Dynamic Modeling HW17 - Do Not Post

1) (i) $x'(t) = 3x(t) - y(t)$

$y'(t) = 2x(t)$

$x''(t) = 3x'(t) - y'(t)$

$x''(t) = 3x'(t) - 2x(t)$

$x''(t) - 3x'(t) + 2x(t) = 0$

$r^2 - 3r + 2 = 0$

$(r-1)(r-2) = 0$

$r = 1, 2$

$x(t) = e^{2t} + e^t$

$\int y'(t) = \int 2e^{2t} + 2e^t dt$

$y(t) = 2(\frac{1}{2}e^{2t}) + 2e^t + c$

$y(t) = e^{2t} + 2e^t + c$

$3 = 1 + 2 + c$

$3 = 3 + c$

$c = 0$

$$x(t) = e^{2t} + e^t$$
$$y(t) = e^{2t} + 2e^t$$

$x(t) = Ae^{2t} + Be^t$

$x'(t) = 2Ae^{2t} + Be^t$

$x(0) = 2$

$x'(0) = 3(2) - 3 = 3$

$2 = A + B$

$-1 \cdot (3 = 2A + B)$

$2 = A + B$

$-3 = -2A - B$

$-1 = -A \Rightarrow A = 1$

$1 + B = 2$

$B = 1$

(ii) $X(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ $A = \begin{bmatrix} 3 & -1 \\ 2 & 0 \end{bmatrix}$ $X'(t) = \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix}$ $X(0) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$\det \begin{bmatrix} 3-\lambda & -1 \\ 2 & -\lambda \end{bmatrix} = -\lambda(3-\lambda) + 2 = 0$ → $(\lambda-1)(\lambda-2)$

$\lambda^2 - 3\lambda + 2 = 0$

$\lambda = 2$

$\begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix} \xrightarrow{-2r_1 + r_2 \to r_2} \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = y(t) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\lambda = 1$

$\begin{bmatrix} 2 & -1 \\ 2 & -1 \end{bmatrix} \xrightarrow{-r_1 + r_2 \to r_2} \begin{bmatrix} 2 & -1 \\ 0 & 0 \end{bmatrix} \xrightarrow{1/2 r_1 \to r_1} \begin{bmatrix} 1 & -1/2 \\ 0 & 0 \end{bmatrix}$ $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = x(t) \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $\vec{v}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = A \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{2t} + B \begin{bmatrix} 1 \\ 2 \end{bmatrix} e^{t}$

$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = A \begin{bmatrix} 1 \\ 1 \end{bmatrix} + B \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$A + B = 2$ → $A + 1 = 2$

$-1(A + 2B = 3)$ → $A = 1$

$-B = -1$

$B = 1$

$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{2t} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} e^{t}$

$\boxed{x(t) = e^{2t} + e^{t} \quad y(t) = e^{2t} + 2e^{t}}$

2) $x'(t) = x(t) + 8y(t)$ $x(0) = 5$

$y'(t) = 4x(t)$ $y(0) = 6$

(i) $x''(t) = x'(t) + 8y'(t)$

$x''(t) - x'(t) - 32x(t) = 0$

$r^2 - r - 32 = 0$

$r = \frac{1 \pm \sqrt{1 - 4(1)(-32)}}{2(1)} = \frac{1 \pm \sqrt{1+128}}{2} = \frac{1 \pm \sqrt{129}}{2}$ * $\frac{1+\sqrt{129}}{2} \approx 6.179$; $\frac{1-\sqrt{129}}{2} \approx -5.179$

$x(t) = Ae^{\frac{1+\sqrt{129}}{2}t} + Be^{\frac{1-\sqrt{129}}{2}t}$ $x(0) = 5$

$x'(t) = \frac{1+\sqrt{129}}{2} Ae^{\frac{1+\sqrt{129}}{2}t} + \frac{1-\sqrt{129}}{2} Be^{\frac{1-\sqrt{129}}{2}t}$ $x'(0) = 5 + 8(6) = 53$

$(5 = A + B) \cdot 5.179$

$+ \ 53 = 6.179A - 5.179B$

$78.895 = 11.358A$

$A = 6.946$

$B = 5 - 6.946 = -1.946$

$\boxed{x(t) = 6.946e^{\frac{1+\sqrt{129}}{2}t} - 1.946e^{\frac{1-\sqrt{129}}{2}t}}$

