Lab assignment #6: ODEs Part 1

Instructor: Nicolas Grisouard (nicolas.grisouard@utoronto.ca)
TA & Marker: Mikhail Schee (mikhail.schee@mail.utoronto.ca)

Due Friday, 22 October 2021, 5 pm

It appears that there is no need for a second room. Should it be incorrect and the room overflows, N.G. will open up the room he had for the previous labs (grab the link on the old question documents).

Room 1 (also for office hour) Mikhail Schee and Nicolas Grisouard,
URL: https://gather.town/app/faemi6rJ9YvAgbx5/MS-PHY407-2
PWD: `phy407-2021-MS`

General Advice

- **Work with a partner!**
- Today's topics revolve around applications of the Fourier transform, and their numerical computations by computers. Compared to other labs, coding proficiency will be less crucial, and we will instead focus on physical applications.
- Read this document and do its suggested readings to help with the pre-labs and labs.
- Ask questions if you don't understand something in this background material: maybe we can explain things better, or maybe there are typos.
- Carefully check what you are supposed to hand in for grading in the section "Lab Instructions".
- Whether or not you are asked to hand in pseudocode, you **need** to strategize and pseudocode **before** you start coding. Writing code should be your last step, not your first step.
- Test your code as you go, **not** when it is finished. The easiest way to test code is with `print('')` statements. Print out values that you set or calculate to make sure they are what you think they are.
- Practice modularity. It is the concept of breaking up your code into pieces that as independent as possible from each other. That way, if anything goes wrong, you can test each piece independently.
- One way to practice modularity is to define external functions for repetitive tasks. An external function is a piece of code that looks like this:

```python
def MyFunc(argument):
    """A header that explains the function
    INPUT:
    argument [float] is the angle in rad
    OUTPUT:
    res [float] is twice the argument"""
    res = 2.*argument
    return res
```

Place them in a separate file called e.g. `MyFunctions.py`, and call and use them in your answer files with:

```python
import MyFunctions as fl2  # make sure file is in same folder
ZehValyou = 4.
ZehDubble = fl2.MyFunc(ZehValyou)
```

Computational background

4$^{\text{th}}$-order Runge-Kutta (RK4) In Lab #1, you have previously seen the Euler and Euler-Cromer methods for integrating systems of ODEs with initial values. A 4$^{\text{th}}$-order method, which is probably the most widely used method for solving systems of ODEs, is the 4$^{\text{th}}$ order Runge-Kutta method (or RK4 for short).

The lecture only covers RK4 for one first-order ODE, but an extension to higher-order and coupled ODEs is relatively straightforward, as discussed in the text (§§ 8.2, 8.3). This method invokes calculating the RHS vector of $d\vec{r}/dt = \vec{f}(\vec{r},t)$ at various intermediate points between steps. Full implementation requires coding the following 5 lines which are iterated over t values:

$$\vec{k}_1 = h\vec{f}(\vec{r},\ t) \tag{1}$$

$$\vec{k}_2 = hf\left(\vec{r}+\frac{1}{2}\vec{k}_1,\ t+\frac{1}{2}h\right) \tag{2}$$

$$\vec{k}_3 = hf\left(\vec{r}+\frac{1}{2}\vec{k}_2,\ t+\frac{1}{2}h\right) \tag{3}$$

$$\vec{k}_4 = hf\left(\vec{r}+\vec{k}_3,\ t+h\right) \tag{4}$$

$$\vec{r}(t+h) = \vec{r}(t)+\frac{1}{6}\left(\vec{k}_1+2\vec{k}_2+2\vec{k}_3+\vec{k}_4\right) \tag{5}$$

Verlet algorithm We will implement the Verlet method for a conservative system (i.e., the molecular dynamics simulation). The algorithm is described in the text on pp. 371-373. Starting from $a = F/m$, or equivalently,

$$\frac{\mathrm{d}^2\vec{r}}{\mathrm{d}t^2} = \vec{f}(\vec{r},t) \tag{6}$$

and denoting the velocity as $\vec{v}$, the Verlet algorithm works as follows.

For the first step only,

$$\vec{v}\left(t+\tfrac{1}{2}h\right) = \vec{v}(t)+\tfrac{1}{2}h\vec{f}\,(\vec{r}(t),\ t)\,, \tag{7}$$

then repeatedly apply the equations

$$\vec{r}(t+h) = \vec{r}(t) + h\vec{v}\left(t+\tfrac{1}{2}h\right), \tag{8}$$

$$\vec{k} = h\vec{f}\left(\vec{r}(t+h),\ t+h\right), \tag{9}$$

$$\vec{v}(t+h) = \vec{v}\left(t+\tfrac{1}{2}h\right) + \tfrac{1}{2}\vec{k}, \tag{10}$$

$$\vec{v}\left(t+\frac{3}{2}h\right) = \vec{v}\left(t+\tfrac{1}{2}h\right) + \vec{k}. \tag{11}$$

This last equation for $\vec{v}\left(t+\frac{3}{2}h\right)$ becomes $\vec{v}\left(t+\frac{1}{2}h\right)$ for the next iteration of Eqn. (8). Eqn. (10), which calculates $\vec{v}(t+h)$, is not required for the timestepping. It is useful if you want to diagnose the velocity at the same time as the $\vec{r}(t+h)$ for the purpose of, for example, calculating the energy.

