- -0 -



A Simulation Tool for Parallel beam Computed Tomography Based on MCNP

Christopher Mutzel, Oguz Semerci, and Eric L. Miller, Tufts University

Contact: christopher.mutzel@tufts.edu, oguz.semerci@tufts.edu, elmiller@ece.tufts.edu

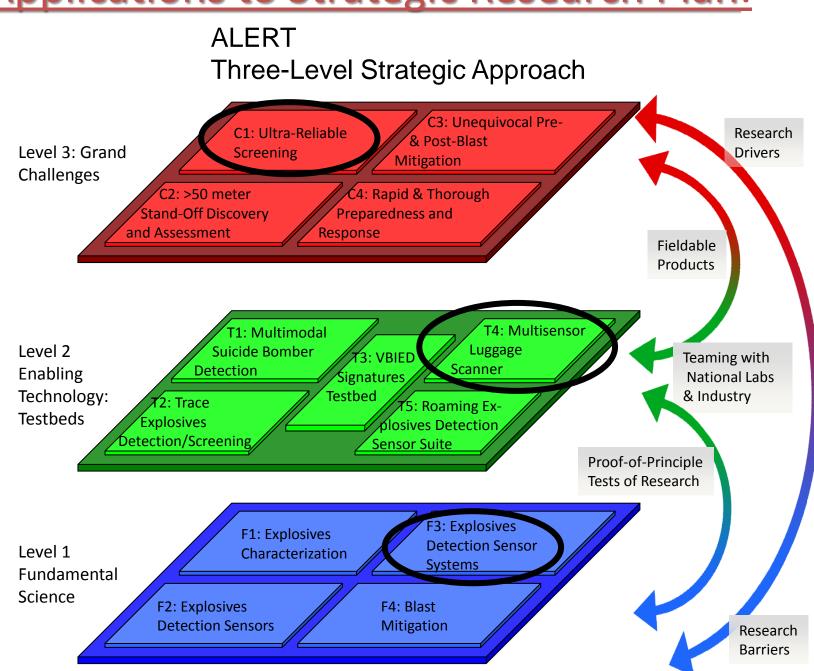


School of Engineering

Abstract

The aim of this work is to create a tool with the ability to simulate parallelbeam photon transport in two dimensions, creating data similar to that produced by operational CT scanners. Data are generated using Monte-Carlo N-Particle (MCNP), a software package distributed by Los Alamos National Laboratory. MCNP provides individual particle tracking and records aspects of their average behavior, making it ideal for constructing datasets that represent x-ray scans of an arbitrary geometry. The tool we are developing will provide the user with a base text file which may be edited to define sample geometry, energy range of the photons, and number of particles. The user is provided with the opportunity to choose the number of rotation angles for the system, physical aspects of the particles, as well as format of the output file. Output data can be in units of particles/time, energy/time, total particles, or total energy. Results may be modified to include the number of collisions that photons undergo during transport, a useful value when scatter is considered. This base input file is then modified and used to produce a sequence of MCNP input files, one for each rotation angle. The program runs each input file, and output files are parsed to build an array of values for each detector over all angles. Final output includes runtime information and statistical variation of the tallies, as well as the data mentioned above. Tests with simple geometries for well-defined materials show that this tool can be used to accurately simulate data, which can then be used to test reconstruction algorithms

Applications to Strategic Research Plan:



State of Art:

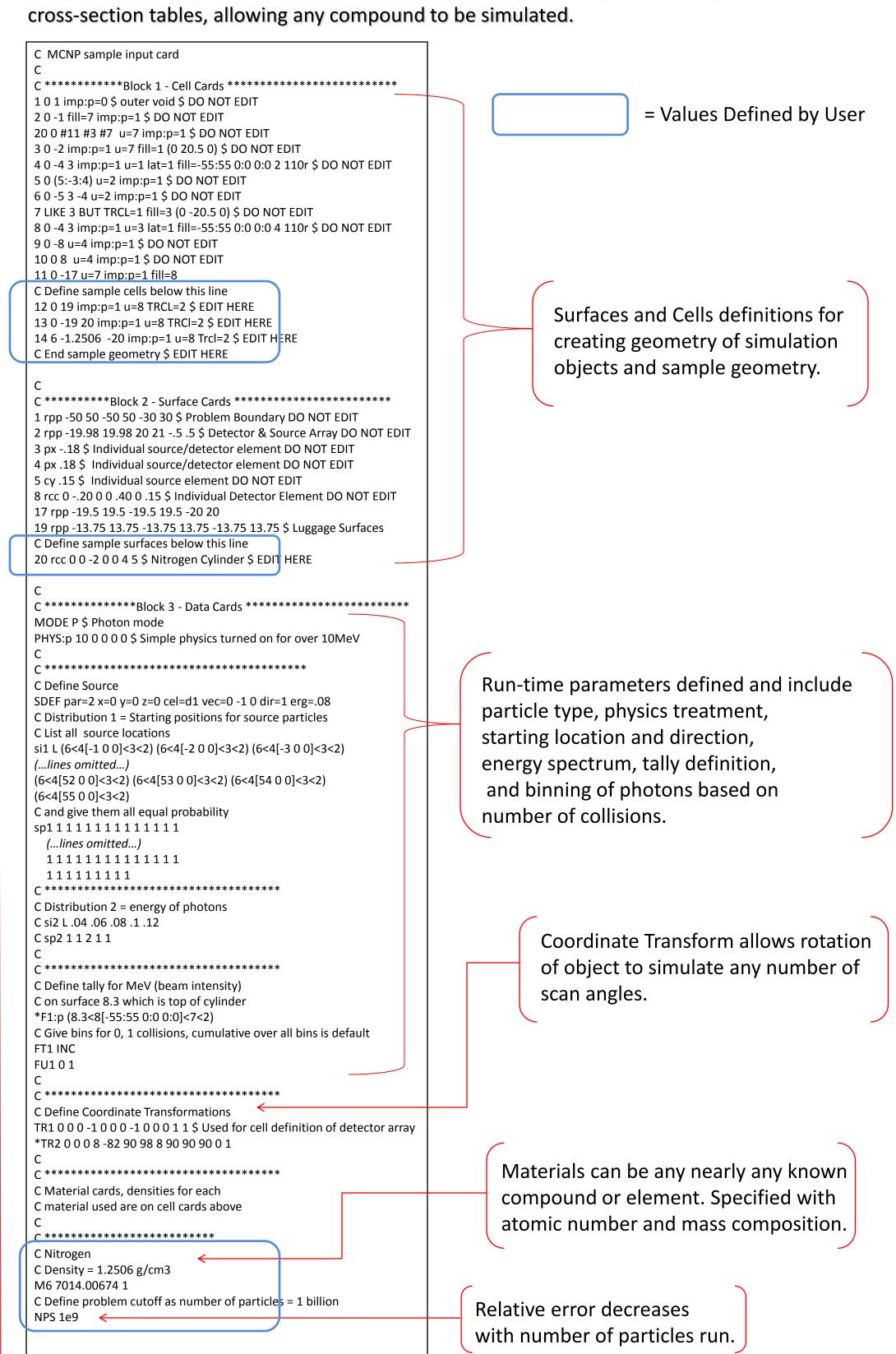
- ❖ Validation of image formation processing requires either the use of advanced simulation tools or access to proprietary specifications and calibration methods for industrial scanners.
- Challenges:
- Scanner information very difficult to obtain for academic research
- Existing simulation codes exist but are very complex requiring months to learn
- Our goal is to provide a user friendly, file based interface to the MCNP code designed specifically for generation of data sets for validation of luggage-screening image formation methods
- MCNP uses updated and experimentally verified cross section data for physics modeling [1].
- Comparison of results with known values show that physics treatment is accurate.
- Advances R2 Thrust area through development of reconstruction methods to optimize the entire detection system.
- Deterministic methods fail to simulate inconsistencies such as (scatter, partial volume effect, beam hardening).