$\int y'(t) = \int 4x(t)$

→ integral done on maple

$x(t) = 6.946e^{\frac{1+\sqrt{129}}{2}t} - 1.946e^{\frac{1-\sqrt{129}}{2}t}$

$y(t)$ = → on maple (not fully matching (iii) due to rounding errors in hand-done math)

(ii) Eigenvectors + Eigenvalues on maple

$$\begin{bmatrix} 5 \\ 6 \end{bmatrix} = A \begin{bmatrix} \frac{8}{-1/2 + \frac{\sqrt{129}}{2}} \\ 1 \end{bmatrix} + B \begin{bmatrix} \frac{8}{\frac{1}{2} - \frac{\sqrt{129}}{2}} \\ 1 \end{bmatrix}$$

$$5 = \frac{8}{-1/2 + \frac{\sqrt{129}}{2}} A + \frac{8}{-\frac{1}{2} - \frac{\sqrt{129}}{2}} B$$

$$6 = A + B$$

→ solved on Maple

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = 4.497 \begin{bmatrix} \frac{8}{-1/2 + \sqrt{129}/2} \\ 1 \end{bmatrix} e^{\frac{1}{2} + \frac{\sqrt{129}}{2}t} + 1.503 \begin{bmatrix} \frac{8}{-1/2 - \frac{\sqrt{129}}{2}} \\ 1 \end{bmatrix} e^{\frac{1}{2} - \frac{\sqrt{129}}{2}t}$$

3) (ii)

$$x_1'(t) = x_1(t) + x_2(t) + x_3(t)$$
$$x_2'(t) = x_1(t) + x_2(t)$$
$$x_3'(t) = x_1(t)$$

$$\Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\lambda_1 = 2.247 + 1\times10^{-10}i, \quad \vec{v}_1 = \begin{bmatrix} 2.247 + 1.515\times10^{-9}i \\ 1.802 + 1.899\times10^{-9}i \\ 1 \end{bmatrix}$$

$$\lambda_2 = -0.802 - 1.866\times10^{-10}i, \quad \vec{v}_2 = \begin{bmatrix} -0.802 + 3.686\times10^{-10}i \\ 0.445 - 5.948\times10^{-10}i \\ 1 \end{bmatrix}$$

$$\lambda_3 = 0.555 - 1.340\times10^{-11}i, \quad \vec{v}_2 = \begin{bmatrix} 0.555 - 2.855\times10^{-11}i \\ -1.247 + 5.909\times10^{-11}i \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = A v_1 e^{\lambda_1 t} + B v_2 e^{\lambda_2 t} + C v_3 e^{\lambda_3 t}$$

