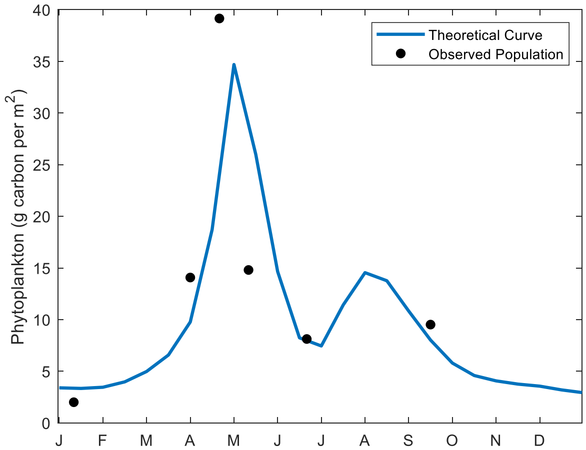
Bethany Fowler | Modeling Problem Set 1

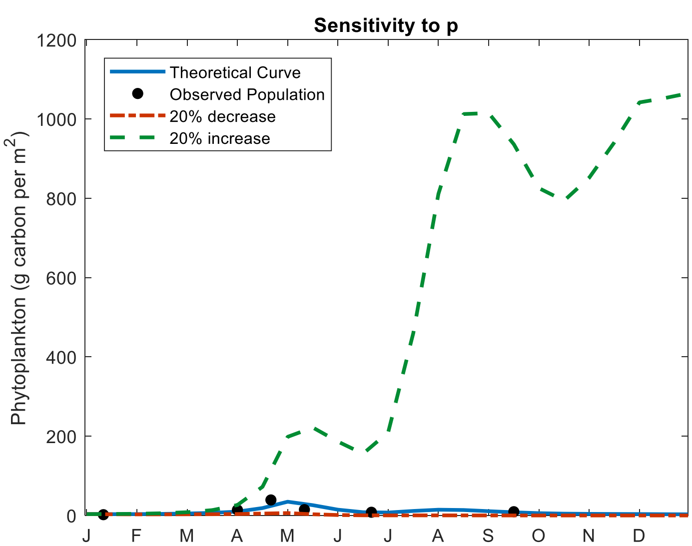
Part 1: Reproduce Figure 21

Code to generate this figure is attached.