About MCNP:

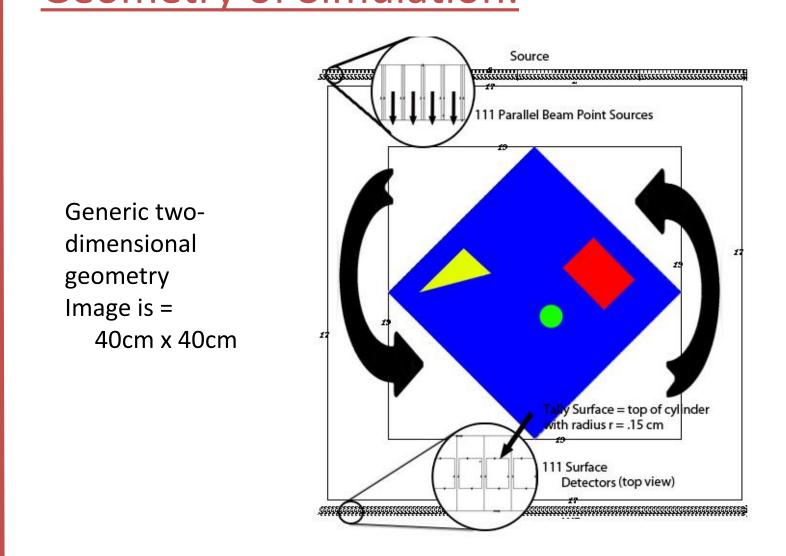
- ❖ Monte-Carlo N-Particle (MCNP) software distributed by Los Alamos National Laboratory.
- Run on command-line along with X-Windowing system for geometry debugging.
- Particles undergo "random-walk" non-deterministic process. Average behavior and individual particle tracked over time.
- **Extensive** documentation for use and modification to user's needs.
- Hardware and software requirements:
 - Distributions for nearly all platforms including Unix, PC, and Mac.
 - ❖Installation requires 2.3gB of hard-disk space.
 - ❖ At least 128 megabytes of RAM recommended.

User Input:

User modifies base input file to their needs. Source and detector geometries are predefined. Sample geometry created using MCNP macro-bodies or surface equations. Material compositions are specified with atomic number, density, and may reference specific nuclear



Geometry of Simulation:



Programs in Package:

Modular programs built to provide functions for various needs. User must currently complete several steps to perform full simulation.

Build input.cpp – Input arguments are file name and number of rotations. The program takes these parameters and creates one MCNP input file for each rotation, each each with a different coordinate transformation. File name is the name of the base input file which was then modified to define sample geometry.

<u>Files Run.sh</u> – Shell script to sequentially submit each input file to MCNP.

<u>Parse out.cpp</u> - Processes MCNP output files (one MCNP output file is created for each angle of revolution). Each file contains thousands of lines of ASCII text. A majority of these lines are results which represent tally info for each detector.

Future goal is to integrate all functions into a MATLAB © GUI for streamlined use.

Output:

