MUSIMAN: MUltiple SImulations MANagement

Marcelino Hermida López

Version 1.0 (January 22, 2017)

Purpose

The MUSIMAN package is a software tool written in MATLAB to ease the parallelization of simulations run with the Monte Carlo code for radiation transport PENELOPE 2014 which use `penmain` as steering main program. It can be freely downloaded from a GitHub repository at https://github.com/marcelinohermida/MUSIMAN.

Description

The Monte Carlo code for radiation transport PENELOPE 2014 [1] includes an auxiliary program named `penmain-sum` to combine the results of independent runs of the same simulation problem, provided that `penmain` is used as main steering program of the simulation. In this way, a simulation may be run using multiple computing cores simultaneously, thus increasing the simulation efficiency linearly with the available number of cores. Each independent run produces a dump file with the partial results of the simulation with accumulated statistics. The tool `penmain-sum` combines all dump files to get the final results of the simulation.

A drawback of this method is that the process of manually generating the simulation files needed for multiple independent runs is tedious and error-prone. The MATLAB scripts from the MUSIMAN package automate this process. Briefly, the scripts take care of:

1. Creating the files needed for an arbitrary number of independent runs of the same simulation problem. A different pair of seeds for the random number generator is assigned to each run, ensuring that each simulation run uses independent sequences of pseudo-random numbers.

2. Launching simultaneously a number of parallel independent simulation runs.

3. Preparing the dump files to be combined with penmain-sum to obtain the final results of the simulation.

Components

The distributed package includes the following files:

- createruns.m: MATLAB script to create an arbitrary number of folders with the files needed for independent runs of the same simulation problem.

- simruns.m: MATLAB script to launch simultaneously a number of simulation runs with the files created by createruns.m.

- combineruns.m: MATLAB script to combine the results of multiple independent runs.

- seeds.dat: data file in MATLAB format with a list of seeds for the random-number generator of PENELOPE 2014. The list was taken from the rita.f source file included in PENELOPE 2014, which was obtained with the algorithms from the work of Badal and Sempau [2]. Each consecutive pair of seeds in the list is separated from the following pair by 10^{14} positions.

- MUSIMAN_instructions.pdf: this file.

In addition, MUSIMAN needs a executable file compiled from the penmain.sum.f Fortran file included in the PENELOPE 2014 distribution. This executable needs to be compiled by the user, and is not included in the MUSIMAN distribution package.

The scripts from the present MUSIMAN version were written with MATLAB R2014a under Windows 10, although they should work in other MATLAB and Windows versions. As an alternative to run the scripts directly from the MATLAB command line, the scripts may be compiled. As an example, to compile createruns.m, run the following command from the MATLAB command line:

```
mcc -m createruns.m
```

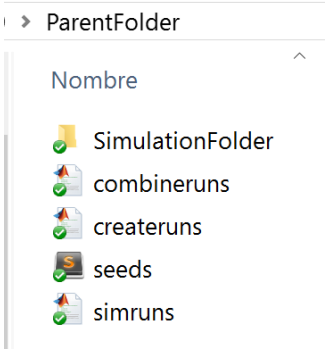
If compiled versions of the scripts are to be used, MATLAB or the MATLAB Compiler Runtime (MCR) must be installed in the simulation computer. The MCR must match the version of MATLAB where the scripts were compiled. MCR for each MATLAB version can be freely downloaded from https://www.mathworks.com/products/compiler/mcr.html.

Note that executables compiled with a 32 bits version of MATLAB will not work in 64 bits Windows systems, and vice versa.

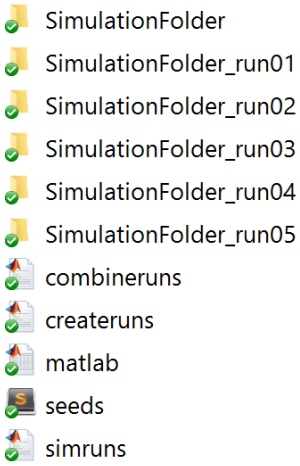
Use

First, we need a folder with all the files needed for a PENELOPE simulation. The main program of the simulation must be the `penmain` included with PENELOPE 2014. It is assumed in the MUSIMAN scripts that only one file with extension `.in` is present in the simulation folder (named SimulationFolder in the examples below). That file must be the input file of the simulation.

The simulation folder must be the only folder located in another folder (ParentFolder). In this parent folder, copy the MUSIMAN files `(.m/.exe` and seeds.mat):



To create the files for multiple independent runs, execute the `createruns.m` script from the MATLAB command line, or the compiled script `.exe` file, from the ParentFolder. The program will ask for the number of runs to be created, the number of histories to be simulated in each run, and the time between dumps. The name of the created folders follow the format: SimulationFolder_runXX:



To launch the simulation runs simultaneously, run the `simruns.m` script from the parent folder. The program will ask the first and last runs to be launched. This feature may be useful to distribute the runs among several computers. For example, we may create 80 run folders, and then launch the runs 1-40 in one computer and the runs 41-80 in another computer.

Once all simulation runs are completed (or stopped by the user), the last step is combining the partial dump files produced by each simulation run. To do so, execute the `combineruns` script from the parent folder. Again, the program asks the first and last runs to be combined. Then, the program copies each dump file to a new folder named RESULTS located at the parent folder. Next, the `penmain-sum.exe` is started to combine all the dump files and obtain the final results of the tallies selected in the simulation.

At this point the run folders can be safely deleted, and the contents of the RESULTS folder can be manually moved to the main simulation folder.

Recommended citation

MUSIMAN was developed to speed-up some of the simulations from the doctoral dissertation of the author. If you find this tool useful for your research, please cite this reference [3]:

M. Hermida-López, “Improvements to the dosimetry of ^{106}Ru/^{106}Rh ophthalmic plaques: Monte Carlo simulations and radiochromic film measurements”. Doctoral dissertation (Universität Duisburg-Essen, 2016). Available at:

https://www.researchgate.net/publication/312578357_Improvements_to_the_dosimetry_of_106Ru106Rh_ophthalmic_plaques_Monte_Carlo_simulations_and_radiochromic_film_measurements

Copyright notice

References

[1] F. Salvat, “PENELOPE-2014: A Code System for Monte Carlo Simulation of Electron and Photon Transport,” 2015.

[2] A. Badal and J. Sempau, “A package of Linux scripts for the parallelization of Monte Carlo simulations,” *Comput. Phys. Commun.*, vol. 175, no. 6, pp. 440–450, Sep. 2006.

[3] M. Hermida-López, “Improvements to the dosimetry of 106Ru/106Rh ophthalmic plaques: Monte Carlo simulations and radiochromic film measurements,” Universität Duisburg-Essen, 2016.

History

2017-01-22 Version 1.0 released.