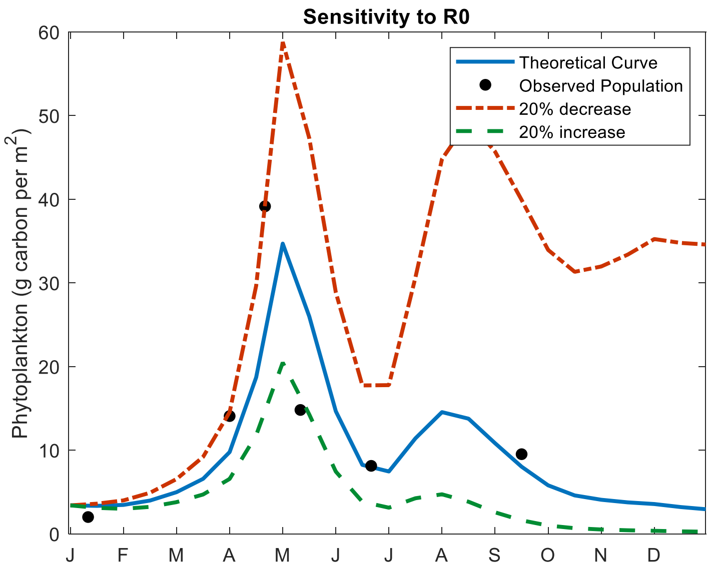


Part 2: Sensitivity Analysis

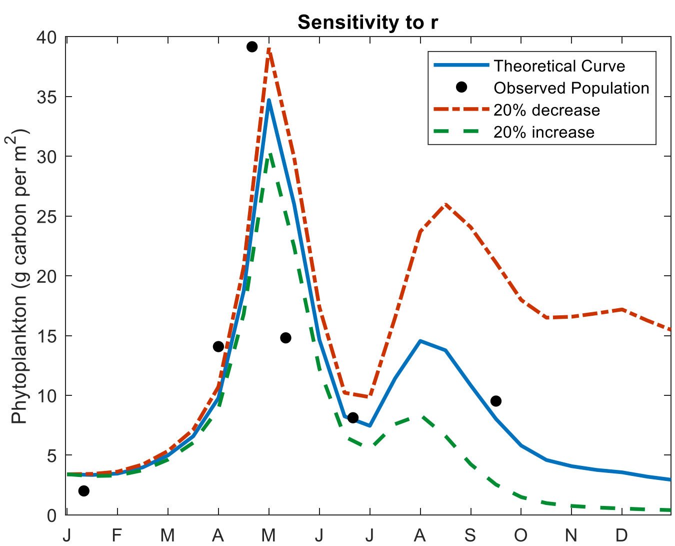
Each plot shows simulation results for the two 20% perturbations to parameters as well as the original curve above. I measure the fit of each simulation by calculating the difference between the simulated values and the observed values across the six samples as a percentage of the observed values. For each parameter, I report the mean value of this percent error for both the 20% increase and decrease scenarios, rounded to the nearest percent. The code for this and the plots is attached.



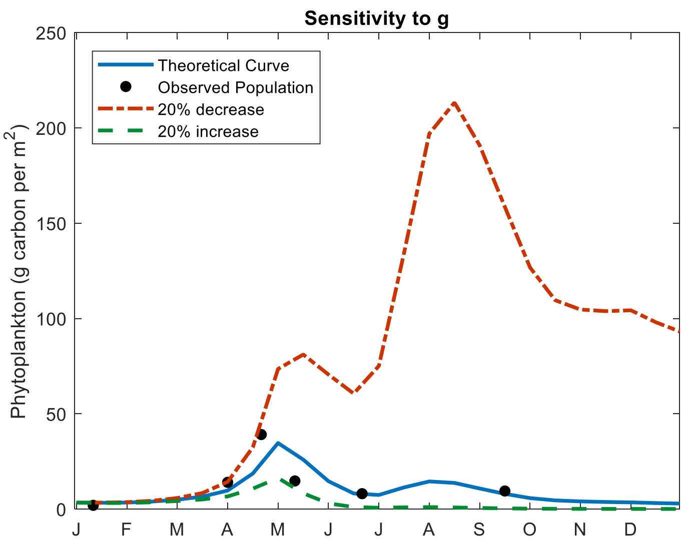
Mean P Error, decreased p = 81%; Mean P Error, increased p = 2,246%



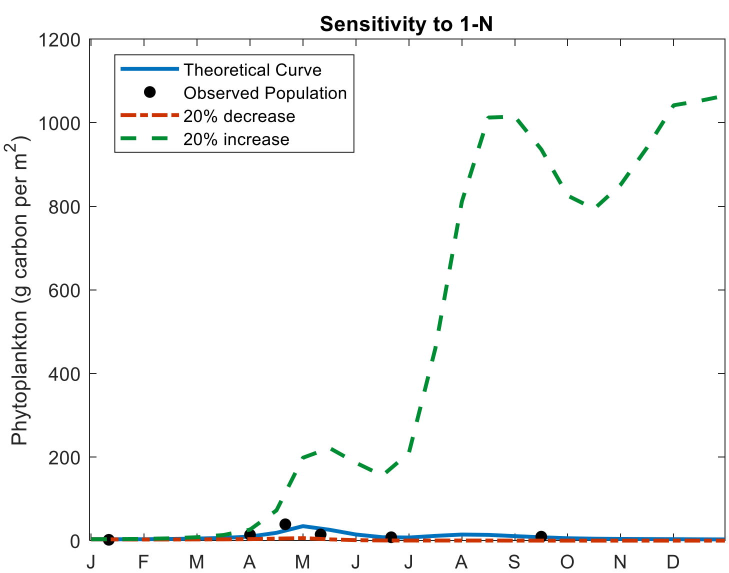
Mean P Error, decreased R0 = 127%;   Mean P Error, increased R0 = 54%