$A = 0.522 - 3.891 \times 10^{-10} i$, $B = -0.495 + 2.020 \times 10^{-11} i$,

$C = -1.027 + 4.093 \times 10^{-10} i$

* these are large numbers, and i didn't want to make a mistake when rewriting it, so I left it here *

```
> #Nikita John, Assignment 17
  #November 1st, 2021
> Help17 :=proc( ) : print(` HW3g(u,v,w,M), HW2g(u,v,M) `) :end:



     #HW3g(u,v,w,M): The Hardy-Weinberg unerlying transformation with (u,v,w),
     GENERALIZED Eqs. with the 3 by 3 matrix M  (53a,53b,53c) in Edelestein-Keshet Ch. 3
  #Based on Anne Somalwar's solution of the bonus problem from hw15, see the end of
  #from https://sites.math.rutgers.edu/~zeilberg/Bio21/HW15posted/hw15AnneSomalwar.pdf
  HW3g :=proc(u, v, w, M) local tot, LI :
  LI := [

  M[1][1]*u^2 + (M[1][2] + M[2][1])/2*u*v + M[2][2]*(1/4)*v^2,

  (M[1][2] + M[2][1])/2*u*v + (M[1][3] + M[3][1])*u*w + M[2][2]/2*v^2
      + (M[2][3] + M[3][2])/2*v*w,

  M[2][2]*1/4*v^2 + (M[2][3] + M[3][2])/2*v*w  + M[3][3]*w^2] :
  tot := LI[1] + LI[2] + LI[3] :
  [LI[1]/tot, LI[2]/tot, LI[3]/tot] :
  end:


     #HW2g(u,v,M): The Generalized Hardy-Weinberg unerlying transformation with (u,v), M is
     the survival matrix. Based on Ann Somalwar's HW3g(u,v,w) (only retain the first two
     components and replace w by 1-u-v)
  HW2g :=proc(u, v, M) local LI, w :
  LI := HW3g(u, v, w, M) :
  normal(subs(w = 1-u-v, [LI[1], LI[2]])) :
  end:





  #OLD STUFF

  Help15 :=proc( ) : print(`HW3(u,v,w), HW2(u,v) , Dis1(F,y,y0,h,A), ToSys(k,z,f,INI) `) :end:



     #ToSys(k,z,f,INI): converts the kth order difference equation x(n)=f(x[n-1],x[n-2],...x[n-k]) to
     a first-order system
  #x1(n)=F(x1(n-1),x2(n-1), ...,xk(n-1))
  #x2(n)=x1(n-1)
  #...

     #xk(n)=x[k-1](n-1). It gives the underlying transformation phrased in terms of z[1],...z[k],
```

```
followed by the initial conditions. Try:
#ToSys:=proc(2,z,z[1]+z[2],[1,1])
ToSys :=proc(k, z, f, INI) local i :
[f, seq(z[i-1], i=2..k)], INI:
end:


#HW3(u,v,w): The Hardy-Weinberg unerlying transformation witu (u,v,w), Eqs. (53a,53b,
53c) in Edelestein-Keshet Ch. 3
HW3 :=proc(u, v, w) : [u^2 + u*v + (1/4)*v^2,  u*v + 2*u*w + 1/2*v^2 + v*w, 1/4
 *v^2 + v*w + w^2] :end:


#HW2(u,v): The Hardy-Weinberg unerlying transformation witu (u,v,w), Eqs. (53a,53b,53c)
in Edelestein-Keshet Ch. 3 using the fact that u+v+w=1
HW2 :=proc(u, v) : expand([u^2 + u*v + (1/4)*v^2 , u*v + 2*u*(1-u-v) + 1/2*v^2
 + v*(1-u-v)]) :end:




#Dis1(F,y,y0,h,A): The approximate orbit of the Dynamical system approximating the 1D for
the autonomous  continuous dynamical  process dy/dt=F(y(t)) , y(0)=y0 with mesh size h
from t=0 to t=A
Dis1 :=proc(F, y, y0, h, A) local L, x, i :
L := Orb(x + h*subs(y=x, F), x, y0, 0, trunc(A/h)) :

L := [seq([i*h, L[i]], i=1..nops(L))] :
end:

##old stuff

#M13.txt: Maple code for Lecture 13 of Dynamical Modesl in Biology, Fall 2021 (taught by
Dr. Z.)
Help13 :=proc() :
 print(`RT2(x,y,d,K), Orb2(F,x,y,pt0,K1,K2), FP2(F,x,y), SFP2(F,x,y), PlotOrb2(L), FP2drz
(F,x,y), SFP2drz(F,x,y) `) :end:

with(LinearAlgebra) :


#RT2(x,y,d,K): A random rational transformation of degree d from R^2 to R^2 with postiive
integer coefficients from 1 to K The inputs are variables x and y and
#the output is a pair of expressions  of (x,y) representing functions. It is for generating examples
#Try:
#RT2(x,y,2,10);
RT2 :=proc(x, y, d, K) local ra, i, j, f, g :
ra := rand(1..K) : #random integer from -K to K
```

```
f := add(add(ra( ) * x^i * y^j, j = 0 ..d - i), i = 0 ..d) / add(add(ra( ) * x^i * y^j, j = 0 ..d - i), i = 0
      ..d) :
g := add(add(ra( ) * x^i * y^j, j = 0 ..d - i), i = 0 ..d) / add(add(ra( ) * x^i * y^j, j = 0 ..d - i), i = 0
      ..d) :
[f, g] :
end:



   #Orb2(F,x,y,pt,K1,K2): Inputs a mapping F=[f,g] from R^2  to R^2 where f and g describe
   functions of x and y, an initial point pt0=[x0,y0]
#outputs the orbit starting at discrete time K1 and ending in discrete time K2. Try
#F:=RT2(x,y,2,10);
#Orb2(F,x,y,[1.1,1.2],1000,1010);
Orb2 :=proc(F, x, y, pt0, K1, K2)  local pt, L, i :
pt := pt0 :

