FDA *Office of the Chief Scientist* through the *Computational Endpoints for Cardiovascular Device Evaluations* project and by the FDA *Office of Women's Health*.

The source code and documentation of penEasy_Imaging are openly and freely distributed at the website: <http://code.google.com/p/peneasy-imaging/>. The following disclaimer notice applies to the code and documentation developed exclusively at the FDA (this disclaimer is provided at the beginning of each file developed at the FDA):

**** DISCLAIMER ****

This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

The penEasy package is actively developed by Josep Sempau at the Universitat Politècnica de Catalunya (Spain). This code provides a modular, user-friendly main program for PENELOPE and many useful tally options and source models. It also has a flexible geometric model in which the simulated objects can be described using quadric surfaces (standard PENELOPE geometry), using voxels, or using a combination of voxels and quadrics. The users should refer to the penEasy README and documentation files for further information. The latest version of this open-source software can be obtained at the website: <http://www.upc.edu/inte/en/descarregues.php>. PenEasy is copyrighted by the Universitat Politècnica de Catalunya:

The PENELOPE subroutine package for the simulation of x-ray, electron and positron transport in matter are developed by Francesc Salvat and Josep Sempau at the Universitat de Barcelona (Spain). The package is distributed by the Nuclear Energy Agency Data Bank. Despite the PENELOPE code is required to compile penEasy, this code is not included in the penEasy_Imaging package (see section 4 for more details). The PENELOPE user manual and instructions on how to obtain the code are provided at the website: www.nea.fr/abs/html/nea-1525.html. PENELOPE is copyrighted by the Universitat de Barcelona:

2 Main features of penEasy_Imaging

PenEasy_Imaging implements two majors extensions to the penEasy code:

- **TALLY IMAGE**

The Tally Image is a tally module to generate radiographic images as the output of the penEasy x-ray transport simulation. The images can be generated using four different imaging formation models:

1. Image produced by scoring the 2D energy deposition distribution inside the detector object with a user-defined uniform grid of pixels. The pixel values have absolute units of eV/cm^2 per history. If the detector material is set to be completely absorbent, the image values will be equivalent to the energy fluence inside each pixel. With this tally the user can choose one of these three options: generate a single output image including all radiation; tally only primaries (the simulation is sped up skipping the tracking of scatter); or tally separately the primary, the secondary (electrons, fluorescence) and the scattered particles, differentiating photons that suffered Rayleigh, Compton, and multiple scattering events in the track.
2. Image generated modeling the transport of optical photons created after each x-ray interaction inside the scintillator. The optical photons are sampled using a triple Gaussian model fitted to an experimentally measured point spread function, as described by Kyprianou et al. [Med. Phys. 35, pp. 4744-4756 (2008)].
3. Analytical ray-tracing model: this option disables the Monte Carlo sampling and utilizes the penEasy transport routines as an analytical ray-tracing engine to generate scatter- and noise-free images in a short computing time. In this model a reduced number of rays are cast from the focal spot to different positions inside each pixel to estimate the probability of x rays arriving at the pixel without interacting in the track. This process is repeated by each energy bin given in the input energy spectrum. The final image is scaled by the probability of emitting the x rays within the solid angle covering each pixel, assuming that the source is collimated to exactly cover the detector surface. The reported pixel values have the same units as model 1.
4. Extension of model 2 in which the optical photons are created using a depth-of-interaction-dependent point spread function Gaussian model based on MANTIS simulations [Kyprianou et al., Proc. SPIE, Vol. 6913, p. 69130 (2008)].

The user has to provide information on the modeled detector in the tally input file, such as the location and orientation of the detector, the size of the detector's sensitive region and the number of (virtual) pixels in which this region is divided. It is important to note that a "real" detector object (usually a thin slab of scintillator material such as CsI) has to be defined in the simulation's geometry using quadric surfaces. Without this quadric object in place, most particles will not interact inside the detector volume and the image will not be correctly generated. The location and orientation of the detector in the quadric geometry have to match the values given in the input file. The output images can be written in ASCII or in binary format. Simple GNUplot scripts are provided to visualize graphically the simulated images.

Sample input file of the image tally:

```
[SECTION TALLY IMAGE v.2010-09-02]
(ON )                # STATUS (ON or OFF)
 5                   # DETECTOR MATERIAL IN THE QUADRIC GEOMETRY
 1                   # DETECTOR MODEL (1=eV/cm^2, 2=Optical, 3=Ideal, 4=OpticalDepth)
 0                   # SCATTER OPTION (0=All, 1=Only-non-scattered, 2=Separate-scatter)
 0                   # OUTPUT FORMAT (0=ASCII, 1=Binary)
image_model_1.dat     # OUTPUT FILE NAME (name ends at first blank)
10.0 250              # X SIZE DETECTOR (cm), NUM PIXELS
10.0 250              # Z SIZE DETECTOR (cm), NUM PIXELS
0.06                 # Y SIZE DETECTOR (PIXEL THICKNESS)
0.0 0.0 0.0          # ROTATION EULER ANGLES (OMEGA, THETA, PHI) [deg]
0.0                  # ROLL ANGLE (detector tilt, usually 0) [deg]
5.0 20.0 5.0         # X, Y, AND Z TRANSLATION DETECTOR CENTER FROM ORIGIN
0.0                  # RELATIVE UNCERTAINTY (%) REQUESTED
[END OF IMG SECTION]
```