Physics background

A stick-slip model Several vibrating phenomena such as earthquakes, creaking doors or the sound of a bow on a violin string can be explained by stick-slip phenomena. Stick-slip happens in non-lubricated friction, when two solids rub against one another parallel. The simplest model that encapsulates the physics of stick-slip is that of a mass m, attached to an ideal spring of stiffness k, the other end of the spring being pulled at constant velocity v_p (see fig. 1).

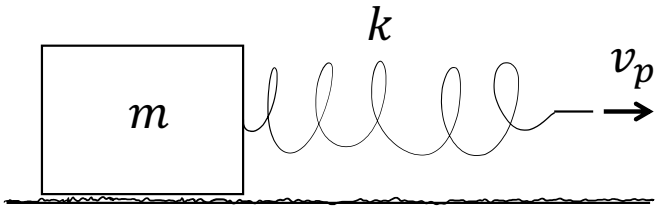

Figure 1: Basic stick-slip setup.

A common approximation for the friction force is that of the "fluid friction", or Stokes-like force,

$$F_f = -\alpha\dot{x}, \tag{12}$$

where x denotes the position of the massive object and $\dot{x} = \mathrm{d}x/\mathrm{d}t$ is the velocity of the mass, which is different than v_p in general.

Assuming that this fluid friction and the spring restoring forces are the only ones to act on the mass, the equation of motion is

$$m\ddot{x} = -k(x - v_p t) - \alpha\dot{x}, \tag{13}$$

with $\ddot{x}$ is the acceleration of the mass.

The equation above admits a particular solution satisfying $\dddot{=}0$, which is $x = \alpha v_p/k + v_p t$. The equation above also admits linear oscillations about this particular solution. That is,

$$x = \frac{\alpha v_p}{k} + v_p t + A\mathrm{e}^{\mathrm{i}\omega t}, \tag{14}$$

with $A \in \mathbb{C}$, is a solution if and only if ω satisfies

$$\omega = \mathrm{i}\frac{\alpha}{2m} \pm \sqrt{\frac{k}{m} - \left(\frac{\alpha}{2m}\right)^2}. \tag{15}$$

That is, if damping is weak ($\alpha^2 < 4km$), the solution is that of damped oscillations with pseudo-frequency $\sqrt{k/m - \alpha^2/4m^2}$, decaying exponentially at a rate $\alpha/(2m)$.

In the presence of static friction however, the mass m will tend to "stick" to the surface unless the tangential pulling force exceeds a certain value, after which the friction against the surface is called "dynamic". Without this static friction force, a ladder couldn't lean against a wall, or we could not walk without slipping.

The transition from static to dynamic friction is discontinuous, depends on microscopic properties of the surface and the mass, and generally not easy to represent with a single ODE. Instead, we model these phenomena with a "pseudo-static friction force"

$$F_s = -\beta \mathrm{e}^{-\dot{x}/v_f}, \tag{16}$$

which is non-zero when $\dot{x} \to 0$.

Vibrations of a building This background material and the corresponding question are inspired by an experiment described at

http://www.shodor.org/~wmyers/curric/workshops/lessons/ise/renee/weave/module1/equations.html#fig:3dof

Consider a simple model of a building consisting of massive rigid floors supported by light flexible columns, as sketched in Fig. 2.

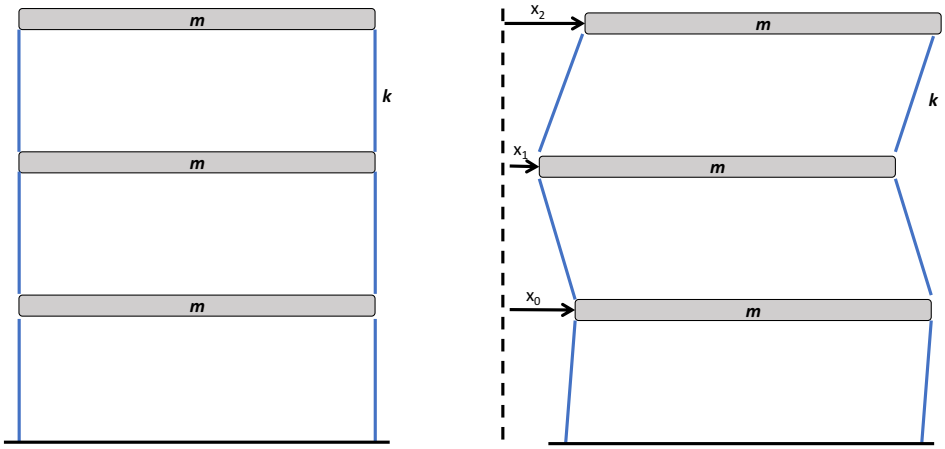

Figure 2: Sketch of the vibrating building.

