```
> #OK to post Homework
  #Jeton Hida, Assignment 20, November 15, 2021
  read "/Users/jeton/Desktop/Math 336/DMB.txt"
```

First Written: Nov. 2021

This is DMB.txt, A Maple package to explore Dynamical models in Biology (both discrete and continuous)
accompanying the class Dynamical Models in Biology, Rutgers University. Taught by Dr. Z. (Doron Zeilbeger)

The most current version is available on WWW at:
http://sites.math.rutgers.edu/~zeilberg/tokhniot/DMB.txt .
Please report all bugs to: DoronZeil at gmail dot com .

For general help, and a list of the MAIN functions,
type "Help();". For specific help type "Help(procedure_name);"

For a list of the supporting functions type: Help1();
For help with any of them type: Help(ProcedureName);

For a list of the functions that give examples of Discrete-time dynamical systems (some famous), type: HelpDDM();
For help with any of them type: Help(ProcedureName);

For a list of the functions continuous-time dynamical systems (some famous) type: HelpCDM();
For help with any of them type: Help(ProcedureName);

------------------------------ **(1)**

```
> #Question 1 i.
> Help(SIRS)
```

SIRS(s,i,beta,gamma,nu,N): The SIRS dynamical model with parameters beta,gamma, nu,N (see section 6.6 of Edelstein-Keshet), s is the number of
Susceptibles, i is the number of infected, (the number of removed is given by N-s-i). N is the total population. Try:

SIRS(s,i,beta,gamma,nu,N); **(2)**

```
> Help(EquP)
```

EquP(F,x): Given a transformation F in the list of variables finds all the Equilibrium points of *the continuous-time dynamical system x'(t)=F(x(t))*

*EquP([5/2*x*(1-x)],[x]);*

EquP([y(1-x-y),x*(3-2*x-y)],[x,y]);* **(3)**

```
> Help(SEquP)
```

SEquP(F,x): Given a transformation F in the list of variables finds all the Stable Equilibrium points of the continuous-time dynamical system x'(t)=F(x(t))

*SEquP([5/2*x*(1-x)],[x]);*

SEquP([y(1-x-y),x*(3-2*x-y)],[x,y]);* **(4)**

```
> Help(TimeSeries)
```

TimeSeries(F,x,pt,h,A,i): Inputs a transformation F in the list of variables x

The time-series of x[i] vs. time of the Dynamical system approximating the the autonomous continuous dynamical process

dx/dt=F(x(t)) by a discrete time dynamical system with step-size h from t=0 to t=A

Try:

TimeSeries([x(1-y),y*(1-x)],[x,y],[0.5,0.5], 0.01, 10,1);* **(5)**

```
> Help(PhaseDiag)
```

PhaseDiag(F,x,pt,h,A): Inputs a transformation F in the list of variables x (of length 2), i.e. a mapping from R^2 to R^2 gives the

The phase diagram of the solution with initial condition x(0)=pt

dx/dt=F[1](x(t)) by a discrete time dynamical system with step-size h from t=0 to t=A

Try:

PhaseDiag([x(1-y),y*(1-x)],[x,y],[0.5,0.5], 0.01, 10);* **(6)**

```
> F:=SIRS(s,i,(.3)*2/1000,5,2,1000)
```

$$F := [-0.0006000000000\, s\, i + 5000 - 5\, s - 5\, i,\ 0.0006000000000\, s\, i - 2\, i] \quad (7)$$

