**Group 4B Codes**

BeckerWeinmann.m

%molecular level, mammalian, mechanistic ode

%% parameter changes

%VB1 can be used as the forcing term

%use 'L' as a function of time, multiply ftransPC

%check other functions see similarity?

%goodwin very abstract, is it sufficient

%BW far more sufficient/ less abstract

%use of BW we can directly affect the clock with other stimulus

%look for discussions on the papers

%% Inital conditions

F0=[0.273;0.347;1.318;1.119;0.830;1.469;1.092];

tspan=[0,96];

N=100000;

vb1=@(t) (9+3\*sin(2\*pi\*t/24));

%% RK4 / Forward euler function

[t,fBW]=BWfunc(vb1,N,tspan,F0);

F=[fBW(1,:);fBW(3,:);fBW(4,:);fBW(7,:);fBW(5,:)+fBW(6,:)+fBW(7,:)];

%%

hold on

plot(t,F), grid on

title('Becker Weinmann Model')

xlabel('Time (hours)')

ylabel('mRNA and Protein Concentration [nM]')

legend({'Cry mRNA','PER2/CRY','Bmall mRNA','BMAL1\*','BMAL1'},'Location','northeast')

xlim([tspan(2)-96,tspan(2)])

BeckerWeinmann3eq.m

%molecular level, mammalian, mechanistic ode

kd=[0.12,0.05,0.12,0.75,0.06,0.12,0.09];

kb=[1,0.3,NaN,3.6,0.24];

kt=[NaN,0.24,0.02,NaN,0.45,0.06];

ka=[NaN,NaN,NaN,NaN,NaN,0.09,0.003];

vb=[9,NaN,NaN,3.6];

p=8;

q=2;

c=0.01;

r=3;

kli=0.56;

%ftransper2cry = (vb(1)\*(y(7)+c))/(kb(1)\*(1+((y(3)/kli)^p))+(y(7)+c))

% ftransbmall = (vb(4)\*y(3)^r)/((kb(4)^r)+(y(3)^r))

DF= @(y,t) [(vb(1)\*(y(7)+c))/((kb(1)\*(1+((y(3)/kli)^p)))+(y(7)+c))-kd(1)\*y(1);...

kb(2)\*(y(1)^q)-kd(2)\*y(2)+kt(3)\*y(3);...

kt(2)\*y(2)-kt(3)\*y(3)-kd(3)\*y(3);...

(vb(4)\*(y(3)^r))/((kb(4)^r)+(y(3)^r))-kd(4)\*y(4);...

kb(5)\*y(4)-kd(5)\*y(5)-kt(5)\*y(5)+kt(6)\*y(6);...

kt(5)\*y(5)-kt(6)\*y(6)-kd(6)\*y(6)+ka(7)\*y(7)-ka(6)\*y(6);...

ka(6)\*y(6)-ka(7)\*y(7)-kd(7)\*y(7)];

F0=[0.3;0.1;1.1;0.8;1;1;1];

tspan=[0,100];

N=100000;

[fend,t,f]=MyIVPBW(DF,F0,tspan,N);

% F=f(5,:)+f(6,:)+f(7,:);

F=f(1,:);

plot(t,F), grid on

BW\_perturbation.m

function [graph,dpsy]=BW\_perturbation(force,tol,N,tspan,k)

%force consists of [magnitude;start;end]

%% GW model for initial forcing

F0=[0.273;0.347;1.318;1.119;0.830;1.469;1.092];

vb1initial=@(t) (9); %a1 is the function of a stimulus eg light

[t,f\_initial]=BWfunc(vb1initial,N,tspan,F0);

f\_initial=f\_initial(k,:);

%% GW model with forcing

vb1\_perturb=@(t) (9+(force(1)\*(force(2)<=t && t<=force(3))));

[~,f\_perturbed]=BWfunc(vb1\_perturb,N,tspan,F0);

f\_perturbed=f\_perturbed(k,:);

%% dpsy

%take fin value of fpert go back 24hrs

%estimate differential

hr24=ceil(N\*24/(tspan(2)-tspan(1))); %estimate for 24hrs in the array

n=N-hr24;

n1=n;

n2=n;

diffestimate=(f\_perturbed(1,n1)-f\_perturbed(1,n1-1)); %ignore the dt since dt is constant between array elements

%check for closest oscillation

while (abs(f\_initial(1,n1)-f\_perturbed(1,N-hr24))>tol(1) || abs(diffestimate-(f\_initial(1,n1+1)-f\_initial(1,n1-1))/2)>tol(2)) && n1>n-hr24

n1=n1-1;

end

while (abs(f\_initial(1,n2)-f\_perturbed(1,N-hr24))>tol(1) || abs(diffestimate-(f\_initial(1,n2+1)-f\_initial(1,n2-1))/2)>tol(2)) && n2<=N

n2=n2+1;

end

nfin=(abs(n1-n)<=abs(n2-n))\*n1 + (abs(n1-n)>abs(n2-n))\*n2;

graph=[f\_initial(1,:);f\_perturbed(1,:);t];

dpsy=t(n)-t(nfin);

if abs(dpsy)>12

dpsy=-24\*sign(dpsy)+dpsy;

end

end

BWfunc.m

function [graph,dpsy]=BW\_perturbation(force,tol,N,tspan,k)

%force consists of [magnitude;start;end]

%% GW model for initial forcing

F0=[0.273;0.347;1.318;1.119;0.830;1.469;1.092];

vb1initial=@(t) (9); %a1 is the function of a stimulus eg light

[t,f\_initial]=BWfunc(vb1initial,N,tspan,F0);

f\_initial=f\_initial(k,:);

%% GW model with forcing

vb1\_perturb=@(t) (9+(force(1)\*(force(2)<=t && t<=force(3))));