for i from 1 to K1 - 1 do
pt := subs( {x = pt[1], y = pt[2]}, F) :
od:

L := [ ] :
for i from K1 to K2 do
L := [op(L), pt] :
pt := normal(subs( {x = pt[1], y = pt[2]}, F) ) :

od:
L :
end:

#FP2(F,x,y): The list of fixed points of the transformation [x,y]->F. Try
#FP2([x-y,x=y],x,y);
FP2 :=proc(F, x, y) local L, i :
L := [solve( {F[1] = x, F[2] = y}, {x, y}) ] :

[seq(subs(L[i], [x, y]), i = 1 ..nops(L) ) ] :
end:


#SFP2(F,x,y): The list of Stable fixed points of the transformation [x,y]->F. Try
#SFP2([(1+x)/(1+y), (1+7*y)/(4+x)],x,y);
SFP2 :=proc(F, x, y) local L, J, S, J0, i, pt, EV :

L := evalf(FP2(F, x, y) ) :
   #F is the list of ALL fixed points of the transformation [x,y]->F using the previous procedure
   FP2(F,x,y), but since we are interested in numbers we take the floating point version using
   evalf

J := Matrix(normal( [ [diff(F[1], x), diff(F[1], y) ], [diff(F[2], x), diff(F[2], y) ]]) ) :
```

```
    #J is the Jacobian matrix in general (in terms of the variables x and y). Note that J is a
    SYMBOLIC matrix featuring variables x and y

S := [ ]:  #S is the list of stable fixed points that starts out empty

for i from 1 to nops(L) do  #we examime it case by case
pt := L[i] : #pt is the current fixed point to be examined

J0 := subs({x=pt[1], y=pt[2]}, J) :
    #J0 is the NUMERICAL matrix obtained by plugging-in the examined fixed pt

EV := Eigenvalues(J0) :
    # We used Maple's command Eigenavalues to find the eigenvalues of this 2 by 2 matrix

if abs(EV[1]) < 1 and abs(EV[2]) < 1  then
 S := [op(S), pt] :
    #If both eigenvalues have absolute value less than 1 it means that they are stable, so we
    append the examined fixed point, pt, to the list of fixed points
fi:

od:
S : #the output is S
end:




###added Oct. 17, 20221
with(plots) :

PlotOrb1 :=proc(L) local i, d :

d := textplot([L[1], 0, 0]) :

for i from 2 to nops(L) do
d := d, textplot([L[i], 0, i-1]) :
od:
display(d) :
end:


PlotOrb2 :=proc(L) local i, d :

d := textplot([op(L[1]), 0]) :

for i from 2 to nops(L) do
d := d, textplot([op(L[i]), i-1]) :
od:
display(d) :
end:
```

```
###End added Oct. 17, 20221


###old stuff
#M11.txt: Maple code for Lecture 11 of Dynamical Models in Biology taught by Dr. Z.
Help11 :=proc( ) : print(` SFPe(f,x), Orbk(k,z,f,INI,K1,K2) `) :end:


   #SFPe(f,x): The set of fixed points of x->f(x) done exactly (and allowing symbolic
   parameters), followed by the condition of stability (if it is netween -1 and 1 it is stable)
#Try: FPe(k*x*(1-x),x);
#VERSION OF Oct. 12, 2021 (avoiding division by 0)
SFPe :=proc(f, x) local f1, L, i, M:
f1 := normal(diff(f, x)) :
L := [solve(numer(f-x), x)] :
M := [ ] :

for i from 1 to nops(L) do
 if subs(x=L[i], denom(f1) ) ≠ 0 then
  M := [op(M), [L[i], normal(subs(x=L[i], f1))]] :
 fi:
od:
M:

end:


#Added after class

   #Orbk(k,z,f,INI,K1,K2): Given a positive integer k, a letter (symbol), z, an expression f of z
   [1], ..., z[k] (representing a multi-variable function of the variables z[1],...,z[k]

   #a vector INI representing the initial values [x[1],..., x[k]], and (in applications) positive
   integres K1 and K2, outputs the

   #values of the sequence starting at n=K1 and ending at n=K2. of the sequence satisfying the
   difference equation
##x[n]=f(x[n-1],x[n-2],..., x[n-k+1]):

   #This is a generalization to  higher-order difference equation of procedure Orb(f,x,x0,K1,K2)
   . For example
#Orbk(1,z,5/2*z[1]*(1-z[1]),[0.5],1000,1010);  should be the same as
#Orb(5/2*z[1]*(1-z[1]),z[1],[0,5],1000,1010);
#Try:
#Orbk(2,z,(5/4)*z[1]-(3/8)*z[2],[1,2],1000,1010);
Orbk :=proc(k, z, f, INI, K1, K2) local L, i, newguy :
L := INI: #We start out with the list of initial values

if not (type(k, integer) and type(z, symbol) and type(INI, list) and nops(INI) = k and type(K1,
```

```
integer) and type(K2, integer) and K1 > 0 and K2 > K1) then
    #checking that the input is OK
 print(`bad input`) :
 RETURN(FAIL) :
fi:

while nops(L) < K2 do
 newguy := subs({seq(z[i]=L[-i], i=1..k)}, f) :