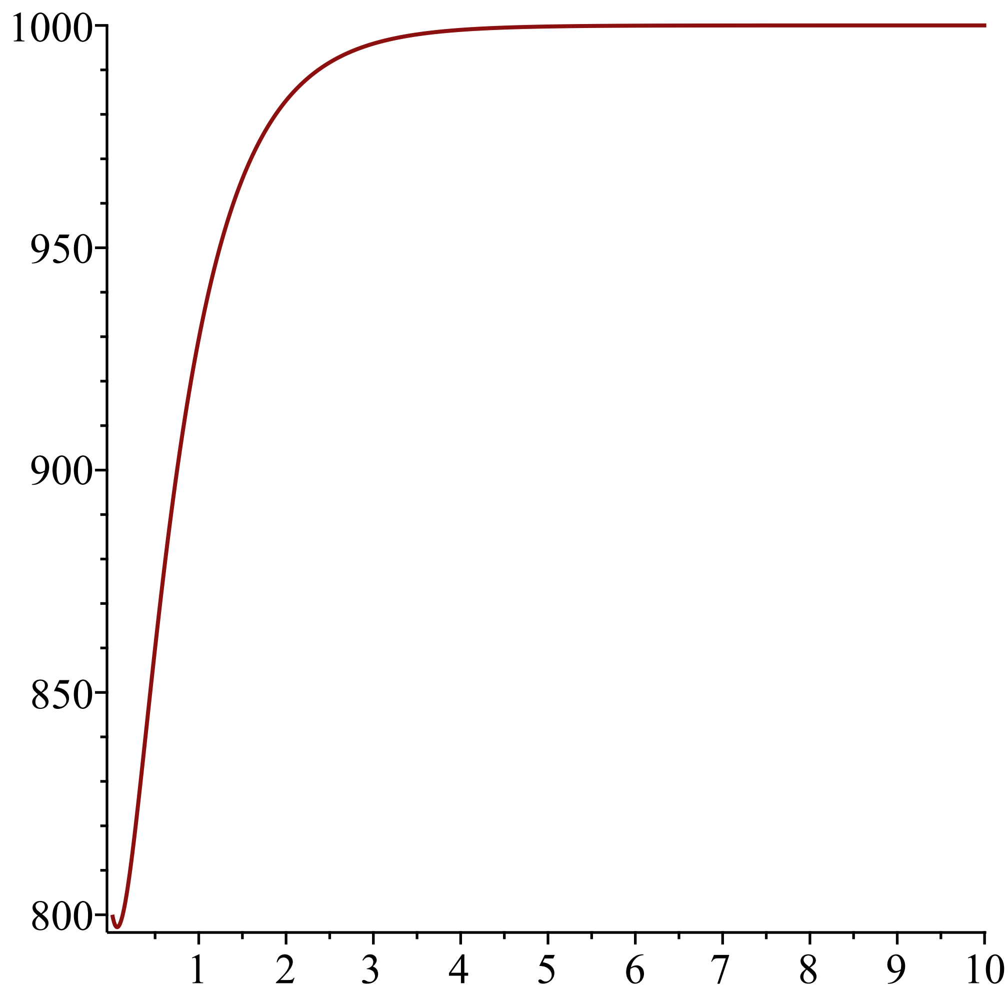
```
> EquP(F,[s,i])
```

$$\{[1000., 0.], [3333.333333, -1666.666667]\} \quad (8)$$

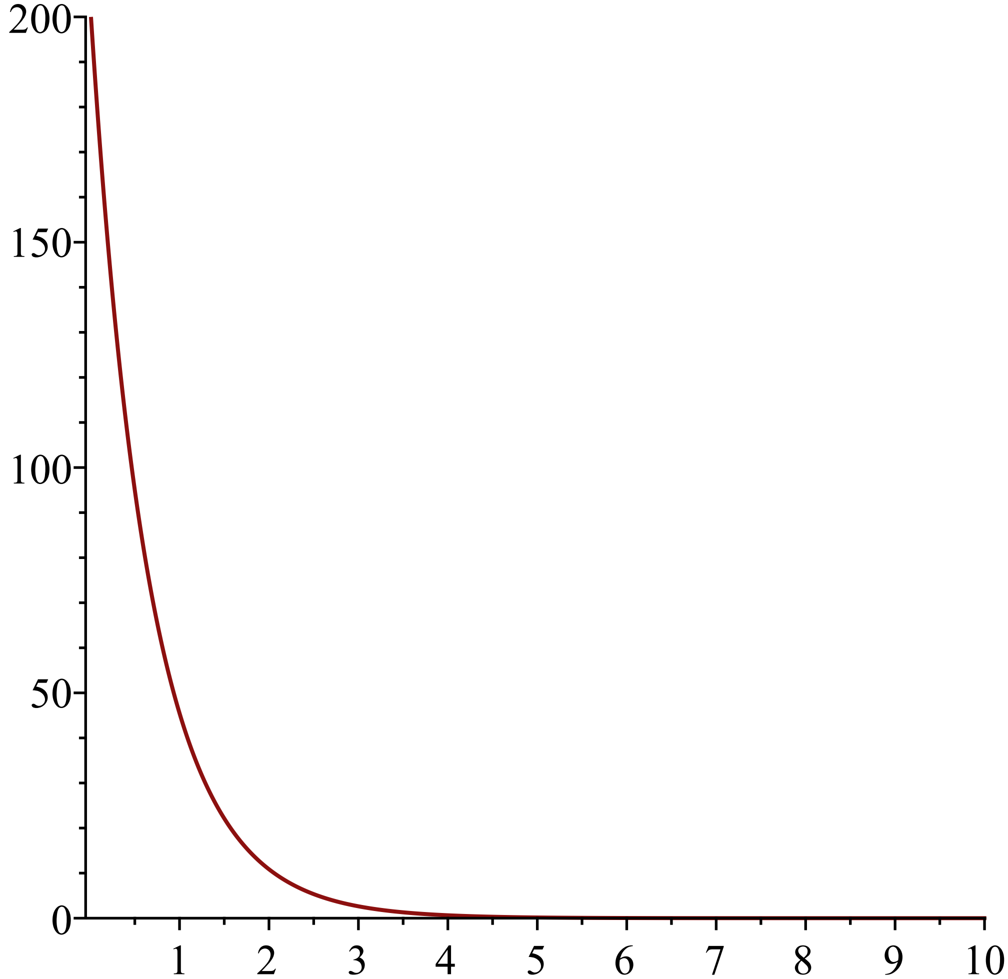
```
> SEquP(F,[s,i])
```

$$\{[1000., 0.]\} \quad (9)$$

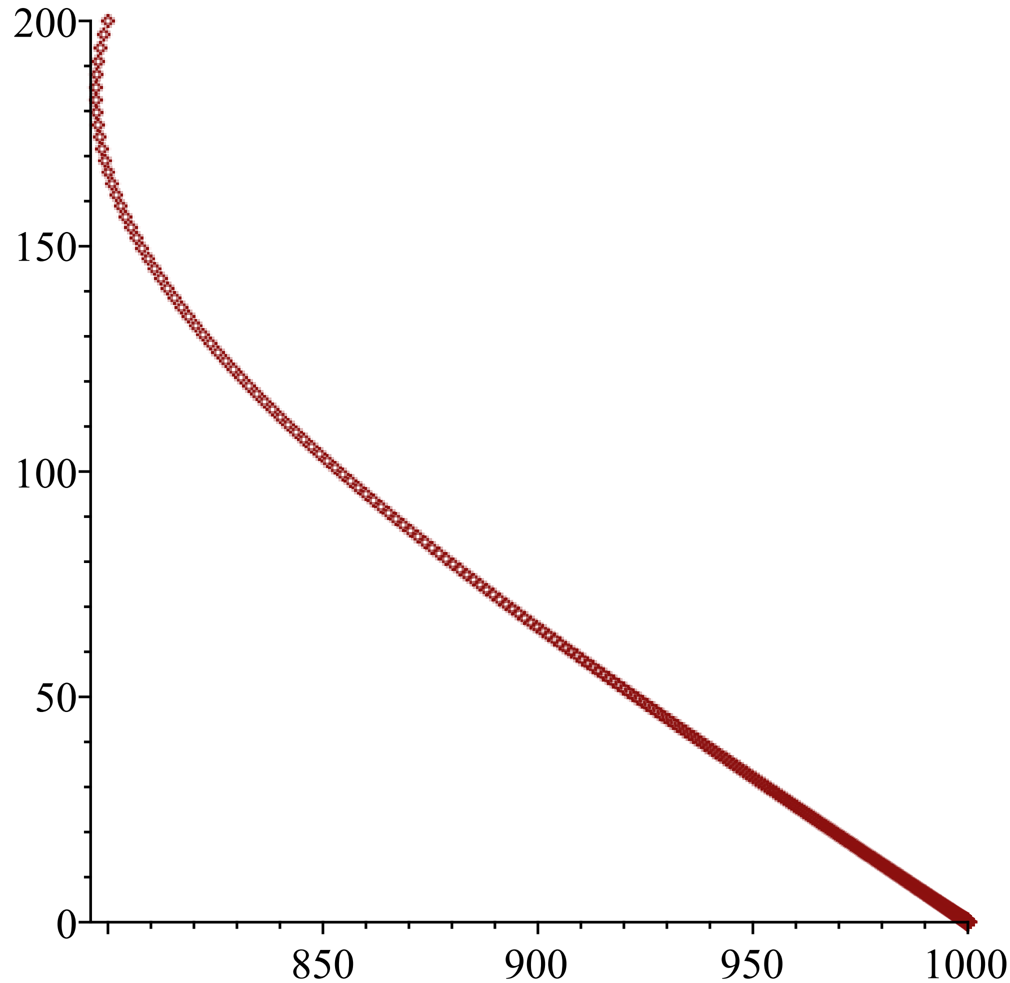
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,(.9)*2/1000,5,2,1000)
```

$$F := [-0.001800000000\, s\, i + 5000 - 5\, s - 5\, i,\ 0.001800000000\, s\, i - 2\, i] \quad \textbf{(10)}$$

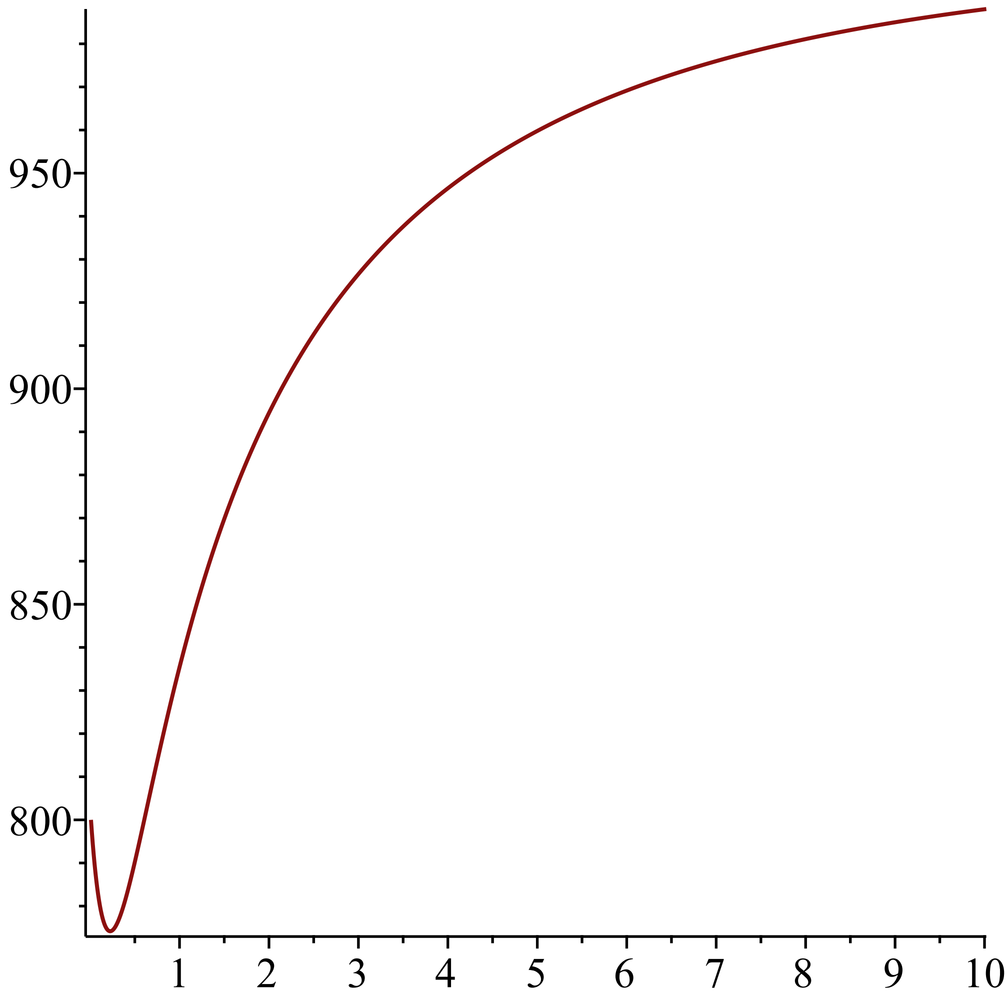
```
> EquP(F,[s,i])
```

$$\{[1000., 0.], [1111.111111, -79.36507937]\} \quad \textbf{(11)}$$

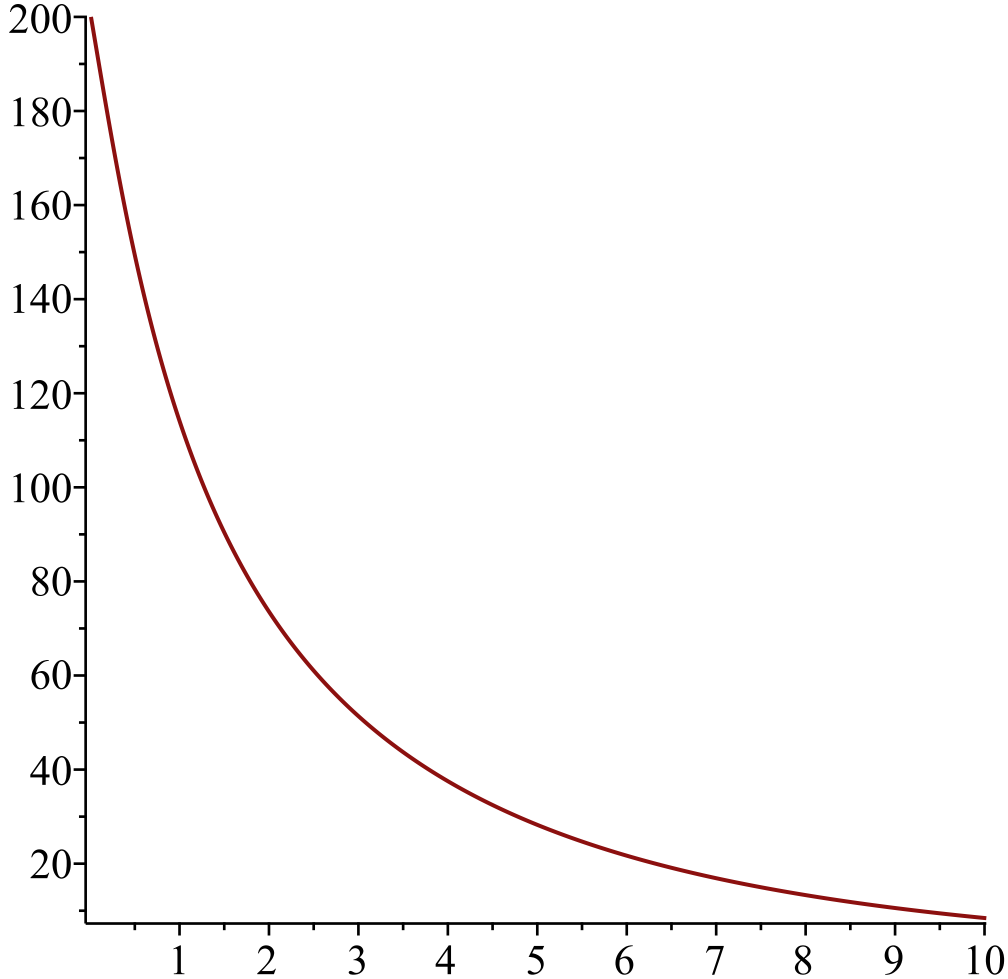
```
> SEquP(F,[s,i])
```

$$\{[1000., 0.]\} \quad \textbf{(12)}$$

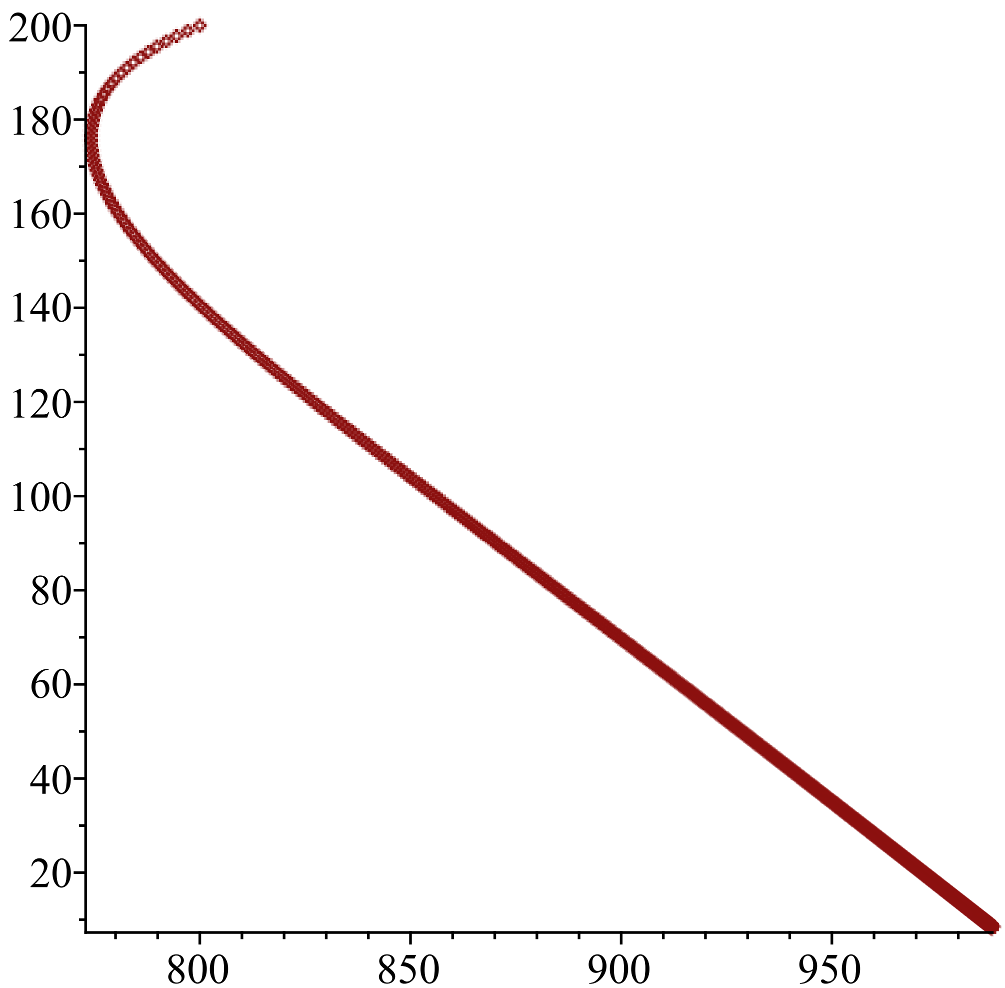
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,(3.9)*2/1000,5,2,1000)
```

$$F := [-0.007800000000\, s\, i + 5000 - 5\, s - 5\, i,\ 0.007800000000\, s\, i - 2\, i] \quad \textbf{(13)}$$

