```
> #Okay to Post
  #Nikita John, Assignment 12, October 17th, 2021
> #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)]):
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:
```

```
> #1(i)
 f1 := x^3 - 6*x^2 + 12*x - 6:
 FP(f1, x);
 SFP(f1, x);
```

$$[1., 2., 3.]$$

$$[2.] \quad (1)$$

> *#1(ii)*

$f2 := x^4 - \frac{13}{36} \cdot x^2 + x + \frac{1}{36}$:

$FP(f2, x)$;

$SFP(f2, x)$;

[0.3333333333, −0.5000000000, 0.5000000000, −0.3333333333]

[0.3333333333, −0.5000000000] **(2)**

>

Dynamic Modeling HW 12 (Okay to Post)

2) (i) $f(x,y) = f(x_0,y_0) + f_x(x_0,y_0)(x-x_0) + f_y(x_0,y_0)(y-y_0)$

$f(x,y) = \sqrt{x+4y}$ @ $(1,2)$

$f(1,2) = \sqrt{1+4(2)} = \sqrt{9} = 3$

$f_x = \frac{1}{2}(x+4y)^{-1/2} \Rightarrow f_x(1,2) = \frac{1}{2}(1+4(2))^{-1/2} = \frac{1}{2}\left(\frac{1}{\sqrt{9}}\right) = \frac{1}{6}$

$f_y = \frac{1}{2}(x+4y)^{-1/2} \cdot 4 = 2(x+4y)^{-1/2} \Rightarrow f_y(1,2) = 2(1+4(2))^{-1/2} = \frac{2}{3}$

$f^*(x,y) = 3 + \frac{1}{6}(x-1) + \frac{2}{3}(y-2)$ ← Linearization

point: $(0.95, 1.02)$

$f(0.95, 1.02) = \sqrt{0.95+4(1.02)} = 2.243$

$f^*(0.95, 1.02) = 3 + \frac{1}{6}(0.95-1) + \frac{2}{3}(1.02-2) = 2.338$

(ii) $f(x,y,z) = f(x_0,y_0,z_0) + f_x(x_0,y_0,z_0)(x-x_0) + f_y(x_0,y_0,z_0)(y-y_0) + f_z(x_0,y_0,z_0)(z-z_0)$

$f(x,y,z) = x^3y^4z^5$ @ $(1,1,1)$

$f(1,1,1) = (1)^3(1)^4(1)^5 = 1$

$f_x = 3x^2y^4z^5 \Rightarrow f_x(1,1,1) = 3(1)^2(1)^4(1)^5 = 3$

$f_y = 4x^3y^3z^5 \Rightarrow f_y(1,1,1) = 4(1)^3(1)^3(1)^5 = 4$

$f_z = 5x^3y^4z^4 \Rightarrow f_z(1,1,1) = 5(1)^3(1)^4(1)^4 = 5$

$f^*(x,y,z) = 1 + 3(x-1) + 4(y-1) + 5(z-1)$ ← Linearization

point: $(1.01, 1.02, 0.99)$

$f(1.01, 1.02, 0.99) = (1.01)^3(1.02)^4(0.99)^5 = 1.061$

$f^*(1.01, 1.02, 0.99) = 1 + 3(0.1) + 4(0.2) + 5(-0.01) = 1.06$

(iii) $f(x_1, x_2, x_3, x_4) = f(x_1^\circ, x_2^\circ, x_3^\circ, x_4^\circ) + f_{x_1}(x_1^\circ, x_2^\circ, x_3^\circ, x_4^\circ)(x_1 - x_1^\circ) +$
$f_{x_2}(x_1^\circ, x_2^\circ, x_3^\circ, x_4^\circ)(x_2 - x_2^\circ) + f_3(x_1^\circ, x_2^\circ, x_3^\circ, x_4^\circ)(x_3 - x_3^\circ)$
$+ f_4(x_1^\circ, x_2^\circ, x_3^\circ, x_4^\circ)(x_4 - x_4^\circ)$