    #Using what we know about the value yesterday, the day before yesterday, ... up to k days
    before yesterday we find the value of the sequence today
 L := [op(L), newguy] : #we append the new value to the running list of values of our sequence
od:

[op(K1..K2, L)] :

end:


####STAFT FROM M9.txt
#M9.txt: Maple Code for "Dynamical models in Biology" (Math 336) taught by Dr. Z., Lecture 9

Help9 :=proc( ) :
     print(` Orb(f,x,x0,K1,K2), Orb2D(f,x,x0,K) , FP(f,x) , SFP(f,x) , Comp(f,x)   `) :end:


    #Orb(f,x,x0,K1,K2): Inputs an expression f in x (desccribing) a function of x, an initial point,
    x0, and a positive integer K, outputs
#the values of x[n] from n=K1 to n=K2. Try: where x[n]=f(x[n-1]), . Try:
#Orb(2*x*(1-x),x,0.4,1000,2000);
Orb :=proc(f, x, x0, K1, K2) local x1, i, L :
x1 := x0 :

for i from 1 to K1 do
 x1 := subs(x=x1, f) :
    #we don't record the first values of K1, since we are interested in the long-time behavior of
    the orbit
od:

L := [x1] :

for i from K1 to K2 do
 x1 := subs(x=x1, f) :   #we compute the next member of the orbit
 L := [op(L), x1] : #we append it to the list
od:

L : #that's the output

end:
```

```
#Orb2D(f,x,x0,K): 2D version of Orb(f,x,x0,0,K), just for illustration
Orb2D :=proc(f, x, x0, K) local L, L1, i :
L := Orb(f, x, x0, 0, K) :
L1 := [[L[1], 0], [L[1], L[2]], [L[2], L[2]] ] :
for i from 3 to nops(L) do
 L1 := [op(L1), [L[i-1], L[i]], [L[i], L[i]]] :
od:
L1 :
end:


#FP(f,x): The list of fixed points of the map x->f where f is an expression in x. Try:
#FP(2*x*(1-x),x);
FP :=proc(f, x)
evalf([solve(f=x, x)]) :
end:


#SFP(f,x): The list of stable fixed points of the map x->f where f is an expression in x. Try:
#SFP(2*x*(1-x),x);
SFP :=proc(f, x) local L, i, f1, pt, Ls :
L := FP(f, x) : #The list of fixed points (including complex ones)

Ls := [ ] :        #Ls is the list of stable fixed points, that starts out as the empty list

f1 := diff(f, x) :  #The derivative of the function f w.r.t. x

for i from 1 to nops(L) do
pt := L[i] :

if abs(subs(x=pt, f1)) < 1 then

 Ls := [op(Ls), pt] :  # if pt,  is stable we add it to the list of stable points

fi:

od:

Ls :   #The last line is the output

end:

#Comp(f,x): f(f(x))
Comp :=proc(f, x) : normal(subs(x=f, f) ) :end:




##added Oct. 17, 2021
#FP2drz(F,x,y): The list of fixed points of the transformation [x,y]->F. Dr. Z.'s way
```

```
#FP2([x-y,x+y],x,y);
FP2drz := proc(F, x, y) local eq, i, L, S1 :
eq := [numer(F[1]-x), numer(F[2]-y)] :

L := Groebner[Basis](eq, plex(x, y)) :

S1 := evalf([solve(L[1], y)]) :
[seq([solve(subs(y=S1[i], L[2]), x), S1[i]], i=1..nops(S1))] :
end:


#SFP2drz(F,x,y): The list of Stable fixed points of the transformation [x,y]->F. Try
#SFP2drz([(1+x)/(1+y), (1+7*y)/(4+x)],x,y);
SFP2drz := proc(F, x, y) local L, J, S, J0, i, pt, EV :

L := FP2drz(F, x, y) :
    #F is the list of ALL fixed points of the transformation [x,y]->F using the previous procedure
    FP2(F,x,y), but since we are interested in numbers we take the floating point version using
    evalf

J := Matrix(normal([[diff(F[1], x), diff(F[2], x)], [diff(F[1], y), diff(F[2], y)]])) :
    #J is the Jacobian matrix in general (in terms of the variables x and y). Note that J is a
    SYMBOLIC matrix featuring variables x and y

S := [ ] :  #S is the list of stable fixed points that starts out empty

for i from 1 to nops(L) do  #we examime it case by case
pt := L[i] : #pt is the current fixed point to be examined

J0 := subs({x=pt[1], y=pt[2]}, J) :
    #J0 is the NUMERICAL matrix obtained by plugging-in the examined fixed pt

EV := Eigenvalues(J0) :
    # We used Maple's command Eigenavalues to find the eigenvalues of this 2 by 2 matrix

if abs(EV[1]) < 1 and abs(EV[2]) < 1 then
 S := [op(S), pt] :
    #If both eigenvalues have absolute value less than 1 it means that they are stable, so we
    append the examined fixed point, pt, to the list of fixed points
fi:

od:
S : #the output is S
end:
> #1 (iii)
> dsolve({diff(x(t), t) = 3·x(t) − y(t), diff(y(t), t) = 2·x(t), x(0) = 2, y(0) = 3}, {x(t), y(t)});
```