Mean P Error, decreased r = 65%;   Mean P Error, increased r = 52%



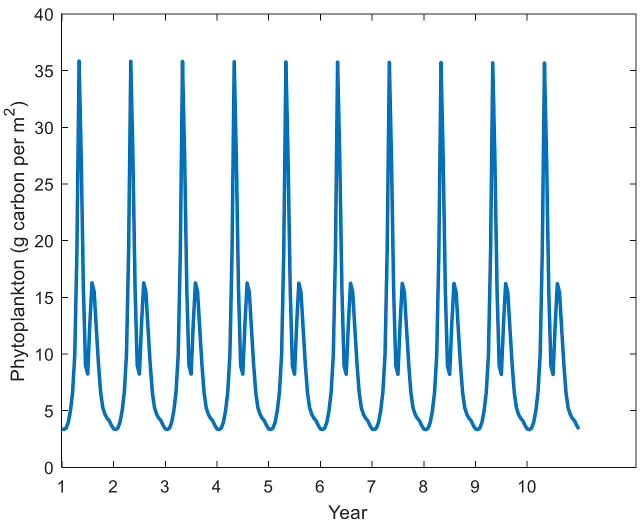
Mean P Error, decreased g = 465% ; Mean P Error, increased g = 66%



Mean P Error, decreased (1-N) = 81%; Mean P Error, increased (1-N) = 2,246%

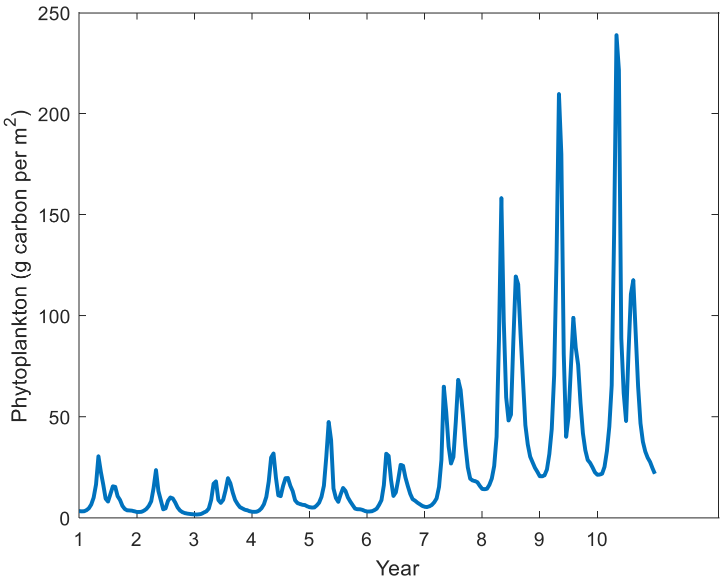
From these simulations, it appears that the model is most sensitive to changes in p and (1-N). Particularly, increasing these values leads to populations of phytoplankton that grow wildly. This makes sense since p is a photosynthetic constant and N is the reduction in photosynthetic rate due to nutrient depletion, so increasing 1-N increases the ability of plankton to grow in low nutrient environments. (Some values of 1-N are now greater than 1, which is all the more unreasonable). The model is also quite sensitive to changes in g. Because g is a grazing rate, changes in g have an inverse effect on the simulated phytoplankton stock. Relative to these parameters, the model is reasonably insensitive to changes in r and R0, which describe the temperature effect on Respiration. Increasing the effect of temperature on respiration leads to decreases in phytoplankton, since more production is lost to respiration, especially at high temperatures. We see that although the model can be well fit to the data, as Riley suggests, a relatively small error in parameter estimation could lead to dramatic errors in predicted Phytoplankton stock – especially for values of p and (1-N).

