Term Project Part A: Proposal
Due Tuesday, April 2

Please note that this is **Part A** of a three-part Term Project. For Part B, you'll need to accomplish most of what you propose to do here, in Part A. For Part C, you will be asked to refine and improve upon Part B, based on feedback you will receive when Part B is handed back.

All parts of the Term Project are worth the same amount. No part can be dropped. So please do a good job!

So here's what you need to do for Part A (this assignment):

1. **Choose your *own* scenario to model with differential equations.** That is, you need to envision, or imagine, or invent, a situation that can be modeled with a system of differential equations, analogous to (but different from!!!) systems we've studied like SIR; fermentation; circadian rhythms; monomers, dimers, and trimers; etc. Describe this scenario, in words, in this proposal.
2. Also explain in words how the rates of change of your variables behave. For example: "The purple monster population grows at a rate proportional to the number of coffee shops present. The number of coffee shops grows logistically, with carrying capacity proportional to the number of travel mugs present. But coffee shops also close, at a rate proportional to the purple monster population. The number of travel mugs present grows at a rate proportional to the product of the purple monster population and the number of coffee shops present."
3. Write down differential equations reflecting the behavior you described above. You do *not*, at this point, need to specify actual numerical values for your parameters. (Just give them letter names, like b, c, k, etc.)

For Part A (this part) of the project, you do *not* need to worry about writing any Sage code.

Later this semester, when you begin working on Part B of this project, you may end up changing around some of what you have done for Part A above. That's OK. The point is to start thinking about how some set of interrelated quantities can be described in terms of their rates of change.

Some hints and notes:

(a) Be ORIGINAL! It's OK to to refer to modeling problems we've already studied, like SIR, fermentation, etc., for guidance and inspiration, but **your scenario should differ in significant, interesting ways from anything you've seen before in this class.**

(b) Be CREATIVE!!! Interpret the adjective "real-world" broadly. You can be as silly, wacky, etc. as you'd like. It's not important that you describe anything particularly realistic. What's important is that you describe a scenario that you can model with differential equations, and will later be able to study with a Sage program.

Besides the second (SIR) and third (fermentation) Mini Projects, you might want to look at the following for inspiration/ideas: the class notes and tutorials from Week 8 and the first part (Monday and Tuesday) of Week 9; Part B of Individual Homework #7 .

For further inspiration, **see the sample Term Projects attached at the end of this assignment.** These are actual *first drafts* of Term Projects handed in by groups from previous semesters. (Again, for this assignment, you just need to *propose* a scenario; you don't need to do a first draft yet. But these samples should give you something of an idea of how things should ultimately look.)

(c) The system you create should involve at least *three* interrelated quantities (as in the SIR or fermentation projects). You can have more than three if you want, but things get very complex (and hard to make work) if you have too many variables.

(d) Again: for now, you needn't worry about writing any Sage code. But keep in mind that you'll need to do so eventually.

(e) Get started *early.* Give your group's brains time to play around with this.

(f) Have fun with it!!!

A sample Term Project

We analyzed a race between three extremely mental athletes. Our first competitor is Discouraged Dan, a snail who enters as the undisputed underdog. Everybody loves a Cinderella story, but for this snail who prefers to go by "DD", an upset is unlikely. Next we have a veteran crowd favorite, Timmy Two-Toes, the sloth. He is one of the animal kingdom's most respected low-speed athletes, and is racing for his world record 10^{th} win. Rounding out the pack is Greasy Garry, a turtle who makes up for his lack of speed with cunning, and trickery. As Timmy once said, "If Garry ain't cheatin', then you've got the wrong Garry, because this Gary is seriously always cheating. "
As mentioned, these are incredibly mental athletes, whose performances depend heavily on their states of mind.

