# Programming in Engineering

# 2015-191158510-3A

# PiE Matlab test

# MSM - CTW

# 6 July 2015

## General instructions

**Report and codes:** Present your results in a report that includes short explanations of your approach, your answers to the questions listed in this exam, and your interpretations of the plots and other output generated by your codes. Please do not include full-length source code in the report; in stead, use the relevant segment(s) of the code to illustrate your point. The report should be a single pdf file, containing all plots, tables and their captions. You are requested to write your report, and the comments in your code, in English. We also want to receive your final m-files, to test the performance of your codes. Do not forget to include your name and student number in your report and all m-files. Please combine the report and m-files into a single compressed archive file. Send this compressed file by e-mail to Dalila Vescovi `d.vescovi@utwente.nl`, before the deadline of *Monday 3$^{rd}$ August 2015, 9:00*.

**Grading:** After receiving your archive file, we will invite you for a brief discussion of your report and code. The goal of the exam is to question you on your codes, to establish that you understand how they work. Please bring along your laptop with the m-files, to show the running code. Your grade will be determined by the quality of your code, the quality of your report, and the short oral exam. To pass the matlab section of Programming in Engineering, you should have solved all problems in this exam and be able to explain your work during the oral exam. For a grade eight, the code should be efficient, in good style and well commented. Top grades are awarded for solutions to the harder problems.

Please do not hesitate to contact the MSM staff (`www.utwente.nl/ctw/msm/people/`) for more information and/or assistance.

## The assignment

### Background

For this assessment you simulate the interaction of some simple atoms on a computer in the context of N-body simulation; using the so-called Molecular Dynamics (MD) simulation method. In MD, the atoms interact for a period of time, giving a view of the motion of the atoms. The trajectories of atoms and molecules are determined by numerically solving the Newton's equations of motion for a system of interacting particles, where forces between the particles and potential energy are defined by interatomic potentials or molecular mechanics force fields. The method was originally conceived within theoretical physics in the late 1950s but is applied today mostly in chemical physics, materials science and the modelling of biomolecules.

Consider the particles (atoms) defined in the txt files *Input1D-1.txt* and *Input1D-2.txt*. The input data files store information about particles: The format is initial $x$ location, initial $u$ velocity, mass. For this assessment you will use the Lennard-Jones potential to model the interactions between the particles. The Lennard-Jones potential is a mathematically simple model that approximates the interaction between a pair of neutral atoms or molecules.

For the Lennard-Jones potential the interaction between the particles $i$ and $j$ is modelled using

$$V(r_{ij}) = 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6} \right]$$

where $\sigma$ is the particle diameter. Set the materials parameters $\epsilon = 1$ and $\sigma = 1$. Moreover $r_{ij}$ denotes the distance between the particles $i$ and $j$.

The interaction force acting between the particles $i$ and $j$, $f_{ij}$, is obtained as the derivative of the potential $V(r_{ij})$ with respect to $r_{ij}$ i.e.

$$f_{ij} = -\frac{\mathrm{d}V(r_{ij})}{\mathrm{d}r_{ij}} \hat{n}_{ij}$$

where $\hat{n}_{ij}$ is the unit vector between the particles. Thus the equation that defines the motion for the particle $i$ is given by

$$m_i \ddot{x}_i = \sum_{j=1 \;\; j \neq i}^{N} f_{ij}, \tag{1}$$

where $N$ is the total number of particles.

## Questions

1. Plot the potential $V_{ij}$ as a function of the distance $r_{ij}$. Plot separately both the attractive and repulsive contribution.

2. Use symbolic differentiation from the matlab symbolic toolbox to derive the magnitude of the force as a function of $r_{ij}$.

3. Plot the force $f_{ij}$ as a function of the distance $r_{ij}$.

4. Write two matlab functions that solve the differential equations, for a 2D system of particles, using Euler Forward algorithm and the Verlet algorithm. *Euler Forward*

$$x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t$$

$$\dot{x}(t + \Delta t) = \dot{x}(t) + \ddot{x}(t)\Delta t$$

*Verlet algorithm*

$$x(t + \Delta t) = -x(t - \Delta t) + 2x(t) + \ddot{x}(t)\Delta t^2$$

$$x(-\Delta t) = x(0) - \dot{x}(0)\Delta t$$

The functions should accept any of the input files and produce the following output:

   (a) Determine the position of the particles during the time interval [0,20] using $\Delta t = 0.005$. Print to a file the position and velocity of all particles at each time step.

(b) Compute both the kinetic and potential energy of the system at each time step and print the data to a file. Also compute the total (kinetic + potential) energy. Note, potential energy is per interaction, not per atom.

(c) Comment on which algorithm you prefer and state why?

5. Write a MATLAB file that

(a) Reads in the data you printed in the files and plot the energies as a function of time.

(b) Read in the data you printed in the files and make movies where you visualise the positions of all the particles overtime. The colour of the particles should change according to their current speed. Plot particles with physical size. Hint *patch* or *fill* command.

6. * [for grades 9 and 10 only]

(a) Consider the 2D particles data sets defined in *Input2D-1.txt* and *Input2D-2.txt*. The format of these dataset is as follows : initial $x$ location, initial $y$ location, initial $v_x$ velocity, initial $v_y$ velocity, mass. Extend your code so that it works in two-dimensions and can deal with either two- or one-dimensional input files. Explain, carefully, the modifications you implemented, to extend your one-dimensional code.

(b) Repeat parts 4 and 5 for these 2D data sets.