Exploring Gonze and Goldbeter's model of the *FRQ* circadian clock in Neurospora with MATLAB

Charlie Ginn, Izaak Cole Wallace, Maddie Bailey-Moyes,
Simraj Sangha, Tyler Buckley, Miguel Alburo

November 2023

Introduction

Circadian rhythms describe the inherent 24 hour cycle within organisms that regulate biological processes throughout the day. Often entrained by day-night cycles, circadian clocks therefore have the ability to adapt to the dynamic environment. In Neurospora *crassa*, the circadian rhythm is controlled by an autoregulatory negative feedback loop of *frq* mRNA . Gene *frq* is transcribed into mRNA in response to the increasing light of day. As time passes, *frq* mRNA is translated into the *FRQ* protein which inhibits mRNA from being transcribed and thus the system eventually resets for the next cycle. The simple mechanism in which circadian rhythms in Neurospora are controlled is ideal for modelling mathematically. In 1999, Didier Gonze and Albert Goldbeter came up with one such model shown in the following system of equations.

$$\frac{dM}{dt} = v_s \frac{K_I^n}{K_I^n + F_N^n} - v_m \frac{M}{K_m + M} \tag{1}$$

$$\frac{dF_C}{dt} = k_s M - v_d \frac{F_C}{K_d + F_C} - k_1 F_C + k_2 F_N \tag{2}$$

$$\frac{dF_N}{dt} = k_1 F_C - k_2 F_N \tag{3}$$

Here, M denotes the concentrations of *frq* mRNA; and F_C and F_N denote the concentrations of the cytosolic and nuclear forms of the protein *FRQ* respectively. Parameter v_s denotes the rate of *frq* transcription which increases with light. In this report, we will closely examine the behaviour of the system after forcing with dynamic v_s values corresponding to different light-dark (LD) cycles. All other parameters will be the same constants used by Gonze and Glodbeter in their paper. K_I relates to the threshold beyond which nuclear *FRQ* represses frequency transcription. v_m and v_d are the rates of *frq* mRNA and *FRQ* degredation. The assumed first-order rate constants: k_s measures the rate of *frq* mRNA translation into *FRQ*; k_1 and k_2 characterise *FRQ* transport in and out of the nucleus. The Michaelis constants are K_m and K_d, relate to v_m and v_d, and the degredation of mRNA and protein respectively. The Hill coefficient n characterises the degree of cooperativity of the repression process.

1 Coding and running the model

In order to use MATLAB to construct plots of the oscillatory dynamics, we first had to create a function which evaluated the system of equations. We defined the global variables in the `Globals.m` script, then created the function file evaluating equations (1), (2) and (3) shown by the following code:

```
f(1)= vs.*((KI .^ n)/(KI .^n + X(3).^ n))- vm .*(X(1)/(Km+X(1)));  % Eqn (1)

f(2)= Ks.*X(1)-vd*(X(2)/(Kd+X(2)))-k1*X(2)+k2*X(3); % Eqn (2)

f(3)= k1.*X(2)-k2.*X(3); % Eqn (3)
```

Once this was done, we used the function `ode45` which numerically integrates the this function with respect to time, and outputs an array or solution showing the state of the system at chronological points in time. We then plotted the solutions in the following graphs for the v_s value which represented constant light and dark conditions analogous to daylight and night intensities.

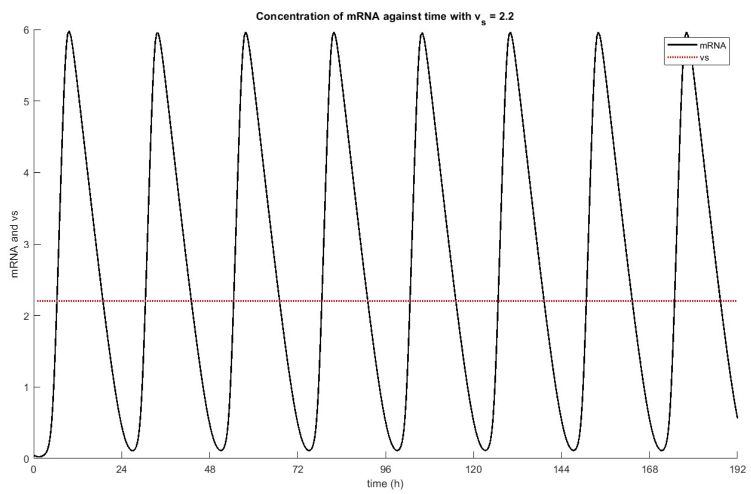

Figure 1: Behaviour of the system under constant light.