[~,f\_perturbed]=BWfunc(vb1\_perturb,N,tspan,F0);

f\_perturbed=f\_perturbed(k,:);

%% dpsy

%take fin value of fpert go back 24hrs

%estimate differential

hr24=ceil(N\*24/(tspan(2)-tspan(1))); %estimate for 24hrs in the array

n=N-hr24;

n1=n;

n2=n;

diffestimate=(f\_perturbed(1,n1)-f\_perturbed(1,n1-1)); %ignore the dt since dt is constant between array elements

%check for closest oscillation

while (abs(f\_initial(1,n1)-f\_perturbed(1,N-hr24))>tol(1) || abs(diffestimate-(f\_initial(1,n1+1)-f\_initial(1,n1-1))/2)>tol(2)) && n1>n-hr24

n1=n1-1;

end

while (abs(f\_initial(1,n2)-f\_perturbed(1,N-hr24))>tol(1) || abs(diffestimate-(f\_initial(1,n2+1)-f\_initial(1,n2-1))/2)>tol(2)) && n2<=N

n2=n2+1;

end

nfin=(abs(n1-n)<=abs(n2-n))\*n1 + (abs(n1-n)>abs(n2-n))\*n2;

graph=[f\_initial(1,:);f\_perturbed(1,:);t];

dpsy=t(n)-t(nfin);

if abs(dpsy)>12

dpsy=-24\*sign(dpsy)+dpsy;

end

end

eq\_track.m

%%

delta=[0.5,0.5,0.5];

a=[5,5,5];

k=1;

alpha=(5^3)/(0.5^3);

XYZ=@(xyzn) [alpha/(1+xyzn(3)^xyzn(4))-xyzn(1);xyzn(1)-xyzn(2);xyzn(2)-xyzn(3)];

% XYZ=@(xyzn) [

dXYZN=@(xyzn) MyJacobian(XYZ,xyzn,1e-5);

stepsize=5e-3;

nmax=5e+4;

[XYZlist,Eiglist]=MTC(XYZ,dXYZN,[0.04;0.6;5.5;30],[0;0;0;-0.01],stepsize,nmax);

%%

Z=XYZlist(3,:);

Eiglist=real(Eiglist);

Stablist=zeros(1,length(Eiglist));

% for n=1:length(Eiglist) %Sets up for the stability of a eq point

% if max(Eiglist(:,n))<0

% Stablist(1,n)=-1;

% elseif min(Eiglist(:,n))>0

% Stablist(1,n)=1;

% else

% Stablist(1,n)=0;

% end

% end

%Stablist gives number of unstable eigenvalues

tiledlayout(2,1)

nexttile

plot(XYZlist(4,:),Z), grid on

title('Location of Equilibria')

% axis([0,2,0,100])

xlabel('n')

ylabel('Z')

nexttile

gscatter(XYZlist(4,:),Z,Stablist), grid on

title('Location of Equilibria and Their Stability')

xlabel('n')

% axis([0,2,0,100])

legend({'stable','unstable','saddle'})

Goodwin.m

clear

delta=[1,1];

a=[2,1];

k=0.5;

df=@(F,t) [(a(1)\*(k/(k+F(2)))-delta(1));(a(2)\*F(1)-delta(2))];

F0=[1;2];

tspan=[0,80];

N=800;

[fend,t,f]=MyIVP(df,F0,tspan,N);

plot(t,f),grid on

% plot(f(1,:),f(2,:)), grid on

Goodwin3var.m

clear

%X mRNA (period 1/2)

%Y protein 1

%Z protein 2

delta=[0.5,0.5,0.5];

a1=@(t) (5); %a1 is the function of a stimulus eg light

% a1=@(t) (5+2.5\*sin(2\*pi\*t/24));

a=[0,5,5]; %a(1) is given above

k=1;

n=10;

df=@(F,t) [(a1(t)\*k^n)/(k^n+F(3)^n)-delta(1)\*F(1);a(2)\*F(1)-delta(2)\*F(2);a(3)\*F(2)-delta(3)\*F(3)]\*7.934787217/24; %X,Y,Z

F0=[0.01;0.2;2.6];

tspan=[0,2400];

N=1e+6;

[fend,t,f]=MyIVP(df,F0,tspan,N);

% plot(t,f(1,:),'.-'),grid on

% plot(f(1,:),f(3,:)), grid on

%%

% tiledlayout(2,2)

% nexttile

% plot(f(1,:),f(2,:)), grid on

% xlabel('mRNA')

% ylabel('Protein 1')

% nexttile

% plot(f(1,:),f(3,:)), grid on

% xlabel('mRNA')

% ylabel('Protein 2 (inhibitor)')

% nexttile

% plot(f(2,:),f(3,:)), grid on

% xlabel('Protein 1')

% ylabel('Protein 2 (inhibitor)')

% %%

% nexttile

%%

n=30;

df=@(F,t) [(a1(t)\*k^n)/(k^n+F(3)^n)-delta(1)\*F(1);a(2)\*F(1)-delta(2)\*F(2);a(3)\*F(2)-delta(3)\*F(3)]\*11.5985/24; %X,Y,Z

[~,tn,fn]=MyIVP(df,F0,tspan,N);

%%

hold on

F1=plot(t,f(1,:));grid on

F2=plot(t,f(2,:));

F3=plot(t,f(3,:));

xlabel('time')

xlim([tspan(2)-240,tspan(2)])

F1a=plot(tn,fn(1,:),'--');

F2a=plot(tn,fn(2,:),'--');

F3a=plot(tn,fn(3,:),'--');

hold off

legend([F1;F2;F3],'mRNA','Protein 1','Protein 2 (inhibitor)')

Goodwinvarforced.m

function [t,fGW]=Goodwin3varforced(N,tspan)

%X mRNA (period 1/2)

%Y protein 1

%Z protein 2

%period approximately 8 in arb time

delta=[0.5,0.5,0.5];