Part 3: Periodicity

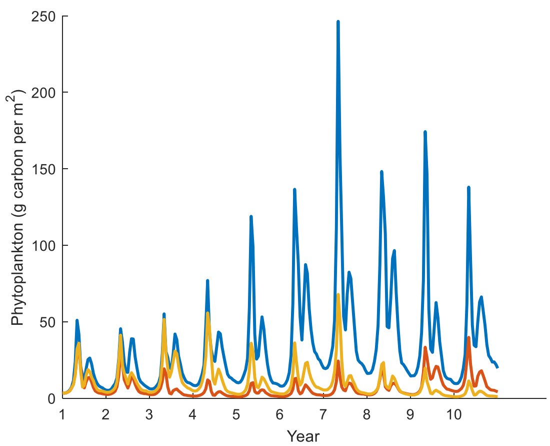


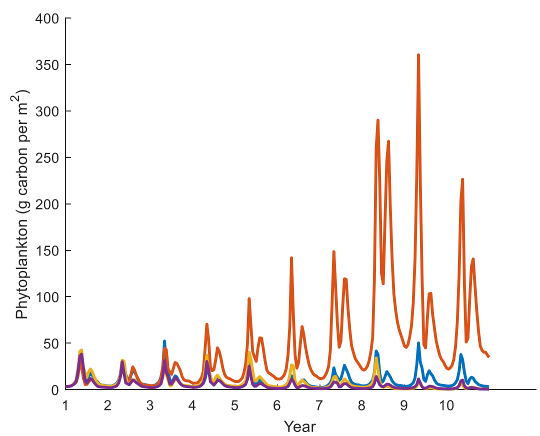
For P to be periodic we need dP/dt to average to 0 over the course of a year, so that the stock returns to its original value. We can generate the periodic solution above by keeping the initial parameters, making g = 0.007435 /day and making the annual inputs periodic (OneMinusN, OneMinusV, k, T, I, Z, z1). Changing the other parameters obviously changes the value of g that makes the solution periodic. As we saw in the sensitivity tests, increasing some parameters has positive effects on phytoplankton growth and increasing others, like g, results in negative effects. So for example, increasing p to 2.6 /day requires us to increase g to 0.0079465 to obtain a periodic solution, because p has a positive effect which must be balanced by a greater negative effect from grazing.

If we add the stochasticity to the Z input as indicated in the assignment we get the following, which is no longer periodic:



Of course, because we have incorporated stochasticity, this result will be different every time. Here are a few more simulations using the same code:





Again, we see the sensitivity of the model to relatively small perturbations. Changing the Zooplankton stock by <20% leads to quite different dynamics, even when environmental conditions are forced to be exactly periodic (which they almost certainly wouldn't be in reality). We also see that by construction the potential for large anomolies is greater in the positive direction, since phytoplankton stock is necessarily bounded below by 0. If the zooplankton stock and consequently grazing rates were allowed to increase in response to P, rather than randomly, we might expect to see fewer simulations where phytoplankton stock attains these superhigh values.

