```
> #work until 11:00 am
> #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:

 #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);
SFPe :=proc( f, x ) local f1, L, i :
f1 := diff ( f, x ) :
L := [solve( f=x, x ) ] :
[seq( [L[i], normal(subs(x = L[i], f1 ) ) ], i = 1 ..nops(L) ) ] :

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:
```

```
> #QUESTION 1:
#For each of the two functions, findall the fixed points, and for
each of
#them, decide whether they are stable fixed points.
```

```
> #(i)   x ⇒ x^3 − 6 x^2 + 12 x − 6
 #
 #

#Fixed points
print("fixed points");
FP(x^3 - 6*x^2 + 12*x - 6, x);
#Stable Fixed Points
print("stable fixed points");
SFP(x^3 - 6*x^2 + 12*x - 6, x);
#Answer: 2 is our ONLY FIXED POINT
#Just to verify:
#When IC is 1
Orb(x^3 - 6*x^2 + 12*x - 6, x, 1.000001,1000,1020);
```

```
#When IC is 1.0001
```

"fixed points"

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

"stable fixed points"

$[2.]$

[2.000000000, 2.000000000] **(1)**

> $\#(ii)\ x \Rightarrow x^4 - \frac{13}{36} x^2 + x + \frac{1}{36}$

```
#
#
#Fixed Points
print("fixed points");
FP(x^4 - 13/36*x^2 + x - 1/36, x);
print("stable fixed points");
#Stable Fixed Points
SFP(x^4 - 13/36*x^2 + x - 1/36, x);
```

"fixed points"

[0.2552723494 I, -0.2552723494 I, 0.6528974525, −0.6528974525]

"stable fixed points"

[−0.6528974525] **(2)**

```
> #QUESTION 2:
#Find the linearizations of the given functions at the designated points, and compare the exact value with the approximate value given by the linearization
```

> # $f(x,y) = \sqrt{x + 4y}$ *at* $\cdot (1, 2)$. *The values at* $\cdot (0.95, 1.02)$

```
> #Exact value
#What tool will we use to find the exact value of the function? probably fsolve
```

> $fun := \sqrt{x + 4};$

$$fun := \sqrt{x+4} \quad \textbf{(3)}$$

> $solve(fun, x);$

#The solve finds the root. so how do we find the place where $\sqrt{x+4} = 1$ *?*

$solve(fun = 1, x)$

#THAT IS THE CORRECT WAY TO USE THE SOLVE COMMAND

-4

-3 **(4)**

```
> #Now we have to use the solve command to find the appropriate x and y (solving a system)
```

> $multifun := \sqrt{x + y};$

(5)

$$multifun := \sqrt{x+y} \quad (5)$$

> $solve(multifun = 1, [x, y]);$
  #Cool I understand how that works.

$$[[x = -y + 1, y = y]] \quad (6)$$

```
> #Now, How do i figure out
```

> $solve(multifun, [x = 1, y = 2]);$

Warning, solving for expressions other than names or functions is not recommended.

$$[\,] \quad (7)$$

```
> #I have to try something else. I know how to do this in my head,
  and with substitution
  #I know that I could just create a proc to fix everything. maybe
  that is best way
```

> $multifunProc := \mathbf{proc}(x, y)\ \mathbf{local}\ F:$
  $F := \sqrt{x+y}\ :$

  end:

No i dont need to do that, I learned how to use the subs command

> $F1 := \sqrt{x+y}$

$$F1 := \sqrt{x+y} \quad (8)$$

```
> #In hindsight, trying to use the solve command for that purpose
  is kinda stupid? YES! Use the subs() command instead!
> #Now for the Linearizations (DO PARTIAL derivatives!)
```

> *#here, we will find the sum* $\sqrt{x_0 + y_0} + \frac{\partial}{\partial x}\left(\sqrt{x+y}\right)(x - x_0) + \frac{\partial}{\partial y}\left(\sqrt{x+y}\right)(y - y_0)$

```
> #Which evaluates to
```

> $Linearized1 := \left(\sqrt{x_0 + y_0}\right) + \frac{1}{2} \cdot (x+y)^{-\frac{1}{2}} \cdot (x - x_0) + \frac{1}{2}(x+y)^{-\frac{1}{2}} \cdot (y - y_0)$

$$Linearized1 := \sqrt{x_0 + y_0} + \frac{x - x_0}{2\sqrt{x+y}} + \frac{y - y_0}{2\sqrt{x+y}} \quad (9)$$