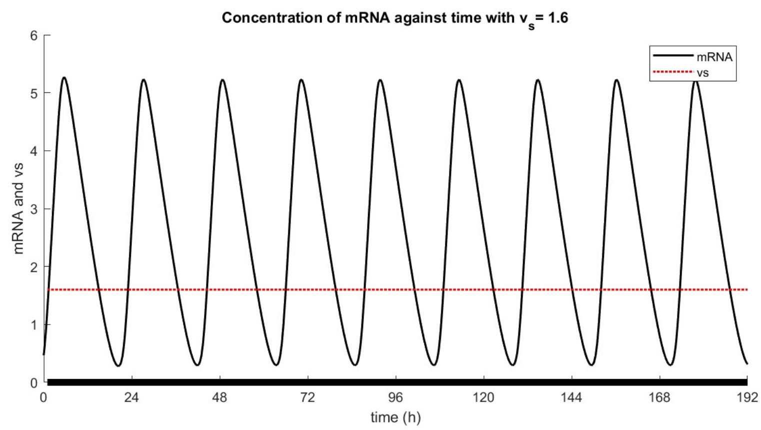

Figure 2: Behaviour of the system under constant dark.

Comparing the two graphs from Fig.1, Fig.2 we can see that they both oscillate uniformly with a period of just under 24 hours, but the system that is in constant light has a higher amplitude and period. In the light system, mRNA concentration oscillates between 0.1 and 6 as opposed to between 0.4 and 5.2 in the dark system. This result shows that *frq* mRNA concentrations will still cycle roughly every 24 hours and the circadian rhythm is maintained, regardless if the organism is exposed to light or not. In fact, what you will see in the following sections is that the system can very easily become chaotic particularly for non-circadian LD cycles, and v_s values depicting unnaturally intense light.

2 Piecewise constant forcing

Now we need to understand how the system will operate when forced under the standard 12 hour LD cycles. To split the 24 hour period into 12 hours of light and 12 hours of dark we modelled v_s as a square wave with minimum $v_{s,min} = 1.6$ and maximum $v_{s,max} = 2.25$. These were done in the files `GGsquare.m` and `CircadianFun1.m` with this additional code allowing for forcing of the v_s square wave. We then plotted these results in another time graph.

```
if mod(t - 1,24) > 12
    vs=vsmin;
elseif mod(t - 1,24) <= 12
    vs=vsmax;
end
```

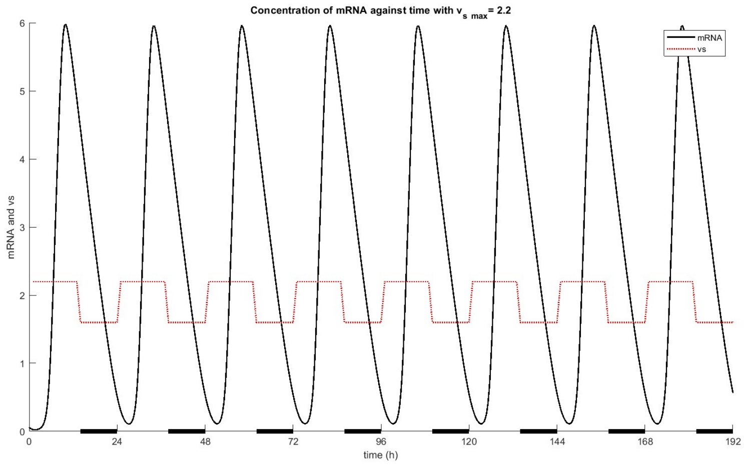

Figure 3: System under 12 hour square wave LD cycles with $v_{s,max} = 2.25$

In this new LD cycle, the system produces circadian oscillations as expected (Fig. 3). Furthermore, after long periods of time we observed that systems with different initial conditions will eventually shift to the same phase with a perfect 24 hour rhythm implying entrainment by the LD cycle. Under the same 12 hour LD cycle but now with $v_{s,max} = 3.1$ revealed very different behaviour as seen in the figure below (Fig 4). The oscillations are now chaotic, with no pattern in amplitude and period.

