# Comparative Analysis of Cerium Nanoparticle Modelling: Montecarlo Algorithm vs. CHARMM-Based Methods

Daniel Sarcanean

## I. Introduction

### A. Context

The synthesis of nanoparticles has emerged as a critical area of research in modern science due to their unique properties and potential for a wide range of applications. Controlling their synthesis for specific applications is to this day still a challenge and due to this many simulation techniques come into play. By using advanced computational models, it is possible to predict and manipulate the formation of nanoparticles; a topic that covers several approaches: *ab initio* iterative methods, such as the one presented here, and other much more physically accurate ones, that will be compared against the former to see its validity.

### B. Objectives

The aim of this work is to present a Montecarlo approach for nanoparticle arrangement from scratch using python and any complementary package, then use an established method to compare the results. An additional simulation step will be tested to see the reliability of the Montecarlo's product by integrating NAMD. Overall, python, CHARMM, VMD and NAMD will be used for a whole description of the process.

## II. Methods

Self-arranged nanoparticles were computed using a Montecarlo Metropolis-Hastings algorithm[1] by introducing Lennard-Jones potential as a discriminator

$$V(r) = 4 \cdot \varepsilon \left( \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} \right) \tag{1}$$

where $\varepsilon$ and $\sigma$ come from an universal approach designed by Elliott and Akerson [1]. Then validating the atom's position based on the Boltzmann distribution factor. In addition, only spherical locations where preferred by a simple radius delimiter included at the code. The conditions for the simulation where discussed at 1000 K, to evade cerium's fusion temperature from a theoretical point but to let Boltzmann's factor to move the atoms as preferred. A simplified 'cristallinity' factor is included into the simulation script for proper FCC arrangement and to observe the validation of different Lennard-Jones values, although its implementation has been omitted for the present work.

The script includes both an appendix to export the visible data to a .pdb and a .xyz file, for whatever purpose it can be used. The first includes a recalculation of the factor B in regard of how distant the atom is to the particle's center, although it is recommended to use directly the formulas provided by VMD's material[2] (if in possesion of the anisotropic information). A normalization factor that acts as a multiplier of 3.56 units was added to consider non-bonding conditions for VMD automatic detection.

### A. Creating the .psf file

Looking for the creation of the .psf file, input topological data was included at the 'psfgen' library from the Interfaces Laboratory[3] directly into VMD. Hence, every single atom from the .pdb was accordingly named to match the final simulation conditions of said force-field input, by labeling each residue as **ICEM**, found at the documentation. Doing this is fundamental, as the simulation can perform an automatic calculation of the bonds and positions following a real model. The force field topological model, named **top_interface.rtf**, is annexed at the github for fast implementation.

### B. Molecular dynamics simulation of the proposed nanoparticle model

Following this, an interaction study was performed for 2 exclusively generated nanoparticles by merging both .pdb and .psf files into a single file and solvating

[1] The whole python script can be read carefully with comments at https://github.com/danielsarcanean

[2] http://www.bmsc.washington.edu/CrystaLinks/man/pdb/part_62.html

[3] https://bionanostructures.com/interface-md/

them into water, all of this thanks to the plugins included in VMD. A NAMD input file was chosen accordingly to perform an equilibrium stabilization and record the particles' behaviour after 12500 steps (25 ps). Other important variables of the NAMD script cover a temperature of 1000 K and no constraints, due to the lack of a proper locked .pdb. A structural image was created for each picosecond and the trajectory was saved in a .dcd type format, then analyzed via VMD.

To further acknowledge the capabilities of the system, a NAMD NpT script was implemented within 500000 steps (1 ns), demonstrating the expansion of the boundary box and the distant interaction between particles.

