Ray Tracing Many Spheres

The following information has been adapted from Chapter 2 of the PhD thesis of Evan F Johnson, Middle East Technical University, February, 2021 [1]

Background

For studying radiation in packed particle beds or powders, a 3D MCRT code is needed to simulate the radiative transfer in beds of thousands to hundreds-of-thousands of spheres. While MCRT codes have been well-studied in general [2][3][5], no open source code was found for this specific application. The code developed has several notable attributes:

A) The code works seamlessly with DEM by reading the xyz particle positions from the output files, allowing ray tracing to be done for realistic groups of particles, as opposed to ordered packing structures like Kaviany [4].
B) It is written in C++ using parallel "message passing interface" (MPI) programming, allowing for simultaneous computations on many cores of a workstation or cluster. Speedup is nearly linear with the number of processors.
C) The particles which emit photons (rays) are specified by the user with an input text file, allowing a user to specify a certain particle or group of particles, from which photons will be emitted.
D) Reflections and absorptions are modeled, leading to the Radiation Distribution Factor (RDF), which allows for a simpler and more accurate calculation of heat transfer compared to approaches using the more typical radiative view factor.
E) The current code uses diffuse view reflections, though specular may be added in the future.

The Ubuntu (Linux) operating system is recommended to run the MCRT code, as LIGGGHTS is typically run in this operating system as well. The code is written in C++. OpenMPI must be installed to enable parallel processing, which most users will already have installed as it is a prerequisite for LIGGGHTS as well. For ray tracing in systems of tens to hundreds of thousands of particles, running simulations in parallel is a necessity; for the simulations of 50,000 particles run in Chapter 4 of the thesis, parallel computing reduced run time from months to several days.

For more details on how it works and validation studies, see the thesis [1] and papers [6][7].

File Structure

Information about each folder and file is described below, and the folders and files for running the simulation are shown in Figure 1.

1. pos_[*number of particles*]_[*solid fraction*]_[*LIGGGHTS time step number*].txt: this file contains the xyz position of the center of each particle, one particle per line, in a space delimited format.
2. home_id_[*number of particles*]_[*solid fraction*]_[*LIGGGHTS time step number*].txt: this file contains a list of 0's and 1's, with one value for each particle. A 1 indicates the particle will be an "emitting" particle ("home" particle), and 0 indicates the particle will not.

- The *RDF_files* folder begins empty, and it is filled with the output files at the end of the simulation.
- The *tmp_data* folder begins empty, and it is used to temporarily store the data output from each processor, before the end of the simulation when data is gathered and consolidated into a single output file in the *RDF_files* folder. The contents can be deleted routinely.
- The text file containing the source code, such as *MCRT_1.6.cpp.*
- The compiled executable, *PW_MCRT_1.6*, is present after the source code has been compiled.

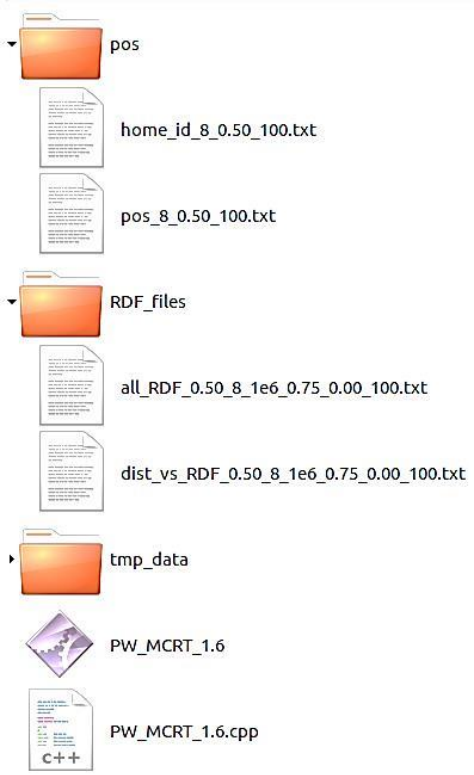

Figure 1. File structure of the MCRT code.

Particle-Particle RDF Example

This example shows how the RDFs are calculated for the 8-particle domain shown in Figure 2 with the geometry defined by the particle center coordinates found in the file *pos_8_0.50_100.txt*. In the file *home_id*, there are eight rows containing a 1, so each of the particles is designated as an emitting particle ("home" particle). The *home_id* file in this case was simple to make manually, but for simulations with thousands of particles, it is recommended to create this file with Matlab or a similar program, as this gives the user full flexibility over which will be designated as home particles. For example, by reading in the particle position file, the particles in a certain region can be designated (e.g. all particles with $-0.1 < x < 0.1$), or the home particles can be designated as a random subset of the particles in a specified region.