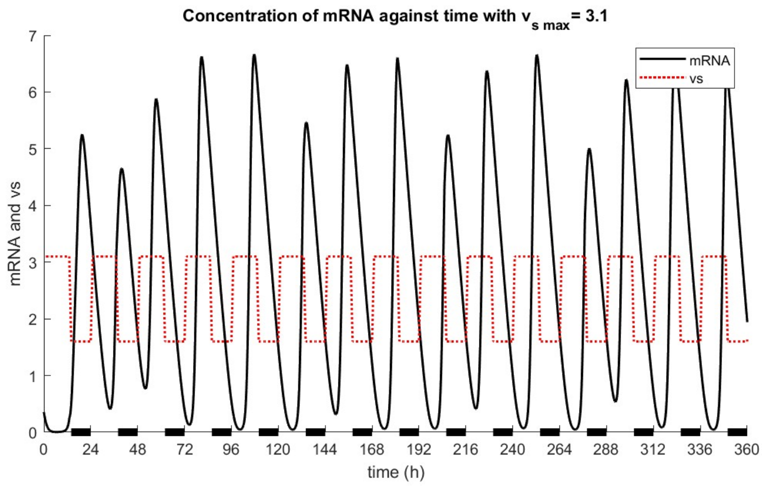

Figure 4: System under 12 hour square wave LD cycles with $v_{v,max} = 3.1$

3 Continuous time forcing

Many lab experiments have historically modelled the LD cycle of day and night using the square wave as was done in the previous section. In the natural world, however, light intensity will gradually increase as the day starts and slowly decrease coming to night. Thus it is more realistic to model LD cycles as sinusoidal rather than square wave (Eq. 4). To do this we adapted our previous code so that v_s changes sinusoidally between $v_{s,min}$ ans $v_{s,max}$ in 24 hour periods and reaches its maximum at 6 hours and minimum at 18 hours (File `GGsine.m` and `CircadianFun2`)

$$v_s(t) = \frac{v_{s,max} - v_{s,min}}{2} \sin(\frac{\pi}{12}t) + \frac{v_{s,max} + v_{v,min}}{2} \tag{4}$$

```
vs=0.5*(vsmax-vsmin)*sin((t*pi)/12)+(vsmax+vsmin)/2;
```

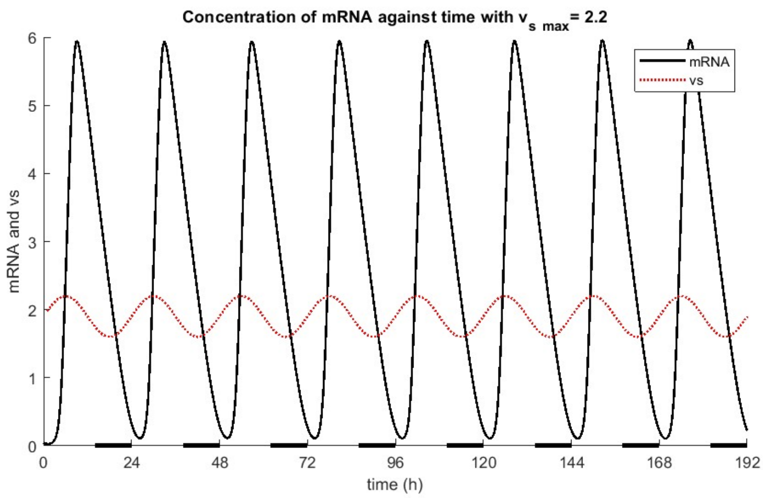

Figure 5: System under sinusoidal LD cycles with $v_{s,max} = 2.25$

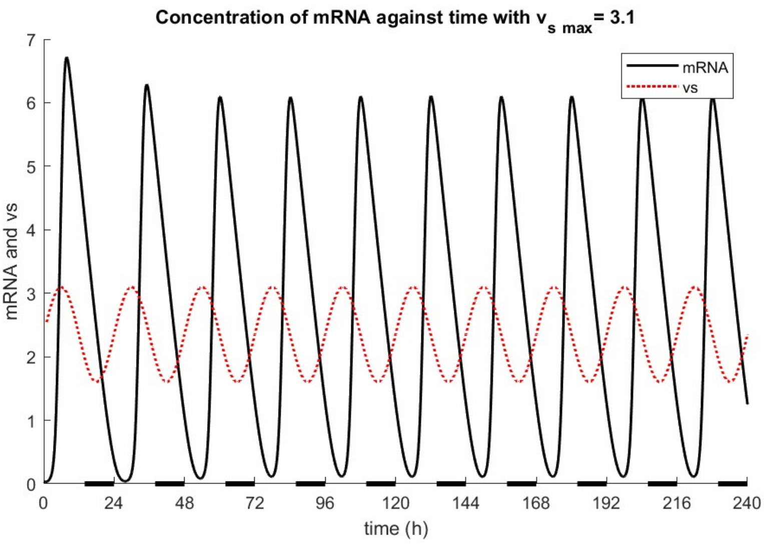

Figure 6: System under sinusoidal LD cycles with $v_{s,max} = 3.1$

Figure 6 is not as chaotic as figure 5, its amplitude varies in the first 72 hours then have a uniform amplitude after this. Looks similar to Figure 5 with $v_{s\ max} = 2.2$, especially after the first 72 hours once it has settled.