## III. Results and discussion

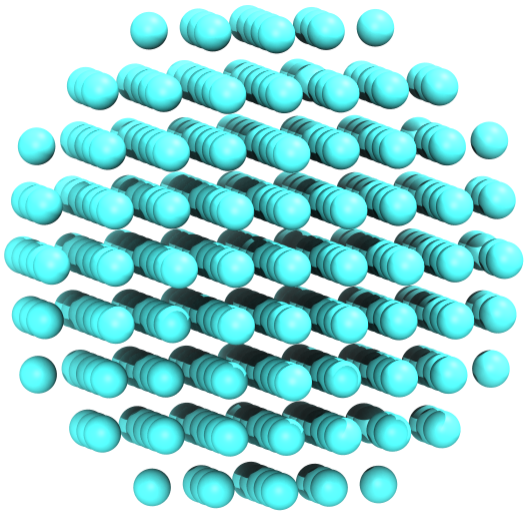

**Fig. 1:** Simulation performed with 2000 *nsteps* and a radius of 16.6 Å, with a time simulation of 10m 9.32s.

By running the script, it is possible to choose the size of the particle by changing directly the **radius**, in angstroms, among other variables of interest. Indeed, as the script is based upon randomness, the selected *nsteps*, which colloquially translates into the atoms pumped into the simulation, can fulfill or not the space for such particle depending on the more values we select. The bigger *nsteps*, less empty will be the nanoparticle but more time will be consumed. As a demonstration, on 1 a simulated nanoparticle can be viewed.

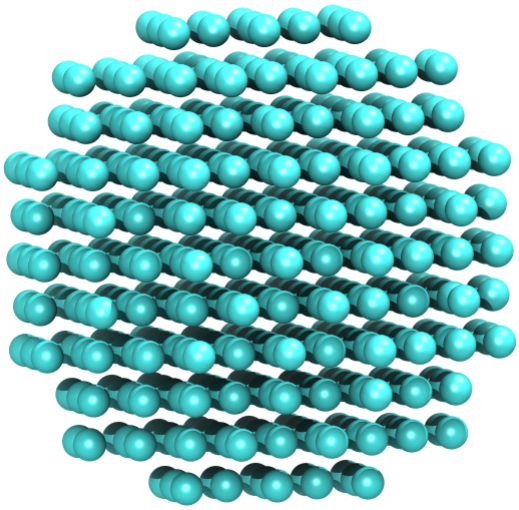

**Fig. 2:** Structure generated by CHARMM-GUI material modeller, where a radius of 15 Å (16.6 Å) was given.

It is clear that a perfect crystal compaction is followed, meaning that the potential discriminator is working accordingly. In fact, if compared to the nanoparticle that appears from the CHARMM-GUI input file at 2, which already makes use of more complex force field documentation, it can be stated that the Montecarlo script has a lot of affinity. The differences among both outputs can be traced as purely geometric mismatches, where at the CHARMM-generated file it is clearly seen that a stacking-based algorithm is followed, in the python script it is a geometrical parameter the one that encloses the sphere. The bonding spaces are corrected afterwards through the force field implementation. These images can be contrasted with realistic cerium particles viewed through a TEM among the literature, observing that either way there is high similitude.

Mounting the initial figure for the last simulation, two spatially separated nanoparticles were arranged and then solvatated into a 30x30x30 water box following their dimensions. The result can be viewed at 4. It is important to remark that both particles have to be completely submerged into the liquid in order to eliminate boundary condition perturbations while running the script. The cell parameters were selected manually via *Tk Console* and inserted into NAMD's input, which means that it has to be changed for each simulation, if the dimensions are switched.

