Manual for LPC3D based Pystencils (Lattice_Porous_Carbon_3D.c)

Contents

1 Introduction

Over the past decade, most of the simulations of supercapacitors were performed at the microscopic scale, using molecular dynamics software such as LAMMPS and ESPResSo. This allowed to understand the adsorption of ions and the effect of surface porosity on some electrochemical properties. However it is well known from experiments that commercial materials are highly inhomogeneous. For example, the typical pore size distributions range from 0.8 to 2 nm. Molecular simulations, where electrode sizes are a few nanometers, only allow for the inclusion of a few pores. It is therefore necessary to simulate electrodes and supercapacitors at larger scales. To this end, LPC3D is a software designed for mesoscopic simulations of capacitive properties of carbon-carbon capacitors, based on a lattice gas model. The code calculates quantities of adsorbed ions, diffusion coefficients and predicts NMR spectra of ions adsorbed in porous carbon matrices. It uses information from molecular dynamics simulations as its main input, so that the parameterization is made at the microscale.

The initial LPC3D code, which was written in C, was a serial code. It allowed routine simulations of systems on a 30×30×30 grid, that corresponds to a dimension of approximately $(1\,\mu\mathrm{m})^3$, i.e. the size of a single - relatively small - particle, much smaller than an electrode and than a supercapacitor device.

The migration from the serial code to Python code based PyStencils has been performed. PyStencils was chosen since it allows to generate highly efficient and scalable C++ and CUDA/HIP code from the mathematical description of the model. This now allows one to run the model in parallel using OpenMP on CPU or hybride CPU-GPU, making use of a full node. This new implementation allows to simulate systems on a 100×100×100 grid (1 million sites), with a very short simulation time of around 0.5 seconds per timestep. It is therefore now possible to simulate routinely systems with several small particles immersed in a bulk electrolyte as well as a full supercapacitor.

Key Features

- Symbolic expressions for defining simulation equations
- Support for high-performance stencil computations
- CPU parallelization using OpenMP
- Hybride CPU-GPU

2 Installation

In this model, carbon electrode, fluid and supercapacitor are simulated. The lattice structure is represented as inter-connected discrete sites, separated by a lattice spacing.

Before running the code, several modules need to be installed; if you want to run the software on CPU, you have to install pystencils, numba, matplotlib, Ipython and vtk, using pip with the command:

pip install pystencils numba Ipython matplotlib pyevtk

If you want to run the software on GPU, you need to install additional modules, such as cupy (make sure to match the module version with your CUDA version) using pip with the command:

pip install cupy-cudaXXX

And also load CUDA module with the following command line:

module load cuda/XXX

3 Running the program

On CPU or hybride CPU-GPU, the program can be run using the following command line:

OMP_NUM_THREADS=NUMBER_OF_THREADS python3 3D-lattice-gas.py

4 Formats of input file

- 1st line: type of processing unit to be used for the computation (> *cpu* or *gpu*)
- 2st line: numbers of lattice sites in x, y and z directions (> Nx, Ny, Nz)
- 3st line: boudary conditions in x, y and z directions, true or false (> *true true true*)
- 4nd line: lattice parameter in Å (> a)
- 5nd line: dwell time in seconds (> *dwell*)
- 6th line: number of steps between measures (> *nsample*)
- 7th line: temperature in Kelvin (> T)
- 8th line: Larmor frequency of the studied nucleus under the magnetic field considered in MHz (> $larmorfreq$)
- 9th line: number of values for the Fourier transform (> $Nvalues$)
  Please use an even number of values.
- 10th line: energy barrier value (> *Enbar*)
- 11th line: equilibration process. 0 if no, 1 if yes (> *equi*)
- 12th line:
  - no 12th line if equi=0
  - if equi=1: number of steps for the equilibration process (> $Nstep$)
- 13th line: number of blocks (> $Nblocks$)
  - We divide the simulation lattice into blocks along the z-axis.
  - PS: All blocks should have the same size.

After this line, the uploaded files differ depending on the block nature. Block nature: bulk or electrode.

4.1 "Bulk" type simulations

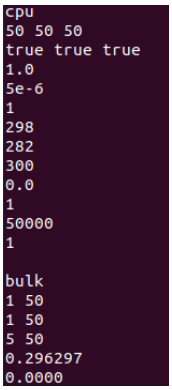
if $Nblocks$ =1:

- 14th line: An empty line
- 15th line: block nature (> *bulk*)
- 16th line: xmin,xmax,ymin,ymax,zmin,zmax coordinates of the block
- 17th line: density of the bulk (> $dens_bulk$)
- 18th line: frequency of the bulk (in ppm) (> $Nfreq$)

4.2 "electrode" type simulations

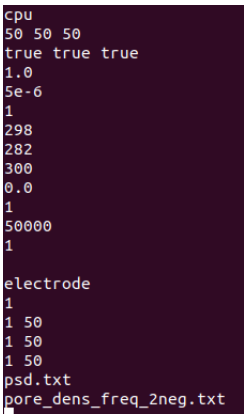
if $Nblocks$ =1:

- 14th line: An empty line
- 15th line: block nature (> *electrode*)
- 16th line: electrode is filled with fluid or not? 0 if no, 1 if yes (> $fielectrode$)
- 17th line: xmin,xmax,ymin,ymax,zmin,zmax coordinates of the block



Input example for "bulk" type simulations.

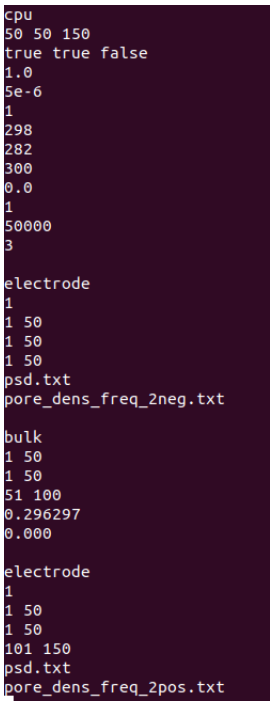
- 18th line: name of the file with pore size distribution ($> name_psd$) The first column should be in Å, the second column is the probability.
- 19th line: name of the file with pores size as a function of density and frequency ($> name_poresize_dens_freq$) The first column should be in Å, the second column is the densities of the fluid and the third column is the frequencies and should be in ppm.



Input example for "electrode" type simulations.

4.3 "supercapacitor" type simulations

simulation of supercapacitor is actually a combination of bulk and electrode constituting 3 blocs, (electrode/bulk/electrode). Here is an input example that illustrates the simulation of supercapacitor. PS: don't forget to make an empty line between blocks.



Input example for “supercapacitor” type simulations.