4 Effect of the amplitude of continuous time forcing

Here we will be looking at the effect of increasing the amplitude of continuous time forcing by simulating the system with sinusoidal forcing. We will use a range of values of $v_{s,max}$ while keeping $v_{s,min}$ fixed at 1.6. The code used is very similar to the previous questions. Note that the lines for v_s have been removed from the plot for legibility.

Figure 7 is for $v_{s,max} \in [2.1, 8]$ this interval approaches a periodic regime with 24 hour period after around 168 hours.

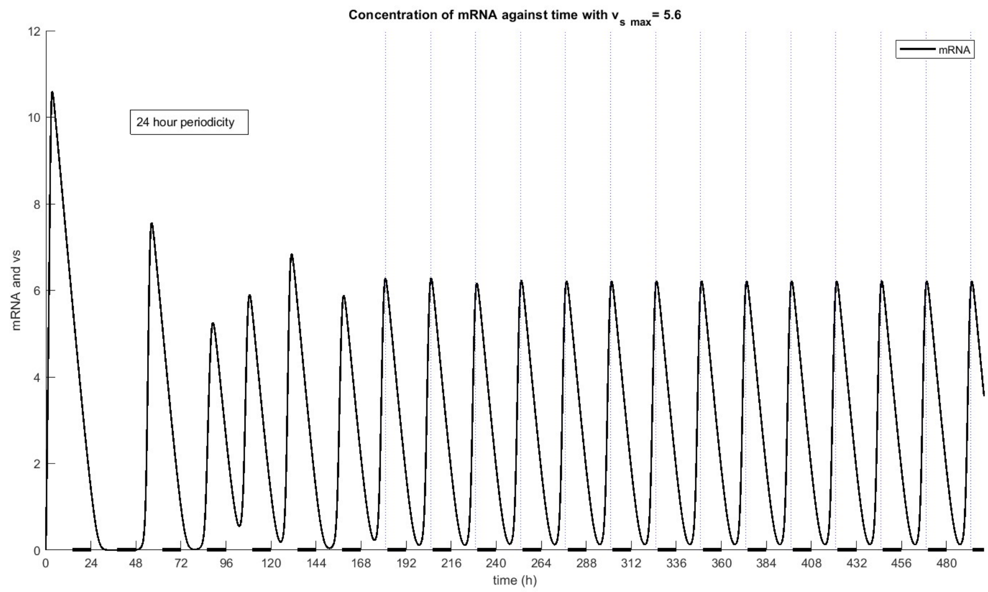

Figure 7: Varied light and dark

From Figure 7 it is clear that with value of v_s oscillating between 1.6 and [2.1,8] (in this case 5.6), the system approaches a limit cycle with a 24 hour period. This is highlighted by the vertical lines at 24 hour intervals. Figure 8 utilises a value of $v_{s,max} \in [9.7, 10.7]$ and behaves interestingly. After roughly 200 hours the system settles in a 72 hour period. Within this period there are 3 different maxima, chaotic behaviour is starting to appear.