The bottom columns are attached to the ground which we assume is at rest. When floors are displaced in the x direction relative to each other, the shear forces in each column can be model as being proportional to the displacement of the column from the vertical, with stiffness constant k.

For example, if x_0 is the displacement of the lowest floor, floor 0, and x_1 the displacement of the next floor up, the lowest floor will experience a shear force $-kx_0$ from the columns below and $k(x_1 - x_0)$ from the columns above. Thus the equation of motion of the lowest floor will be

$$m\ddot{x}_0 = (-kx_0) + k(x_1 - x_0) = k(x_1 - 2x_0). \tag{17}$$

Continuing for the other floors, the equations of motion will be of the form

$$\ddot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}, \tag{18}$$

where $\boldsymbol{x} = (x_0, x_1, x_2)^T$ and

$$\boldsymbol{A} = \frac{k}{m}\begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}. \tag{19}$$

We can find the modes of vibration of this system for oscillatory solutions of the form $x_i = \hat{x}_i \cos(\omega t + \phi_i)$. These are known as the *normal frequencies* and *normal modes* for the building. Substituting this form into the equations of motion, we obtain the eigenvalue problem

$$\boldsymbol{A}\hat{\boldsymbol{x}} = \lambda\hat{\boldsymbol{x}}, \tag{20}$$

where the eigenvalues $\lambda = -\omega^2$ will be the negative of the square of the frequencies and the amplitudes $\hat{\boldsymbol{x}} = (\hat{x}_0, \hat{x}_1, \hat{x}_2)$ will be the eigenvectors describing the normal modes.

Questions

1. **[60%] Stick-slip phenomenon**

   (a) This time, we will work on establishing and understanding a few properties of the stick-slip phenomenon from an analytical point of view.

   If you get stuck on this first, mostly analytical part, move on to the next computational part and come back to the analytical part later.

   i. Plot the total absolute frictional force $|F(\dot{x})| = |F_f(\dot{x}) + F_s(\dot{x})|$ for $\dot{x} > 0$. To see what is going on, you may want to vary the ratio α/β as well as v_f. Give an brief physical interpretation for β and v_f. Analytically show that when $\dot{x} < v_f$, one needs to have $\beta > \alpha v_f$ in order for the total frictional force to first decay when $\dot{x}$ increases from zero. Based on what we are trying to model, why do we want to have such a property?

   ii. Looking at eqn. (16), what happens when $\dot{x} < 0$? How would you fix this problem?

   For the rest of the analytical calculations in this question, you may assume that $\dot{x} > 0$. However, your simulations must include the fix for $\dot{x} < 0$.

   iii. Write or type the equation of motion of the system. Introduce

   $$\omega_0^2 = \frac{k}{m}, \quad \tau = \frac{m}{\alpha} \quad \text{and} \quad \gamma = \frac{\beta}{m}. \tag{21}$$

   Give the units of ω_0, τ and γ and their physical meaning. Re-write the condition $\beta > \alpha v_f$ with these new parameters.

iv. What condition must the initial position x_0 satisfy in order to have a constant-velocity solution? I.e., such that $\dot{x} = C$, with C a constant to be determined. Write the full solution.

v. We are now considering a solution of the form

$$x(t) = x_0 + Ct + u(t), \tag{22}$$

with x_0 and C as derived in the previous part, and $u(t)$ an unknown deviation from $x_0 + Ct$. Write down the ODE that $u(t)$ is a solution of.

Assuming $|\dot{u}| \ll v_f$ (small velocity fluctuations), find the condition that v_p, v_f, τ and γ must satisfy in order for $u(t)$ to follow simple harmonic motion. *Hint: we are looking for an equality.*

SHOW PLOTS AND WRITTEN ANSWERS SO FAR.

(b) You will now write a program to solve the ODE you found in Q1(a)iii (*not* Q1(a)v; the ODE for x is more straightforward and therefore easier to code; that way you can semi-independently validate your calculations of Q1(a)v by isolating u in your numerical solution).