Here's what we know:

--We can count on Greasy Garry timing the starting shot all too well, and getting a (relative) running start, at a (relatively) high speed of 15 meters/hour.
--Dan complains his way into a sliding start, and with a boost from Tanner Ten Point, the buck, our underdog snail starts the race at a cool 10 meters/hour.
--We see so many athletes use the "everyone does it" excuse, but expect Timmy Two-Toes to start standing still, in a display of true veteran class.
--Absent of other factors, Dan's speed increases logistically with a growth rate of .001, with a maximum speed that is proportional to Garry's speed. The mental aspect of racing comes into play here. Discouraged Dan doesn't get his name for just any reason. The faster his opponents get, the worse he performs. His maximum performance is essentially limited by Garry's success. Discouraged Dan really looks up to Timmy, and is negatively affected by the sloth's stellar performance. Specifically, he loses speed at a rate proportional to .001 of Timmy's speed, in proportion to his own speed.
--Timmy displays the extreme drive necessary for success in the competitive low-speed racing world. He couldn't care less about the slow snail to his right. But, Garry got a head start, and Timmy Two Toes is motivated and prepared. The determined sloth increases his speed at a rate proportional to 2% of Garry's speed.
--Garry normally couldn't keep the speed with which he started. Absent outside factors, he would lose his speed at a rate of .25 meters/hour per foot. However, he is a competitive animal as well. As Timmy gets started, and especially as he accelerates quickly behind, Garry speeds up accordingly. In an attempt to hold his lead, Garry increases his speed at a rate proportional to 4% of Timmy's speed.

<u>Differential Equations</u>

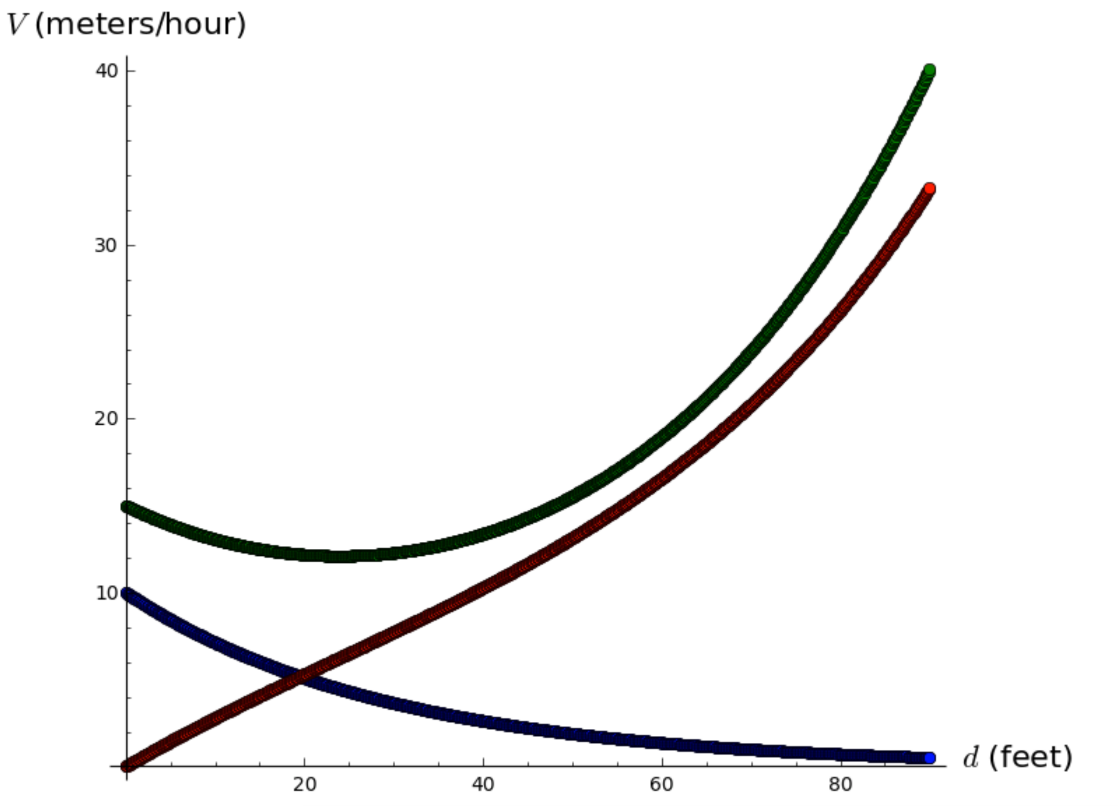
```
Dprime=k*D*(1-D/(b*G))-(a*A*D)
Tprime=z*G
Gprime=-e+(.04*T)
```

