Attached written notes are my pseudo code for the problems.

8.4 a.

(8.45) $\frac{d\theta}{dt} = w$

(8.46) $\frac{dw}{dt} = -\frac{g}{l}\sin\theta$

$l = 10\,cm$

$t = 0 \quad \theta = 179°$

Use 4th order Runge-Kutta method

pseudo code:

```
import everything
g = 9.81
l = 0.1
def f(r,t):
    θ = r[0]
    w = r[1]
    f(θ) = w
    f(w) = -g/l · sin θ.
    return ( ).
a = 0.
b = 100.
N = 1000
h = (b-a)/N.
t_values = arange(a, b, h).
θ_values = [ ]
w_values = [ ]

r = array([179°, 0], float).
for t in t_values:
    θ_values.append(r[0])
    w_values.append(r[1]).
    k1 =
    k2 =
    k3 =
    k4 =
    r+ = (k1 + 2k2 + 2k3 + k4)/6.

plot(t_values, θ_values).
```

8.5 a.

transform a 2nd order ODE into 2 1st order ODE.

$\frac{d^2\theta}{dt^2} = f$

$\frac{d\theta}{dt} = w$

$\frac{dw}{dt} = \frac{d^2\theta}{dt^2} = f$

$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin\theta.$

$\frac{d^2\theta}{dt^2} + \frac{g}{l}\sin\theta = 0.$

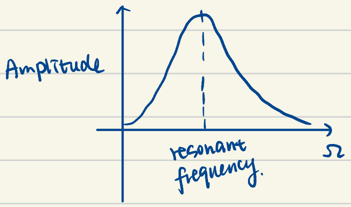
$\sim \frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = 0$

$F = C\cos\theta\sin\Omega t$

same pseudo-code as 8.4 a) but with a slight change to w equation & initial conditions.

```
import everything
g = 9.81
l = 0.1
C = 2
Ω = 5
def f(r,t):
    θ = r[0]
    w = r[1]
    f(θ) = w
    f(w) = -g/l · sin θ + C cosθ sin Ωt
    return ( ).
... (same as 8.4 a) ).
```

8.5b PART ①.




```
maxθ_values = []
... (same as a))
Ω = np.linspace(5, 30, 100).
... (same code).
For i in Ω:
   t_values = arange(a, b, h).
   θ_values = [ ]
   w_values = [ ]
   r = array([0, 0], float)
   for t in t_values:
        θ_values.append(r[0])
        w_values.append(r[1]).
        k1 =
        k2 =
        k3 =
        k4 =
        r+ = (k1 + 2k2 + 2k3 + k4) / 6.
   maxθ = max(θ_values)
   maxθ_values.append(maxθ).

plt.(Ω, maxθ_values)
```

Set Ω = Ω [maxθ_value_index]
Use the same code in part a) to plot θ v.s. t.

8.2 rabbit $\frac{dx}{dt} = \alpha x - \beta x y$

fox $\frac{dy}{dt} = \gamma x y - \delta y$

As shown in 8.4 a), I used the same code but change the constants, functions, and initial conditions.

$\alpha = 1$

$\beta = \gamma = 0.5$

$\delta = 2$

```
def f(r,t):
    x = r[0]
    y = r[1]
    f(x) = αx - βxy
    f(y) = γxy - δy
    return ( ).
    ... same as before.
r = array([2, 2], float)
    ... same as 8.4 a).
```

53·54·5 = 112

112/117

PHYS 304 AS8

Xiyue Shen*

Haverford College Department of Physics

(Dated: March 29, 2024)

No Collaborators. I have four Python files.

PHYS_304_AS8_8.2_Xiyue_Shen.py file is the Predator-prey problem, numbered as 8.2 in the textbook.

PHYS_304_AS8_8.4a_Xiyue_Shen.py
PHYS_304_AS8_8.4a_Xiyue_Shen.py
PHYS_304_AS8_8.4a_Xiyue_Shen.py are the files for pendula problem, numbered as 8.4*a*, 8.5*a*, and 8.5*b* in the textbook

The pseudo-code is attached at the beginning.