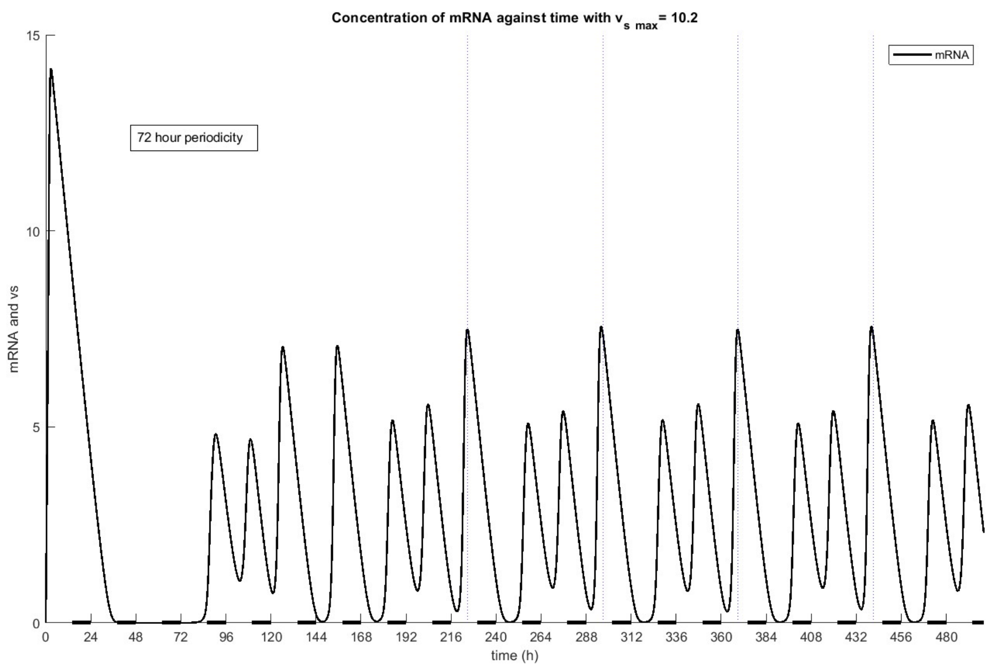

Figure 8: Varied light and dark

Figure 9 utilises a value of $v_{s,max} \in [19, 20]$

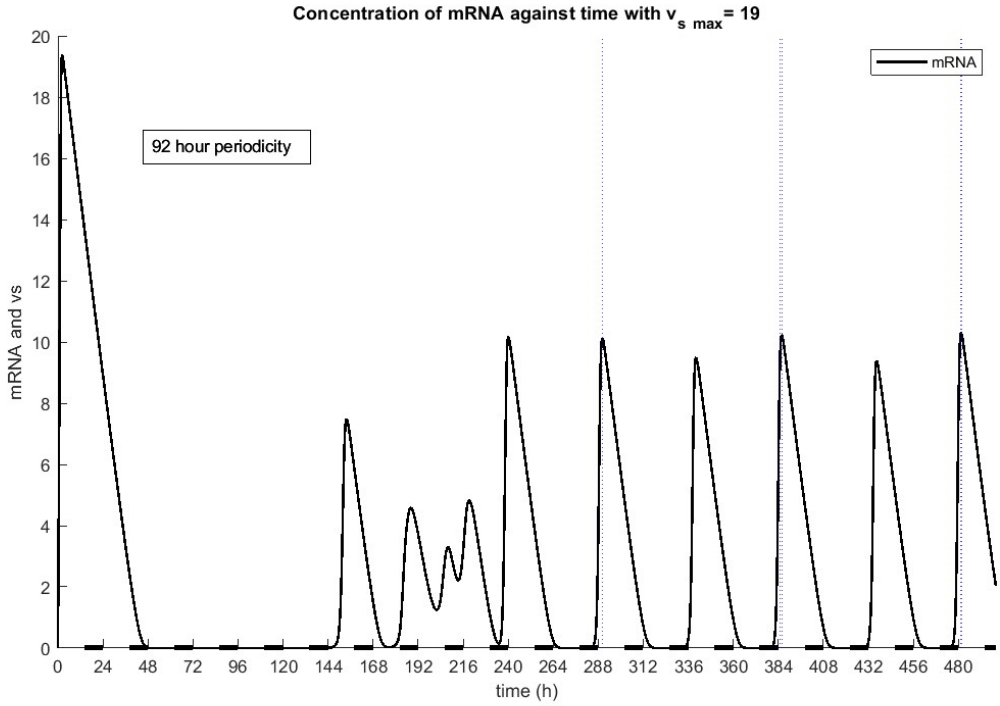

Figure 9: Varied light and dark

For a value of $v_{s\ max}$ between [19,20], the system oscillates unpredictably before settling into the 96 hour period. Different values of $v_{s\ max}$ were tested and they each settle add different speeds, this is due to the chaotic nature of the system with such high variation in v_s.

5 Conclusion

In this report we saw a range of interesting behaviour exhibited by Gonze and Goldbeter's model for Neurospora's circadian rhythm system. From quasiperiodic oscillations that occurred in constant dark and light conditions; to entrainment caused by square and sinusoidal LD cycles which would help explain how circadian rhythms are synchronised to the time of day. We also found how incredibly high $v_{s,max}$ values would still eventually lead to periodic oscillations with 3 and 4 day periods. This aligned with Gonze and Goldbeter's hypothesis (Figure 7 of their paper) that predicted an entrained system regardless of increasing $v_{s,max}$ if the LD cycles were completely sinusoidal. This project was particularly useful as it gives an insight into why circadian rhythms adapt so well in response to changes in the environment. Though mathematically focused, use of these models could also be of interest to a plant scientist trying to optimise LD cycles for maximum growth; showing the potential for real-world application.

A References

[1]Gonze D, Goldbeter A. Entrainment versus chaos in a model for a circadian oscillator driven by light-dark cycles. Journal of Statistical Physics. 2000 Oct;101:649-63.

B MATLAB Codes

B.1 Global.m

To reduce troubleshooting, a function that initialises all the global variables ensures consistency.

```
global n vm vd KI Km Ks Kd k1 k2 range vsmax vsmin

n = 4;
range = 300;
vsmin = 1.6;
vsmax = 2.25;
vm = 0.505;
vd = 1.4;
KI = 1;
Km = 0.5;
Ks = 0.5;
Kd = 0.13;
k1 = 0.5;
k2 = 0.6;
```

B.2 Light and dark

The code of the light and dark case takes one value of v_s of 1.6 and 2.2 for the dark and light case respectively. The MATLAB function `Light_dark.m` takes user input for the value of v_s, modelling the circadian rhythm in the two extremes.