Sage Code

```
# Race Speed Analysis

# Starting and ending points, stepsize, and total number of observation points
dstart=0
dfin=90
stepsize=0.1
length=(dfin-dstart)/stepsize+1

# Next, specify the intial speed of Dan; and the growth rate k; initial speed of Garry;initial speed
of Timmy; and constants a, z, and e
D=10
T=0
G=15
a=.001
b=.4
z=.02
e=.25
k=.001
d=dstart

# Next we create lists to store our computed values of d and D and T and G
Dvalues=[]
Tvalues=[]
Gvalues=[]
dvalues=[]



for i in range(length):

    # Store current values
    Dvalues.append(D)
    Tvalues.append(T)
    Gvalues.append(G)
    dvalues.append(d)
```

```
    # Store current values
    Dvalues.append(D)
    Tvalues.append(T)
    Gvalues.append(G)
    dvalues.append(d)

    # Compute rate of change using logistic equation; compute T prime; compute G prime
    Dprime=k*D*(1-D/(b*G))-(a*A*D)
    Tprime=z*G
    Gprime=-e+(.04*T)

    # Net change equals rate of change times stepsize
    DeltaD=Dprime*stepsize
    DeltaT=Tprime*stepsize
    DeltaG=Gprime*stepsize

    # New values equal current values plus net change
    D=D+DeltaD
    T=T+DeltaT
    G=G+DeltaG
    d=d+stepsize
    # Next time through the loop, the above new values play the role of current values

# Zip the d values with the S/I/R values into lists of ordered pairs, and create plots of these
Dplot=list_plot(list(zip(dvalues,Dvalues)),plotjoined=True,marker='o',color='blue')
Tplot=list_plot(list(zip(dvalues,Tvalues)),plotjoined=True,marker='o',color='red')
Gplot=list_plot(list(zip(dvalues,Gvalues)),plotjoined=True,marker='o',color='green')

# Now plot the computed Y G and T values against the corresponding points in the domain
show(Dplot+Tplot+Gplot,axes_labels=['$d$ (feet)','$V$ (meters/hour)'])
```

Graph and Analysis



Garry got his head start, but then began to slow, as predicted. But, once Timmy started to accelerate we see Garry pick his speed up. Timmy has the fastest acceleration, as the slope of his speed is the greatest, but with Garry's head start, Timmy never catches up. His speed never even reaches a higher point than Garry's. Dan, our discouraged snail, lives up to his name. He too had a head start, but it didn't do him much good. As soon as he saw his competitors take off, he decelerated quite rapidly. Timmy Two-Toes will live to race another day, and hopefully get his world record. But for now, Greasy Garry is the low speed champion of the animal kingdom.

Another sample Term Project

The College Student: A system of sleep, coffee, grumpiness and productivity over 16 hours

The real-world scenario that I chose to model was a college student's traits of grumpiness and productivity when those traits are affected by the amount of sleep and coffee the student has had. Since the timeframe being measured is 16 hours, this was intended to be tracing the student during a typical day that this individual was awake. Therefore, the differential equation for sleep decreased linearly with time, as the units being measured are "hours since sleep." The amount of sleep was not influenced by any of the other factors in the system. However, the amount of coffee (caffeine in centigrams) in the bloodstream was a function that was influenced by the other factors of the system. The amount of coffee that this individual drinks is proportional to the level of grumpiness they have and also was modeled by a sin wave. Grumpiness was a measure of the number of times per hour that the college student thought of unhappy, or stressful things. Productivity was measured in the amount of tasks completed per hour. Both grumpiness and productivity are related to the amount of sleep and amount of coffee in the bloodstream. Grumpiness increased proportional to both the amount of sleep and coffee, but also does not reach below 10 unhappy thoughts per hour, which is the base/lowest level of grumpiness for this college student. Productivity increased exponentially as the day went on, but also had a limit to how many things could be completed in an hour. Therefore the upper limit is around 20 tasks per hour, but is also proportional to the amount of grumpiness.