Now just substitute

```
> plugged1 := subs({x__0=0.95,y__0=1.02},Linearized1);
  print("answer to 2 part (i)");
  print("1st degree linear approximation");
  evalf(subs({x = 1,y = 2},plugged1));
  print("maple default");
  evalf(subs({x=1,y=2},F1));
```

$$plugged1 := 1.403566885 + \frac{x - 0.95}{2\sqrt{x+y}} + \frac{y - 1.02}{2\sqrt{x+y}}$$

"answer to 2 part (i)"

"1st degree linear approximation"

1.700902274

"maple default"

1.732050808 **(10)**

> $F2 := x^3 \cdot y^4 \cdot z^5;$

$$F2 := x^3 y^4 z^5 \quad \textbf{(11)}$$

```
> Linearized2 := subs({x=x__0,y=y__0,z=z__0},F2)+1/2*(diff(F2,x)*
(x-x__0)+diff(F2,y)*(y-y__0)+diff(F2,z)*(z-z__0));
```

$$Linearized2 := x_0^3 y_0^4 z_0^5 + \frac{3 x^2 y^4 z^5 (x - x_0)}{2} + 2 x^3 y^3 z^5 (y - y_0) + \frac{5 x^3 y^4 z^4 (z - z_0)}{2} \quad \textbf{(12)}$$

```
> plugged2 := subs({x__0 = 1.01,y__0 = 1.02 ,z__0 = 0.99},
Linearized2);
```

$$plugged2 := 1.060573524 + \frac{3 x^2 y^4 z^5 (x - 1.01)}{2} + 2 x^3 y^3 z^5 (y - 1.02) + \frac{5 x^3 y^4 z^4 (z - 0.99)}{2} \quad \textbf{(13)}$$

```
> print("answer to 2 part (ii)");
print("1st degree linear approximation");
evalf(subs({x=1,y=1,z=1},plugged2));
print("maple default");
evalf(subs({x=1,y=1,z=1},F2));
```

"answer to 2 part (ii)"

"1st degree linear approximation"

1.030573524

"maple default"

1. **(14)**

> $F3 := \sqrt{x1 + x2 + x3 + x4};$

$$F3 := \sqrt{x1 + x2 + x3 + x4}$$

$$\sqrt{1 + x2 + x3 + x4} \quad \textbf{(15)}$$

```
> Linearized3 :=subs({x1=x1__0,x2=x2__0,x3=x3__0,x4=x4__0},F3)+1/2*
(diff(F3,x1)*(x1-x1__0)+diff(F3,x2)*(x2-x2__0)+diff(F3,x3)*(x3-
x3__0)+diff(F3,x4)*(x4-x4__0));
```

$$Linearized3 := \sqrt{x1_0 + x2_0 + x3_0 + x4_0} + \frac{x1 - x1_0}{4\sqrt{x1 + x2 + x3 + x4}} + \frac{x2 - x2_0}{4\sqrt{x1 + x2 + x3 + x4}} + \frac{x3 - x3_0}{4\sqrt{x1 + x2 + x3 + x4}} + \frac{x4 - x4_0}{4\sqrt{x1 + x2 + x3 + x4}} \quad \textbf{(16)}$$

```
> plugged3 := subs({x1__0=1.01,x2__0=1.01,x3__0=0.99,x4__0=0.99},
Linearized3);
```

$$plugged3 := 2.000000000 + \frac{x1 - 1.01}{4\sqrt{x1 + x2 + x3 + x4}} + \frac{x2 - 1.01}{4\sqrt{x1 + x2 + x3 + x4}} \quad \textbf{(17)}$$

$$+\frac{x3-0.99}{4\sqrt{x1+x2+x3+x4}}+\frac{x4-0.99}{4\sqrt{x1+x2+x3+x4}}$$

```
> print("answer to 2 (iii)");
print("first degree linear approximation");

evalf(subs({x1=1,x2=1,x3=1,x4=1},plugged3));
print("maple default");
evalf(subs({x1=1,x2=1,x3=1,x4=1},F3));
```

"answer to 2 (iii)"

"first degree linear approximation"

2.000000000

"maple default"