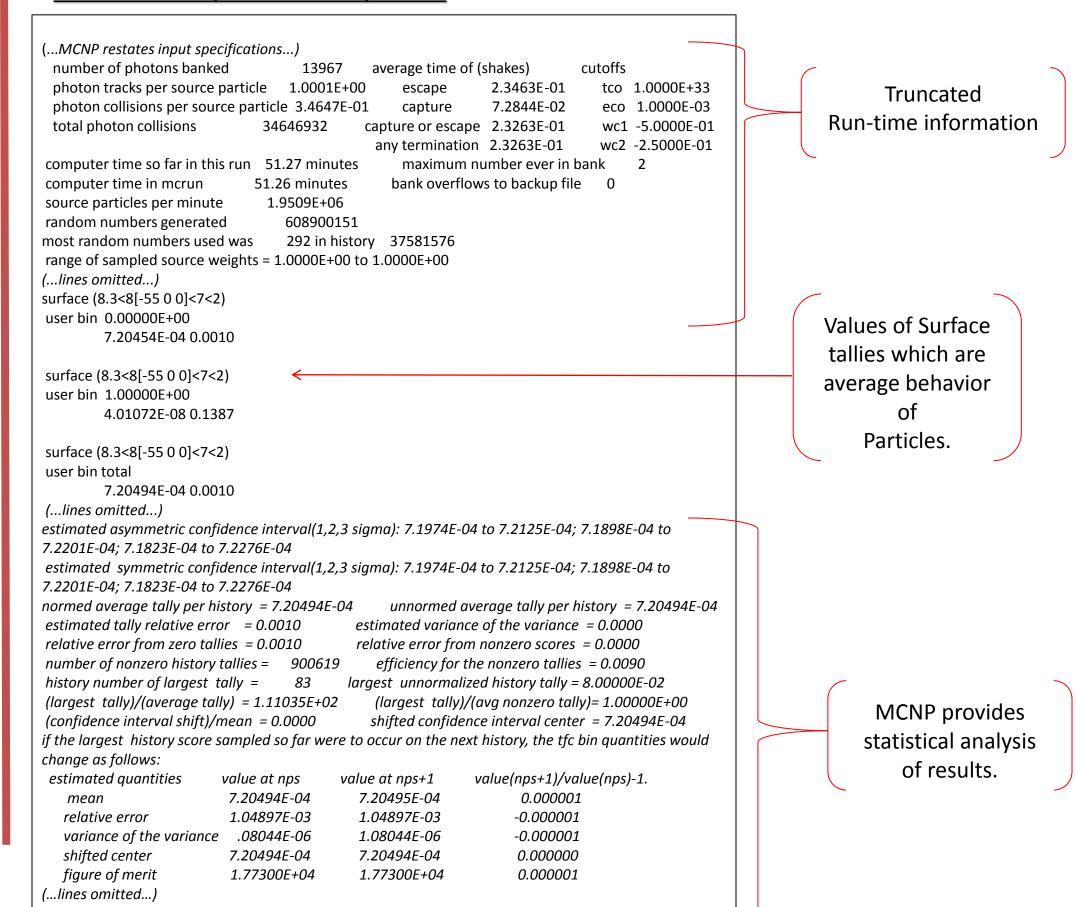
Parse_out.cpp creates several arrays which represent scan data. Each array represents tallies from particles that have undergone a specified number of collisions. A complete sinogram can be produced by including tallies from all photons with any number of collisions. Examining 1st and 2nd second scatter can be useful in analyzing the ability of reconstruction algorithms to treat noise.

Tally units are decided upon in input file:

They make take the form of: particles, MeV, particles/unit time, MeV/unit time.

MCNP output files provide statistical information for evaluating validity of results including relative error of each tally, variance and variance of variance, and an MCNP specific Figure of Merit.

Below is a sample MCNP output file.



Verification of Results:

Case 1: Comparison of Single Beam Attenuation with Deterministic Results

Known Mass attenuation coefficient of Nitrogen = .1639 cm2/g Beam through 10cm of N2 with density (ρ) of 1.2506 g/cm3 MCNP simulation gives intensity values of:

Photons at 80 keV

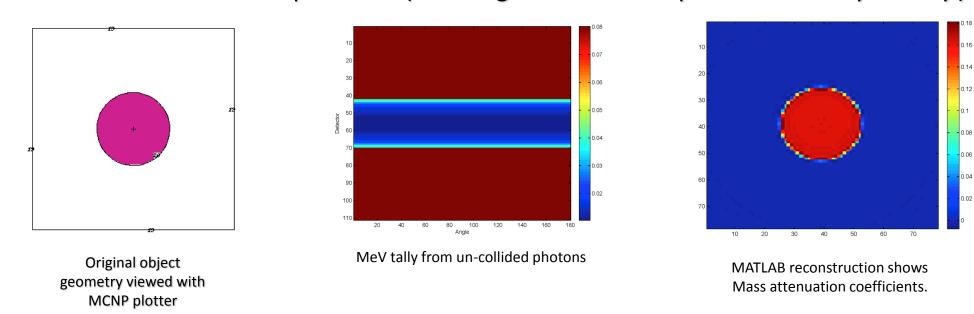
Nitrogen Sample $\ell = 10 \text{ cm}$ Detector

I = 0.0802 MeV for un-collided beam and I = 0.0103 MeV for attenuated beam.

