PHYS 619 (Computational Physics) — Lab & Homework 02 (due 02/07)

Helmut G. Katzgraber and Amin Barzegar
Department of Physics and Astronomy, Texas A&M University

General Instructions — Homework is due one week after it was discussed in the lab. I ask you to please send a tarball of your version control directory using the TAMU filex system [https://filex.tamu.edu]. In the title please add the following string

```
LN_PHYS619_HWYY
```

where `LN` is your last name. To generate the tarball, simply issue the following commands:

```
cd dir_where_exercise_solution_is
tar cvzf LN_PHYS619_HWYY.tgz LN_PHYS619_HWYY
```

then upload the file

```
LN_PHYS619_HWYY.tgz
```

where `YY` is the weekly exercise index – in this case `YY=01`. Problems marked with `L` are meant to be completed during the lab time. You may finish these at home (because you might need part of it for subsequent exercise sets), but that will not count towards the lab grade. Problems marked with `H` are homework. However, if you have time, we encourage you to start with them during the lab time because you will have the unique opportunity to ask questions.

Problems marked with `E` are meant as extra credit problems. The numbers in parenthesis correspond to the percentage this question counts towards the grade. Example:

```
L1 [20]
```

means 'solve in lab, 20% of total grade.'

L1 [60] Bisection & Newton Method — The goal is to compute the roots of the following third-order polynomial:

$$x^3 - 4x + 2 = 0$$

You might have to guess some good starting values to find the roots, e.g., by using gnuplot:

```
gnuplot> plot [-10:10] [-5:10] x**3 - 4*x + 2
```

a) [30] Find the two positive roots using the bisection method to 4 decimal places. Make sure you bracket the roots.

b) [30] Determine the two positive roots of the polynomial using the Newton method up to 14 decimal places. Compare the convergence rate of both methods.

L2 [40] Ballistic Trajectory — Consider a classical point particle of unit mass launched into a ballistic trajectory. This particle follows the equation of motion

$$\frac{d^2\vec{r}}{dt^2} = \vec{F} \tag{1}$$

where $\vec{F} = (0, -g)$ represents the gravitational force and, for simplicity, we set $g = 1$. The initial position is $\vec{r}_0 = (0, 1)$, i.e., the projectile is fired from a cliff of height $y = 1$ at distance $x = 0$.

a) [20] Simulate its trajectory using the Euler method until the projectile crosses the horizontal axis. The initial velocity is $\vec{v}_0 = (v_0 \cos\alpha, v_0 \sin\alpha)$ with $v_0 = 1$ and $\alpha = 45°$.

b) [10] Again, simulate the trajectory using the Euler method, but this time, use $\alpha = 0°$.

c) [10] In gnuplot, plot both trajectories (x vs y).

H3 [30] Secant, Fixed Point, and Newton Method — Compute $\sqrt{3}$ up to 14 decimal places using

a) [10] The Newton method.

b) [10] The secant method.

c) [10] Fixed-point iteration.

H4 [30] Anharmonic Oscillator — Consider a point particle of unit mass moving in a (one-dimensional) potential

$$V(x) = \frac{x^4}{4}.$$

Note: If you write your code in a modular way where the force function is a separate routine and the integration is done by a separate routine, then part (b) of the problem will be extremely easy. All you will have to do is replace the integration engine from RK4 to velocity Verlet and re-run the code! This is why modular programming is very important.

a) [15] Using 4*th*-order Runge-Kutta integrate the equation of motion of the anharmonic oscillator. Generate a velocity–position phase plot for at least one orbit and store that information with your code. Furthermore, plot $x(t)$ for a few periods. Initial conditions to be used: $v_0 = 10$, $x_0 = 0$. Make sure that the energy is conserved during one orbit. This shall tell you that your step size is small enough.

b) [15] Using the velocity Verlet algorithm integrate the equation of motion of the anharmonic oscillator up to $t_{\max} = 200$ with a time step of $h = 0.125$. Again, start from $v_0 = 10$, $x_0 = 0$. Make a plot of the total energy $E = (1/2)v^2 + (1/4)x^4$ as a function of time comparing the results from the velocity Verlet algorithm to the RK4 implementation you used in part (a).

H5 [40] Kepler 2-body problem — The goal is to compare integrators for solving the Kepler 2-body problem. Consider a particle with the equation of motion

$$\frac{d^2\vec{r}}{dt^2} = -\frac{\vec{r}}{r^3}$$

that can be thought of as orbiting a unit mass at the origin. The particle has the initial conditions $\vec{r}_0 = (10, 0)$ and $\vec{v}_0 = (0, 0.1)$. The exact solution is given by

$$r(\theta) = \frac{a(1-\varepsilon^2)}{1 \pm \varepsilon \cos\theta}\,,$$

where the eccentricity is given as $\varepsilon = \sqrt{1 + 2EL^2}$ and the semi-major axis is given as $a = -1/(2E)$. $E = (1/2)v^2 - 1/r$ is the total energy and $L = |\vec{r} \times \vec{v}|$ the total angular momentum. Determine the position of the particle and the total energy of the system over 10 periods with a time step of $10^{-3}T$ (Remember that the period is given by $T = 2\pi a^{3/2}$ in dimensionless units)...

a) [15] ...with the Verlet algorithm,

b) [15] ...and with the 4*th* order Runge-Kutta algorithm.

c) [10] In gnuplot, plot both trajectories including the exact solution as x vs y. In a separate figure, plot the total energy E vs time (in number of periods). Explain what your observe.