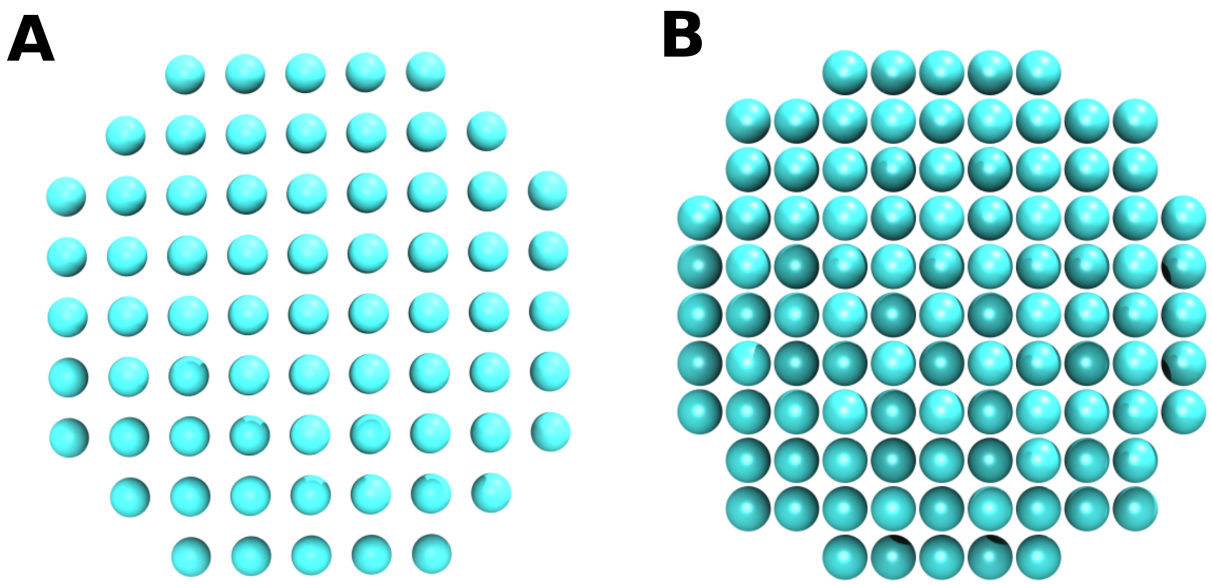


**Fig. 3:** Comparison of the Montecarlo simulated particle **A** and the CHARMM-GUI implementation **B**.

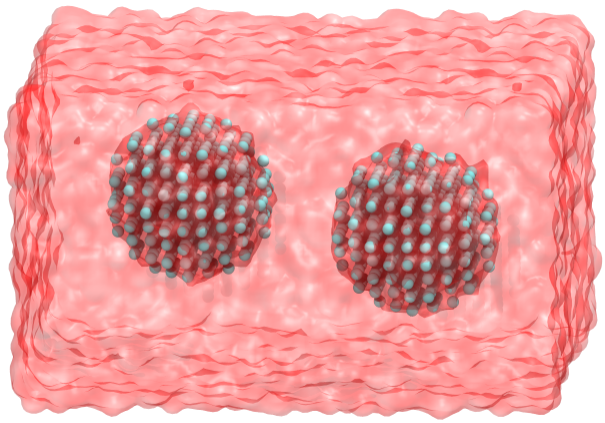

**Fig. 4:** System with both particles immersed in water.

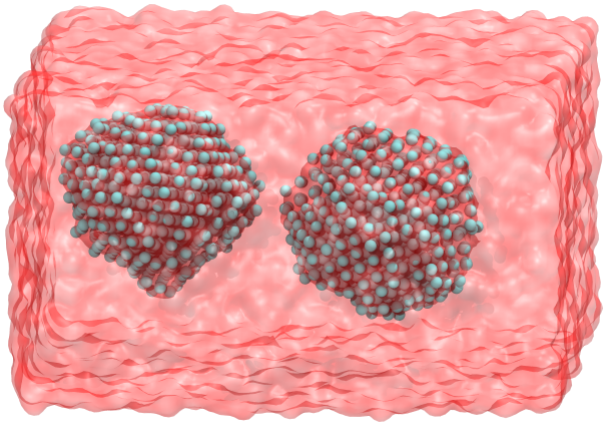

**Fig. 5:** Simulation's output after 12500 steps, or 25 ps. It lasted 1935.17 s of CPU-time with a 16-thread processor.

The product of the equilibrium script can be observed at 5. In it, we see that the atomic layout of the cerium nanoparticles has moved little but coherently, due to the crystal's composition not presenting any substantial charges. Although it is arguable to say that the crystal has been subjected to some lattice movement in pro to minimize interactive energies, hence why is seen an altered structure and not the initial one.

Last but not least, a final NpT simulation was completed, seeing both particles distant in a much bigger box where the pressure should be of 1 bar (as the conditions involved in the input indicate). The expansion of the system means that the first covered simulation in this chapter was extensively over-pressured, therefore it is logical to assume that the particles experimented those lattice displacements. The last frame of the simulated trajectory can be observed at 6.