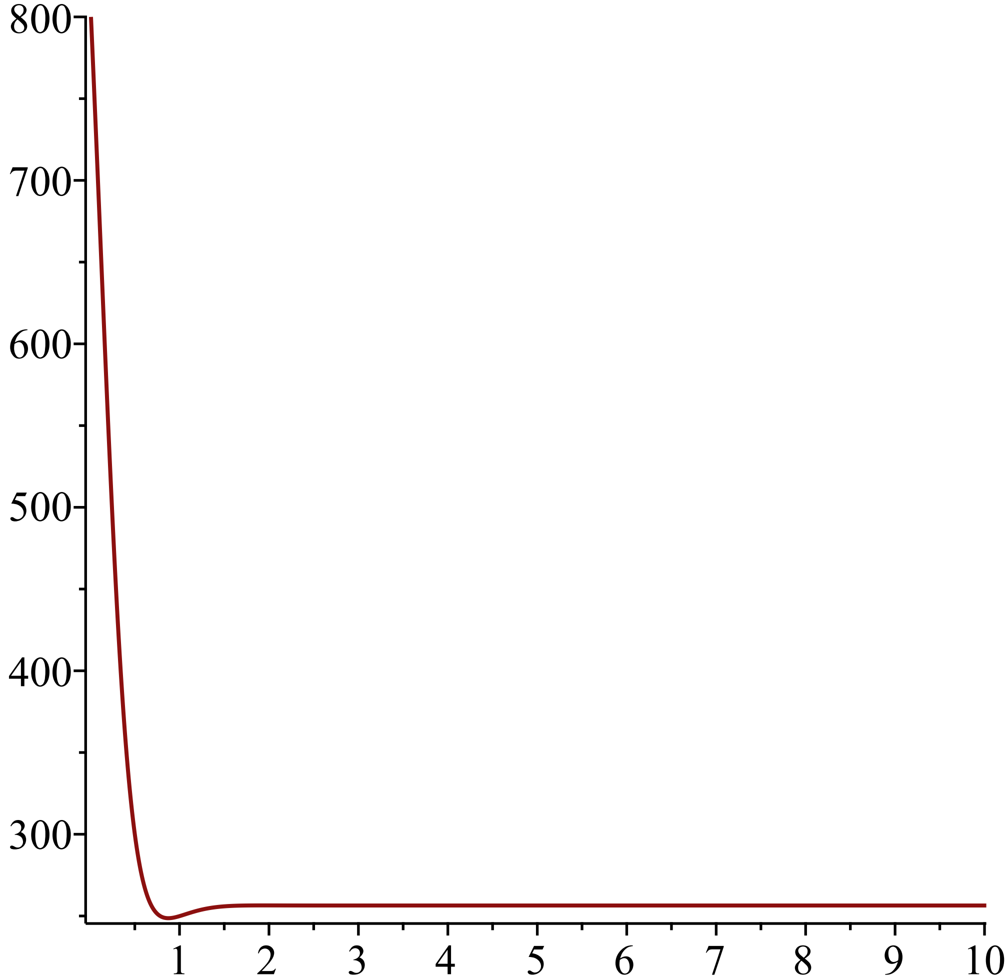
```
> EquP(F,[s,i])
```

$$\{[256.4102564, 531.1355311], [1000., 0.]\} \quad \textbf{(14)}$$

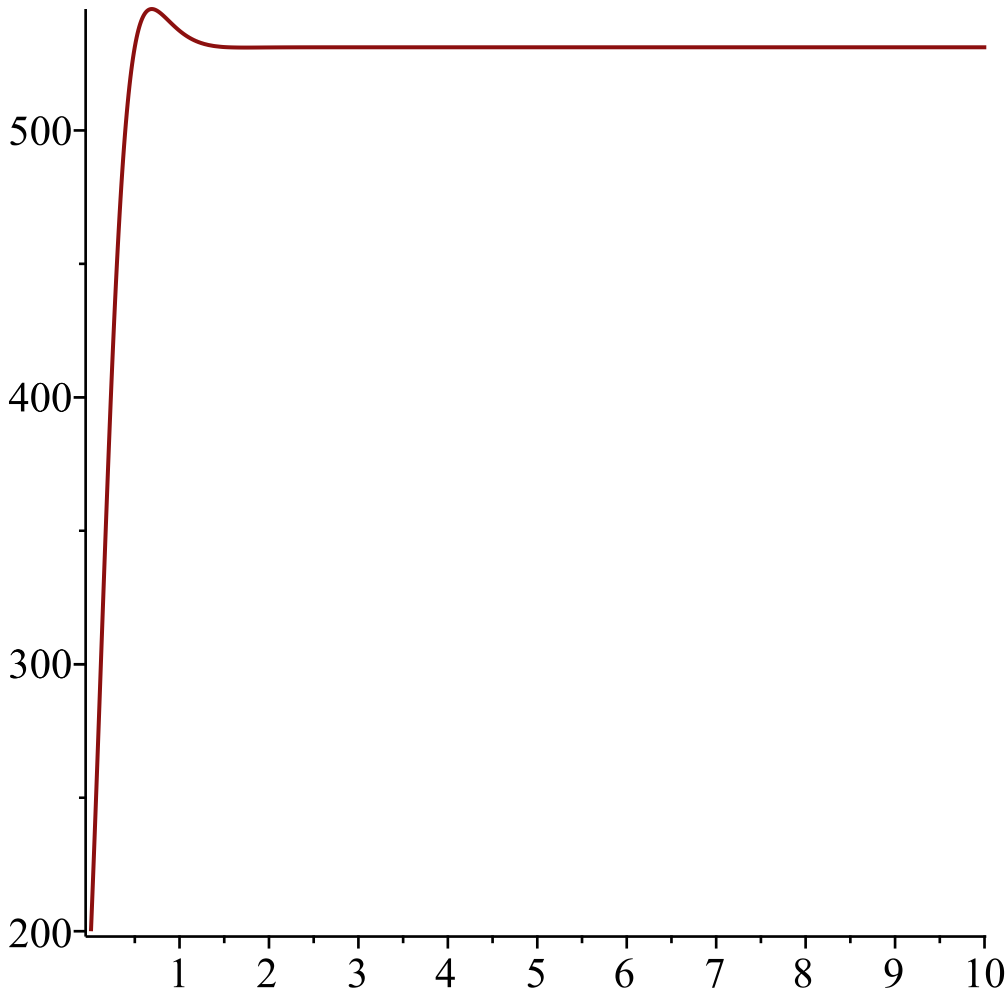
```
> SEquP(F,[s,i])
```

$$\{[256.4102564, 531.1355311]\} \quad \textbf{(15)}$$

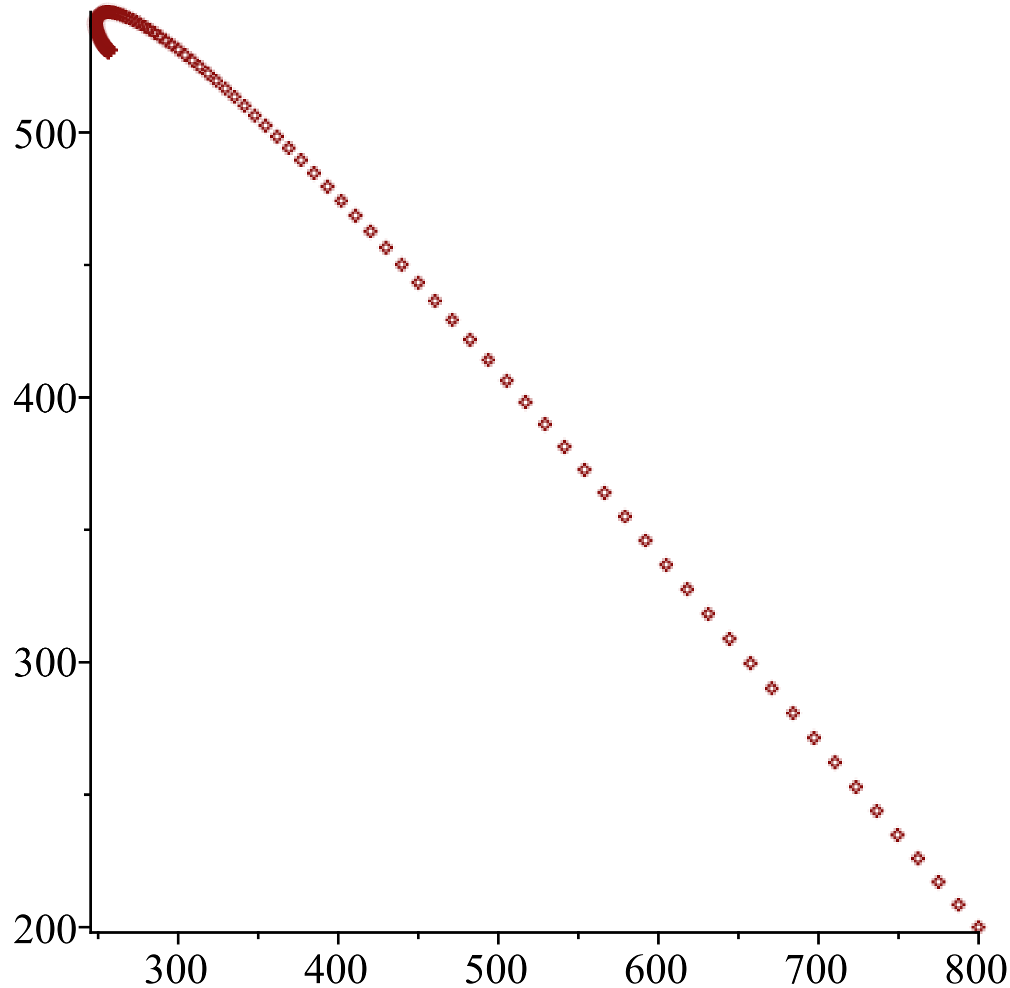
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> #ii.
> F:=SIRS(s,i,(.3)*3/1000,6,3,1000)
```

$$F := [-0.0009000000000\, s\, i + 6000 - 6\, s - 6\, i,\ 0.0009000000000\, s\, i - 3\, i] \quad (16)$$

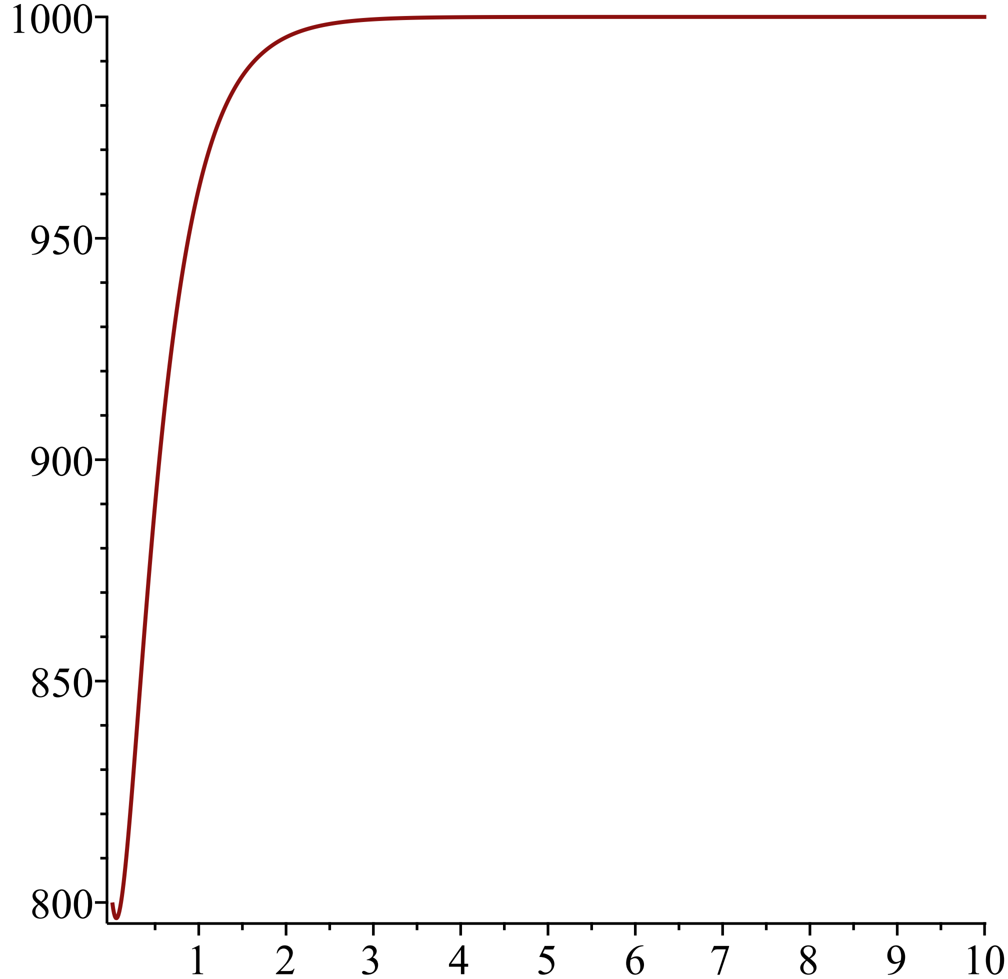
```
> EquP(F,[s,i])
```

$$\{[1000.,\ 0.],\ [3333.333333,\ -1555.555556]\} \quad (17)$$

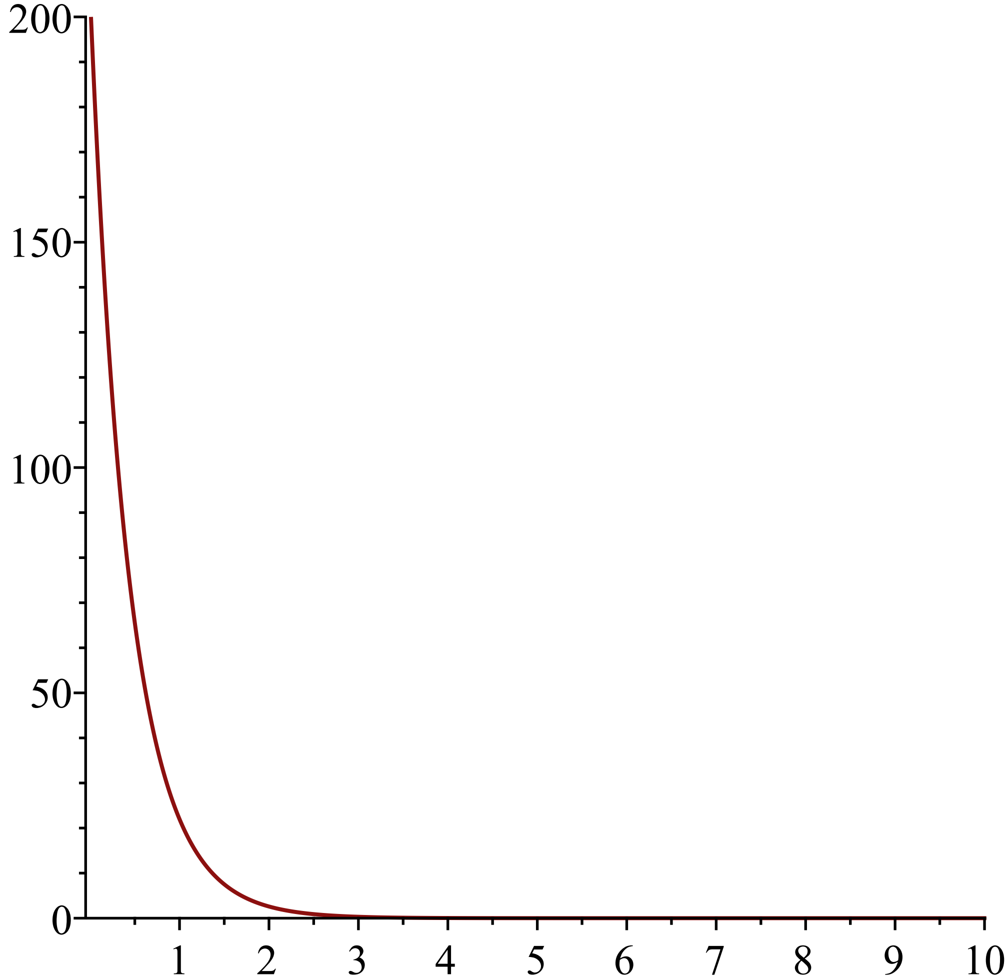
```
> SEquP(F,[s,i])
```

$$\{[1000.,\ 0.]\} \quad (18)$$

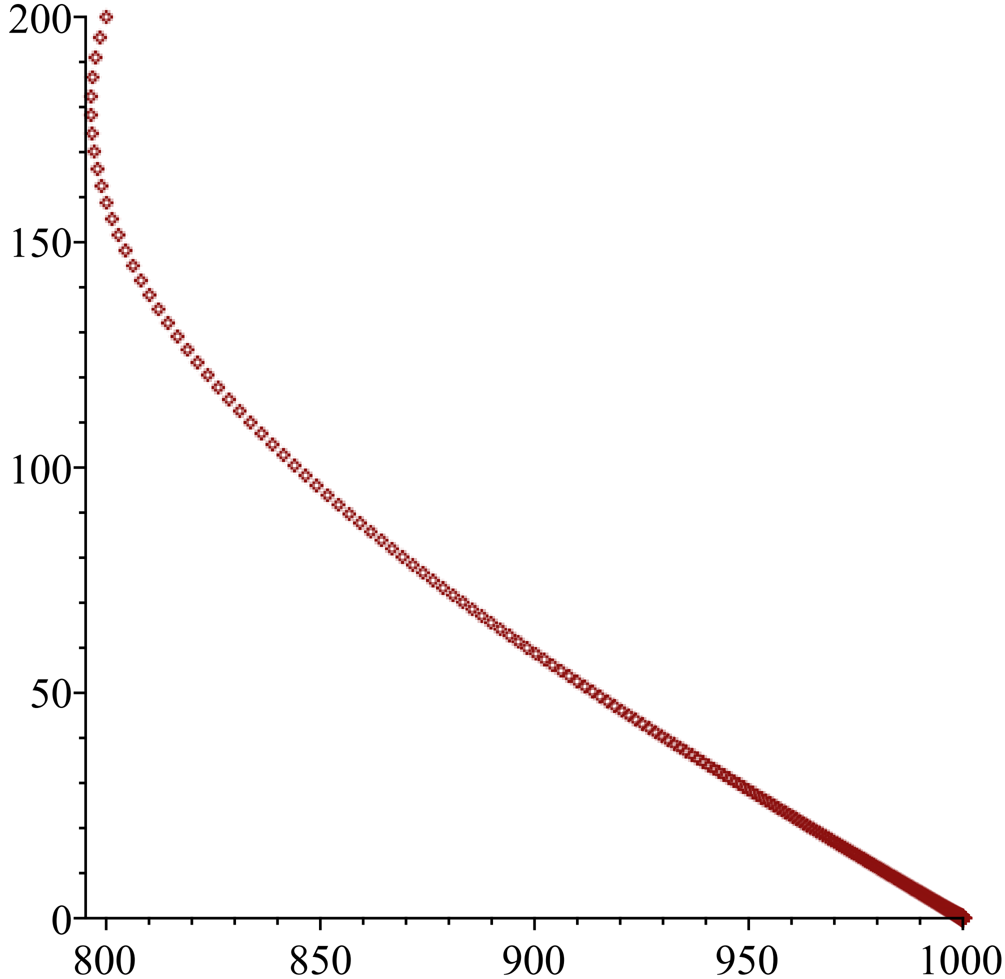
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,(.9)*3/1000,6,3,1000)
```

$$F := [-0.002700000000\, s\, i + 6000 - 6\, s - 6\, i,\ 0.002700000000\, s\, i - 3\, i] \quad (19)$$

