In-Class Work: Molecular Dynamics Simulation

In last class we rewrote Newton's second law (2nd order DE)

$$F_x^{\mathrm{net}} = ma_x$$

in the form of two 1st order differential equations:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v_x$$
$$\frac{\mathrm{d}v_x}{\mathrm{d}t} = a_x = \frac{F_x}{m}$$

With the Euler Method, the time updates are

$$x(t + \Delta t) = x(t) + v_x(t)\Delta t$$
$$v_x(t + \Delta t) = v_x(t) + F_x(t)/m\Delta t$$

2. Newton's Second Law

We applied this Euler step to the case of free fall

$$F_x^{\mathrm{net}} = -mg$$

with $g = 9.8, \Delta t = 0.2, t_{\mathrm{max}} = 20.0, x_0 = 5.0, v_{x0} = 2.3$. We printed into a file $t, x(t), v_x(t)$.
And also obtained the numerical result for $x(t)$ for $\Delta t = 0.1$ and $\Delta t = 0.01$.
The analytical solution is

$$x(t) = x_0 + v_{x0}\,t + 0.5gt^2$$
$$v_x(t) = v_{x0} + g\,t$$

You find a program for this in `~kvollmay/share.dir/inclass2023.dir/md2.py`

3. Harmonic Oscillator & Surprise

3a. Numerically integrate this time for the case of an harmonic oscillator, so $F_x^{\rm net} = -kx$. We can also analytically solve this equation. Let's choose $t_0 = 0.0, x_0 = 5.0, v_{x0} = 0.0$, then the theoretical solution is

$$x(t) = 5.0\cos(\omega_0 t) \qquad\qquad v_x(t) = -5.0\omega_0 \sin(\omega_0 t)$$

where $\omega_0 = \sqrt{(k/m)}$. So we know the period $T = 2\pi/\omega_0$. Let's choose $\Delta t = T/n_{\rm div}$. Integrate $F_x^{\rm net} = -kx$ for $k = m = 1$ and for $n_{\rm div} = 100$ and do $n_{\rm max} = 10 n_{\rm div}$ MD steps. Print also the analytical solution and compare. **Note: Before you update $x(t)$, you need to copy the value of $x(t)$ into a temporary variable for example `xold=x` only then you can update x and then v. For v you need to use `xold`.** Try also with $n_{\rm div} = 1000$. What happens?

3b. Also determine the theoretical and numerical total energy

$$E_{\rm tot} = \frac{1}{2}kx^2 + \frac{1}{2}mv_x^2$$

as function of time, so $E_{\rm tot}(t)$. If you print for example into `out3sim100.dat` $t, x, v_x, E_{\rm tot}$ and similarly into the other files, you can compare the $E(t)$ results with

```
xmgrace -block out3theory.dat -bxy 1:4 -block out3sim100.dat -bxy 1:4 -block out3sim1000.dat  -bxy 1:4
```

Get me, when you have the results.

4. Euler-Cromer

Read in the Gould et al. book the first page of chapter 3. Change your program from 3b to use the Euler-Cromer step instead of the Euler step. Repeat the integration and compare again with the theoretical solution.

5. Integration Methods

If time is left, start reading the Appendix 3A which is near the end (page 30 out of 41 pages) of Chapter 3 of the Gould et al. book.

6. Driven Damped Pendulum Intro & Trajectory

6a. I will give you an introduction to the equation of motion for the driven damped pendulum. We just derived the essential equation for the simulation of the driven, damped pendulum to be

$$\frac{d^2\theta}{dt^2} = \tilde{A}\cos\left(\tilde{\omega}_{\rm D}\, t\right) - \sin(\theta) - \tilde{\gamma}\,\frac{d\theta}{dt} \qquad (3)$$

where we replaced $\tilde{t}$ by t simply for the convenience of notation. In the computer simulation we solve this equation numerically, i.e. our goal is to determine $\theta(t)$ and $\dot{\theta}(t)$.

Using the Euler method as written on the white board, program this driven damped pendulum. Use

$$\theta_0 = -2.5 \quad \omega_0 = 0.0 \quad \tilde{A} = 0.95 \quad \tilde{\omega}_{\rm D} = 2.0/3.0 \quad \tilde{\gamma} = 0.5 \quad \Delta t = 0.01 \quad n_{\rm max} = 10000$$

You may use the just presented solution to the MD-inclass step 4.
`~kvollmay/share.dir/inclass2023.dir/md4.py`

Print only every 10th MD-step $t, \theta(t), \omega(t)$. (In the following I will call this `nprint=10`.) Look at $\theta(t)$ and $\omega(t)$. If your data are in the file `out6sim.dat` you can do this with

```
xmgrace -block out6sim.dat -bxy 1:2 -bxy 1:3 -legend load
```

6b What is the energy of the driven damped pendulum? Since we chose as time unit $1/\omega_0$ and as torque unit $I\omega_0^2$ our energy unit is also $I\omega_0^2$ and this means that in the program you want to determine $\tilde{E} = \frac{E}{I\omega_0^2}$. Please get me when you have your expression for $\tilde{E}$. Then add the determination of $\tilde{E}$ to your program and print $\tilde{E}(\tilde{t})$ and look at your results with `xmgrace`. Get me also when you have your result. We will discuss the interpretation of your result.