# **CT Scanner Model in FLUKA – Documentation**

## **Contents**

1. **Requirements**
2. **Preprocessing**
3. **FLUKA Model**
4. **Parallelization**
5. **Postprocessing**
6. **Benchmark Problems**

## Requirements:

* 1. FLUKA – Installed in HPCF
  2. FLAIR - Installed in HPCF
  3. MATLAB
  4. NX – Linux Client for Windows

## Preprocessing:

Geometry

1. For Voxelized geometry, use FLAIR’s DICOM routine to convert DICOM images to VOXELS (.vxl file).
2. For voxelizing the DICOM images, two files are required:
3. HU to Material (.mat), also includes maximum and minimum density values for the HU range.
4. Material Definition (.inp) – Includes weight fraction of the elements for each material.
5. Calibration curve for the CT scanner is generated following Schnider’s work (See paper).
6. Details of the calibration and Matlab script used are in folder:

Preprocessing/HU\_Calibration

1. Matlab script to fuse CT table into Phantom or Patient DICOM for table representation is located in:

Preprocessing/Table\_preprocessing

X-ray Spectrum

1. The computational tool Spektr is used to generate the X-Ray tube spectra.

2. HVL and Exposure measurements are then used to optimize the spectra for the scanner.

3. Carbon and Aluminum filters are assumed and their thickness is varied to match the HVL of the measured spectrum.

4. 120 Kvp with Large Filter and 100 Kvp with Medium Filter measurements and their corresponding equivalent spectra are available.

Bowtie Filter

1. To characterize the bowtie filter, the method suggested in Turner’s paper was used.

2. Based on the measurements along the bowtie filter axis, the aluminum thickness was optimized to match the measured exposure.

3. The X-ray spectrum at the center is attenuated by the filter thickness and the output is converted to CDF for use in FLUKA.

4. Currently 50 Bowtie bins each of 1cm length is generated.

5. Each bowtie bin will have its own X-ray spectrum and the corresponding CDF will be used to sample the initial energy of the photon.

Heel Effect

1. From the beam profile measurements made for QA at each tube collimation, CDFs are constructed and used in the source particle sampling.

2. Including this in the simulation helps to replicate the beam profile of the tube more accurately in the model.

e. Tube Current Modulation

1. A script reads DICOM images and obtain the current values for each slice and constructs the TCM CDF.

2. This file can be provided to FLUKA input for TCM simulation.

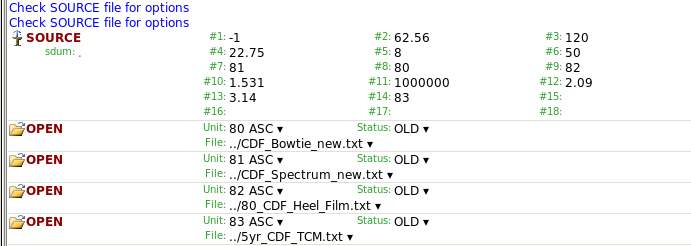
f. Source Strength Calculation

1. The total number of photons emitted for a scan can be calculated from the spectrum generated and the scanner settings used.