%a1=5 gives period 7.934787217 without extra term on df

a=[0,5,5]; %a(1) is given above

k=1;

n=10;

df=@(F,t) [(a1\_forced(t)\*k^n)/(k^n+F(3)^n)-delta(1)\*F(1);a(2)\*F(1)-delta(2)\*F(2);a(3)\*F(2)-delta(3)\*F(3)]\*7.934787217/24; %X,Y,Z

F0=[0.0079;0.18;2.5];

[~,t,fGW]=MyIVP(df,F0,tspan,N);

end

Goodwinvarfunc.m

function [t,fGW]=Goodwin3varfunc(a1,N,tspan,F0)

%X mRNA (period 1/2)

%Y protein 1

%Z protein 2

delta=[0.5,0.5,0.5];

%a1=5 gives period 7.934787217 without extra term on df

a=[0,5,5]; %a(1) is given above

k=1;

n=10; %10

df=@(F,t) [(a1(t)\*k^n)/(k^n+F(3)^n)-delta(1)\*F(1);a(2)\*F(1)-delta(2)\*F(2);a(3)\*F(2)-delta(3)\*F(3)].\*(7.934787217/24); %X,Y,Z

[~,t,fGW]=MyIVP(df,F0,tspan,N);

end

Goodwinvarfunc\_diff.m

function [t,fGW]=Goodwin3varfunc(a1,N,tspan,F0)

%X mRNA (period 1/2)

%Y protein 1

%Z protein 2

delta=[0.5,0.5,0.5];

%a1=5 gives period 7.934787217 without extra term on df

a=[0,5,5]; %a(1) is given above

k=1;

n=10; %10

df=@(F,t) [(a1(t)\*k^n)/(k^n+F(3)^n)-delta(1)\*F(1);a(2)\*F(1)-delta(2)\*F(2);a(3)\*F(2)-delta(3)\*F(3)].\*(7.934787217/24); %X,Y,Z

[~,t,fGW]=MyIVP(df,F0,tspan,N);

end

Goodwinvarfunc\_nvar.m

function [t,fGW]=Goodwin3varfunc\_nvar(a1,N,tspan,F0,n,freq)

%X mRNA (period 1/2)

%Y protein 1

%Z protein 2

delta=[0.5,0.5,0.5];

%a1=5 gives period 7.934787217 without extra term on df

a=[0,5,5]; %a(1) is given above

k=1;

df=@(F,t) [(a1(t)\*k^n)/(k^n+F(3)^n)-delta(1)\*F(1);a(2)\*F(1)-delta(2)\*F(2);a(3)\*F(2)-delta(3)\*F(3)]\*(7.934787217/24)\*freq; %X,Y,Z

[~,t,fGW]=MyIVP(df,F0,tspan,N);

end

GW\_lightpoll.m

clear

tspan=[0,800];

F1=[0.0079;0.18;2.5];

N=1e+5;

%% to get a stable oscillation/initial conditions

tspan1=[0,240];

a1\_initial=@(t) (5+2.5\*sin(2\*pi\*t/24)); %a1 is the function of a stimulus eg light

[t1,f\_initial]=Goodwin3varfunc(a1\_initial,N,tspan1,F1);

F0=f\_initial(:,end);

%% no light pollution

a1\_initial=@(t) (5+2.5\*sin(2\*pi\*t/24)); %a1 is the function of a stimulus eg light

[t,f\_natural]=Goodwin3varfunc(a1\_initial,N,tspan,F0);

%% with light pollution

const=0.5; %const\*2.5 is the minimal forcing on top of the 5

%ie light never drops below const\*2.5

a1\_lpoll=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(sin(2\*pi\*t/24)>const)+...

(5+2.5\*const)\*(sin(2\*pi\*t/24)<=const);

[~,f\_lpoll]=Goodwin3varfunc(a1\_lpoll,N,tspan,F0);

%fplot(a1\_lpoll) for a graph of the forcing term

%% combining initial to other plots

f\_natural=[f\_initial,f\_natural];

f\_lpoll=[f\_initial,f\_lpoll];

t\_total=[t1,t+t1(end)];

%% plots

hold on

plot(t\_total,f\_natural(1,:)), grid on

plot(t\_total,f\_lpoll(1,:),'--')

xlim([216 400])

title('Effects of Light Pollution')

xlabel('Time (hours)')

ylabel('Concentration')

legend({'No Pollution','Pollution'},'Location','southeast')

Goodwin\_perttest.m

clear

%force consists of [magnitude;start;end]

%[0;12;13] for the graphs

force=[0;25;24];

%tolerence of f and df see while loops in func

tol=[1e-4,0.05];

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

m=1;

%force of light perturbations

%if too many it can slow the code significantly

lightforce=(0:1:15);

duration=[0.5,1,2,3,4];

dpsy=zeros(1,length(lightforce));

for ptime=duration

force(3)=force(2)+ptime;

n=1;

for pert=lightforce

force(1)=pert;

[graph,dpsy(m,n)]=GW\_perturbation(force,tol,N,tspan);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1;

end

m=m+1

end

%%

plot(lightforce,dpsy), grid on

title('Effect of light intensity on phase shift')

xlabel('Intensity of light')

ylabel('Phase Shift')

% plot(graph(3,:),graph(1:2,:))

legend({'30mins','1hr','2hrs','3hrs','4hrs'},'location','northwest')

GW\_perturbation.m

function [graph,dpsy]=GW\_perturbation(force,tol,N,tspan)

%force consists of [magnitude;start;end]

%% GW model for initial forcing

F0=[0.0079;0.18;2.5];

a1initial=@(t) (5); %a1 is the function of a stimulus eg light

[t,f\_initial]=Goodwin3varfunc(a1initial,N,tspan,F0);