1. PENDULA

1.1. Exercise 8.4a

In this problem, we are given two first-order differential equations,

$$\frac{d\theta}{dt} = \omega \tag{1}$$

$$\frac{d\omega}{dt} = -\frac{g}{l}\sin\theta \tag{2}$$

In this way, we simplify a second-order differential equation $\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin\theta$ to a solvable situation. Following the hint code provided, I added several steps, including dividing the time-space, setting the initial conditions, initiating the differential equation, and making the plot (shown in the pseudo-code). The result is shown in figure 1

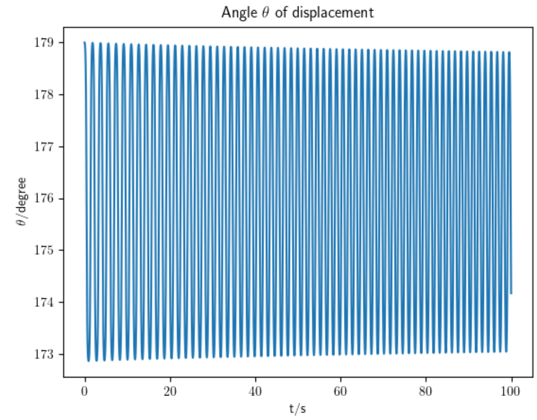


FIG. 1: [Angle θ of displacement with initial position 179°]

One interesting thing about the result is that we observe some degree of damping in the oscillation, while no damping term is included in the equations. One possibility is that our solving method is not accurate enough. In the textbook, there's another method called leap-frog, which will conserve the system's energy, while the Runge-Kutta method makes the system gradually lose energy.

1.2. Exercise 8.5a

In this problem, we have an extra term in the differential equation. A driving force keeps the pendulum from damping to zero. The differential equation is shown as,

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin\theta + C\cos\theta\sin\Omega t \tag{3}$$

Similarly, we split this second-order differential equation into two first-order differential equations,

$$\frac{d\theta}{dt} = \omega \tag{4}$$

$$\frac{d\omega}{dt} = -\frac{g}{l}\sin\theta + C\cos\theta\sin\Omega t \tag{5}$$

Using the same code with a slight adjustment in the constants and equations, the result is shown in figure 2

C = ? Ω = ?

1.3. Exercise 8.5b

Now, we want to find the resonant frequency. A large displacement in the oscillation is expected by setting the

*Electronic address: xshen2@brynmawr.edu; URL: Optionalhomepage

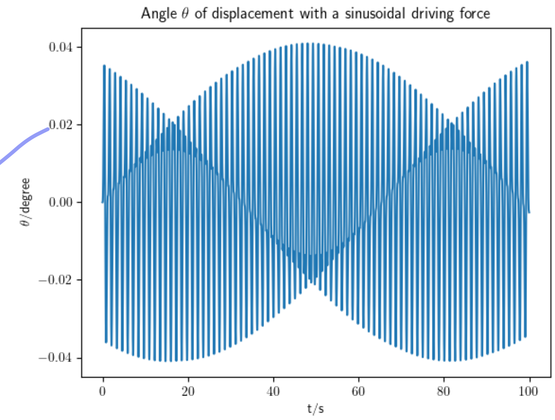


FIG. 2: [Angle θ of displacement under a driving force. The initial conditions are $\theta = 0$ and $d\theta/dt = 0$]

driving frequency Ω equal to this value. To find the resonant frequency, I give an array of Ω values, collect the maximum θ value, and make them into another array. A curve is observed as shown in figure 3

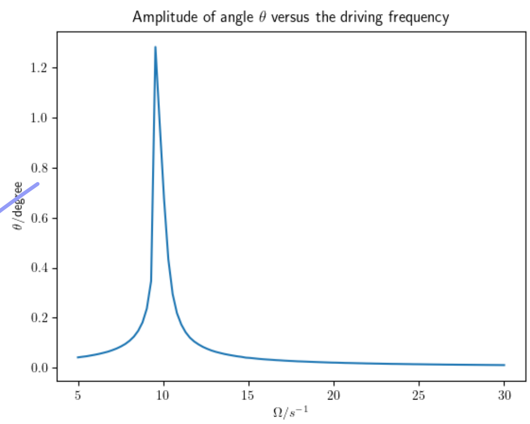


FIG. 3: [Amplitude with respect to Ω, driving frequency]

Then I set the Ω equal to the value where the θ amplitude is maximum, around 9.5454. Then, make another plot for the oscillation at this driving frequency. The plot is shown in figure 4

As we observed the amplitude of the oscillation is indeed around 1.2, as indicated in figure 3.

1.4. Physics behind pendula

One interesting thing about his system is that the initial conditions for both θ and ω are zero, which means that the system will stay silent without the driving part. The driving role here is to start the oscillation, and gradually, the driving frequency and the system's internal frequency are out of phase.

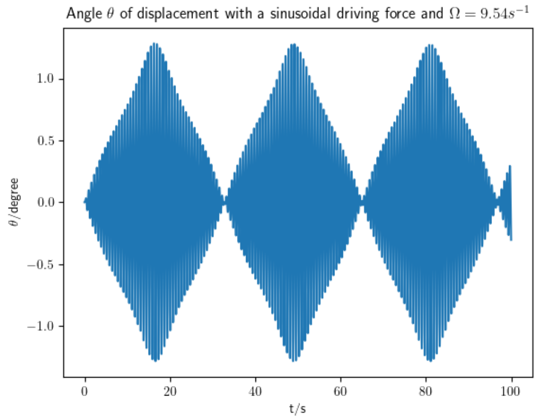


FIG. 4: [Angle θ of displacement under a driving force at resonant frequency.]

