PYTHON CODE FROM TEXT "CALCULUS IN CONTEXT".

M. AUTH

1. Calculus in Context

1.1. **Program: SIR.**

```
t = 0  # intital time

# intial conditions
S = 45400.0
I = 2100.0
R = 2500.0

# stepsize = h = deltat
deltat = 1

print(t,S,I,R) # print inital conditions

# for loop implements Euler's method
for k in range(3):
    # System of ODE
    Sprime = -.00001*S*I
    Iprime = .00001*S*I - I/14
    Rprime = I/14

    # Approximate the rate of change
    deltaS = Sprime*deltat
    deltaI = Iprime*deltat
    deltaR = Rprime*deltat

    # Update all values each time through loop
    t += deltat
    S += deltaS
    I += deltaI
    R += deltaR

    print(t,str(round(S,1)),str(round(I,1)),str(round(R,1)))
```

1.2. **Program: SIRPLOT.**

```
from pylab import plot, show

# intial conditions
t = 0  # intital time
S = 45400.0 # intial Susceptibles
I = 2100.0 # initial Infecteds
R = 2500.0 # initial Recovereds

# make lists in order to graph
tList = [t]
sList = [S]
iList = [I]
rList = [R]

# stepsize = h = deltat
deltat = 0.1

# for loop implements Euler's method
for k in range(30):
    # System of ODE
    Sprime = -.00001*S*I
    Iprime = .00001*S*I - I/14
    Rprime = I/14

    # Approximate the rate of change
    deltaS = Sprime*deltat
    deltaI = Iprime*deltat
    deltaR = Rprime*deltat

    # Update all values each time through loop
    t += deltat
    S += deltaS
    I += deltaI
    R += deltaR

    # update each list each time through loop
    tList.append(t)
    sList.append(S)
    iList.append(I)
    rList.append(R)

# plot functions of time, S(t), I(t), R(t).
plot(tList,sList)
plot(tList,iList)
plot(tList,rList)
show()
```

1.3. **Program: SEQUENCE.**

```
# Exercise 11 from section 2.2 on page 87
from pylab import plot, show
from math import exp

# the exact solution
def f(t):
    return 5*exp(t)/(4 + exp(t))

deltat = 0.1 # stepsize = h = deltat for exact
t = 0 # initial time

# Exact Solution
tList = [t]
yExact = [f(0)]  # list of exact solution

# for loop implements Euler's method
for k in range(30):
    t += deltat

    # update each list each time through loop
    tList.append(t)
    yExact.append(f(t))

plot(tList,yExact) # plot exact solution

# for loop to implement several Euler's methods
# intial conditions
for k in range(3):
    t = 0  # intital time
    y = 1  # initial value y(0)

    # make lists in order to graph
    tList = [t]
    yList = [y]
    deltat = (.5)**k
    n = 3*2**k

    for k in range(n):
        # System of ODE
        yprime = 0.2*y*(5-y)

        # Approximate the rate of change
        deltay = yprime*deltat

        # Update all values each time through loop
        t += deltat
        y += deltay
```

```
        # update each list each time through loop
        tList.append(t)
        yList.append(y)
        # plot approximate solution
    plot(tList,yList)

show()
```

1.4. **Program: BABYLON.**

```
a = 5
x = 2  # initial guess
n = 6  # number of steps

for k in range(n):
    x = (x + a/x)/2
    print(str(round(x,6)))
```

References