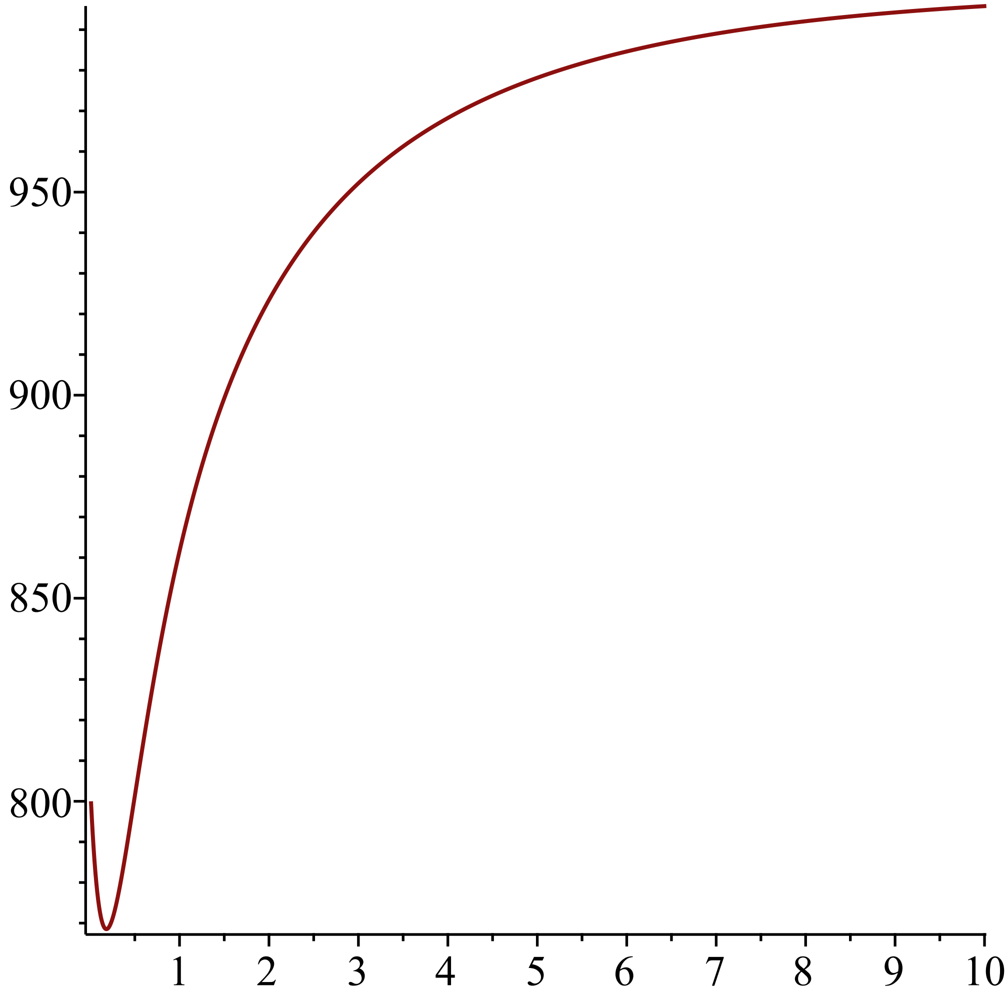
```
> EquP(F,[s,i])
```

$$\{[1000.,\ 0.],\ [1111.111111,\ -74.07407407]\} \quad (20)$$

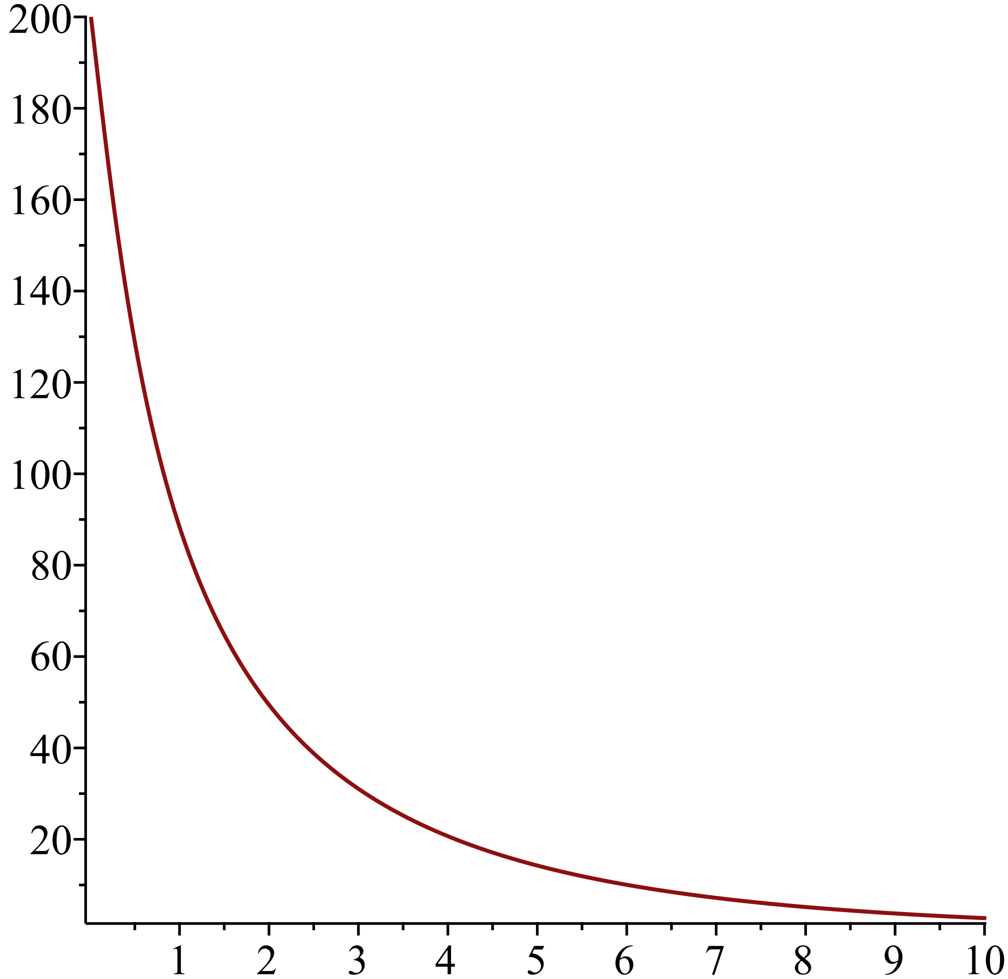
```
> SEquP(F,[s,i])
```

$$\{[1000.,\ 0.]\} \quad (21)$$

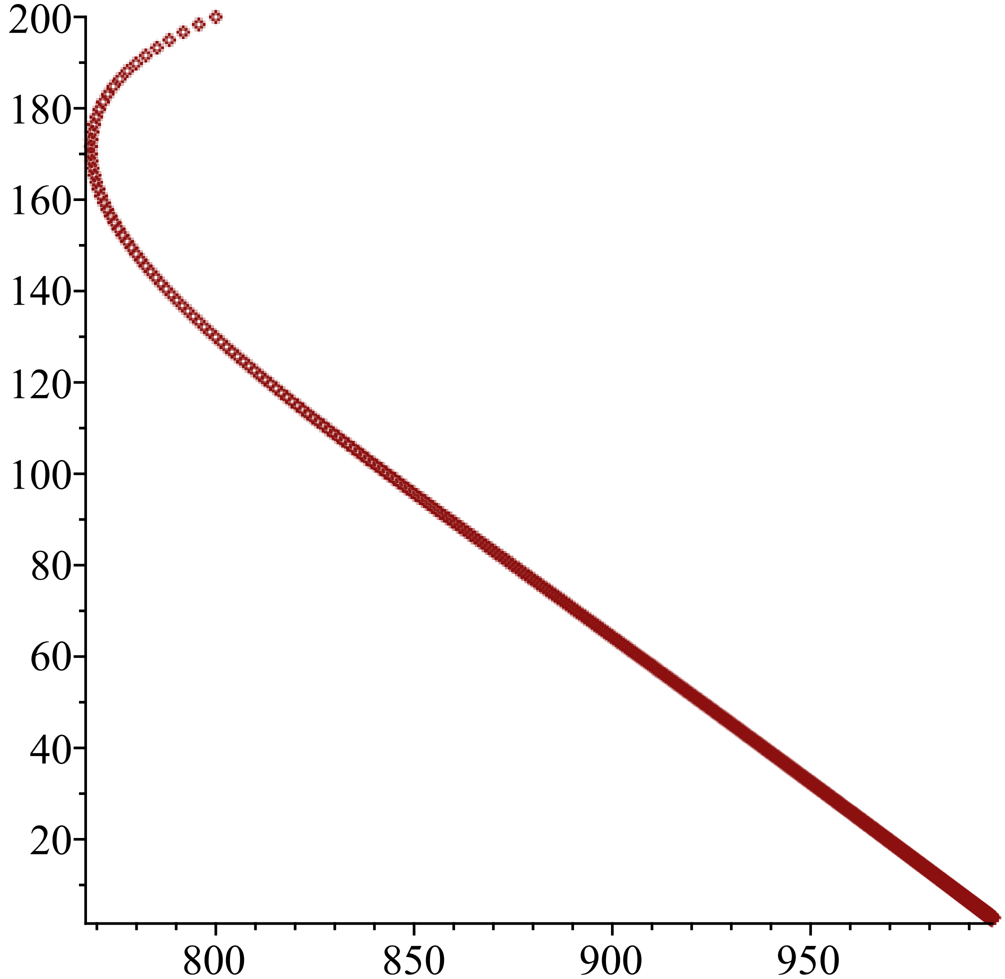
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,(3.9)*3/1000,6,3,1000)
```

$$F := [-0.01170000000\, s\, i + 6000 - 6\, s - 6\, i,\ 0.01170000000\, s\, i - 3\, i] \quad \textbf{(22)}$$

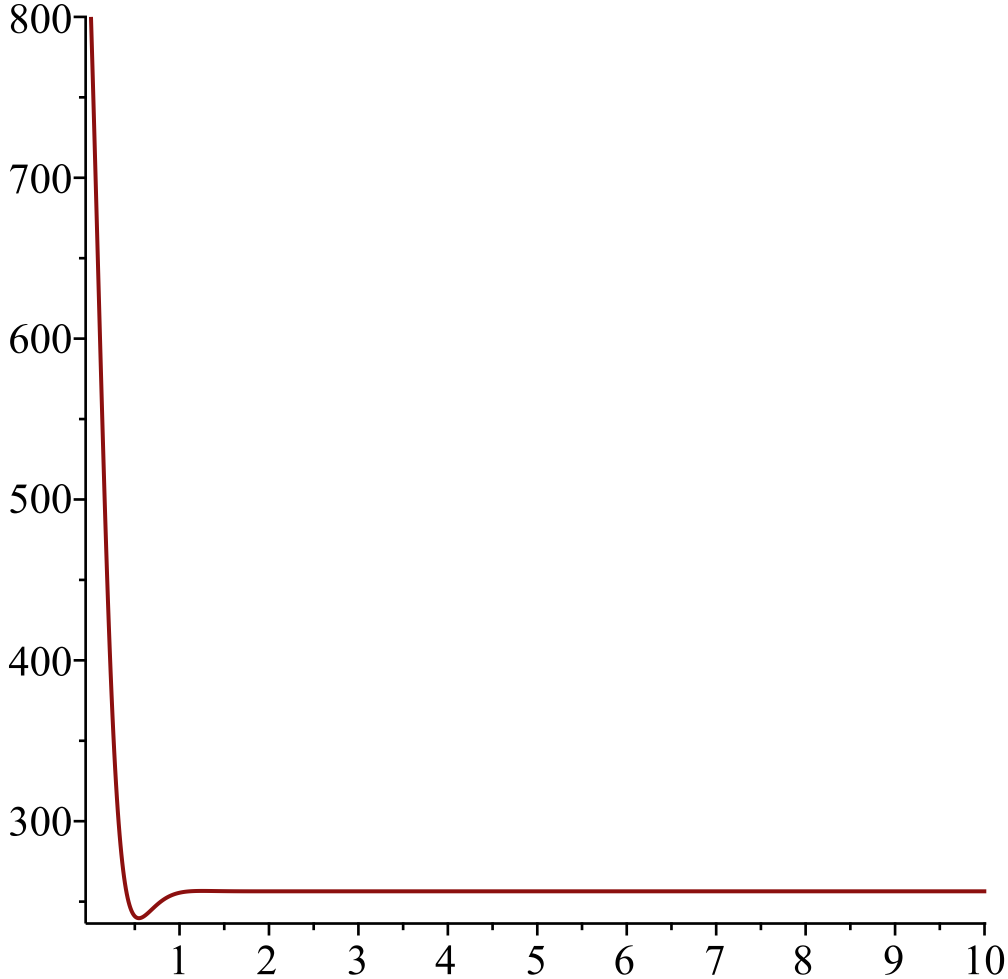
```
> EquP(F,[s,i])
```

$$\{[256.4102564, 495.7264957], [1000., 0.]\} \quad \textbf{(23)}$$

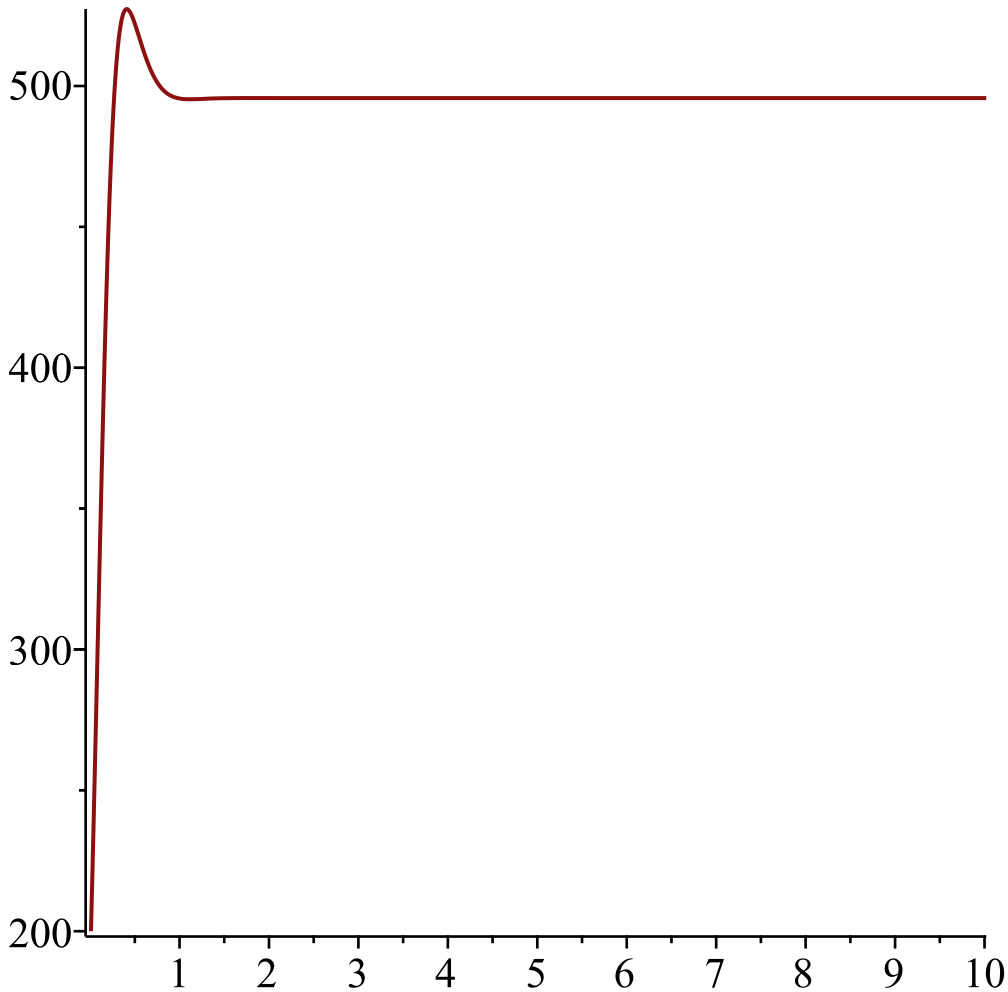
```
> SEquP(F,[s,i])
```

$$\{[256.4102564, 495.7264957]\} \quad \textbf{(24)}$$

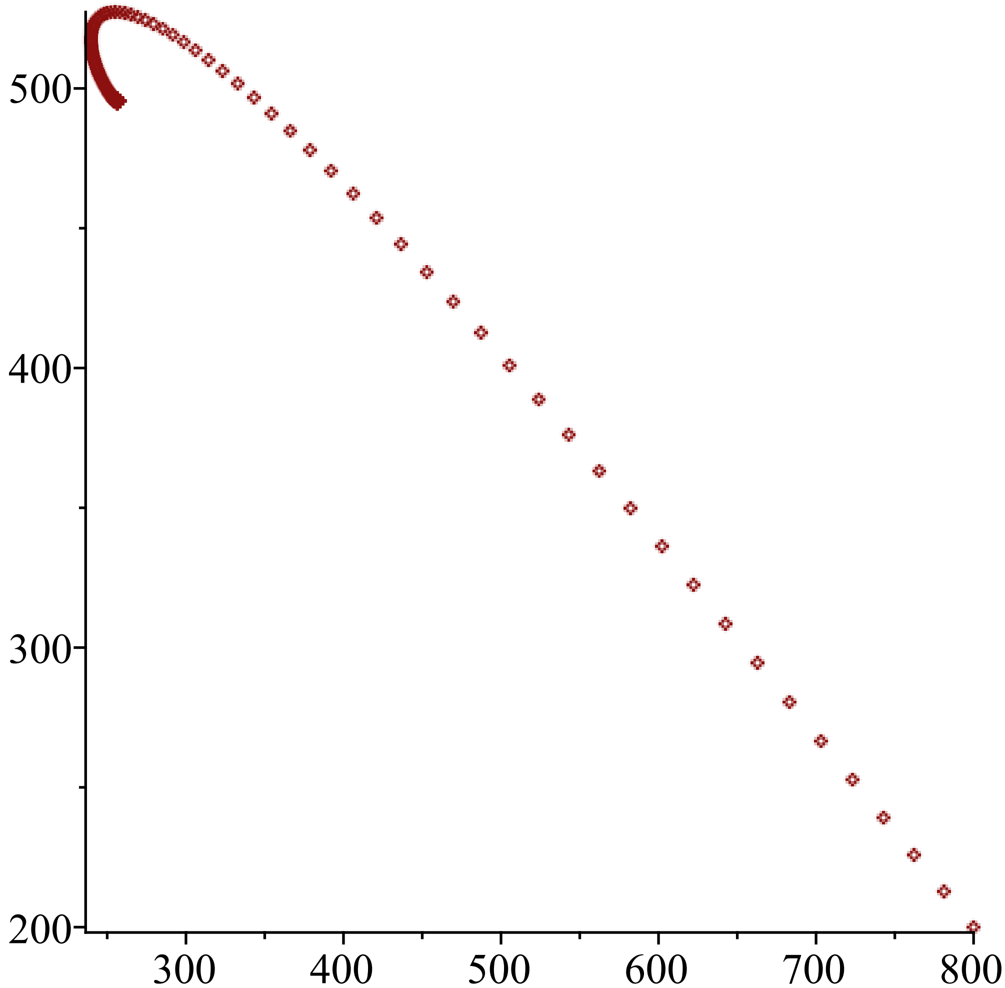
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> #iii.
> F:=SIRS(s,i,.3*4/1000,1,4,1000)
```

$$F := [-0.001200000000\, s\, i + 1000 - s - i,\ 0.001200000000\, s\, i - 4\, i] \quad (25)$$