Nice!

Recall what we have learned from 213 Waves and Optics, the amplitude is a function as,

$$A(\omega_d) = \frac{F_0/m}{\sqrt{(\omega_0^2 - \omega_d^2)^2 + (\gamma\omega_d)^2}} \tag{6}$$

When the denominator gets smaller, the amplitude is higher, when the driving frequency is in the area of the system's natural frequency. The natural frequency is around $\sqrt{g/l} \approx \sqrt{98.1} \approx 10$. We can narrow down the range of Ω around this value and have a more accurate calculation.

2. EXERCISE 8.2 THE LOTKA-VOLTERRA EQUATIONS

In this problem, we have a mathematical model that can describe the predator-prey interactions. We are given two first-order differential equations,

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{7}$$

$$\frac{dy}{dt} = \gamma xy - \delta y \tag{8}$$

$$\tag{9}$$

$\alpha = ?$ $\beta = ?$ $\gamma = ?$ $\delta = ?$

The $\frac{dx}{dt}$ refers to the rabbit population, while the $\frac{dy}{dt}$ indicates the fox population. Interpreting the equations, the rabbit population growing rate is positively proportional to the overall population and inversely proportional to the interaction between rabbit and fox. The old rabbit will give birth to a new rabbit, causing an increase. For the interaction, the fox will consume the rabbit, leading to a decrease. The fox population is in the opposite situation. The old fox will compete with the new fox for food. The more foxes there, the less likely the population will increase.

In part a, we are asked to solve the two differential equations and make a plot from $t = 0$ to $t = 30$, assuming the unit is day. Just as with the code we used in previous problems, change the constants, rewrite the equations, reset the time spaces, and make the plot. The result is shown in figure 5

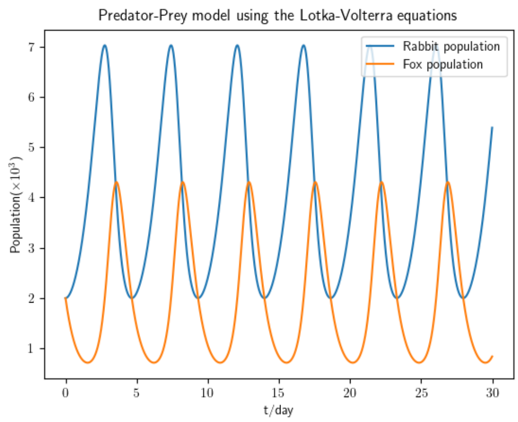


FIG. 5: [Predator-prey model using the Lotka-Volterra equations]

In part b, we are asked to explain the plot. In the beginning, we have 2000 rabbits and foxes. Immediately after this, the rabbit population begins to drop; at the same time, we can see the fox population going up. This means the foxes are consuming the rabbits. Until the point that the rabbit population approaches zero, meaning there's no food for the foxes. This is followed by a maximum value in the fox population, inducing much competition for food in the fox group. The old fox population is significant, too, due to the proportionality of the fox population. The fox population begins to drop. Because the predator is decreasing, the rabbit population grows until the competition among foxes is significantly reduced. This is where the two populations are about to intersect. At this time, the rabbit population reaches maximum. The food reservoir is filled again, and the possibility of a fox gaining food is substantial due to the xy term. Then, the rabbit population drops again. They reach the point where they start. Such a cycle will repeat.

3. SURVEY

15

I spent roughly 6.5 hours on this homework. I learned several methods to do differential equations. I like the driving oscillating pendulum most. I thought it would be tough to describe the driving situation because of the beating induced. But, it indeed works, which I'm very surprised. I practiced using the fourth-order Runge-Kutta method. I think this set is about the right length.

4. UNGRADED PART

I have done all of the required and ungraded work.

8.4+8.5

53/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

3 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

2 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)? missing units -1

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

2 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? Name missing -1

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

2 . Are key figures and numbers from the problem given? (+ 3 points) numerically define variables ($C = ?$, $\Omega = ?$) -1

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2 . Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)

1 . Are collaborators clearly acknowledged? (+1 point)

2 . Are any outside references appropriately cited? (+2 point)

8.2 54/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

3 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

---

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

2 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? Name missing -1

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

2 . Are key figures and numbers from the problem given? (+ 3 points) numerically define variables (α, β, etc.) -1

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)