Worksheet 7: Error Analysis

June 12, 2017

General Remarks

- The deadline for handing in the worksheets is **Monday, June 19th, 2017, 12:00 noon**.
- For this worksheet, you can achieve a maximum of 10 points.
- To hand in your solutions, send an email to your tutor:
  - Johannes Zeman `zeman@icp.uni-stuttgart.de` (Tue 15:45–17:15)
  - Michael Kuron `mkuron@icp.uni-stuttgart.de` (Wed 15:45–17:15)
  - Kai Szuttor `kai@icp.uni-stuttgart.de` (Thu 14:00–15:30)
- Please try to only hand in a single file that contains your program code for all tasks. If you are asked to answer questions, you should do so in a comment in your code file. If you are asked for graphs or figures, it is sufficient if your code generates them. You may as well hand in a separate PDF document with all your answers, graphs and equations.
- The worksheets are to be solved in groups of two or three people.

The data required for all tasks in this worksheet is provided in the file `ws8.pkl.gz`, which can be downloaded on the lecture website.

Task 7.1: Error Analysis and the Autocorrelation Function (4 points)

The file `ws7.pkl.gz` contains five artificially generated time series with well-defined autocorrelation times. The example program `ws7.py` contains code to read the file contents and plot a part of the data series.

- **7.1.1** (2 points) Compute and plot the interesting parts of the autocorrelation functions of the different time series.
- **7.1.2** (1 point) Estimate the autocorrelation time τ_C of the different series by fitting an exponential function to the autocorrelation data.
  To compute the fit parameters, use `scipy.optimize.curve_fit()` or a similar least-squares fitting procedure.
- **7.1.3** (1 point) Compute the mean values of the different series and their statistical errors from the estimated correlation times.

Hints

- You may want to use the functions `numpy.mean()`, `numpy.std()` or `numpy.var()`.
- If the mean value of a data series is non-zero, its autocorrelation function will not converge to zero. Thus, you should subtract the mean value prior to computing the autocorrelation.
- The autocorrelation function decays very quickly, so you will have to zoom the function at small τ.
- When fitting the exponential, you should only fit at small τ.

Task 7.2: Error Analysis of Real Simulation Data (2 points)

The data file also contains data obtained from a Molecular Dynamics simulation of a charged colloidal particle (charge +300 e, radius 50 nm) in a solution of monovalent and multivalent ions (charges -1 e, +1 e and -3 e).

The sample program shows how to load the data from the file, where `ts` contains the time axis and `n1`, `n2`, and `n3` are the number of ions of the different types close to the surface of the colloid.

What one wants to know are the mean values of these observables in thermodynamic equilibrium. Note that the simulation is not started in thermodynamic equilibrium – initially, the different ions were randomly distributed around the colloid. Therefore, the simulation first has to equilibrate, i.e. it has to run for a while until equilibrium is reached. In the observables, this manifests itself in clear trends until the measurements only fluctuate around their equlibrium mean values. In all of the following tasks, only use the data where the system is in equilibrium. This should be the case for $t > 20000$ for all observables.

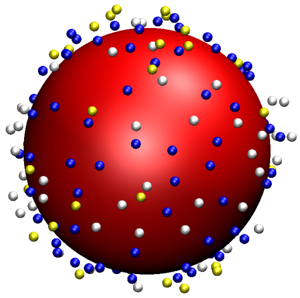

Figure 1: Charged colloid with ions near its surface. [1]

- **7.2.1** (1 point) Plot the interesting parts of the autocorrelation function of the equilibrated data.
- **7.2.2** (1 point) As in the previous task, estimate the mean values, the statistical errors and the correlation times of the equilibrated data.

Task 7.3: Binning Analysis (4 points)

- **7.3.1** (2 points) Implement a Python function that computes the binning error estimate of a given data set for a given bin size.
- **7.3.2** (1 point) Plot the binning error estimate versus the bin size for the different data sets (both the artificial and the simulation data) and read off the binning error estimates and the correlation times.
- **7.3.3** (1 point) Compare the binning autocorrelation times with the estimates from the fitted autocorrelation function in the previous tasks. Do they match? Which of them do not match? Do you have an idea why?

References

[1] O. Lenz and C. Holm. Simulation of charge reversal in salty environments: Giant overcharging? *Eur. Phys. J. E*, 26:191–195, 2008.