$$\{x(t) = e^{2t} + e^{t}, y(t) = e^{2t} + 2\,e^{t}\} \quad (1)$$

```
> #2 (ii)
```

```
#a1 = 1, a2 = 8, a3 = 4, a5 = 0, a7 = 5, a8 = 6
with(LinearAlgebra) :
```

> #2 (i)

$$evalf\left(dsolve\left(\left\{ diff(y(t), t) = 4\cdot\left(6.946\cdot\exp\left(\left(\frac{1}{2}+\frac{\sqrt{129}}{2}\right)\cdot t\right) - 1.946\cdot\exp\left(\left(\frac{1}{2}-\frac{\sqrt{129}}{2}\right)\cdot t\right)\right), y(0)=6 \right\}, \{y(t)\} \right) \right);$$

$$y(t) = 1.503019455\, e^{-5.178908345\, t} + 4.496587171\, e^{6.178908345\, t} + 0.000393374 \qquad \textbf{(2)}$$

> #2 (ii)

$A := Matrix([[1, 8], [4, 0]]);$

$$A := \begin{bmatrix} 1 & 8 \\ 4 & 0 \end{bmatrix} \qquad \textbf{(3)}$$

> $Eigenvectors(A);$

$$\begin{bmatrix} \frac{1}{2}+\frac{\sqrt{129}}{2} \\ \frac{1}{2}-\frac{\sqrt{129}}{2} \end{bmatrix}, \begin{bmatrix} \frac{8}{-\frac{1}{2}+\frac{\sqrt{129}}{2}} & \frac{8}{-\frac{1}{2}-\frac{\sqrt{129}}{2}} \\ 1 & 1 \end{bmatrix} \qquad \textbf{(4)}$$

> $sys1 := \left\{ \frac{8}{-\frac{1}{2}+\frac{\sqrt{129}}{2}}\cdot x + \frac{8}{-\frac{1}{2}-\frac{\sqrt{129}}{2}}\cdot y = 5, x + y = 6 \right\}:$

$evalf(\ solve(sys1, \{x, y\}));$

$$\{x = 4.496766540, y = 1.503233460\} \qquad \textbf{(5)}$$

> #2 (iii)