Code for this Problem Set

```
%% define simulation function for Riley model

function [Pvec] = Riley(numsteps, p, R0, r, g, P, I, k,
OneMinusN, OneMinusV, T, Z, z1)

%time step variables
tstep = 15;

%save output vector for P time series
Pvec = P;

for s = 1:numsteps
    %use equation 8 from the paper
    dPdt = P*((p*I(s)/(k(s)*z1(s)))*(1-exp(-
k(s)*z1(s)))*OneMinusN(s)*OneMinusV(s) - R0*exp(r*T(s)) -
g*Z(s));

    %adjust P accordingly
    P = exp(15*dPdt / P) * P;

    %save P to output
    Pvec = [Pvec P];
end


%% Part 1: Run model just like Riley's
clear all

load('Riley_dat.mat')
%I, k, OneMinusN, OneMinusV, T, Z, z1
numsteps = 24; %run for 1 "year"

%initial condition
P = 3.4; %gCm^(-2)

%params
p = 2.5; %/day
R0 = 0.0175; %/day
r = 0.069;  %/degC
g = 0.0075; %/day

[Pvec] = Riley(numsteps, p, R0, r, g, P, I, k, OneMinusN,
OneMinusV, T, Z, z1);
```

```
%generate ~figure 21
plot([1:15:370], Pvec, 'linewidth', 2)
axis([0 360 0 40])
xticks([1:30:370])
xticklabels({'J'; 'F'; 'M'; 'A'; 'M'; 'J'; 'J'; 'A'; 'S'; 'O';
'N'; 'D'})
ylabel('Phytoplankton (g carbon per m^{2})')
hold on
scatter((Obs_time+1), Obs_P, 'black', 'filled')
legend('Theoretical Curve', 'Observed Population')

clear P



%% Part 2: Sensitivity Analysis
%change these for each parameter test
titleval = 'Sensitivity to p'
p1 = .8.*p; % 20% decrease
p2 = 1.2 .*p; % 20% increase

%params
p = 2.5; %/day
R0 = 0.0175; %/day
r = 0.069;  %/degC
g = 0.0075; %/day


p = p1 %Change this
P = 3.4; %gCm^(-2)

[Pvec1] = Riley(numsteps, p, R0, r, g, P, I, k, OneMinusN,
OneMinusV, T, Z, z1);

hold on
%add simulation to graph
plot([1:15:370], Pvec1, '-.', 'color', [.8 .2 0], 'linewidth',
2)

clear P;
% --- simulate again with new param---

p = p2; %Change this
P = 3.4; %gCm^(-2)
```

```
[Pvec2] = Riley(numsteps, p, R0, r, g, P, I, k, OneMinusN,
OneMinusV, T, Z, z1);

hold on
%add simulation to graph
plot([1:15:370], Pvec2,'--', 'color', [0 .55 .2], 'linewidth',
2)
title(titleval)
legend('Theoretical Curve', 'Observed Population', '20%
decrease', '20% increase')
axis('auto')

%% Part 3: Calculate Error

%first interpolate simulation so we have points at same time as
data

X1 = interp1([1:15:370], Pvec1, (Obs_time+1), 'linear');
pErr = abs(X1 - Obs_P) ./ Obs_P * 100;
X2 = interp1([1:15:370], Pvec2, (Obs_time+1), 'linear');
pErr2 = abs(X2 - Obs_P) ./ Obs_P * 100;

meanpErr1 = mean(pErr)
meanpErr2 = mean(pErr2)



%% Part 4: When it is periodic
clear all

load('Riley_dat.mat')
%I, k, OneMinusN, OneMinusV, T, Z, z1

%change inputs so they are long enough for 10 year simulation?
OneMinusN = repmat(OneMinusN, 10);
OneMinusN = OneMinusN(:,1);
OneMinusV = repmat(OneMinusV, 10);
OneMinusV = OneMinusV(:,1);
T = repmat(T, 10);
T = T(:,1);
Z = repmat(Z, 10);
Z = Z(:,1);
z1 = repmat(z1, 10);
z1 = z1(:,1);
k = repmat(k, 10);
k = k(:,1);
I = repmat(I, 10);
```

```
I = I(:,1);

%now add randomness to Z
for i = 1:length(Z)
    Z(i) = Z(i)*(.8 + 0.4*rand);
end


numsteps = 240;
%initial condition
P = 3.4; %gCm^(-2)
%params
p = 2.5; %/day
R0 = 0.0175; %/day
r = 0.069;  %/degC
g = 0.007435; %/day

[Pvec] = Riley(numsteps, p, R0, r, g, P, I, k, OneMinusN,
OneMinusV, T, Z, z1);

hold on
%generate ~figure 21
plot([1:15:3614], Pvec, 'linewidth', 2)
axis('auto')
ylabel('Phytoplankton (g carbon per m^{2})')
hold on
xticks([1:360:3600])
xticklabels([1:10])
xlabel('Year')
clear P Pvec
```