%% GW model with forcing

a1\_perturb=@(t) (5+(force(1)\*(force(2)<=t && t<=force(3))));

[~,f\_perturbed]=Goodwin3varfunc(a1\_perturb,N,tspan,F0);

%% dpsy

%take fin value of fpert go back 24hrs

%estimate differential

hr24=ceil(N\*24/(tspan(2)-tspan(1))); %estimate for 24hrs in the array

n=N-hr24;

n1=n;

n2=n;

diffestimate=(f\_perturbed(1,n1)-f\_perturbed(1,n1-1)); %ignore the dt since dt is constant between array elements

%check for closest oscillation

while (abs(f\_initial(1,n1)-f\_perturbed(1,N-hr24))>tol(1) || abs(diffestimate-(f\_initial(1,n1+1)-f\_initial(1,n1-1))/2)>tol(2)) && n1>n-hr24

n1=n1-1;

end

while (abs(f\_initial(1,n2)-f\_perturbed(1,N-hr24))>tol(1) || abs(diffestimate-(f\_initial(1,n2+1)-f\_initial(1,n2-1))/2)>tol(2)) && n2<=N

n2=n2+1;

end

nfin=(abs(n1-n)<=abs(n2-n))\*n1 + (abs(n1-n)>abs(n2-n))\*n2;

graph=[f\_initial(1,:);f\_perturbed(1,:);t];

dpsy=t(n)-t(nfin);

end

GWforcedtest.m

clear

tol=2e-2;

N=1e+4;

lag1=48;

k=1;

for dt=0.1:0.1:24

lag=[lag1,lag1+dt]; %start/end of time pause

[graph,recov(:,k)]=Response\_GW(lag,tol,N);

Dt(k)=dt;

k=k+1;

end

%%

dpsy=recov(2,:)-lag(1);

scatter(Dt,dpsy,'x'), grid on

% hold on

% plot(graph(4,:),graph(1,:),'linewidth',0.6), grid on

% plot(graph(4,:),graph(2,:),'-','color','#77AC30','linewidth',0.6)

% plot(graph(4,:),graph(3,:),'--','color','#A2142F','linewidth',0.6)

% xlim([24,(graph(4,end))])

GWforcedvsunforced

clear

%X mRNA (period 1/2)

%Y protein 1

%Z protein 2

delta=[0.5,0.5,0.5];

F0=[0.01;0.2;2.6];

N=1e+5;

tspaninit=[0,240];

specific=[10,20,30;7.934787217,10.2990909091,11.5985];

k=1;

n=specific(1,k);

freqforn=specific(2,k);

freq=1.01; %1.01 -> 1% more oscillations or 23hr 45min circadian rhythm

%%

%ensure that both are in sync

a1init=@(t) (5+2.5\*sin(2\*pi\*t/24));

[tini,GW]=Goodwin3varfunc(a1init,N,tspaninit,F0);

F0=GW(:,end);

tspan=[0,960]; %960

%%

%unforced

a1=@(t) (5); %a1 is the function of a stimulus eg light

[t,ufGW]=Goodwin3varfunc\_diff(a1,N,tspan,F0,freq,n,freqforn);

%light dark forced

a1=@(t) (5+2.5\*sin(2\*pi\*t/24));

[~,fGW]=Goodwin3varfunc\_diff(a1,N,tspan,F0,freq,n,freqforn);

%24hr rythm under forcing

a1=@(t) (5+2.5\*sin(2\*pi\*t/24));

[~,natfGW]=Goodwin3varfunc\_diff(a1,N,tspan,F0,1,n,freqforn);

%24hr rythm without forcing

a1=@(t) 5;

[~,natufGW]=Goodwin3varfunc\_diff(a1,N,tspan,F0,1,n,freqforn);

%%

hold on

F1=plot(t,ufGW(1,:),'color','#0072BD'); grid on

F2=plot(t,ufGW(2,:),'color','#D95319');

F3=plot(t,ufGW(3,:),'color','#EDB120');

F1a=plot(t,fGW(1,:),'--','color','#0072BD');

F2a=plot(t,fGW(2,:),'--','color','#D95319');

F3a=plot(t,fGW(3,:),'--','color','#EDB120');

F3b=plot(t,natufGW(3,:),'color','#7E2F8E');

F3c=plot(t,natfGW(3,:),'--','color','#7E2F8E');

xlabel('Time (hours)')

ylabel('Concentration')

title('Effects of Forcing on the Goodwin Model')

xlim([tspan(2)-240,tspan(2)])

hold off

legend([F1;F2;F3;F3b;F1a;F2a;F3a;F3c],'mRNA','Protein 1','Protein 2 (inhibitor)','Protein 2 24hr rhythm','mRNA under forcing','Protein 1 under forcing','Protein 2 under forcing','Protein 2 24hr rhythm under forcing')

Intracellular.m

rm=1;

rp=1;

qm=0.21;

qp=0.21;

n=2;

m=3;

tau=4;

k=1;

rate=@(M,P,t) [rm/(1+(P/k)^n)-qm\*P;rp\*M\*(t-tau)^m-qp\*P]; %dM and dP respectively

F0=[1;1];

tspan=[0,80];

N=800;

[fend,t,f]=MyIVP(df,F0,tspan,N);

plot(t,f),grid on

JetLag.m

clear

tol=1.2e-2;

N=1e+4;

lag1=48;

k=1;

lag=[36,44]; %start/end of time pause

[graph,recov]=Response\_GW(lag,tol,N);

%%

hold on

plot(graph(4,:),graph(1,:),'linewidth',0.6), grid on

plot(graph(4,:),graph(2,:),'-','color','#77AC30','linewidth',0.6)

plot(graph(4,:),graph(3,:),'--','color','#A2142F','linewidth',0.6)

scatter(recov(2),recov(1),'o')

xlim([0 200])

title('Effects of Jetlag')

xlabel('Time (hours)')

ylabel('Concentration')

legend({'Old Time Zone','Effects of Changing Time Zone','New Time zone'},'Location','northeast')