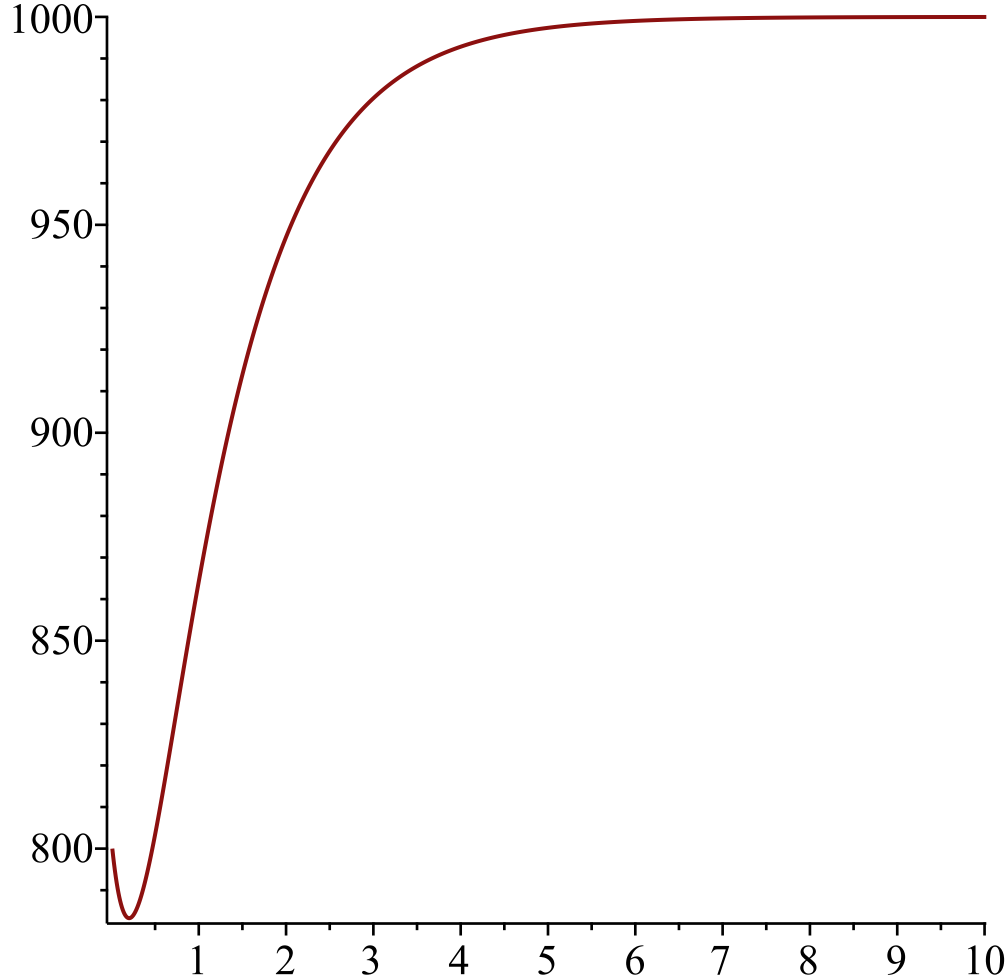
```
> EquP(F,[s,i])
```

$$\{[1000., 0.], [3333.333333, -466.6666667]\} \quad (26)$$

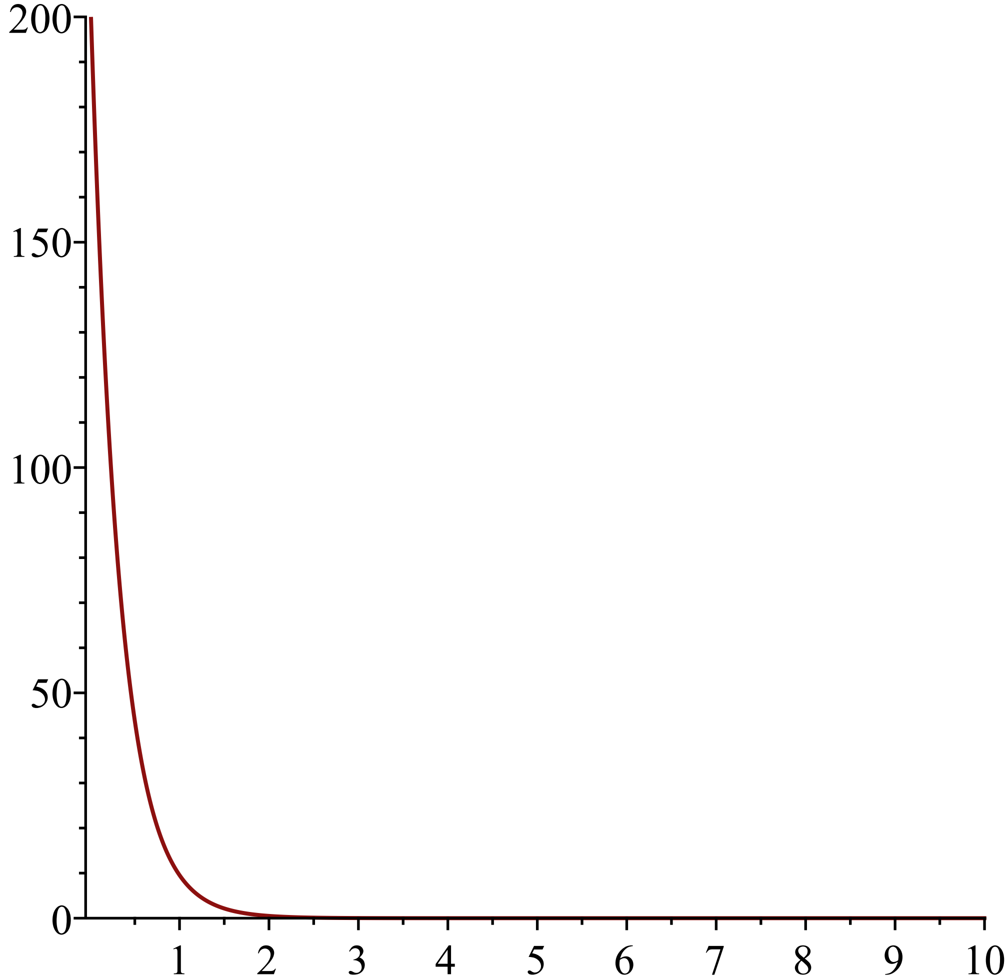
```
> SEquP(F,[s,i])
```

$$\{[1000., 0.]\} \quad (27)$$

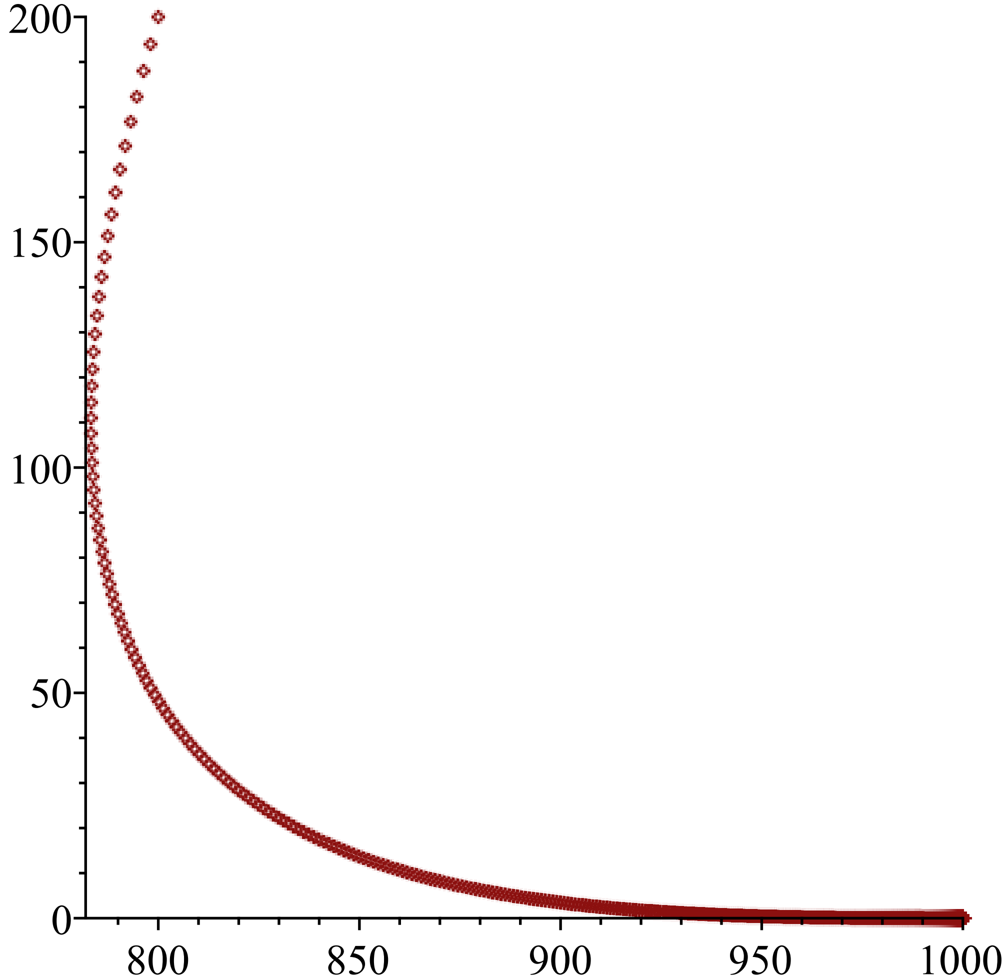
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,.9*4/1000,1,4,1000)
```

$$F := [-0.003600000000\, s\, i + 1000 - s - i,\ 0.003600000000\, s\, i - 4\, i] \quad \textbf{(28)}$$

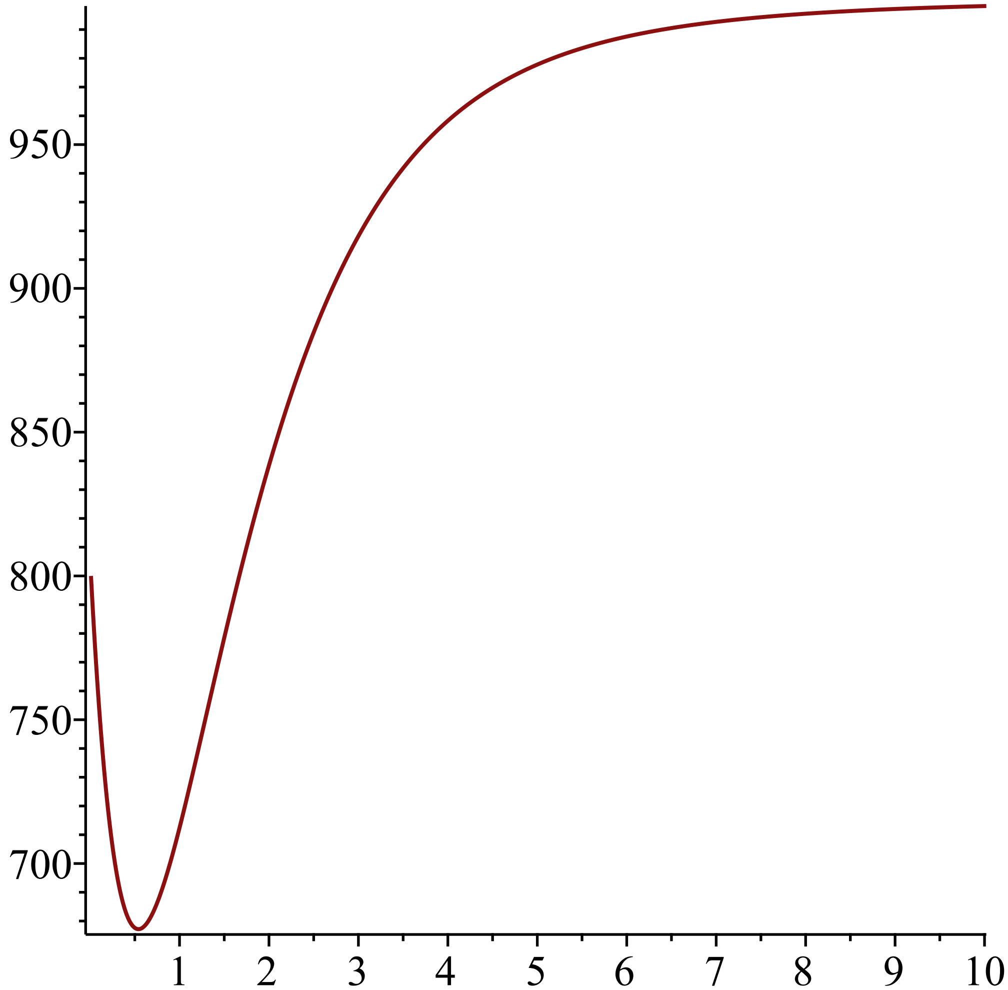
```
> EquP(F,[s,i])
```

$$\{[1000., 0.], [1111.111111, -22.22222222]\} \quad \textbf{(29)}$$

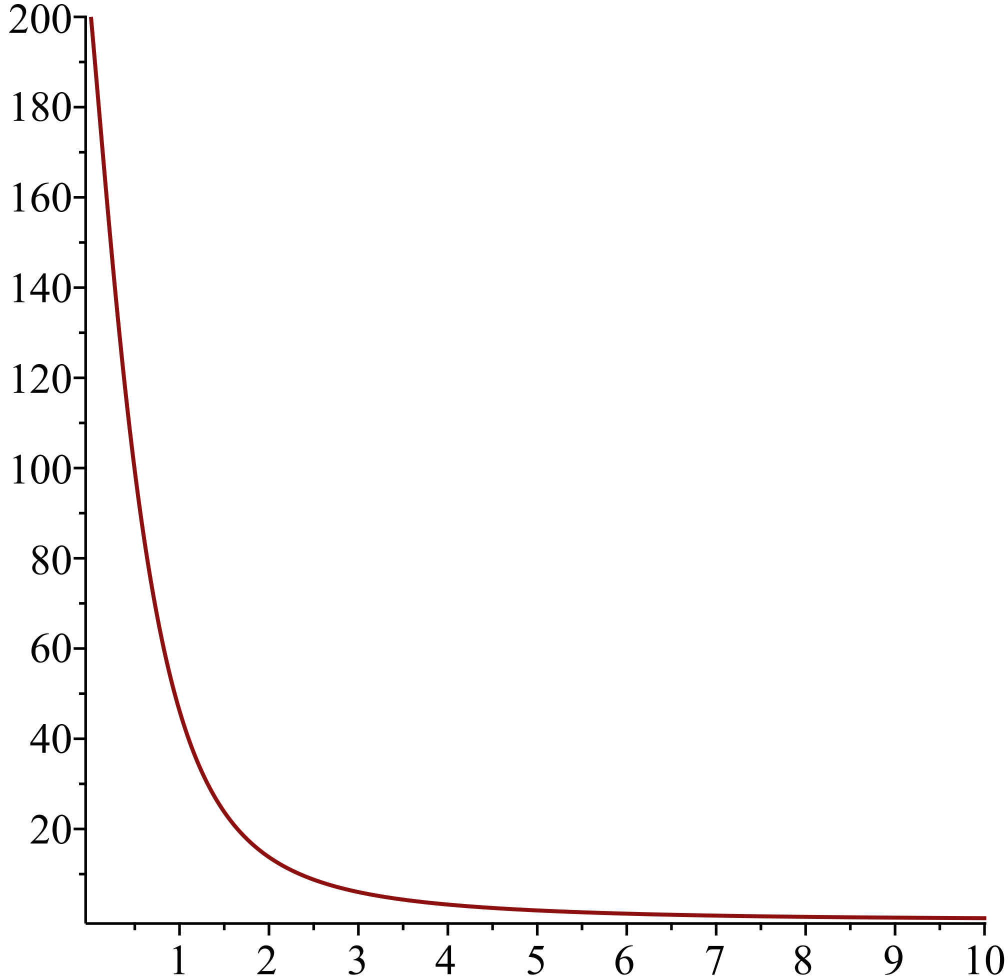
```
> SEquP(F,[s,i])
```

$$\{[1000., 0.]\} \quad \textbf{(30)}$$

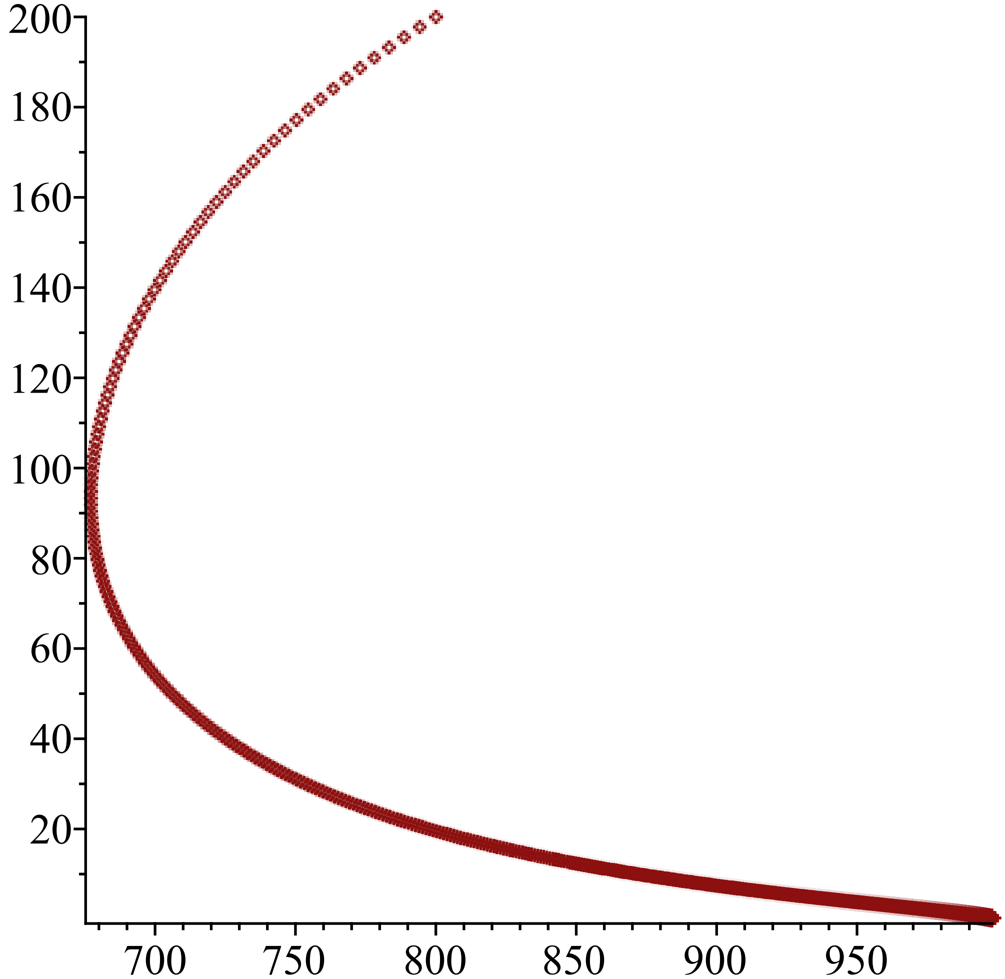
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,3.9*4/1000,1,4,1000)
```