2.000000000 **(18)**

```
> #QUESTION 3
#What is the Jacobian MATRIX (NOT Jacobian determinant) of the following transformation
```

$$> \#(x,y)\rightarrow\left(\frac{x}{y+1},\frac{y}{x+1}\right)$$

```
> #My guess is that we first create a system of equations by using the solve command to get our fixed points?
#Is finding a fixed point even relevant? Do we even need to make a system of equations
> #Since we are dealing with 2 unknowns x and y, we should have 2 equations
```

$$> solve\left(\left\{\frac{x}{y+1},\frac{y}{x+1}\right\},\{x,y\}\right);$$

$$\{x=0, y=0\} \quad \textbf{(19)}$$

```
> #OK great, but there is more than one solution, not just {x=0,y=0}. we can have {x=y,y=x} work for all values except x=y=-1
#Although, does x=y tell us anything? Of course it is not the original transformation, but is it special in any way? #Not relevant to problem
> #Maybe the question is much less complicated than I imagine it to be. *THE ANSWER IS BELOW*
```

$$> \# \begin{bmatrix} \frac{\partial}{\partial x}\left(\frac{x}{y+1}\right) & \frac{\partial}{\partial y}\left(\frac{x}{y+1}\right) \\ \frac{\partial}{\partial x}\left(\frac{y}{x+1}\right) & \frac{\partial}{\partial y}\left(\frac{y}{x+1}\right) \end{bmatrix}$$ *is the CORRECT JACOBIAN MATRIX*

\# *There is no need to do further computation WRONG I still need to evaluate it at the point (1,1)*

#COMPUTED MATRIX ↓↓↓

$$\# \begin{bmatrix} \frac{1}{y+1} & -\frac{x}{(y+1)^2} \\ -\frac{y}{(x+1)^2} & \frac{1}{x+1} \end{bmatrix}$$

#Which at point (1,1) evaluates to ↓↓↓

$$\#\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

```
> #QUEStion 4
  #What is the jacobian matrix of
```

$(x, y, z) \to (x + y + z,\ \ x^2 + y^2 + z^2,\ \ x^3 + y^3 + z^3)$ *at point*$\cdot$ (1, 1, 1)

```
> #our Jacobian matrix before evaluating the derivatives is:
```

$$\begin{bmatrix} \frac{\partial}{\partial x}(x+y+z) & \frac{\partial}{\partial y}(x+y+z) & \frac{\partial}{\partial z}(x+y+z) \\ \frac{\partial}{\partial x}(x^2+y^2+z^2) & \frac{\partial}{\partial y}(x^2+y^2+z^2) & \frac{\partial}{\partial z}(x^2+y^2+z^2) \\ \frac{\partial}{\partial x}(x^3+y^3+z^3) & \frac{\partial}{\partial y}(x^3+y^3+z^3) & \frac{\partial}{\partial z}(x^3+y^3+z^3) \end{bmatrix}$$

Which evaluates to

$$\begin{bmatrix} 1 & 1 & 1 \\ 2x & 2y & 2z \\ 3x^2 & 3y^2 & 3z^2 \end{bmatrix}$$

Which at point (1, 1, 1) evaluates to ↓↓↓

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

```
> QUESTION 5: In your words explain why it makes that a fixed point
  VVV
```

> #\`\` (x_0, y_0)

```
> #Of a transformation
```

> $\#(x, y) \to (f(x, y), g(x, y))$

```
> #i.e. a point in R^2 such that
```

> \# $x_0 = f(x_0,\ y_0)$ **and** $y_0 = g(x_0,\ y_0)$ *creates a stable fixed point* $\cdot (x_0, y_0)$
> **if** *all of the eigenvalues are less than* 1

The fixed point is a stable fixed point if all of the eigenvalues of a matrix have absolute value less than 1 because if the eigenvalue diagonal matrix A of the Jacobian matrix is part of a recurrence
$A(n) = A(n-1)(x, y)$
 the solutions will tend to zero, implying stability

```
> #Questions for Dr Z:
     #Are there any instances when imaginary fixed points are
  important?
     #My best guess could be yes, but not in biology, but what if
  there is an important application of imaginary fixed points in
```

biology

#Dr Z said that fixed points being imaginary numbers appear in electrodynamics. ex. impedance

#On Stack exchange, they say that the Jacobian can be used to linearize stuff at complex