JetLag\_theoreticalsoln.m

clear

tol=1.2e-2;

N=1e+4;

k=1;

tspan=[0,720];

pert=[6,8.5,4]; %magnitude,start,duration

lag=[24,32]; %start/end of time pause

[graph,~]=Response\_GW\_theoreticalforcing(lag,N,pert,tspan,tol);

hold on

plot(graph(end,:),graph(2,:),'-','color','#77AC30','linewidth',0.6), grid on %jet lag without perturbation

plot(graph(end,:),graph(3,:),'--','color','#A2142F','linewidth',0.6) %circ rythm for new location

plot(graph(end,:),graph(4,:),'-.','color','#7E2F8E','linewidth',0.6) %jet lag with perterbation

xlim([0 120])

legend('natural effects of jet lag','circadian rythm for new location','effects of jet lag after perturbation')

xlabel('time (hours)')

ylabel('concentration of X')

title('Theoretical solution to jet lag')

MTC.m

function [ylist,EIG]=MTC(userf,userdf,y0,ytan,s,nmax)

z0=zeros(length(y0),1);

z0(end)=1;

Y(:,1)=y0;

EIG=zeros(length(y0)-1,3);

%% s=0 attempt

J=2;

s0=0;

ypred=Y(:,J-1)+s0\*ytan;

f\_ytan=@(y)[userf(y);(y-ypred).\*ytan];

df\_ytan=@(y)MyJacobian(f\_ytan,y,1e-7);

[~,~,Y(:,J)]=MySolve(f\_ytan,ypred,df\_ytan,1e-7,20);

f=@(y) userf([y;Y(end,J)]);

DF=MyJacobian(f,Y(1:end-1,J),1e-5);

% EIG(:,1)=eig(DF);

Z=[userdf(Y(:,J));ytan']\z0;

ytan=Z/norm(Z,inf)'\*sign(Z'\*ytan);

%%

for j=2:nmax

ypred=Y(:,j-1)+s\*ytan; %initial guess

f\_ytan=@(y)[userf(y);(y-ypred)'\*ytan]; %concatenation of (1) and (2)

df\_ytan=@(y)MyJacobian(f\_ytan,y,1e-7);

[~,~,Y(:,j)]=MySolve(f\_ytan,ypred,df\_ytan,1e-7,20); %Solves equation set created by f\_ytan

Z=[userdf(Y(:,j));ytan']\z0;

ytan=Z/norm(Z,inf).\*sign(Z'\*ytan);

f=@(y) userf([y;Y(end,j)]);

DF=MyJacobian(f,Y(1:end-1,j),1e-5);

% EIG(:,j)=eig(DF);

end

ylist=Y;

end

MyIVP.m

function [xend,t,xt]=MyIVP(f,x0,tspan,N)

%f is such that df=f(x,t)

%x0 initial point

%tspan is [min(t),max(t)]

%N number of steps

h=(tspan(2)-tspan(1))/N; %stepsize

xt=zeros(length(x0),N+1);

xt(:,1)=x0;

t=tspan(1):h:tspan(2);

for step=2:N+1

% xt(:,step)=xt(:,step-1)+h\*f(xt(:,step-1),t(step-1)); %fore euler

k(:,1)=h\*f(xt(:,step-1),t(step-1)); %RK4 scheme

k(:,2)=h\*f(xt(:,step-1)+k(:,1)/2,t(step-1)+(h/2));

k(:,3)=h\*f(xt(:,step-1)+k(:,2)/2,t(step-1)+(h/2));

k(:,4)=h\*f(xt(:,step-1)+k(:,3),t(step-1)+h);

xt(:,step)=xt(:,step-1)+(k(:,1)+2\*k(:,2)+2\*k(:,3)+k(:,4))/6;

end

xend=xt(:,end);

end

MyIVPBW.m

function [xend,t,xt]=MyIVPBW(f,x0,tspan,N)

%f is such that df=f(x,t)

%x0 initial point

%tspan is [min(t),max(t)]

%N number of steps

h=(tspan(2)-tspan(1))/N; %stepsize

xt=zeros(length(x0),N+1);

xt(:,1)=x0;

t=tspan(1):h:tspan(2);

for step=2:N+1

% xt(:,step)=xt(:,step-1)+h\*f(xt(:,step-1),t(step-1)); %fore euler

k(:,1)=h\*f(xt(:,step-1),t(step-1)); %RK4 scheme

k(:,2)=h\*f(xt(:,step-1)+k(:,1)/2,t(step-1)+(h/2));

k(:,3)=h\*f(xt(:,step-1)+k(:,2)/2,t(step-1)+(h/2));

k(:,4)=h\*f(xt(:,step-1)+k(:,3),t(step-1)+h);

xt(:,step)=xt(:,step-1)+(k(:,1)+2\*k(:,2)+2\*k(:,3)+k(:,4))/6;

end

xend=xt(:,end);

end

pertendtime.m

%force consists of [magnitude;start;end]

force=[5;12;12];

%tolerence of f and df see while loops in func

tol=[1e-4,0.05];

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

n=1;

%force of light perturbations

%if too many it can slow the code significantly

lighttime=(0:0.5:24);

dpsy=zeros(1,length(lighttime));

for pert=lighttime

force(3)=force(2)+pert;

[graph,dpsy(1,n)]=GW\_perturbation(force,tol,N,tspan);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1

end

plot(lighttime,dpsy), grid on

pertstart\_length.m

%force consists of [magnitude;start;end]

force=[5;12;12.25];

%tolerence of f and df see while loops in func

tol=[1e-5,0.05];

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

m=1;