I am not writing any units to not spoil Q1(a)iii, but all quantities are dimensional.

i. Write a program to solve the motion of x with the 4^{th}-order Runge-Kutta method. Why can't you use the Verlet method in this case?

SUBMIT YOUR WRITTEN ANSWER TO THE LAST QUESTION.

ii. Let's do a sanity check first: assume $\omega_0 = 1$, cancel the total frictional force F, and assume $v_p = 0$. Your ODE should become that of a simple harmonic oscillator with a well-know solution. Check that your program gives the expected solution. To choose your time step dt, compute the energy of the system with time. You should find that the energy decays, but that the decay is smaller as dt gets shorter. Try various values for dt between 10^{-5} s and 1 s until you are comfortable with how little the energy decays after, say, 10 oscillations. We could show that the energy of this system is

$$E \propto \omega_0^2 (x - v_p t)^2 + \dot{x}^2 \tag{23}$$

up to a constant factor $m/2$.

There is no single "correct" answer, only well-justified ones. Whichever dt you choose, you may keep it for your subsequent simulations.

SUBMIT YOUR PLOTS AND YOUR WRITTEN ANSWERS.

iii. Now set $\tau = 1$, $\gamma = 0.5$ and $v_f = 0.1$, which satisfy the condition laid out in Q1(a)i and Q1(a)iii.

Choose your initial conditions appropriately in order to verify that

- your answer for Q1(a)iv is indeed a constant-velocity solution, and

- your answer for Q1(a)v is indeed a simple harmonic motion superposed on top of the previous translational motion.

Hints:

- *Choose what you plot wisely in order to reveal what you want to show.*
- *Numerical solutions of ODEs are rarely perfect. They could be imperfect for two reasons: you did not find the correct analytical solution, or the integration method has errors, as all numerical methods do. The latter can be identified by seeing the error shrink as the time step decreases. If it is not the case, then the former option appears more likely, and needs to be fixed.*

SUBMIT YOUR CODE, PLOTS AND WRITTEN ANSWERS.

(c) Let us now compute solutions to more challenging cases. Set both your initial position and velocity to zero ($x|_{t=0} = 0$, $\dot{x}|_{t=0} = 0$), keep all other parameter values the same as the last question except for v_p, which you will vary.

Run a few simulations with v_p ranging from $0.1 v_f \ln(\gamma\tau/v_f)$ to $1.5 v_f \ln(\gamma\tau/v_f)$. Describe your results. What is the connection with the systems we mentioned in the Physics background, such as the violin or a creaking door?

SUBMIT YOUR CODE IN A SEPARATE FILE FROM THAT OF Q1B

2. **[40%] Coupled oscillating system: vibrating building**

Consider an N-storey building as a generalization of the three-storey one described in the Physics background. We will solve this system using the Verlet method, compare this solution to the eigenmodes determined by finding the eigenvalues and eigenmodes, and analyze some aspects of the dynamics of this system.

(a) Set up a system of $2N$ first-order ODEs to describe the motion of the N floors and solve them using the Verlet algorithm with the following information.

- Use $k/m = 400$ rad s^{-2}. This ratio is known as the stiffness to mass ratio, and will suggest normal modes with a few cycles per second (Hz).
- Use a time step of $dt = 1$ ms.
- Use an initial value of $x_0 = 10$ cm for the initial position of the lowest floor, $x_i = 0$ for all of the upper floors, and start from a state of rest.
- Write your program such that it can take any N as an argument. To do so, we want you to take advantage of the matrix structure of the problem. This will allow us, in principle, to model the vibrations of a very tall skyscraper with dozens of storeys. Describe, in your explanatory notes, how you implemented this problem taking advantage of the matrix structure. If you want, you can include the relevant lines of code in your explanatory notes.

Plot time series of the motion of all floors over a few cycles, and describe the behaviour you see during the initial evolution, and then its evolution over a longer time period. Do so with $N = 3$ and $N = 10$.

Hint: You ware looking to identify a signal that initially propagates upward. To visualize it, you need to plot your curves together while keeping it legible.

HAND IN YOUR CODE (YOU MAY COMBINE IT WITH THE CODE OF THE NEXT PART), PLOTS AND WRITTEN ANSWERS.

(b) Another way to characterize this system is to find its normal modes of vibration. For $N = 3$, use Python to find the characteristic normal frequencies (in Hz) and describe the normal modes of the system that vibrate with these frequencies (similar to the kind of analysis you would have done in PHY254). Then confirm that these are normal modes by initializing the system in each of these modes, starting from rest with $\hat{\boldsymbol{x}}$ set to each of the eigenvectors, in turn, and plotting the time series of the solutions that result. Are the frequencies in your simulation the same as predicted from the eigenvalue analysis?

HAND IN YOUR CODE, PLOTS AND WRITTEN ANSWERS.