From the *pos* and *home_id* file names, it can be seen that there are 8 particles, a solid fraction of 0.50, and a LIGGGHTS time step ID of 100. When naming the files in this simple example, the solid fraction is not actually calculated, so a value of 0.50 is chosen arbitrarily. Likewise, the particle domain was created manually, so the LIGGGHTS (DEM) time step is simply assigned a value of 100. These parameters are only used in the naming of the input and output files, but when working with many different LIGGGHTS time steps and solid fraction files, maintaining these naming conventions is essential to keep the inputs and outputs organized. In contrast, the number of particles in the file name *must* match the actual number of particles to be simulated.

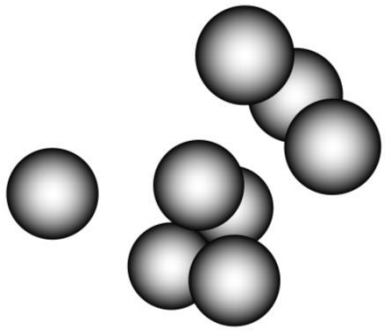

Figure 2. Eight-particle simulation domain.

To compile the code, a terminal window is opened in the main folder, and the command is issued: *mpicxx -std=c++11 MCRT_1.6.cpp -o MCRT_1.6*, which compiles the code in the text file *MCRT_1.6.cpp* into the executable, called *MCRT_1.6*. Next, the simulation is started either in serial with the command *./MCRT_1.6*, or in parallel with *mpirun -np 4 MCRT_1.6*, where 4 is the number of processors specified. As shown in Figure 3, the program asks for simulation parameters, including the number of photons, the volume fraction (solid fraction), the absorptivity (emissivity), the particle radius, and the outputs desired. It then asks if the wall should be modeled, which will be covered in the next section.

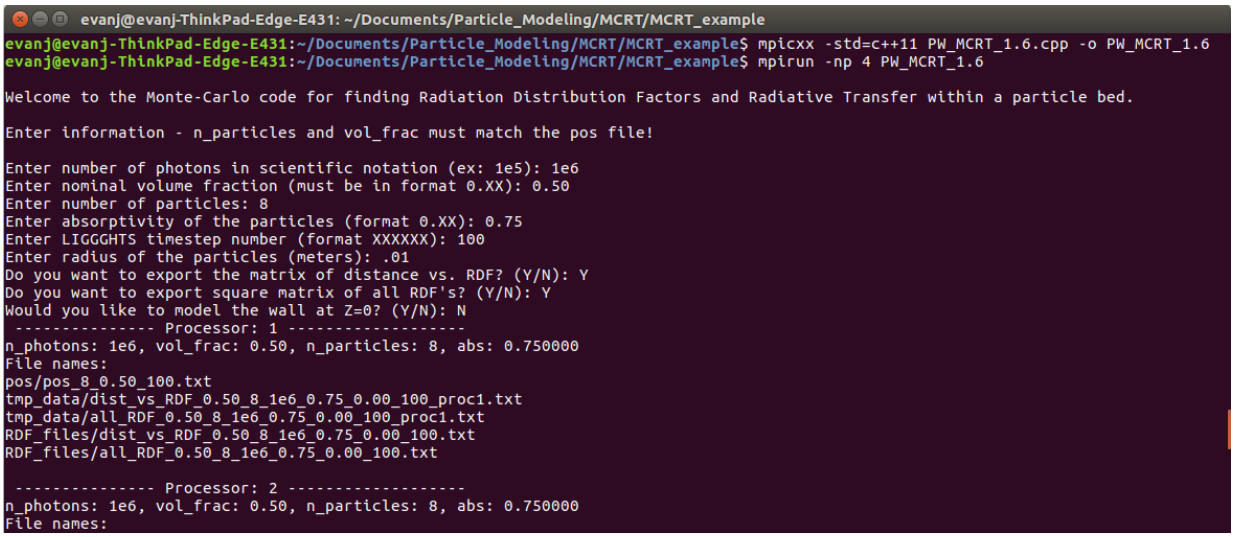

Figure 3. Parameters input to initiate MRCT code.