%force of light perturbations

%if too many it can slow the code significantly

%24:0.5:48

lightstart=(24:0.125:48);

lighttime=[0.5,1,2,4,6];

% lighttime=[4,6,8,10];

dpsy=zeros(1,length(lightstart));

for ptime=lighttime

n=1;

for pert=lightstart

force(2)=pert;

force(3)=force(2)+ptime;

[graph,dpsy(m,n)]=GW\_perturbation(force,tol,N,tspan);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1;

end

m=m+1

end

%%

tiledlayout(2,1)

nexttile

hold on

plot(lightstart,dpsy), grid on %response curve based on when perturbation occurs

legend('Perturbation Length 30mins','Perturbation Length 1hr','Perturbation Length 2hr','Perturbation Length 3hr','Perturbation Length 4hr')

xlim([lightstart(1) lightstart(end)])

title('Different Light Perturbation on Phase Shift')

xlabel('Time perturbation Starts (hours)')

ylabel('Phase Shift (hours)')

hold off

nexttile

a1=@(t) 5+2.5\*sin(2\*pi\*t/24);

fplot(a1), grid on %light dark cycle

xlim([lightstart(1) lightstart(end)])

title('Day/Night Light Levels')

xlabel('Time (hours)')

ylabel('Arbitrary Light Level')

pertstart\_lengthBW.m

clear

%%

% k=2; %1-7 dependant on which chemical we want

%not good with k=1

fig1=figure;

fig2=figure;

fig3=figure;

for k=1:7

%%

%force consists of [magnitude;start;end]

force=[6;12;12.25];

%tolerence of f and df see while loops in func

tol=[1e-3,5e-4];

%number of time steps

N=5e+4;

%time span ran over

tspan=[0,240];

m=1;

%%

%force of light perturbations

%if too many it can slow the code significantly

%24:0.5:48

lightstart=(24:0.25:48);

lighttime=[1,2,4];

dpsy=zeros(1,length(lightstart));

for ptime=lighttime

n=1;

for pert=lightstart

force(2)=pert;

force(3)=force(2)+ptime;

[g,dpsy(m,n)]=BW\_perturbation(force,tol,N,tspan,k);

% graph(n,:,m)=g(1,:);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1;

end

m=m+1

end

%%

% tiledlayout(2,1)

% nexttile

figure(fig1);

hold on

plot(lightstart,dpsy(1,:)), grid on %response curve based on when perturbation occurs

% plot(lightstart,abs(dpsy),'--') %taking the absolute value of the above graph only use for k=1

legend('y1','y2','y3','y4','y5','y6','y7')

xlim([lightstart(1) lightstart(end)])

title('Different Light Perturbation on phase shift 1hr')

xlabel('Time perturbation starts (hours)')

ylabel('Phase shift (hours)')

hold off

%%

figure(fig2);

hold on

plot(lightstart,dpsy(2,:)), grid on %response curve based on when perturbation occurs

% plot(lightstart,abs(dpsy),'--') %taking the absolute value of the above graph only use for k=1

legend('y1','y2','y3','y4','y5','y6','y7')

xlim([lightstart(1) lightstart(end)])

title('Different Light Perturbation on phase shift 2hrs')

xlabel('Time perturbation starts (hours)')

ylabel('Phase shift (hours)')

hold off

%%

figure(fig3);

hold on

plot(lightstart,dpsy(3,:)), grid on %response curve based on when perturbation occurs

% plot(lightstart,abs(dpsy),'--') %taking the absolute value of the above graph only use for k=1

legend('y1','y2','y3','y4','y5','y6','y7')

xlim([lightstart(1) lightstart(end)])

title('Different Light Perturbation on phase shift 4hrs')

xlabel('Time perturbation starts (hours)')

ylabel('Phase shift (hours)')

hold off

end

% nexttile

% a1=@(t) 5+2.5\*sin(2\*pi\*t/24);

% fplot(a1), grid on %light dark cycle

% % plot(g(3,:),g(1,:))

% xlim([lightstart(1) lightstart(end)])

% title('Day/Night light levels')

% xlabel('Time (hours)')

% ylabel('Arbitrary light level')

%%

% gr=permute(graph(1,:,1:2:3),[3 2 1]);

% plot(g(3,:),gr)

pertstart\_lengthBW\_singlegraph.m

clear

%%

k=4; %1-7 dependant on which chemical we want

%not good with k=1

%%

%force consists of [magnitude;start;end]

force=[6;12;12.25];

%tolerence of f and df see while loops in func

tol=[1e-3,5e-4];

%number of time steps

N=8e+4;

%time span ran over

tspan=[0,240];

m=1;

%%

%force of light perturbations

%if too many it can slow the code significantly

%24:0.5:48

lightstart=(24:0.2:48);

lighttime=[1,2,4];

dpsy=zeros(1,length(lightstart));

for ptime=lighttime

n=1;

for pert=lightstart

force(2)=pert;

force(3)=force(2)+ptime;

[g,dpsy(m,n)]=BW\_perturbation(force,tol,N,tspan,k);

graph(n,:,m)=g(1,:);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1;

end

m=m+1

end

%%

hold on

plot(lightstart,dpsy), grid on %response curve based on when perturbation occurs

legend('perturbation length 1hr','perturbation length 2hr','perturbation length 4hr')

xlim([lightstart(1) lightstart(end)])

title('Different Light Perturbation on phase shift')

xlabel('Time perturbation starts (hours)')

ylabel('Phase shift (hours)')

hold off

pertstarttime.m

%force consists of [magnitude;start;end]

force=[5;12;12.25];

%length of perturbation

ptime=10;

%tolerence of f and df see while loops in func

tol=[1e-4,0.05];

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

%if too many it can slow the code significantly

lightstart=(12:0.5:36);

dpsy=zeros(1,length(lightstart));

n=1;

for pert=lightstart

force(2)=pert;

force(3)=force(2)+ptime;