$$F := [-0.01560000000\, s\, i + 1000 - s - i,\ 0.01560000000\, s\, i - 4\, i] \quad \textbf{(31)}$$

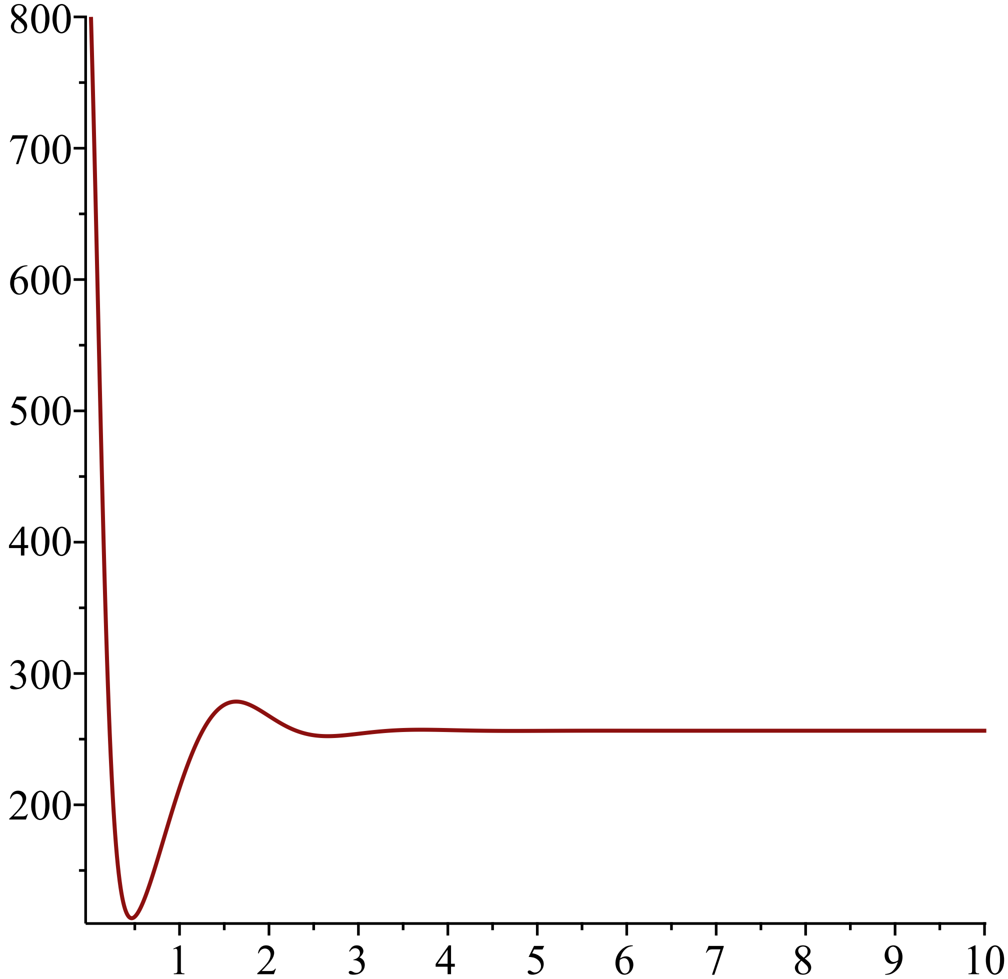
```
> EquP(F,[s,i])
```

$$\{[256.4102564, 148.7179487], [1000., 0.]\} \quad \textbf{(32)}$$

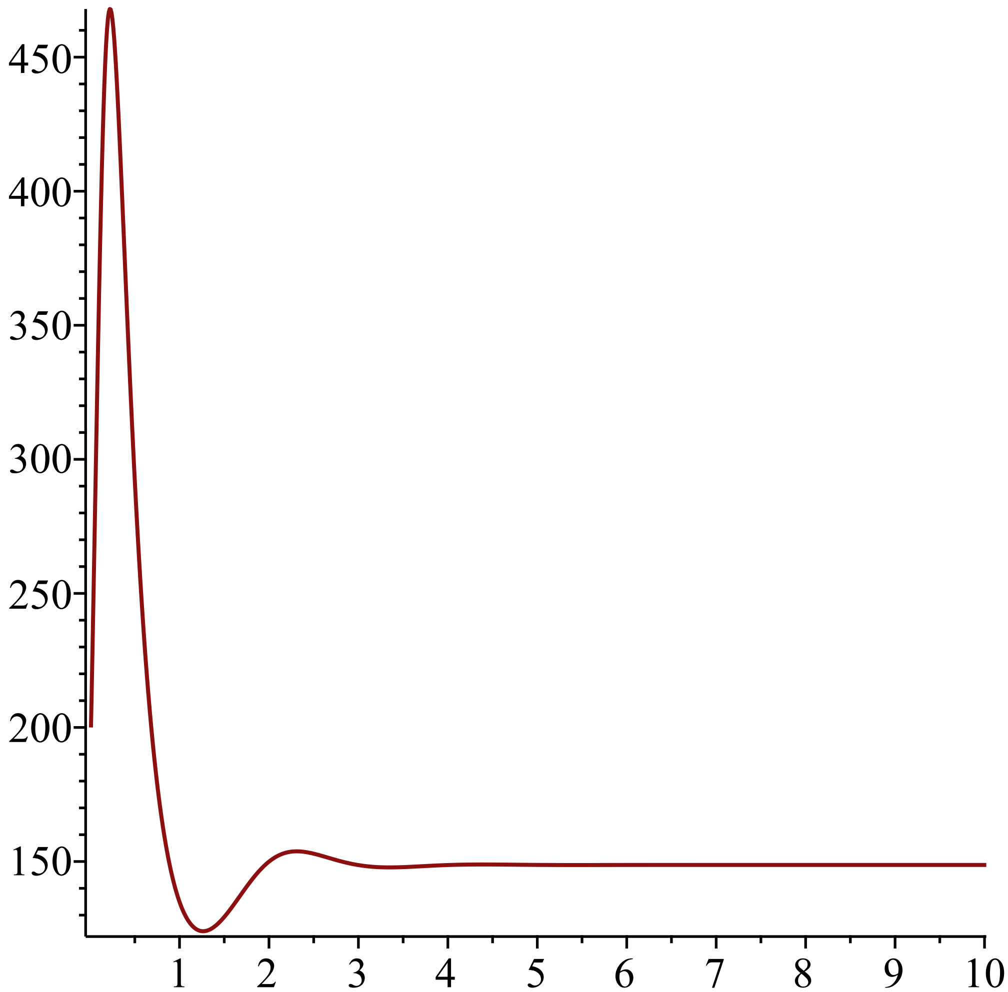
```
> SEquP(F,[s,i])
```

$$\{[256.4102564, 148.7179487]\} \quad \textbf{(33)}$$

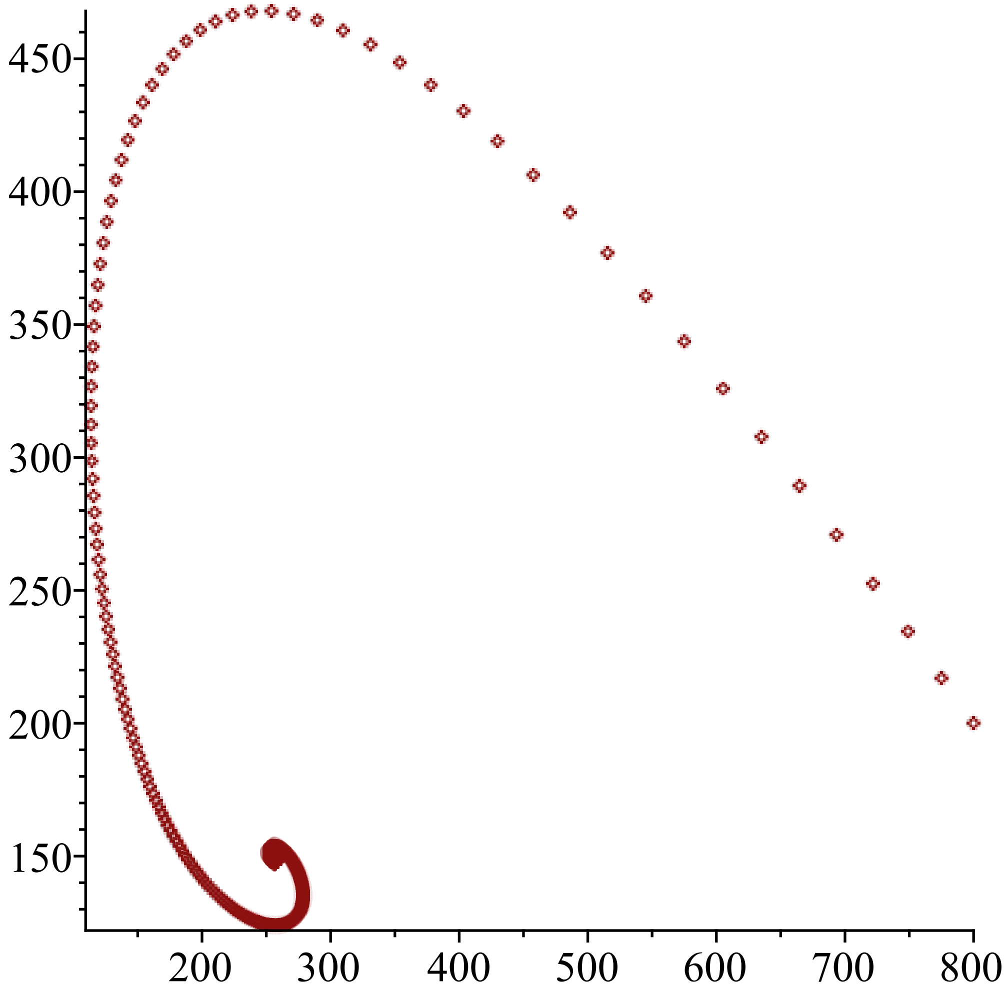
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> #iv.
> F:=SIRS(s,i,.3*7/1000,10,7,1000)
```

$$F := [-0.002100000000\, s\, i + 10000 - 10\, s - 10\, i,\ 0.002100000000\, s\, i - 7\, i] \quad (34)$$

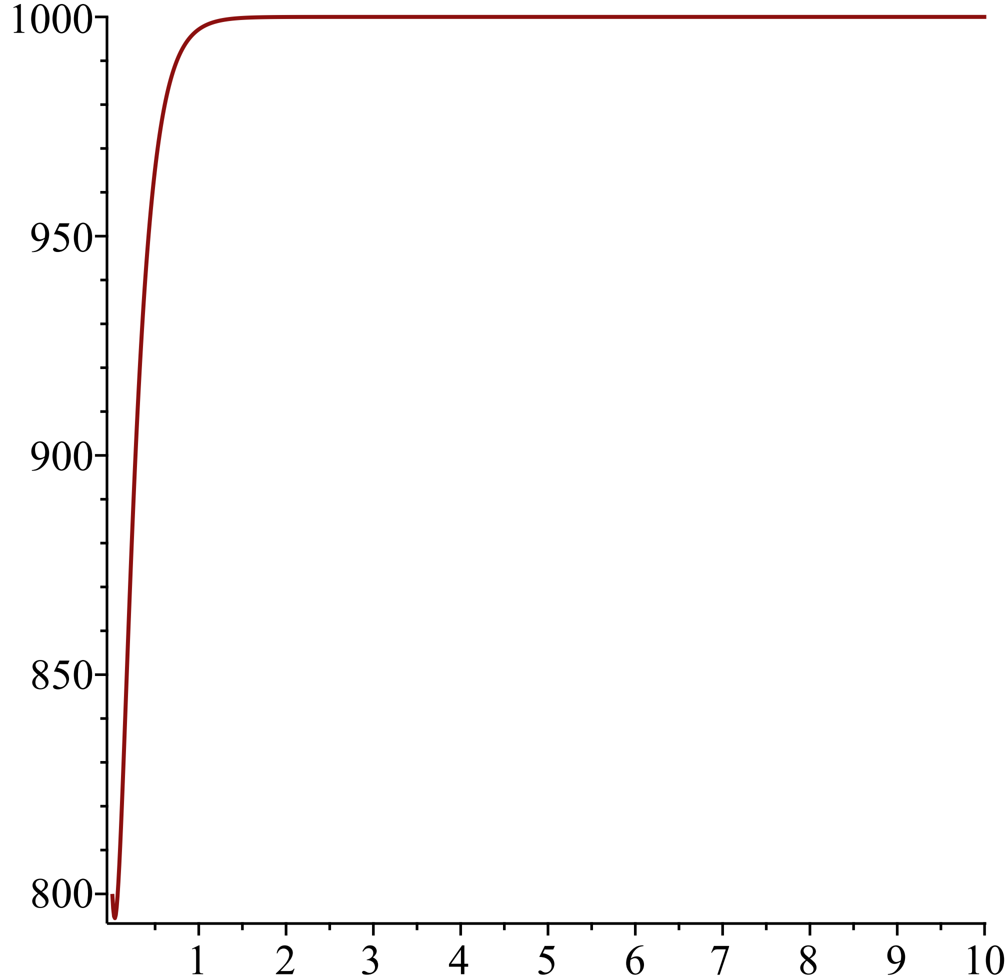
```
> EquP(F,[s,i])
```

$$\{[1000., 0.], [3333.333333, -1372.549020]\} \quad (35)$$

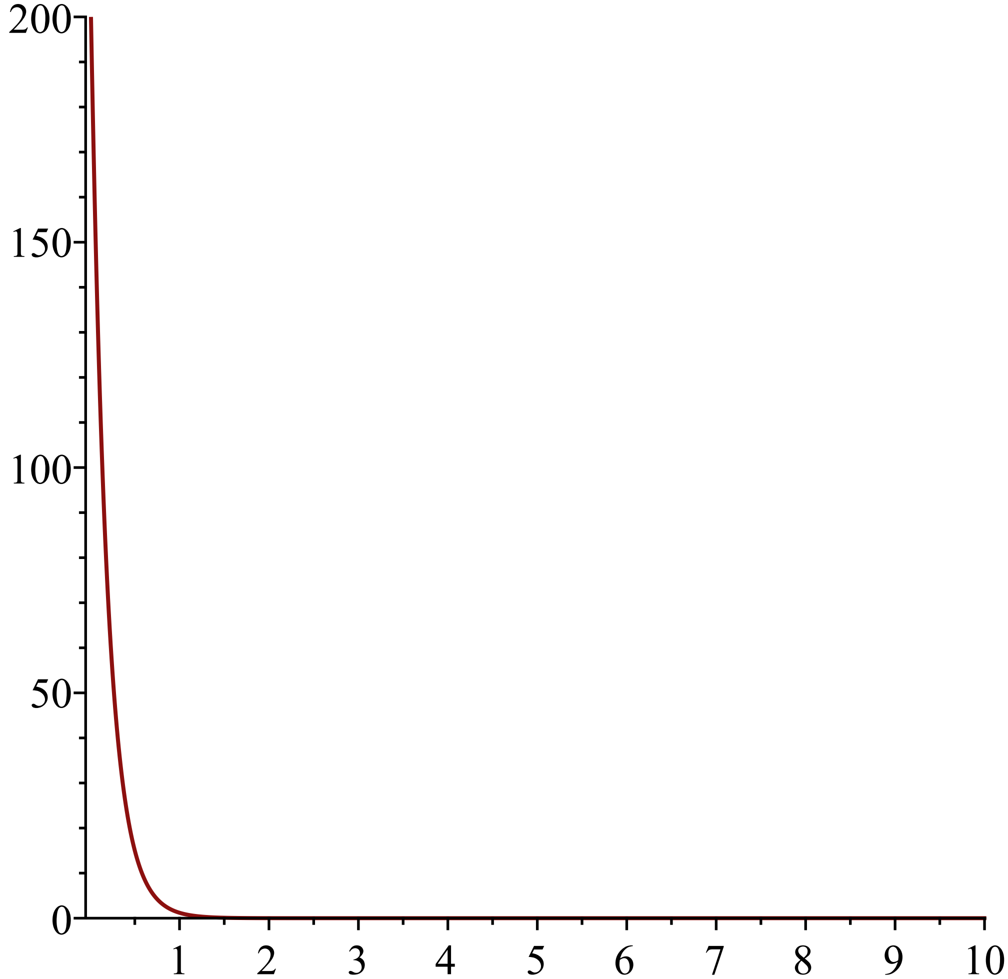
```
> SEquP(F,[s,i])
```

$$\{[1000., 0.]\} \quad (36)$$

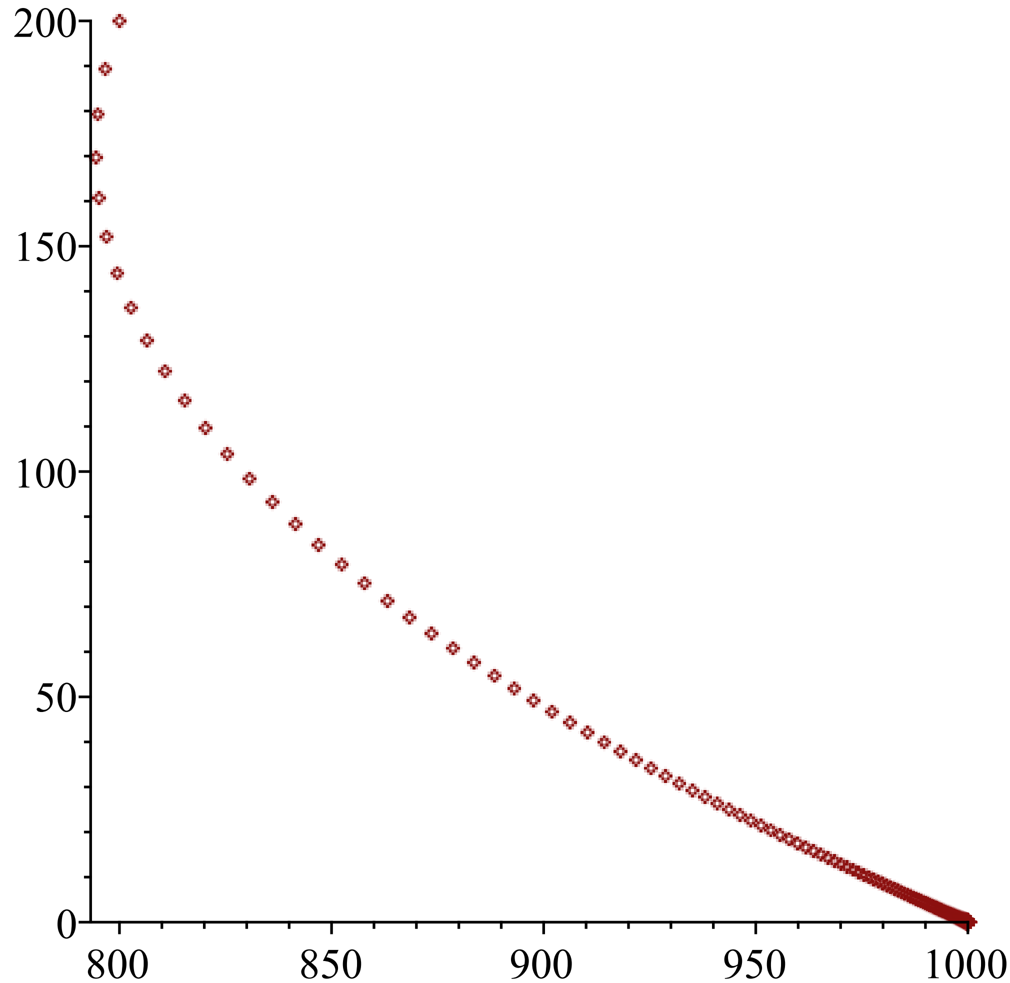
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,.9*7/1000,10,7,1000)
```

$$F := [-0.006300000000\, s\, i + 10000 - 10\, s - 10\, i,\ 0.006300000000\, s\, i - 7\, i] \quad (37)$$