After the run, two output files appear in folder *RDF_files*. The *all_RDF* output style is gives the RDF from every particle to every other particle in the domain. In contrast, the *dist_vs_RDF* output style is more convenient when building the distance vs. RDF curves as done in [6]. The output files are:

1) *all_RDF_0.50_8_1e6_0.75_0.00_100.txt:* This file contains a square matrix of RDFs from each particle to each other particle. The naming convention is (in order) solid fraction, number of particles, number of photons, particle absorptivity, wall absorptivity, and LIGGGHTS time step number. In this case the wall emissivity is zero since the wall is not modeled. For the eight-particle system the *all_RDF* file is shown in Table 1, where headers are added for clarity. As expected, the matrix is symmetric, with small differences due to the inherent errors in the MCRT which diminish with increasing numbers of photons. There is one extra column for the wall, which is not modeled, so it is simply left as a column of zeros.

Table 1. Data in the *all_RDF* output file, showing the RDF from each particle to each other particle.

	Particle j								Wall
Particle i	1	2	3	4	5	6	7	8	
1	0.011975	0.060233	0.045472	0.021374	0.014346	0.007688	0.00539	0.008687	0
2	0.059895	0.014824	0.044933	0.045351	0.002983	0.000937	0.000166	0.018380	0
3	0.045612	0.045622	0.011462	0.036811	0.015029	0.022761	0.013724	0.020551	0
4	0.020964	0.044701	0.036431	0.006866	0.007684	0.009880	0.002730	0.007548	0
5	0.014388	0.002772	0.015116	0.008005	0.001919	0.165793	0.011853	0.002311	0
6	0.007697	0.000999	0.022712	0.009745	0.165081	0.005542	0.089800	0.004596	0
7	0.005349	0.000151	0.013608	0.002727	0.011647	0.089601	0.004403	0.006626	0
8	0.008798	0.018454	0.020660	0.007444	0.002173	0.004470	0.006838	0.001280	0

2) *dist_vs_RDF_0.50_8_1e6_0.75_0.00_100.txt:* This file contains columns for the emitting particle ID, the absorbing particle ID, the distance between the particle centers, and the RDF between the two (Table 2). The first line shows both emitting and receiving IDs are 0, a PP distance of 0.0000, and an RDF of 0.011975. This indicates that 1.1975% of the photons emitted from the surface of particle 0 were reflected back and absorbed by the same home particle. RDFs from particle 0 to the others are shown in the next seven lines. A receiving ID of -1 indicates the wall, which in this case is not modeled so it is left as zero.

Table 2. The *dist_vs_RDF* output file, with table truncated to 12 rows.

Emitting ID	Absorbing ID	Distance (m)	RDF
0	0	0.0000000	0.011975
0	1	0.0200000	0.060233
0	2	0.0223607	0.045472
0	3	0.0300000	0.021374
0	4	0.0374166	0.014346
0	5	0.0424264	0.007688
0	6	0.0512348	0.005390
0	7	0.0438748	0.008687
0	-1	0.0100000	0.000000
1	0	0.0200000	0.059895
1	1	0.0000000	0.014824
1	2	0.0223607	0.044933
(table continues with more rows)			

Particle-Wall RDF Example

The same particle positions are used, but this time a wall on the $z = 0$ plane is modeled, which extends over $-0.05 < x < 0.05$ and $-0.05 < y < 0.05$, as shown in Figure 4. In the MCRT code, the only option is to have the wall situated on the $z = 0$ plane. Photons are still only released from particles, never from the wall. With the same *pos* and *home_id* files, the code is started again as shown in Figure 5, but this time the wall is chosen to be modeled, and the xy wall bounds and absorptivity are entered.