Light_dark.m

```
global vs
vs = input("What is the value of vs? ");

% code to stop values other than 1.6 or 2.2
if vs ~= 1.6 && vs ~=2.2
    print("Please choose 1.6 or 2.2 for the dark and light case")
    quit
end

% Starting variables
range = 300;
X0 = zeros(3,1);
tspan = [0 range];

% Using ODE45 to solve the partial differential equations
options = odeset('RelTol',1e-5,'AbsTol',1e-7);
[t,X] = ode45(@fun, tspan, X0);

% Initialise the plot variables
vsplot = zeros(range,1);
tsize = zeros(range,1);
axisline = zeros(range,1);

% Initialise the variables for the three functions
M = X(:,1);
Fc = X(:,2);
Fn = X(:,3);

% Couting through every time step initialising the plot of vs
for i = 1:range
   tsize(i, 1) = i;
   vsplot(i, 1) = vs;
end
% The visual axis identify that shows if it is light or dark
for j=1:range
    tsize(j, 1) = j;
```

```
    if vs == 1.6
        axisline(j, 1) = 0;
    elseif vs == 2.2
        axisline(j,1) = NaN;
    end
end
% Plotting results
hold on
% Skipping 24 hours whilst the system settles to its 24 hour periodicity
% similar to how is done in Gonze and Goldbeter
plot(t - 24, M, 'k', 'LineWidth', 1.5)
plot(tsize, vsplot, 'r:', 'LineWidth', 1.5)
plot(tsize, axisline, 'k', 'LineWidth', 5)
% 24 hour step itervals matching the Gonze and Goldbeter paper
xticks( 0 :24 : 192);
xlim([0 192])
title(['Concentration of mRNA against time with v_s = ' num2str(vs)])
xlabel('time (h)')
ylabel('mRNA and vs')
legend('mRNA', 'vs')
hold off
function f = fun(~,X)

global vs n vm vd KI Km Ks Kd k1 k2

f = zeros(3,1);
f(1)=vs*((KI^n)/(KI^n+X(3)^n))-vm*(X(1)/(Km+X(1))); %f(1) is dMdt

f(2)=Ks*X(1)-vd*(X(2)/(Kd+X(2)))-k1*X(2)+k2*X(3); %f(2) is dFcdt

f(3)=k1*X(2)-k2*X(3); %f(3) is dFndt
end
```

To output the plot seen in Figure 2 of Gonze and Goldbeter's paper for the complete dark case, where $v_s = 1.6$ type this into the command window:

`Light_dark` the program with request a value for v_s, inputting 1.6 will produce the dark case. For the constant light case typing `Light_dark` then inputting `2.2` will result in a plot of the light case.

B.3 $v_{s\ max}$ oscillating with a square wave

The square wave is plotted using two files. CircadianFun1 contains the 3 differential equations (equation (1)) and the formula for v_s. GGsquare solves the equations.

CircadianFun1.m code:

```
function f = CircadianFun1(t,X)

% Initialise the function for the three differential equations
f = zeros(3,1);

global n vs vm vd KI Km Ks Kd k1 k2 vsmin vsmax

%calculation of v_s for a square wave, with since MATLAB has the first
%index as 1 rather than 0

if mod(t - 1,24) > 12
    vs=vsmin;
elseif mod(t - 1,24) <= 12
```

```
    vs=vsmax;
end

f(1)= vs.*((KI .^ n)/(KI .^n + X(3).^ n))- vm .*(X(1)/(Km+X(1))); %f(1) is dMdt

f(2)= Ks.*X(1)-vd*(X(2)/(Kd+X(2)))-k1*X(2)+k2*X(3); %f(2) is dFcdt

f(3)= k1.*X(2)-k2.*X(3); %f(3) is dFndt

end
```