- **SOURCE RECTANGULAR BEAM**

The Source Rectangular Beam, also called fan-beam source, is a new source model that emits a collimated beam of radiation that produces a rectangular field of radiation on the detector plane. The beam emitted by this source is equivalent to a cone beam emitted from a point focal spot and collimated with 100% attenuating blocks. The user can choose the polar and azimuthal apertures of the fan beam and also the tilting of the rectangular beam (by default the field is perpendicular to the XY plane). The energy spectrum of the emitted fan beam is read from an external file, and it is possible to define different energy spectra for different angular intervals (*e.g.*, to reproduce a bow-tie filter). This source model is not compatible with the ray-tracing model of the Image Tally.

Sample input file of the rectangular beam source:

```
[SECTION SOURCE RECTANGULAR BEAM v.2010-09-02]
(ON )                # STATUS (ON or OFF)
 2                   # PARTICLE TYPE (1=electron, 2=photon, 3=positron)
energy_spectra.dat    # NAME FILE CONTAINING THE ANGLE/ENERGY SPECTRA
70.0 5.0 5.0         # POINT FOCAL SPOT COORDINATES (cm)
-1.0 0.0 0.0         # DIRECTION VECTOR, NO NEED TO NORMALIZE
 2.0 10.0 0.0        # POLAR AND AZIMUTHAL APERTURE AND TILT ANGLE [deg]
[END OF RECTANGULAR BEAM SECTION]
```

3 *Package contents*

The penEasy_Imaging package contains the complete penEasy (2008) package with some minor modifications, the code of the imaging extensions, some auxiliary files, and a sample simulation. The user should carefully read the original documentation from penEasy to learn how the program is executed and how the voxelized and quadric geometries are managed by the main program.

The following files are included in the distributed package:

/fortranCode: folder containing the source code. The files containing the imaging extensions are **tallyImage_2010-09-02.f** and **sourceRectangularBeam_2010-09-02.f**.

README_penEasy_Imaging_2010-09-02.pdf: penEasy_Imaging documentation file.

README_penEasy_2008-06-15.txt: original documentation files from penEasy.
/documentation

Makefile_penEasy_Imaging_2010-09-02_gcc: sample make file scripts.

Makefile_penEasy_Imaging_2010-09-02_intel

Makefile_penEasy_Imaging_2010-09-02_largeMemory_intel

penEasy_Imaging_2010-09-02_intel.x: sample compiled versions of the code.

penEasy_Imaging_2010-09-02_gcc.x

penEasy_Imaging_2010-09-02_largeMemory_intel.x

command.in: penEasy run-time control file.

/SAMPLE_RUN_penEasy_Imaging_2010-09-02: sample simulation.

penEasy_Imaging_2010-09-02_sample.in: sample input file

energy_spectra.dat: sample energy- and angle-dependent spectra for the rectangular beam source.

/voxGeoSample: sample voxelized geometry file and code to upgrade files from previous versions.

/gnuplotScripts: GNUplot scripts to visualize the simulation results.

combine_images_parallel_runs_scatter.py: python script to combine the images from parallel executions of the code.

sed_replace_maxvox_maxmat.sh: sed (Stream Editor) script to modify the number of materials and voxels defined in the penEasy and PENELOPE source code.

4 Code compilation

The penEasy_Imaging code can be easily compiled in a Linux workstation using the provided make scripts: re-name the appropriate make script to “Makefile” and execute the command `make` in the Linux shell. Sample executable files statically compiled with the GCC and the Intel compiler are provided.

Notice that the user must obtain the files **pengeom.f**, **penvared.f**, **penelope.f** from the standard PENELOPE 2006 package and copy them to the /fortranCode folder before the compilation. The PENELOPE files are NOT distributed in this package (contact the author for more information on this topic).

To be able to write binary files in a consistent manner, the Fortran code calls two function written in C (from file **write_binary.c**). Thus, the compilation of the code requires both a Fortran and a C compiler. If a C compiler is not available, the user can disable the binary output commenting out the functions "writefloat" and "writeus" in the source file **tallyImage_2010-09-02.f**.

5 Contact information

Questions, bug reports, feature suggestions, etc, can be posted in the “Issues” section at the code distribution website <http://code.google.com/p/peneasy-imaging/>.

For further information, the users can also directly contact the code developer at the address:

Andreu.Badal-Soler@fda.hhs.gov.

All users of the code are encouraged to cite penEasy in their scientific publications. The tentative citation of penEasy is the following:

J. Sempau, A. Badal and L. Brualla, “penEasy and penEasyLinac, a penelope-based system for the Monte Carlo simulation of linacs and voxelised geometries. Application to far-from-the-axis fields”, submitted to Physics in Medicine and Biology (2010)