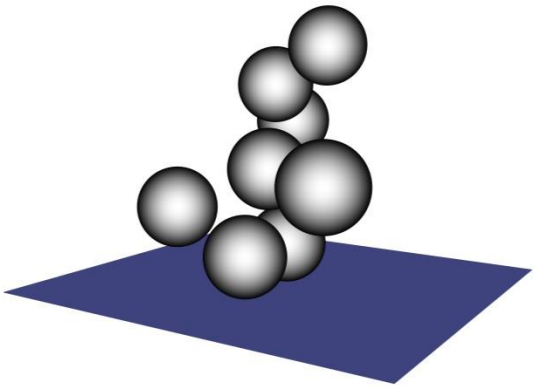

Figure 4. Particle and wall positions used in the example simulation.

```
evanj@evanj-ThinkPad-Edge-E431: ~/Documents/Particle_Modeling/MCRT/MCRT_example
evanj@evanj-ThinkPad-Edge-E431:~/Documents/Particle_Modeling/MCRT/MCRT_example$ mpirun -np 4 PW_MCRT_1.6

Welcome to the Monte-Carlo code for finding Radiation Distribution Factors and Radiative Transfer within a particle bed.

Enter information - n_particles and vol_frac must match the pos file!

Enter number of photons in scientific notation (ex: 1e5): 1e6
Enter nominal volume fraction (must be in format 0.XX): 0.50
Enter number of particles: 8
Enter absorptivity of the particles (format 0.XX): 0.75
Enter LIGGGHTS timestep number (format XXXXXX): 100
Enter radius of the particles (meters): 0.01
Do you want to export the matrix of distance vs. RDF? (Y/N): Y
Do you want to export square matrix of all RDF's? (Y/N): Y
Would you like to model the wall at Z=0? (Y/N): Y
Enter absorptivity of the wall (format 0.XX): 0.85
Enter min x value of wall. (Photons hitting Z=0 wall outside of this will be ignored): -.05
Enter max x value of wall. (Photons hitting Z=0 wall outside of this will be ignored): .05
Enter min y value of wall. (Photons hitting Z=0 wall outside of this will be ignored): -.05
Enter max y value of wall. (Photons hitting Z=0 wall outside of this will be ignored): .05

Check that the values were read in correctly and file names are right:
 --------------- Processor: 1 -------------------
n_photons: 1e6, vol_frac: 0.50, n_particles: 8, abs: 0.750000
```

Figure 5. Parameters for the particle-wall simulation.

In addition to the *dist_vs_RDF* and *all_RDF* output files, another file is automatically generated for a PW simulation, the *PW_RDF* file (Table 3) with three columns: the emitting particle ID, the PW distance, and the PW RDF. This is especially useful for finding correlations for PW RDF as a function of distance. The same data are available in the *dist_vs_RDF* file, but it is in a much more concise form in the *PW_RDF* file.

Table 3. Data output in the *PW_RDF* file.

Emitting ID	PW Distance (m)	PW RDF
0	0.01	0.348029
1	0.01	0.334898
2	0.03	0.120732
3	0.02	0.216053
4	0.04	0.139871
5	0.05	0.058603
6	0.06	0.063357
7	0.03	0.150329

MCRT Conclusion

In this chapter, a MCRT code was developed which is built to be used directly after modeling the particle positions with a DEM code such as LIGGGHTS. The output files give the RDF values either as a square matrix or alongside the distance between the particle-particle or particle-wall pair. It can be used to easily find the RDF (or view factor, if absorptivity is set to 1) between spheres numbering in the hundreds of thousands. This code can be used by others to simulate radiative heat transfer in static beds of small, conductive particles (see the uniform particle temperature assumption), and it can be used to develop or validate particle-scale radiation models such as those used in DEM simulations.

REFERENCES

[1] E.F. Johnson, Advances in Modeling High Temperature Particle Flows in the Field of Concentrating Solar Power, 2021.

[2] J.R. Mahan, The Monte Carlo Ray-Trace Method in Radiation Heat Transfer and Applied Optics, Wiley-ASME Press, 2019. doi:10.1002/9781119518471.

[3] J.R. Howell, The Monte Carlo Method in Radiative Heat Transfer, Mech. Eng. J. Heat Transf. 120 (1998) 547–560.

[4] B.P. Singh, M. Kaviany, Effect of solid conductivity on radiative heat transfer in packed beds, Int. J. Heat Mass Transf. 37 (1994) 2579–2583. doi:10.1016/0017-9310(94)90295-X.

[5] A. V. Gusarov, Radiative transfer, absorption, and reflection by metal powder beds in laser powder-bed processing, J. Quant. Spectrosc. Radiat. Transf. 257 (2020) 107366. doi:10.1016/j.jqsrt.2020.107366.

[6] E.F. Johnson, İ. Tarı, D. Baker, Radiative heat transfer in the discrete element method using distance based approximations, Powder Technol. 380 (2021) 164–182. doi:10.1016/j.powtec.2020.11.050.

[7] E. Johnson, İ. Tarı, D. Baker, A Monte Carlo method to solve for radiative effective thermal conductivity for particle beds of various solid fractions and emissivities, J. Quant. Spectrosc. Radiat. Transf. 250 (2020). doi:https://doi.org/10.1016/j.jqsrt.2020.107014.