Real World Problem Solving

Andey Robins & Dr Borowczak

April 18, 2023

Real World Problem Solving

Motivation

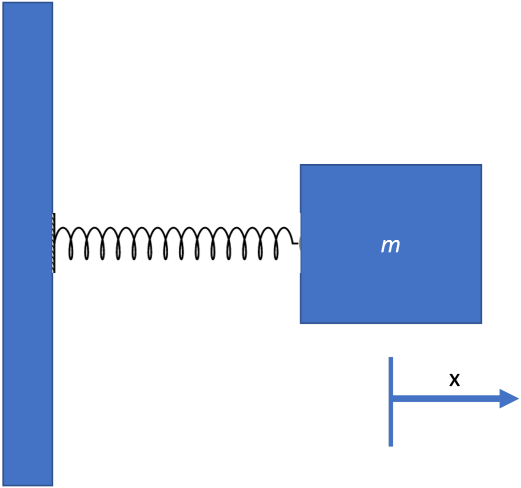
What good does writing code for some abstract problem do?

practice, for fun, just cause

Computer science is all about being able to use computers to solve meaningful problems

Example: Springs

I want to determine how a spring would behave under various loads. I have an equation which models the motion over time, but I want to see how various changes to values impact the underlying system.



$g = 0$

Displacement only in x (no y or z)

$x = A \cos(\omega t + \phi)$

$\omega = (k/m)^{(1/2)}$

k = spring constant

Define Constants

```
A = 0.05
k = 2.0
phi = 0.0

t = np.arange(0, 10, 0.1)

m = 2
```

Model the Spring

```
mass = m
omega = np.sqrt(k / mass)
x = A * np.cos(omega * t + phi)
```

What does the spring look like?

```
array([ 0.05      ,  0.04975021,  0.04900333,
        0.04387913,  0.04126678,  0.03824211,
        ...
       -0.04985861, -0.04923439, -0.04811824,
       -0.04652131, -0.04445956])
```

And this tells us everything we wanted to know about our model. The problem is, that's not a very human readable result. Wouldn't it be nice if we could graph it or something?

External Code

Turns out, there is a lot of code out there [citation needed].

Instead of re-writing simple behavior everytime it is needed, it makes more sense to leverage what has been already done.

"Don't reinvent the wheel."

Load Other Files

Importing is the process by which we load code from one file into another.

Two files: `a.py` and `b.py`

Example of Multiple Files

a.py

```
def function_x(x):
    return x * x
```

b.py

```
import a

print(a.function_x(2))
"4"
```

Types of Imports

```
# standard import
import os

# partial import
from psb2 import PROBLEMS

# named import
import numpy as np
```

PyPi

The **Py**thon **P**ackage **I**ndex is the place where we share most python packages.

Look:

- matplotlib
- numpy
- psb2
- turtle

Adding External Packages

pip is short for "pip installs packages." It's what is called a "package manager." It's a tool which allows us to easily manage code. You'll use it briefly in lab this week.

```
pip install <package-name-here>
```

Retrospective: Autograders

This idea of importing functionality is how all of the autograders work! Let's examine a couple and see how it all breaks down.

```python
import unittest
from toki import translate

class TestTranslator(unittest.TestCase):
  def test_translate(self):
    answers = {
      "Someone is good.": "jan li pona.",
      ...
    }
    for key in answers:
      print(f'Expected: {answers[key]}')
      print(f'Got:      {translate(key)}')
      self.assertTrue(translate(key) == answers[key])
      print()

if __name__ == '__main__':
  unittest.main()
```

Example: Graphing

The most common way to do graphs in python is with the package `matplotlib.pyplot`. This is very commonly imported as `plt`

```
import matplotlib.pyplot as plt
```

Setup

```
plt.plot(x, y)
plt.xlabel('X (units)')
plt.ylabel('Y (units)')
plt.title('Graph Title')
```

Return to Spring Problem

```
figure, ax = plt.subplots(len(m), 1, figsize=(12,7), sharex
for experiment in range(len(m)):
    mass = m[experiment]
    omega = np.sqrt(k/mass)
    x = A * np.cos(omega * t + phi)

    plt.subplot(len(m),1,  experiment+1)
    plt.plot(t,x)
    plt.xlabel('Time (sec)')
    plt.ylabel('Displacement (m)')
    plt.title('Displacement for a ' + str(mass) + 'kg Mass
```

Checkpoint

At this point, we will examine and manipulate the code available on Codio.

Example: Problem Identification

```
volume = 1
moles = 1
R = 8.314 # constant
temp = 273

print((moles * R * temp) / volume)
```

Example: Chemistry

$$pV = nRT$$

$$p = \frac{nRT}{V}$$

Lets model how the pressure of a gas changes with temperature.

Modeling

```
y_pressure = []
t = np.arrange(273, 273+100, 1)
for temperature in t:
    temp = temperature
    y_pressure.append((moles * R * temp) / volume)
```

Graphing

```
plt.plot(t,y_pressure)
plt.xlabel('Temperature (K)')
plt.ylabel('Pressure (bar)')
plt.title('Pressure for a {mole} Mole & {volume} Litre Syst
```

Retrospective: Turtles

Remember the turtles we used to draw shapes all the way back in Quest 1?

```
import turtle

sam = turtle.Turtle()
```

We can now understand this code as importing the module called `turtle` which provides access to all the code relating to turtles!

Next Time

- Learn how to win a gameshow
- Estimate the number of civilizations in the galaxy
- Breed super-mutant rodents
- Prepare for 1030