Physik auf dem Computer **SS 2013**

Worksheet 6: Error Analysis

May 28, 2013

General Remarks

- Deadline is **Tuesday, 4th June 2013, 10:00**
- On this worksheet, you can achieve a maximum of 10 points.
- To hand in your solutions, send an email to
  - Olaf (`olenz@icp.uni-stuttgart.de`; Wednesday, 14:00–15:30)
  - Elena (`minina@icp.uni-stuttgart.de`; Wednesday, 15:45–17:15)
  - Tobias (`richter@icp.uni-stuttgart.de`; Friday, 15:45–17:15)
- Attach all required files to the mailing. If asked to write a program, attach the *source code* of the program. If asked for a text, send it as PDF or in the text format. We will *not* accept MS Word files!
- The worksheets are to be solved in groups of two or three people.
- The tutorials take place in the CIP-Pool of the ICP in Allmandring 3.

All files needed can be found in the file `data.tar.gz` on the home page.

Task 6.1 (4 points): Error Analysis and the Autocorrelation Function

The file `artificial.npy` contains five (artificially generated) time series with a well-defined autocorrelation time. To read the data from the file, use the following command:

```
d0, d1, d2, d3, d4 = numpy.load('articifial.npy')
```

- 6.1.1 (2 points) Compute and plot the interesting parts of the autocorrelation functions of the different time series.
- 6.1.2 (1 points) Estimate the autocorrelation time τ_C of the different series from the plots, by fitting the corresponding exponential function to the autocorrelation function. You can do this either visually (*i.e.* by trying out different values of τ_C and see how it fits) or by any other method (*e.g.* `scipy.optimize.curve_fit()`).
- 6.1.3 (1 points) Compute the mean values and the statistical errors of the different series from the estimated autocorrelation times.

Hints

- You may want to use the functions `numpy.mean()`, `numpy.std()` or `numpy.var()`.
- The autocorrelation function will not converge towards 0 when the mean value of the functions does not converge to 0.
- 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 6.2 (2 points): Error Analysis of Real Simulation Data

As in the previous worksheet, the file `simulation.npy` contains data of an actual Molecular Dynamics simulation of a charged colloidal particle (Charge +300, Radius 50 nm) in a solution of monovalent and multivalent ions (Charges -1, $+1$ and -3).

To read the data from the file, use the following command:

```
t, n1, n2, n3 = numpy.load('simulation.npy')
```

In the previous worksheet, you have determined the equilibration time of the time series. In all of the following tasks, use only the data *after* the equilibration time (*i.e.* the *equilibrated* data).

- 6.2.1 (1 points) Plot the interesting parts of the autocorrelation function of the equilibrated data.
- 6.2.2 (1 points) As in the previous task, estimate the mean values, the statistical errors and the autocorrelation time of the equilibrated data.

Task 6.3 (4 points): Binning Analysis

- 6.3.1 (2 points) Implement a Python function that computes the binning error estimate for a given bin size of a given data set.
- 6.3.2 (1 points) 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 autocorrelation times.
- 6.3.3 (1 points) Compare the binning autocorrelation times with the estimated autocorrelation times from tasks 6.1 and 6.2. Do they match? Which of them do not match? Do you have an idea why?