[graph,dpsy(1,n)]=GW\_perturbation(force,tol,N,tspan);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1;

end

%%

tiledlayout(2,1)

nexttile

hold on

plot(lightstart,dpsy), grid on %response curve based on when perturbation occurs

xlim([lightstart(1) lightstart(end)])

hold off

nexttile

a1=@(t) (5+2.5\*sin(2\*pi\*t/24));

fplot(a1), grid on %normal goodwin model

xlim([lightstart(1) lightstart(end)])

phone\_func.m

function [graph]=phone\_func(N,a1,tspan)

%% GW model for initial forcing

F0=[0.0079;0.18;2.5];

a1initial=@(t) (5+2.5\*sin(2\*pi\*t/24)); %a1 is the function of a stimulus eg light

[t,f\_initial]=Goodwin3varfunc(a1initial,N,tspan,F0);

%% GW model with pause in forcing + perturbation

[~,f\_phone]=Goodwin3varfunc(a1,N,tspan,F0);

%%

graph=[f\_initial(1,:);f\_phone(1,:);t];

end

phones.m

clear

%force consists of [magnitude;start;end]

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

%a1 for phone forcing

%0600 is t=0

force=[16;13;17];

a1=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(0==(sin(2\*pi\*t/24)<sin(2\*pi\*force(2)/24) && sin(2\*pi\*t/24)>sin(2\*pi\*force(3)/24) && sin(4\*pi\*t/24)>0))+...

force(1)\*(sin(2\*pi\*t/24)<sin(2\*pi\*force(2)/24) && sin(2\*pi\*t/24)>sin(2\*pi\*force(3)/24) && sin(4\*pi\*t/24)>0);

% force=[16;17;24];

% a1=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(0==(sin(2\*pi\*t/24)<sin(2\*pi\*force(2)/24) && sin(2\*pi\*t/24)>sin(2\*pi\*force(3)/24) && sin(4\*pi\*t/24)<0))+...

% force(1)\*(sin(2\*pi\*t/24)<sin(2\*pi\*force(2)/24) && sin(2\*pi\*t/24)>sin(2\*pi\*force(3)/24) && sin(4\*pi\*t/24)<0);

[graph]=phone\_func(N,a1,tspan);

%%

% plot(lightforce,dpsy), grid on

hold on

plot(graph(3,:),graph(1,:)), grid on

plot(graph(3,:),graph(2,:),'--')

xlim([120,240])

R\_square.m

clear

c=1; %c can be +-1

zh=@(n) (8/(n-8))^(1/n);

rsquare=@(n) [((2\*zh(n)\*(n-8)\*c)/((88/63)\*((n-17)^2)-(n^2-51\*n+434))); 126/(25\*n)];

k=1;

nspan=(8:0.001:100);

N=zeros(2,length(nspan));

for n=nspan

N(:,k)=rsquare(n);

k=k+1;

end

hold on

plot(nspan,N(1,:)), grid on

plot(nspan,N(2,:),'--')

ylim([0,0.15])

ylabel('R^{2}')

xlabel('n')

title('Amplitude of R^{2} as a function of n')

Response\_GW.m

function [graph,recov]=Response\_GW(lag,tol,N)

tspan=[0,720];

%% GW model for initial forcing

F0=[0.0079;0.18;2.5];

a1initial=@(t) (5+2.5\*sin(2\*pi\*t/24)); %a1 is the function of a stimulus eg light

[t,f\_initial]=Goodwin3varfunc(a1initial,N,tspan,F0);

%% GW model with pause in forcing

a1\_lag=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(t<=lag(1))+...

(5+2.5\*sin(2\*pi\*lag(1)/24))\*(lag(1)<t && t<=lag(2))+...

(5+2.5\*sin(2\*pi\*(t-(lag(2)-lag(1)))/24))\*(lag(2)<t);

[~,f\_forced]=Goodwin3varfunc(a1\_lag,N,tspan,F0);

%% GW model for new forcing

%initial conditions for new light cycle

tspan2=[0,(0-(lag(2)-lag(1)))];

[~,F]=Goodwin3varfunc(a1initial,N,tspan2,F0);

F0=F(:,end);

a1new=@(t) (5+2.5\*sin(2\*pi\*(t-(lag(2)-lag(1)))/24));

[~,f\_new]=Goodwin3varfunc(a1new,N,tspan,F0);

%%

n=N;

while abs(f\_new(1,n)-f\_forced(1,n))<tol && n>ceil(N\*lag(1)/(tspan(2)-tspan(1)))

n=n-1;

end

graph=[f\_initial(1,:);f\_forced(1,:);f\_new(1,:);t];

recov=[f\_forced(1,n);t(n)];

end

Response\_GW\_theoreticalforcing.m

function [graph,recov]=Response\_GW\_theoreticalforcing(lag,N,pert,tspan,tol)

%% GW model for initial forcing

F0=[0.0079;0.18;2.5];

a1initial=@(t) (5+2.5\*sin(2\*pi\*t/24)); %a1 is the function of a stimulus eg light

[t,f\_initial]=Goodwin3varfunc(a1initial,N,tspan,F0);

%% GW model with pause in forcing

a1\_lag=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(t<=lag(1))+...

(5+2.5\*sin(2\*pi\*lag(1)/24))\*(lag(1)<t && t<=lag(2))+...

(5+2.5\*sin(2\*pi\*(t-(lag(2)-lag(1)))/24))\*(lag(2)<t);

[~,f\_forced]=Goodwin3varfunc(a1\_lag,N,tspan,F0);

%% GW model for new forcing

%initial conditions for new light cycle

tspan2=[0,(0-(lag(2)-lag(1)))];

[~,F]=Goodwin3varfunc(a1initial,N,tspan2,F0);

F0\_=F(:,end);

a1new=@(t) (5+2.5\*sin(2\*pi\*(t-(lag(2)-lag(1)))/24));

[~,f\_new]=Goodwin3varfunc(a1new,N,tspan,F0\_);

%% GW model with pause in forcing + perturbation

a1\_lag\_pert=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(t<=lag(1))+...

