FB 08 Informatik
Lehrstuhl f. Software-Technik und Bioinformatik

Prof. Dr. Andreas Hildebrandt
Dr. Marco Carnini

BIOINFORMATICS II - SS 16

6. EXERCISE SHEET

TO BE DELIVERED NOT LATER THAN 05-06-2016

	Exercise	Points
Theoretical	1	10
Theoretical/Practical	2	10+10

Exercise 1: Numerical integration (10 Points)
Consider a system composed by an harmonic oscillator in three dimensions, with mass and elastic constant put equal to one. Assume that that the initial condition is:

$$\begin{cases} \mathbf{x}(0) = 1.0 \\ \mathbf{v}(0) = 1.0 \end{cases}$$

Using Euler and velocity-Verlet algorithms, solve the differential equations. Explore possible values for the time step, and find the maximum value for which the simulation is not affected. Write for each time step the time, the kinetic energy

$$E_{kin} = \frac{1}{2}\sum_i m v_i^2,$$

and the impulse

$$\mathbf{P}_{ges} = \sum_i m_i \mathbf{v}_i$$

into a textfile.
Afterwards, plot each measurement as a function of time and compare the results obtained with the two algorithms.
Consider this notation for the following algorithms:

$$\mathbf{v}_n = \mathbf{v}(t_0 + n \cdot h)$$
$$h \equiv \Delta t \quad (= \text{one time step})$$

The steps for the *Euler algorithm* are

(a) Force: $\mathbf{F}_n = \mathbf{F}(\mathbf{x}_{n-1})$,

(b) Velocity: $\mathbf{v}_n = \mathbf{v}_{n-1} + \frac{h}{m}\mathbf{F}_n$,

(c) Location: $\mathbf{x}_n = \mathbf{x}_{n-1} + h\mathbf{v}_{n-1}$,

and of the *Velocity-Verlet algorithm*:

(a) Location: $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n h + \frac{\mathbf{F}}{2m} h^2$,

(b) Force: $\mathbf{F}_{n+1}(\mathbf{x}_{n+1})$,

(c) Velocity: $\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{h}{2m}(\mathbf{F}_{n+1} + \mathbf{F}_n)$.

Exercise 2: Local Minimizer – C++ (10+10 Points)
Given the function

$$f(x) = \mathrm{e}^{-x^2-y^2} \tag{1}$$

where $\mathbf{x} \in [-4,4] \times [-4,4]$, find the minima and the maxima for the function analytically.
Then implement an algorithm that finds the local minima for the same function. The basic outline of the algorithm is as follows:

(a) Choose a starting point $\mathbf{x}_0$ and evaluate the function at this point.

(b) Determine a search direction and update the current position (x_i) to the new position (x_{i+1}) where i is the number of iterations in the algorithm

(c) Iterate until $\|\mathbf{x}_{i+1} - \mathbf{x}_i\| \leq \varepsilon$ being ε a fixed threshold.

Use the following formulas to determine the search direction:

(a) Using the gradient

$$\mathbf{x}_{i+1} = \mathbf{x}_i - K\nabla f(x_i) \tag{2}$$

where K is the constant which defines the size of the step in the direction of the gradient. Assume K = 0.01. Test the algorithm with different K values and interpret your results.

(b) Using the *Hessian matrix*:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (H(f(\mathbf{x}_i)))^{-1}\nabla f(x_i) \tag{3}$$

The algorithm outputs :

- $f^* = \min_{x \in [0,2\pi]} f(\mathbf{x})$
- $\mathbf{x}^* = \operatorname{argmin}_{x \in [0,2\pi]} f(\mathbf{x})$

Comment how the initial solution $f(\mathbf{x}_0)$ influences your result in both the cases (*Gradient method* and *Hessian matrix*).