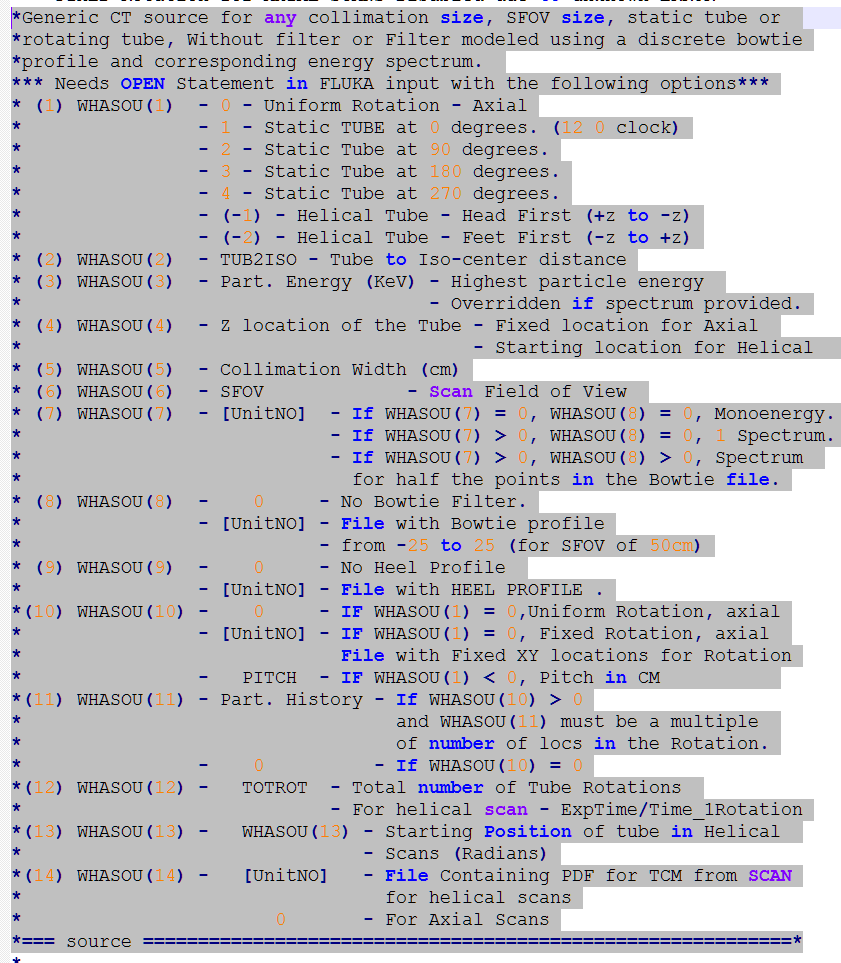
2. The script is located under “ “.

## 3. FLUKA Model

1. The executable for FLUKA with the CT source is located under “RunFluka /myfluka”.
2. The source Fortran file (.f) used to compile FLUKA is under “RunFluka/Fluka\_Scripts/Source” folder.
3. The FLUKA input file (.inp) should have the following source card format for the current FLUKA executable [CTTCM]. The input files are in the “Project” folder for each simulation.



1. The options for the source card are defined as below:



1. To compile a new FLUKA executable if you modify the source file, the script compile\_Fsource.sh under “Fluka\_scripts” folder can be used.

**Syntax:** *compile\_Fsource.sh filename.inp executable*

\*filename.inp is the fortran input file

\*executable is the name of the FLUKA executable that will be created

1. The .vxl files generated for the VOXEL geometry should be less than 7 characters long. Check the .inp files in the project folders to get an idea about the FLUKA input deck.

## Parallelization

1. To launch FLUKA in parallel, the scripts are located under “Fluka\_scripts” folder.
2. The script launch\_Fluka\_Tally.sh should be used to execute a parallel simulation.

**Syntax:** *launch\_Fluka\_Tally.sh filename executable N*

\*filename should be given without the .inp extension (the script will add it)

\*CTTCM is the current executable, any executable located under “myfluka” folder can be used

\* N is the number of cores in which the simulation will be launched.

1. A base input deck should be present in the launch folder.
2. The code parses the base input and assigns a randomly generated random number seed.
3. Depending on the number of cores, the new input files are then launched in their own folders with prefix “run”.
4. If the run is successful, the code moves the output files (usrtrack or usrbin tallies) to “output” folder, merges the output from all the runs and produces the final output in the base folder.
5. As a good rule of thumb, execute the input deck in FLAIR on a single thread, if it runs successfully, then launch in parallel.

## Postprocessing

1. Several MATLAB scripts are available to read the FLUKA output files.
2. If USRBIN tally is used, the script “Read\_USRBIN.m” will read the dose and error values and saves in .mat format.
3. If energy dependent fluence values are tallied using USRTRACK, “Fluka\_Efluence\_reader.m” can be used.
4. To determine the actual dose values for tally locations, the script “process\_plot.m” can be used. This script requires the photon source strength value, which should be calculated at the preprocessing stage.

## Benchmark Problems

1. Inairruns –Simulate exposure measurements at the isocenter.
2. CTDI-Phantom – Simulate CTDI phantom measurements.
3. TG195 – Benchmark problems in TG195 AAPM report for CT.
4. FLUAXIAL, FLUHELICAL, FLUTCM – are simulations for the 5 year old phantom.