(5+2.5\*sin(2\*pi\*lag(1)/24))\*(lag(1)<t && t<=lag(2))+...

(5+2.5\*sin(2\*pi\*(t-(lag(2)-lag(1)))/24))\*(lag(2)<t)+...

(pert(1)\*(pert(2)<=t && t<=pert(2)+pert(3)));

[~,f\_forced\_pert]=Goodwin3varfunc(a1\_lag\_pert,N,tspan,F0);

%%

n=N;

while abs(f\_new(1,n)-f\_forced(1,n))<tol && n>ceil(N\*lag(1)/(tspan(2)-tspan(1)))

n=n-1;

end

recov=[f\_forced(1,n);t(n)];

graph=[f\_initial(1,:);f\_forced(1,:);f\_new(1,:);f\_forced\_pert(1,:);t];

end

Response\_GW\_theoreticalforcing\_SW.m

function [graph,recov]=Response\_GW\_theoreticalforcing\_SW(shift,N,pert,tspan,tol,k)

%% GW model for initial forcing

F0=[0.0079;0.18;2.5];

a1initial=@(t) (5+2.5\*sin(2\*pi\*t/24)); %a1 is the function of a stimulus eg light

[t,f\_initial]=Goodwin3varfunc(a1initial,N,tspan,F0);

%% GW model with shift

a1\_shift=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(t<=shift(1) || t>shift(2))+...

(shift(3)+5)\*(shift(1)<t && t<=shift(2));

[~,f\_forced]=Goodwin3varfunc(a1\_shift,N,tspan,F0);

%% GW model with pause in forcing + perturbation1

a1\_shift\_pert=@(t) (5+2.5\*sin(2\*pi\*t/24))\*(t<=shift(1) || t>shift(2))+...

(shift(3)+5)\*(shift(1)<t && t<=shift(2))+...

(pert(1,1)\*(pert(1,2)<=t && t<=(pert(1,2)+pert(1,3))));

[~,f\_forced\_pert]=Goodwin3varfunc(a1\_shift\_pert,N,tspan,F0);

%%

n=N;

while abs(f\_initial(k,n)-f\_forced(k,n))<tol && n>ceil(N\*shift(1)/(tspan(2)-tspan(1)))

n=n-1;

end

recov=[f\_forced(k,n);t(n)];

graph=[f\_initial(k,:);f\_forced(k,:);f\_forced\_pert(k,:);t];

end

shiftwork.m

%force consists of [magnitude;start;end]

force=[5;0;9];

%tolerence of f and df see while loops in func

tol=[1e-4,0.01];

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

n=1;

%force of light perturbations

%if too many it can slow the code significantly

shiftstart=(0:0.5:24);

shiftlength=9;

dpsy=zeros(1,length(shiftstart));

for pert=shiftstart

force(2)=pert;

force(3)=force(2)+shiftlength;

[graph,dpsy(1,n)]=GW\_perturbation(force,tol,N,tspan);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1

end

%%

a1=@(t) 5+2.5\*sin(2\*pi\*t/24);

tiledlayout(2,1)

nexttile

hold on

plot(shiftstart,dpsy), grid on %response curve based on when perturbation occurs

xlim([shiftstart(1) shiftstart(end)])

title('Phase Shift Against Beginning of a Work Shift')

xlabel('Start of the Work Shift (hours)')

ylabel('Phase Shift (hours)')

hold off

nexttile

fplot(a1), grid on %normal goodwin model

xlim([shiftstart(1) shiftstart(end)])

title('Day/Night Light Levels')

xlabel('Time (hours)')

ylabel('Arbitrary Light Level')

ShiftWork\_theoreticalsoln.m

%force consists of [magnitude;start;end]

force=[5;0;9];

%tolerence of f and df see while loops in func

tol=[1e-4,0.01];

%number of time steps

N=1e+5;

%time span ran over

tspan=[0,240];

n=1;

%force of light perturbations

%if too many it can slow the code significantly

shiftstart=(0:0.5:24);

shiftlength=9;

dpsy=zeros(1,length(shiftstart));

for pert=shiftstart

force(2)=pert;

force(3)=force(2)+shiftlength;

[graph,dpsy(1,n)]=GW\_perturbation(force,tol,N,tspan);

%unsilence n=n=1 to get an idea of how fast its running

n=n+1

end

%%

a1=@(t) 5+2.5\*sin(2\*pi\*t/24);

tiledlayout(2,1)

nexttile

hold on

plot(shiftstart,dpsy), grid on %response curve based on when perturbation occurs

xlim([shiftstart(1) shiftstart(end)])

title('Phase Shift Against Beginning of a Work Shift')

xlabel('Start of the Work Shift (hours)')

ylabel('Phase Shift (hours)')

hold off

nexttile

fplot(a1), grid on %normal goodwin model

xlim([shiftstart(1) shiftstart(end)])

title('Day/Night Light Levels')

xlabel('Time (hours)')

ylabel('Arbitrary Light Level')

stab\_diagram.m

H=@(n) ((8/(n-8))^(1/n))\*n/(n-8);

hold on

fplot(H,[8 100])

% xlim([7,10])

xline(8,'--')

% xlabel('n')

% ylabel('\bf\alpha')

% title('Stablility Diagram')

test.m

H=@(n) ((8/(n-8))^(1/n))\*n/(n-8);

hold on

fplot(H,[8 100])

% xlim([7,10])

xline(8,'--')

% xlabel('n')

% ylabel('\bf\alpha')

% title('Stablility Diagram')