Department of Physics
Freie Universität Berlin

Problem set 3: Computational Molecular Physics

Published: 7 June 2016 *Please send the solution by email to irtaza06@zedat.fuberlin.de*
by 14 June 2016, 12pm

1. Auto-corelation function (ACF) (25 points)
The txt files provided for problem set 4, NN distances, CC distances, psiA dihedrals, phiL dihedrals contain trajectories (time series) of different internal coordinates from a simulation of Alanine-Leucine. The first column is the time step, the second is atom distances or dihedral angles respectively.

Compute and plot the respective auto-correlation functions for NN distances, psiA dihedrals, CC distances and phiL dihedrals using any of your favorite programming language. Discuss your results.

You can use `correlate` from numpy in python or `autocorr` in matlab.

2.Principal compenent analysis (PCA) (25 points)
The attached file traj.xyz.ar.gz contains 1000 frames of a time series of a collection of seven atoms (Alanin-Leucine backbone atoms). Each frame is written in the format

number_of_atoms
title_line
atom_name x-coordinate y-coordinate z-coordinate

Perform a principal component analysis (pca) on the Alanine-Leucine trajectory py computing the following steps (using our favourite programming language):

a) Calculate the mean value μ for each of the 3*7 coordinates

b) Calculate the mean position for each of the seven atoms, expressed as x,y,z coordinates

c) Set up the covariance matrix $\mathbf{C}$ of all the 21 coordinates (r=x,z, or y) with $C_{ij} =< (r_i - \mu_i)(r_j - \mu_j) >$

d) Calculate eigenvalues and eigenvectors of the covariance matrix

e) Choose two eigenvectors to reduce the dimensionality of the system to. Project the trajectory, i.e. each frame of the time series, onto the chosen pincipal components with

$$\mathbf{PA} = \mathbf{B} \tag{1}$$

where the rows of $\mathbf{P}$ are the pincipal components, PC, $\mathbf{A}$ is a matix (21x1000) where each column corresponds to the full coordinates of a frame, and $\mathbf{B}$ is the matrix containing the projected trajectory.

f) Plot the data poins of the projected two-dimensional trajectory as PC1 vs PC2

g) Plot the time series of the data poins projected only onto PC1 and PC2, respectively (PC1 vs. time, and PC2 vs. time)

You can use e.g. the `PCA()` class from the matplotlib.mlab library (in python) or `princomp` from the statistics toolbox in matlab.