 $I = I_O e^{-(\mu/\rho)\rho\ell} \text{ where } \ell \text{ is the distance and } (\mu/\rho) \text{ is the mass attenuation coefficient.}$ $(\mu/\rho) = \ln \left(I_O/I\right)/\rho\ell = \ln(.0802/.0103)/(10 \cdot 1.2506) = .1642 \text{ g/cm3}$

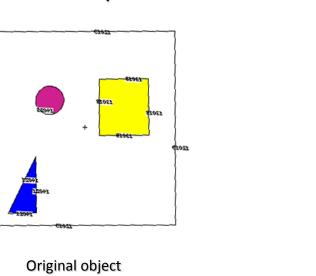
Runtime Environment: Two Sun AMD Athlon-based servers running 64-bit Red hat Linux Enterprise Linux 5. Shared lightweight processing units designed for email and user logins. Uptime = Less than One Minute for Relative Error ≅ .002

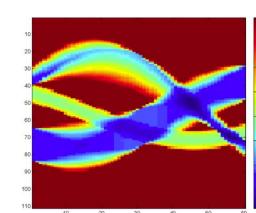
<u>Case 2: Reconstruction of Data Without Scatter Contribution Using MATLAB iradon() Function</u>
<u>Modified to Accommodate Geometry.</u> 180 Degrees of Rotation used for reconstruction but only one run in MCNP of 1 billion particles. (Each angle has same output because of symmetry.)

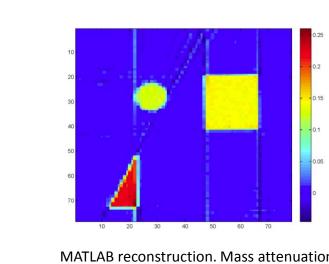


Runtime Environment: 4 Quad-Core 2.3 gHz processors (AMD Opteron 8356) share between users. Running 64-bit Red hat Linux Enterprise Linux 5 Uptime = About 50 minutes (for one angle of rotation) for Relative Error = .0010

Case 3: Reconstruction of Data with Scatter Contribution Using MATLAB iradon() function. 60 different angles of projection, three materials: water, nitrogen, RDX. Run for each rotation stopped at 10⁸ particles.







Coefficients are not right in this example

With the proper system matrix an accurate

reconstruction could be accomplished.

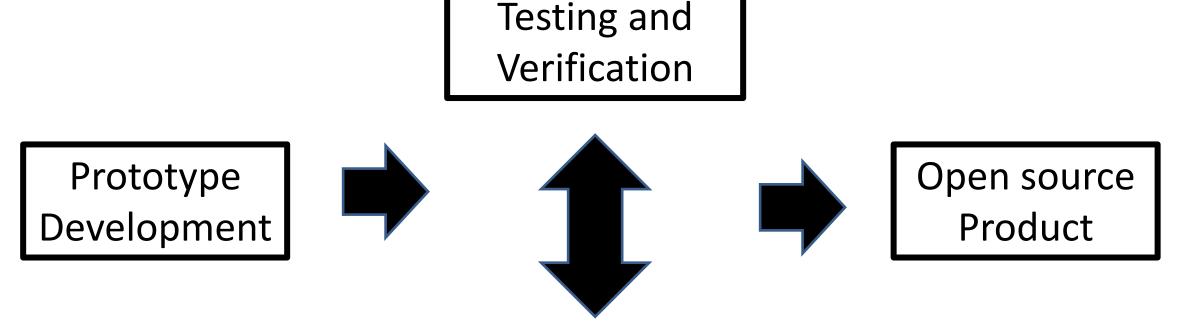
Runtime Environment: 4 Quad-Core 2.3 gHz processors(AMD Opteron 8356) share between users. Running 64-bit Red hat Linux Enterprise Linux 5 Uptime = About 50 hours (50 minutes for each angle) for Relative Error = .0010

Output sinogram.

MeV tally from all photons

Research to Reality:

The goal of this research is to provide an easy to use tool for algorithm verification. It proves the ability of monte-carlo programs, such as MCNP, to simulate photon transport for computed tomography. A further tested and modified tool could result in an open source package available to researchers, within or outside of the educational community. This software would be extremely useful to those researchers interested in testing machine-independent inversion techniques.



Modifications

References:

[1] X-5 Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5, Volume I: Overview and Theory," *Los Alamos National Laboratory*, 2009

Acknowledgements:

This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2008-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.