$evalf(\ dsolve(\{diff(x(t), t) = x(t) + 8\cdot y(t), diff(y(t), t) = 4\cdot x(t), x(0) = 5, y(0) = 6\}, \{x(t), y(t)\}));$

$$\{x(t) = 6.946277074\, e^{6.178908345\, t} - 1.946277077\, e^{-5.178908345\, t}, y(t) = 4.496766540\, e^{6.178908345\, t} + 1.503233460\, e^{-5.178908345\, t}\} \qquad \textbf{(6)}$$

> #3

$B := Matrix([[1, 1, 1], [1, 1, 0], [1, 0, 0]]);$

$evalf(Eigenvectors(B));$

$$B := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 2.246979605 + 1.\times 10^{-10}\, I \\ -0.8019377358 - 1.866025404\times 10^{-10}\, I \\ 0.5549581322 - 1.339745960\times 10^{-11}\, I \end{bmatrix}, [[2.246979634 + 1.514675242\times 10^{-9}\, I, -0.8019377350 + 3.686305552\times 10^{-10}\, I, 0.5549581323 - 2.254559307\times 10^{-11}\, I], \qquad \textbf{(7)}$$

$[1.801937769 + 1.888769131 \times 10^{-9}\,\mathrm{I},\ 0.4450418682 - 5.947994638 \times 10^{-10}\,\mathrm{I},$
$-1.246979604 + 5.809451696 \times 10^{-11}\,\mathrm{I}],$
$[1., 1., 1.]]$

> $sys2 := \{(2.246979634 + 1.514675242 \times 10^{-9}\,\mathrm{I}) \cdot x + (-0.8019377350 + 3.686305552 \times 10^{-10}\,\mathrm{I}) \cdot y + (0.5549581323 - 2.254559307 \times 10^{-11}\,\mathrm{I}) \cdot z = 1, (1.801937769 + 1.888769131 \times 10^{-9}\,\mathrm{I}) \cdot x + (0.4450418682 - 5.947994638 \times 10^{-10}\,\mathrm{I}) \cdot y + (-1.246979604 + 5.809451696 \times 10^{-11}\,\mathrm{I}) \cdot z = 2, x + y + z = -1\}:$
> $evalf(solve(sys2, \{x, y, z\}));$

$\{x = 0.5218275374 - 3.891127619 \times 10^{-10}\,\mathrm{I}, y = -0.4952588736 - 2.020115403 \times 10^{-11}\,\mathrm{I}, z = -1.026568664 + 4.093139159 \times 10^{-10}\,\mathrm{I}\}$ **(8)**

```
> #4 (i)
  M := [[1, 1, 1], [1, 1, 1], [1, 1, 1]]:
  F := [u^2 + u*v + (1/4)*v^2 , u*v + 2*u*(1-u-v) + 1/2*v^2 + v*(1-u-v)]:
  HW2g(0.2, 0.8, M);
  Orb2(F, u, v, [0.05, 0.5], 1000, 1010);
```

[0.3600000000, 0.4800000000]

[[0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000], [0.09000000000, 0.4200000000]] **(9)**

```
> HW2g(0.63, 0.37, M);
  Orb2(F, u, v, [0.5, 0.05], 1000, 1010);
```

[0.6642250000, 0.3015500000]

[[0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000], [0.2756250000, 0.4987500000]] **(10)**

```
> HW2g(0.25, 0.75, M);
  Orb2(F, u, v, [1.5, 0.5], 1000, 1010);
```

[0.3906250000, 0.4687500000]

[[3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000], [3.062500000, −2.625000000]] **(11)**

```
> HW2g(0.366, 0.634, M);
  Orb2(F, u, v, [0.5, 0.35], 1000, 1010);
```

[0.4664890000, 0.4330220000]

(12)

[[0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000], [0.4556250000, 0.4387500000]] **(12)**

```
> HW2g(0.11, 0.89, M);
  Orb2(F, u, v, [0.5, 0.5], 1000, 1010);
```

[0.3080250000, 0.4939500000]

[[0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000]] **(13)**

```
> #4 (ii)
  M1 := [[0.1, 0.2, 0.14], [0.75, 0.5, 1], [0.55, 1, 0.1]]:
  HW2g(0.1, 0.1, M1);
  Orb2(F, u, v, [0.5, 0.5], 1000, 1010);
```

[0.02375296912, 0.4833729216]