GGsquare.m code:

```
% Initialise variables globally over multiple functions
global n vm vd KI Km Ks Kd k1 k2 range vsmax vsmin plotrange

n = 4;
range = 500;
plotrange = 360;
vsmin = 1.6;
vsmax = 3.1;
vm = 0.505;
vd = 1.4;
KI = 1;
Km = 0.5;
Ks = 0.5;
Kd = 0.13;
k1 = 0.5;
k2 = 0.6;


% Initialise the plot variables
vsplot = zeros(range,1);
tsize = zeros(range,1);
axisline = zeros(range,1);

% Starting variables
X0 = rand(3,1);
tspan = [0 range];

% Using ODE45 to solve the partial differential equations
options = odeset('RelTol',1e-5,'AbsTol',1e-5);
[t,X] = ode45(@CircadianFun1, tspan, X0);

% Initialise the variables for the three functions
M = X(:,1);
Fc = X(:,2);
Fn = X(:,3);

% Loop that varies vs in a square function every 12 hours having a full loop
% every 24 hours
for i = 1:range
   tsize(i, 1) = i;
    if mod(i - 1 , 24) > 12
        vsplot(i, 1) = vsmin;
    elseif mod(i - 1, 24) <= 12
        vsplot(i,1) = vsmax;
```

```
    end
end

% Visual axis indicator, the -1 is due to MATLAB having the first index as
% 1 not 0
for j=1:range
    tsize(j, 1) = j;
    if mod(j - 1 ,24) > 12
        axisline(j, 1) = 0;
    elseif mod(j - 1, 24) <= 12
        axisline(j,1) = NaN;
    end
end

% Plotting results
hold on
plot(t - 24, M, 'k', 'LineWidth', 1.5)
plot(tsize, vsplot, 'r:', 'LineWidth', 1.5)
plot(tsize, axisline, 'k', 'LineWidth', 5)
% 24 hour step itervals matching the Gonze and Goldbeter paper
xticks( 0 :24 : plotrange);
xlim([0 plotrange])
title(['Concentration of mRNA against time with v_s_ max= ' num2str(vsmax)])
xlabel('time (h)')
ylabel('mRNA and vs')
legend('mRNA', 'vs')
hold off
```

B.4 $v_{s\ max}$ oscillating with a sinusoidal wave

The sinusoidal wave is plotted using two files. CircadianFun2 contains the 3 differential equations (equation (1)) and the formula for v_s. GGsine solves the equations.

CircadianFun2.m code:

```
function f = CircadianFun2(t,X)

% Initialise the function for the three differential equations
f = zeros(3,1);

global n vs vm vd KI Km Ks Kd k1 k2 vsmin vsmax

%calculation of v_s for a sinusoidal wave

vs=0.5*(vsmax-vsmin)*sin((t*pi)/12)+(vsmax+vsmin)/2;

f(1)=vs*((KI^n)/(KI^n+X(3)^n))-vm*(X(1)/(Km+X(1))); %f(1) is dMdt

f(2)=Ks*X(1)-vd*(X(2)/(Kd+X(2)))-k1*X(2)+k2*X(3); %f(2) is dFcdt

f(3)=k1*X(2)-k2*X(3); %f(3) is dFndt

end
```

GGsine.m code:

```
% Initialise variables globally over multiple functions
global  range vsmax vsmin plotrange

range = 250;
vsmin = 1.6;
vsmax = 2.2;

% Initialise the plot variables
vsplot = zeros(range,1);
tsize = zeros(range,1);
axisline = zeros(range,1);

% Starting variables
X0 = rand(3,1);
tspan = [0 range];

% Using ODE45 to solve the partial differential equations
options = odeset('RelTol',1e-5,'AbsTol',1e-5);
[t,X] = ode45(@CircadianFun2, tspan, X0);

% Initialise the variables for the three functions
M = X(:,1);
Fc = X(:,2);
Fn = X(:,3);

% Sine funstion that has maximum and minimum v_s at 6 ad 18 hours
% respectively
for k=1:range
    tsize(k,1)=k;
    vsplot(k,1)=0.5*(vsmax-vsmin)*sin((tsize(k,1)*pi)/12)+(vsmax+vsmin)/2;
end

% Visual axis indicator, the -1 is due to MATLAB having the first index as
% 1 not 0
for j=1:range
    tsize(j, 1) = j;
    if mod(j - 1 ,24) > 12
        axisline(j, 1) = 0;
    elseif mod(j - 1, 24) <= 12
        axisline(j,1) = NaN;
    end
end

% Plotting results
hold on
plot(t -24, M, 'k', 'LineWidth', 1.5)
plot(tsize, vsplot, 'r:', 'LineWidth', 1.5)
plot(tsize, axisline, 'k', 'LineWidth', 3)
xticks( 0 :24 : 192);
xlim([0 192])
%% 24 hour step itervals matching the Gonze and Goldbeter paper
title(['Concentration of mRNA against time with v_s _max= ' num2str(vsmax)])
xlabel('time (h)')
ylabel('mRNA and vs')
legend('mRNA', 'vs')
hold off
```