Differential equations that model this system:

U=sleep, C=coffee, P=productivity, G=grumpiness

Uprime=a

Cprime=5*sin(g*t)*(f*G)

Pprime=k*P*(1-(P/(b*G)))

Gprime=((e*C)+(e*U))*(1-(G/d))

Analysis of the Results

The results of this model are flawed because it does not incorporate the quality of sleep that the individual had the previous night or how many hours. There are also many more factors that would influence the three variables that all influence one another. However, the graph is relatively easy to decipher and turned out to model this situation pretty well.

Sage Code

```
# The College Student, a system of sleep, coffee, grumpiness, and productivity over the course of 16 hours of the day.


# First, specify the starting and ending points, stepsize, and total number of observation points

tstart=0

tfin=16

stepsize=0.5

length=(tfin-tstart)/stepsize+1


# Next, specify all the intitial values for the traits in question including the initial amount of time since waking up, U, the initial productivity level, P, the initial amount of coffee in the individual's system, C, and the initial level of grumpiness of the individual, G. Also specify the values of any constants, a, b, d, e, g, k and the initial start t=tstart, which is defined above.

U=16

P=1

C=0

G=25

a=-1

b=2

d=10

e=.2

f=.08

g=.8

k=0.5

t=tstart


# Next we create lists to store our computed values of t, U, P, C, and G.

Uvalues=[]

Pvalues=[]
```

```
Cvalues=[]

Gvalues=[]

tvalues=[]


# The following loop does three things:

# (1) stores the current values of U, P, C, G and t into the lists created above;

# (2) computes the next value of U, P, C, or G using Euler's method;

# (3) increases t by the stepsize


for i in range(length):


    # Store current values for each of the traits we are measuring.

    Uvalues.append(U)

    Pvalues.append(P)

    Cvalues.append(C)

    Gvalues.append(G)

    tvalues.append(t)


    # Compute rate of change for time since sleep, productivity, coffee level in the individual, and grumpiness using logistic equations. Notice that U or time since sleep and C or coffee in the system are values that are not affected by the rates of changes' of the other values, although the values for U and C both influence productivity and grumpiness.

    Uprime=a

    Pprime=k*P*(1-(P/(b*G)))

    Cprime=5*sin(g*t)*(f*G)

    Gprime=((e*C)+(e*U))*(1-(G/d))


    # Net change equals rate of change times stepsize

    DeltaU=Uprime*stepsize

    DeltaP=Pprime*stepsize
```

```
DeltaC=Cprime*stepsize

DeltaG=Gprime*stepsize


# New values equal current values plus net change

U=U+DeltaU

P=P+DeltaP

C=C+DeltaC

G=G+DeltaG

t=t+stepsize

# Next time through the loop, the above new values play the role of current values


# Zip the t values with the U, P, C, and G values into lists of ordered pairs, and create plots of these

Uplot=list_plot(list(zip(tvalues,Uvalues)),plotjoined=True,marker='o',color='purple')

Pplot=list_plot(list(zip(tvalues,Pvalues)),plotjoined=True,marker='o',color='green')

Cplot=list_plot(list(zip(tvalues,Cvalues)),plotjoined=True,marker='o',color='blue')

Gplot=list_plot(list(zip(tvalues,Gvalues)),plotjoined=True,marker='o',color='red')


# Now plot the computed sleep, coffee, grumpiness, and productivity values against the corresponding
points in the domain

show(Cplot+Pplot+Gplot+Uplot,axes_labels=['$t$ (hours)','Percieved Level'])
```

Graph of the System