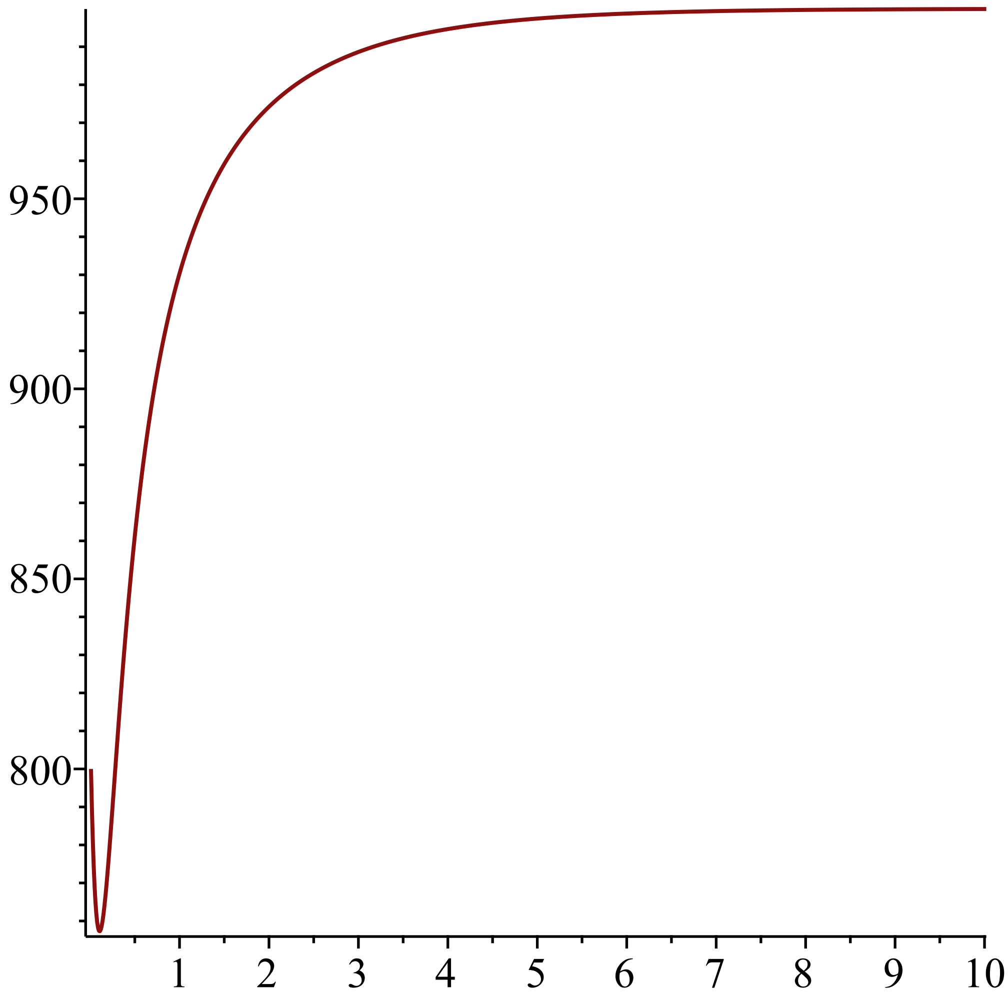
```
> EquP(F,[s,i])
```

$$\{[1000.,\ 0.],\ [1111.111111,\ -65.35947712]\} \quad (38)$$

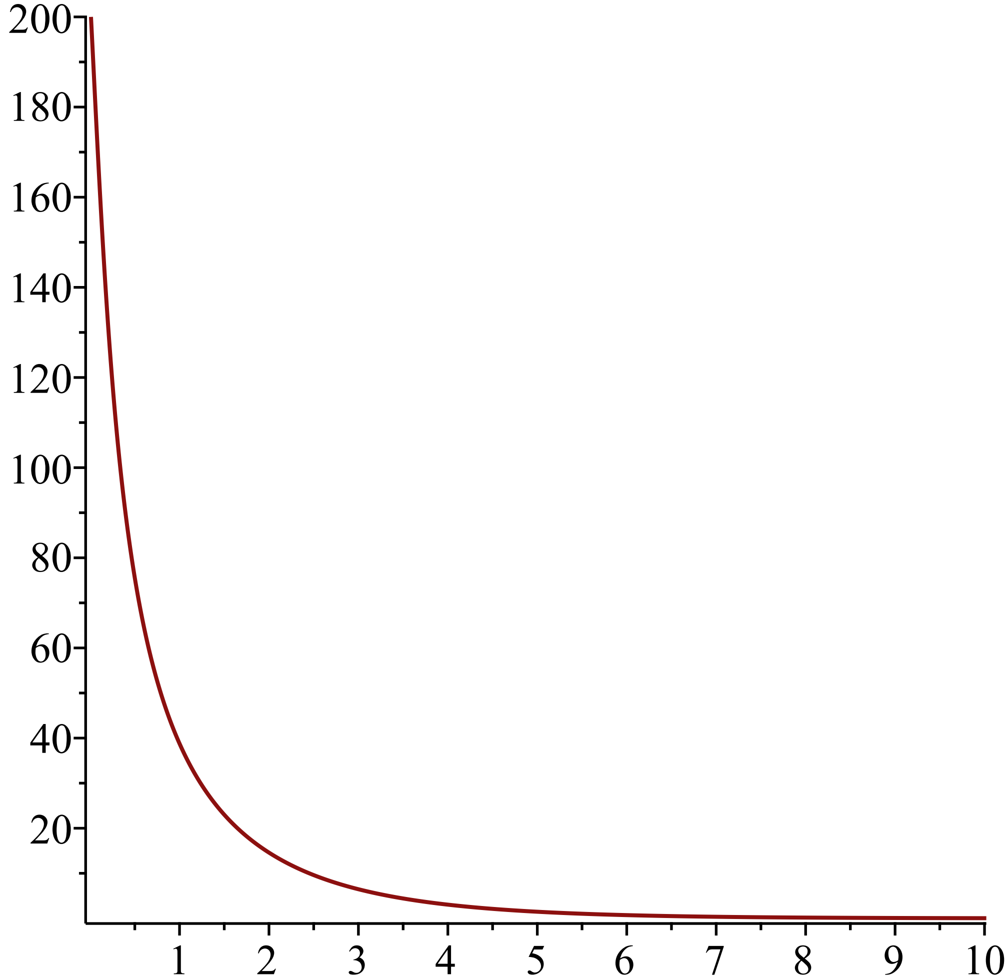
```
> SEquP(F,[s,i])
```

$$\{[1000.,\ 0.]\} \quad (39)$$

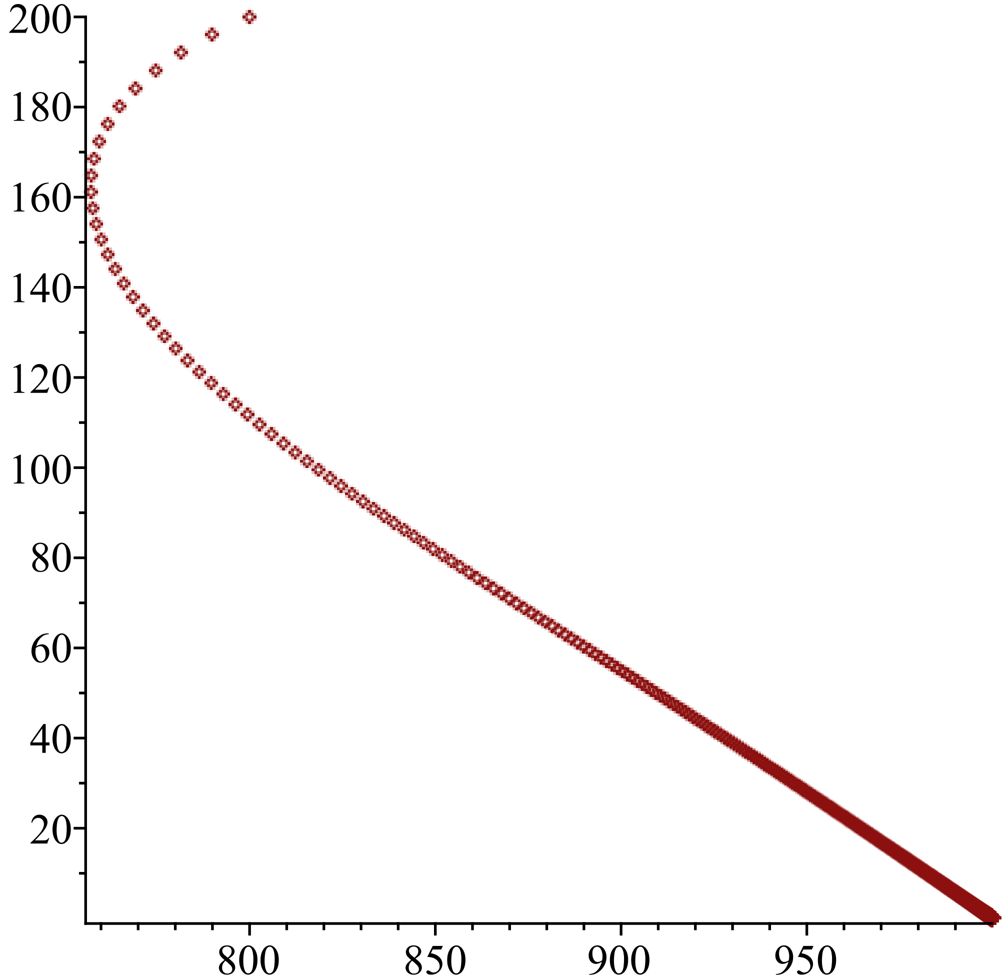
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> F:=SIRS(s,i,3.9*7/1000,10,7,1000)
```

$$F := [-0.02730000000\, s\, i + 10000 - 10\, s - 10\, i, 0.02730000000\, s\, i - 7\, i] \quad \textbf{(40)}$$

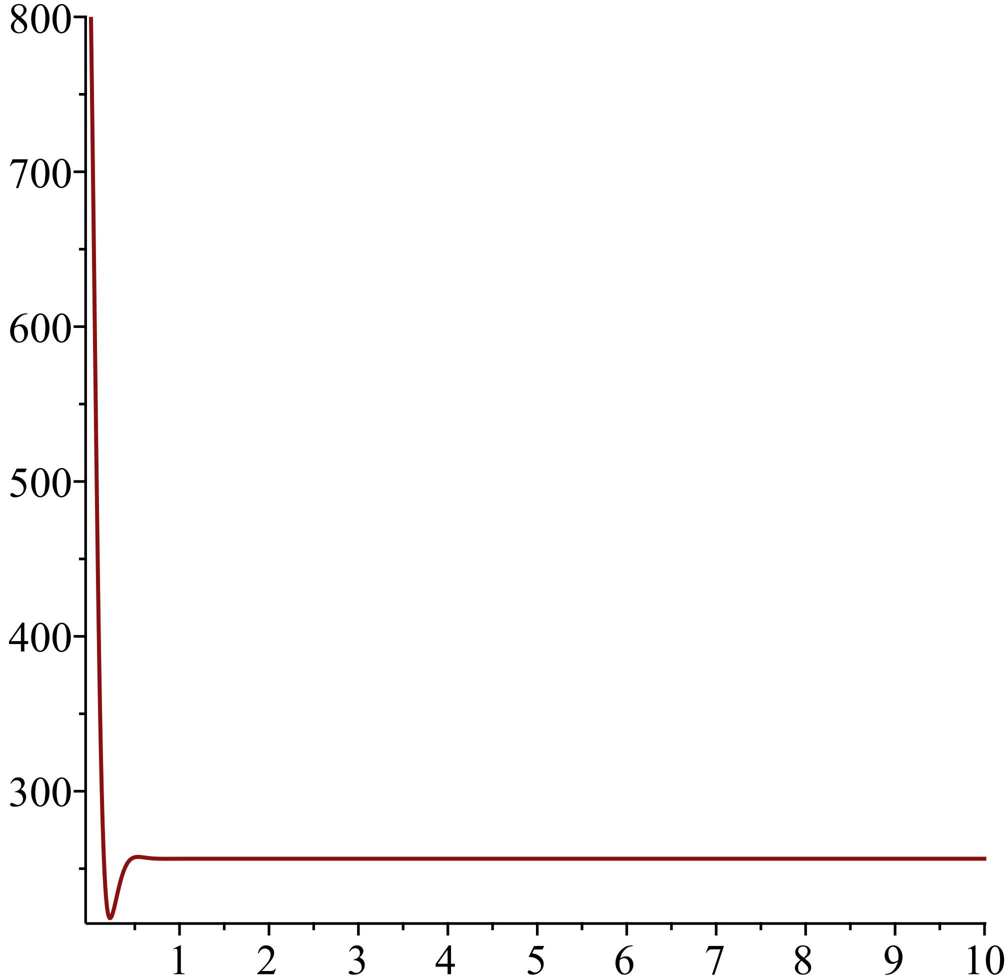
```
> EquP(F,[s,i])
```