$f(x_1, x_2, x_3, x_4) = \sqrt{x_1 + x_2 + x_3 + x_4}$ @ $(1,1,1,1)$

$f(1,1,1,1) = \sqrt{1+1+1+1} = \sqrt{4} = 2$

$f_{x_1} = \frac{1}{2}(x_1 + x_2 + x_3 + x_4)^{-1/2} \Rightarrow f_{x_1}(1,1,1,1) = \frac{1}{2}(1+1+1+1)^{-1/2} = \frac{1}{2}\left(\frac{1}{\sqrt{4}}\right) = \frac{1}{4}$

$f_{x_2} = \frac{1}{2}(x_1 + x_2 + x_3 + x_4)^{-1/2} \Rightarrow f_{x_2}(1,1,1,1) = \frac{1}{2}(1+1+1+1)^{-1/2} = \frac{1}{4}$

$f_{x_3} = f_{x_4} = f_{x_2} = f_{x_1} = 1/4$

↓ Linearization

$f^*(x_1, x_2, x_3, x_4) = 2 + \frac{1}{4}(x_1 - 1) + \frac{1}{4}(x_2 - 1) + \frac{1}{4}(x_3 - 1) + \frac{1}{4}(x_4 - 1)$

point: $(1.01, 1.01, 0.99, 0.99)$

$f(1.01, 1.01, 0.99, 0.99) = \sqrt{2(1.01) + 2(0.99)} = 2$

$f^*(1.01, 1.01, 0.99, 0.99) = 2 + \frac{1}{4}(0.01) + \frac{1}{4}(0.01) + \frac{1}{4}(0.01) + \frac{1}{4}(0.01) = 2.0025$

3) Jacobian Matrix: $\begin{bmatrix} f_x(x_0,y_0) & f_y(x_0,y_0) \\ g_x(x_0,y_0) & g_y(x_0,y_0) \end{bmatrix}$ point: $(1,1)$

$f(x,y) = \frac{x}{y+1}$

$f_x = \frac{1}{y+1} \Rightarrow f_x(1,1) = \frac{1}{1+1} = \frac{1}{2}$

$f_y = -\frac{x}{(y+1)^2} \Rightarrow f_y(1,1) = -\frac{1}{(1+1)^2} = -\frac{1}{4}$

$g(x,y) = \frac{y}{x+1}$

$g_x = -\frac{y}{(x+1)^2} \Rightarrow g_x(1,1) = \frac{-1}{(1+1)^2} = -\frac{1}{4}$

$g_y = \frac{1}{x+1} \Rightarrow g_y(1,1) = \frac{1}{1+1} = \frac{1}{2}$

Jacobian: $\begin{bmatrix} \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} \end{bmatrix}$

4) Jacobian Matrix: $\begin{bmatrix} f_x(x_0,y_0,z_0) & f_y(x_0,y_0,z_0) & f_z(x_0,y_0,z_0) \\ g_x(x_0,y_0,z_0) & g_y(x_0,y_0,z_0) & g_z(x_0,y_0,z_0) \\ h_x(x_0,y_0,z_0) & h_y(x_0,y_0,z_0) & h_z(x_0,y_0,z_0) \end{bmatrix}$ point: $(1,1,1)$

$f(x,y,z) = x + y + z$

$f_x = 1$

$f_y = 1$

$f_z = 1$

$g(x,y,z) = x^2 + y^2 + z^2$

$g_x = 2x \Rightarrow g_x(1,1,1) = 2$

$g_y = 2y \Rightarrow g_y(1,1,1) = 2$

$g_z = 2z \Rightarrow g_z(1,1,1) = 2$

$h(x,y,z) = x^3 + y^3 + z^3$

$h_x = 3x^2 \Rightarrow h_x(1,1,1) = 3$

$h_y = 3y^2 \Rightarrow h_y(1,1,1) = 3$

$h_z = 3z^2 \Rightarrow h_z(1,1,1) = 3$

Jacobian: $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$

5) It makes sense that a fixed point is a stable fixed point if its Jacobian Matrix has eigenvalues w/ absolute value less than 1 because when all the eigenvalues are less than 1 then the points in the neighborhood of the fixed point will go back to the fixed point.