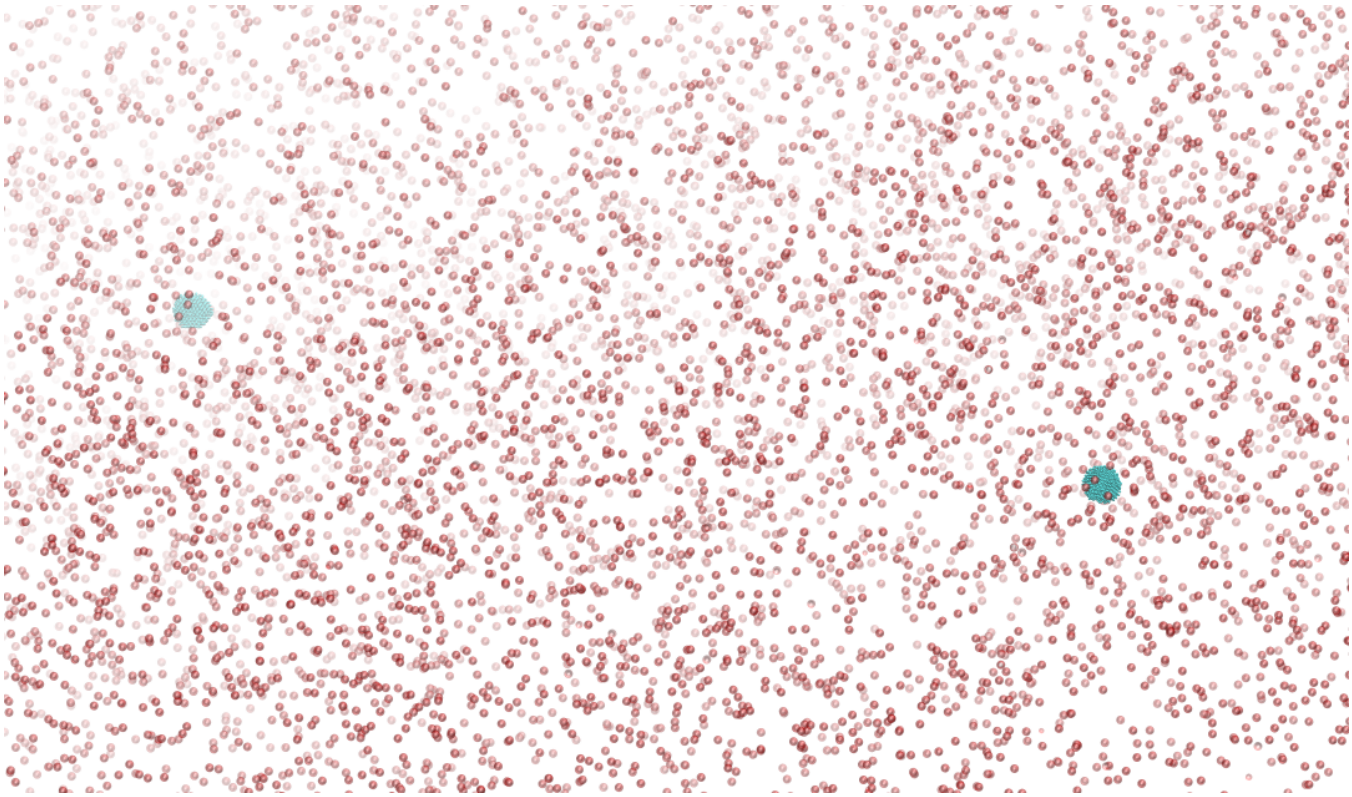

**Fig. 6:** Result of the NpT calculation. CPU-time of 10006.31 s.

## IV. Conclusions

The objectives established in the first part where accomplished, generating a proper Cerium nanoparticle with different space margins than those generated through CHARMM's GUI but with the same propagation axis. This result can be regarded as a stable particle in water, once simulated through a molecular dynamics equilibrium model. In the same manner, the particles experimented latticial displacements as a realistic model under big stress, hence proving that the simulated output matches the purpose that it was designed for.

In conclusion, throughout this last work many individual skills were developed, such as python programming and CHARMM force field adjustments. Creation of formatting data was also a big obstacle to be considered, which was successfully overstepped by controlling every little detail of the generation scripts.

## References

[1] R. S. Elliott, "Efficient 'universal' shifted Lennard-Jones model for all KIM API supported species developed by Elliott and Akerson (2015) v003." OpenKIM, https://doi.org/10.25950/962b4967, 2018.