GGsinelong.m

```
% Initialise variables globally over multiple functions
global range vsmax vsmin

val = input("Which part do you wish to solve? Part 1 , 2 or 3: ");
if val == 1
    vsmax = 5.6;
elseif val == 2
    vsmax = 10.2;
elseif val == 3
    vsmax = 19;
else
    print("Question 4 has only three parts, please try again. Try 1 , 2 or 3 this time")
    return
end
range = 500;

% Initialise the plot variables
vsplot = zeros(range,1);
tsize = zeros(range,1);
axisline = zeros(range,1);

% Starting variables
X0 = zeros(3,1);
tspan = [0 range];
% Using ODE45 to solve the partial differential equations
options = odeset('RelTol',1e-5,'AbsTol',1e-5);
[t,X] = ode45(@CircadianFun2, tspan, X0);
% Initialise the variables for the three functions
M = X(:,1);
Fc = X(:,2);
Fn = X(:,3);

% Sine funstion that has maximum and minimum v_s at 6 ad 18 hours
% respectively
for k=1:range
    tsize(k,1)=k;
    vsplot(k,1)=0.5*(vsmax-vsmin)*sin((tsize(k,1)*pi)/12)+(vsmax+vsmin)/2;
end

% Visual axis indicator, the -1 is due to MATLAB having the first index as
% 1 not 0
for j=1:range
    tsize(j, 1) = j;
    if mod(j - 1 ,24) > 12
        axisline(j, 1) = 0;
    elseif mod(j - 1, 24) <= 12
        axisline(j,1) = NaN;
    end
end

% Plotting results
hold on
plot(t, M, 'k', 'LineWidth', 1.5)
plot(tsize, axisline, 'k', 'LineWidth', 3)
xticks( 0 :24 : 500);
xlim([0 500])
```

```
if val == 1
% Visual line to show 24 hour periodicity for v_s = 5.6
xline(181 : 24: length(M),'b:');
annotation('textbox',[0.2 0.5 0.3 0.3],'String',"24 hour periodicity",'FitBoxToText','on');
elseif val == 2
% Visual line to show 72 hour periodicity for v_s = 10.2
xline(225 : 72: length(M),'b:')
annotation('textbox',[0.2 0.5 0.3 0.3],'String',"72 hour periodicity",'FitBoxToText','on');
elseif val == 3
% Visual line to show 92 hour periodicity for v_s = 19.1
xline(290 : 96: length(M),'b:');
annotation('textbox',[0.2 0.5 0.3 0.3],'String',"92 hour periodicity",'FitBoxToText','on');
end


% 24 hour step itervals matching the Gonze and Goldbeter paper
title(['Concentration of mRNA against time with v_s _max= ' num2str(vsmax)])
xlabel('time (h)')
ylabel('mRNA and vs')
legend('mRNA')
hold off
```

C Command window input

C.1 Question 1

In question 1, we plot two graphs with v_s being either 1.6 or 2.2. To get both graphs copy and paste each paragraph of code. We begin by initialising global variables reducing troubleshooting.

```
%% Iitialise the global variables %%
Global
%% The light case %%
Light_dark
```

This will make the computer send a prompt `What is the value of vs?`. For the light case we type 2.2 this will produce the light graph.

```
2.2
%% press any button to clear the graph fo the next question %%
pause
close
```

```
%% The dark case %%
Light_dark
```

This will make the computer send a prompt `What is the value of vs?`. For the dark case we type 21.6 this will produce the dark graph, seen in Gonze and Goldbeter's paper .

```
1.6
%% press any button to clear the graph fo the next question %%
pause
close
```

C.2 Question 2

Copying and pasting this into the command window will produce a graph where v_s oscillates every 12 hours between 1.6 and 2.2.

```
%% Initialise the global variables %%
Global
%% The square alternating light and dark case %%
```

```
GGsquare
%% press any button to clear the graph fo the next question %%
pause
close
```

C.3 Question 3

Copying and pasting this into the command window will produce a graph where v_s oscillates between a minima of 1.6 and maxima of 2.2 in a sinusoidal manner with peaks and troughs at 6 and 18 hours respectively.

```
%% Iitialise the global variables %%
Global
%% The sinusodial light and dark oscillation %%
GGsine
%% press any button to clear the graph fo the next question %%
pause
close
```

C.4 Question 4

Expanding upon question 3, we push the boundaries of $v_{s\ max}$, were periodic regimes appear. Copy and paste each paragraph individually to produce all three plot.

```
%% Question 4 part one, 24 hour periodicity for vs = [2.1,8] %%
GGsinelong
```

You will be prompted with "`Which part do you wish to solve?  Part 1 , 2 or 3:  `".

```
1
pause
close
```

Now for part 2 with 72 hour periodicity.

```
%% 72 hour periodicity for vs = [9.7,10.7] %%
GGsinelong
```

You will be prompted with "`Which part do you wish to solve?  Part 1 , 2 or 3:  `".

```
2
pause
close
```

Now for part 2 with 96 hour periodicity.

```
%% 96 hour periodicity for vs = [19,20] %%
GGsinelong
```

You will be prompted with "`Which part do you wish to solve?  Part 1 , 2 or 3:  `".

```
3
pause
close
```