[[0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000], [0.5625000000, 0.3750000000]] **(14)**

```
> M2 := [[0.1, 0.02, 0.014], [0.075, 0.05, 0.11], [0.55, 0.2, 0.1]]:
  HW2g(0.1, 0.1, M2);
  Orb2(F, u, v, [0.1, 1.1], 1000, 1010);
```

[0.01173278580, 0.4271100682]

[[0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000], [0.4225000000, 0.4550000000]] **(15)**

```
> M3 := [[1, 0.2, 0.75], [0.75, 0.5, 1], [0.55, 0.6, 0.1]]:
  HW2g(0.5, 0.5, M3);
  Orb2(F, u, v, [0.1, 1.5], 1000, 1010);
```

[0.6530612244, 0.2959183673]

[[0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, **(16)**

0.2550000000], [0.7225000000, 0.2550000000], [0.7225000000, 0.2550000000],
[0.7225000000, 0.2550000000]]

```
> M4 := [[1, 1, 0.14], [0.075, 1, 0.11], [1, 0.2, 1]]:
  HW2g(0.1, 0.1, M4);
  Orb2(F, u, v, [0.5, 1.1], 1000, 1010);
```

[0.02272005084, 0.1448681284]

[[1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000], [1.102500000, −0.1050000000]] **(17)**

```
> M5 := [[1, 0.5, 0.4], [0.35, 1, 0.81], [1, 0.2, 0.97]]:
  HW2g(0.1, 0.81, M5);
  Orb2(F, u, v, [0.67, 1.1], 1000, 1010);
```

[0.2514366084, 0.4968294501]

[[1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000]] **(18)**

```
> M6 := [[1, 0.5, 0.68], [0.35, 0.44, 0.81], [1, 0.88, 0.9]]:
  HW2g(0.3, 0.7, M6);
  Orb2(F, u, v, [0.67, 1.1], 1000, 1010);
```

[0.4816153687, 0.4070439992]

[[1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000], [1.488400000, −0.5368000000]] **(19)**

```
> M7 := [[0.77, 0.5, 0.3], [0.5, 1, 0.66], [0.82, 0.78, 0.97]];
  HW2g(0.5, 0.81, M7);
  Orb2(F, u, v, [0.7, 0.22], 1000, 1010);
```

M7 := [[0.77, 0.5, 0.3], [0.5, 1, 0.66], [0.82, 0.78, 0.97]]

[0.6887657352, 0.2170414461]

[[0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000], [0.6561000000, 0.3078000000]] **(20)**

```
> M8 := [[0.22, 0.22, 0.22], [0.22, 0.22, 0.22], [0.22, 0.22, 0.22]];
```

```
HW2g(0.2, 0.71, M8);
Orb2(F, u, v, [0.67, 0.33], 1000, 1010);
```

M8 := [[0.22, 0.22, 0.22], [0.22, 0.22, 0.22], [0.22, 0.22, 0.22]]

[0.3080250000, 0.4939500000]

[[0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000], [0.6972250000, 0.2755500000]] **(21)**

```
> M9 := [[0.65, 0.56, 0.65], [0.56, 0.65, 0.56], [0.65, 0.56, 0.65]];
HW2g(0.25, 0.61, M9);
Orb2(F, u, v, [1.23, 0.79], 1000, 1010);
```

M9 := [[0.65, 0.56, 0.65], [0.56, 0.65, 0.56], [0.65, 0.56, 0.65]]

[0.3071442805, 0.4935233161]

[[2.640625002, −2.031250002], [2.640625003, −2.031250005], [2.640625001, −2.031250002], [2.640625000, −2.031249999], [2.640625002, −2.031250003], [2.640625002, −2.031250002], [2.640625003, −2.031250005], [2.640625001, −2.031250002], [2.640625000, −2.031249999], [2.640625002, −2.031250003], [2.640625002, −2.031250002]] **(22)**

```
> M10 := [[0, 0.44, 0.4], [0.35, 0.2, 0], [0, 0.2, 0.97]];
HW2g(0.6, 0.5, M10);
Orb2(F, u, v, [0.23, 0.55], 1000, 1010);
```

M10 := [[0, 0.44, 0.4], [0.35, 0.2, 0], [0, 0.2, 0.97]]

[0.4986676818, 0.4358583936]

[[0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000], [0.2550250000, 0.4999500000]] **(23)**

```
>
```