$$\{[256.4102564, 437.4057315], [1000., 0.]\} \quad \textbf{(41)}$$

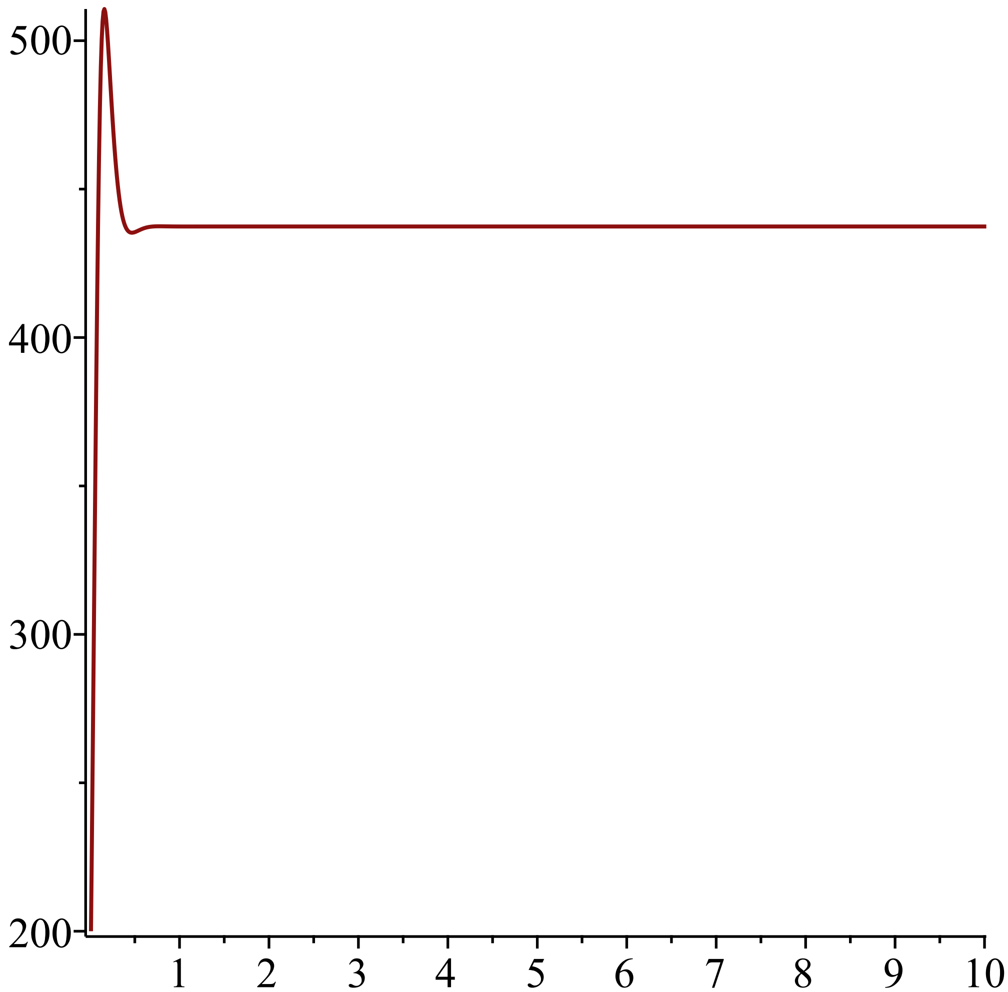
```
> SEquP(F,[s,i])
```

$$\{[256.4102564, 437.4057315]\} \quad \textbf{(42)}$$

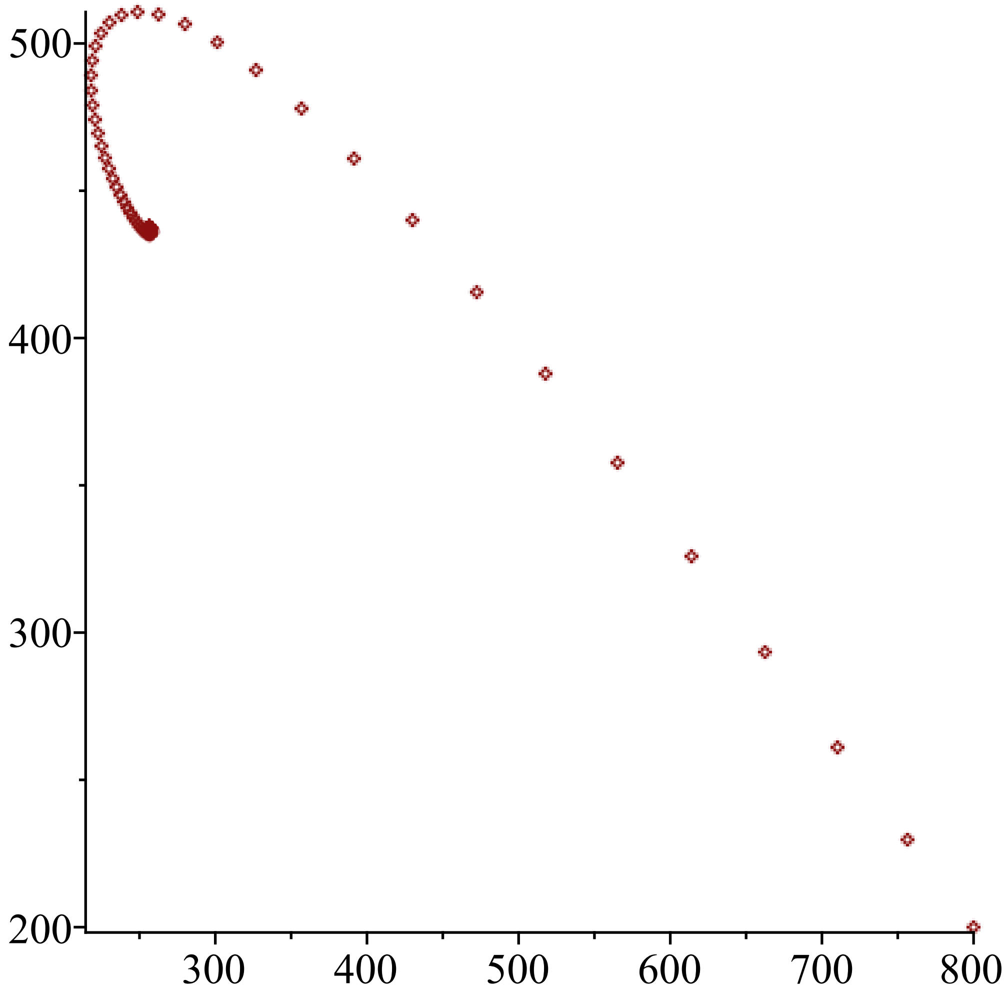
```
> TimeSeries(F,[s,i],[800,200],.01,10,1)
```




```
> TimeSeries(F,[s,i],[800,200],.01,10,2)
```




```
> PhaseDiag(F,[s,i],[800,200],.01,10)
```




```
> #Plots are consistent with the SEquP command.
```

```
> #Question 2
```

```
> F:=RandNice([x,y],3)
```

$$F := [(3-2x-2y)(1-x-3y), (1-2x-y)(2-3x-3y)] \quad (43)$$

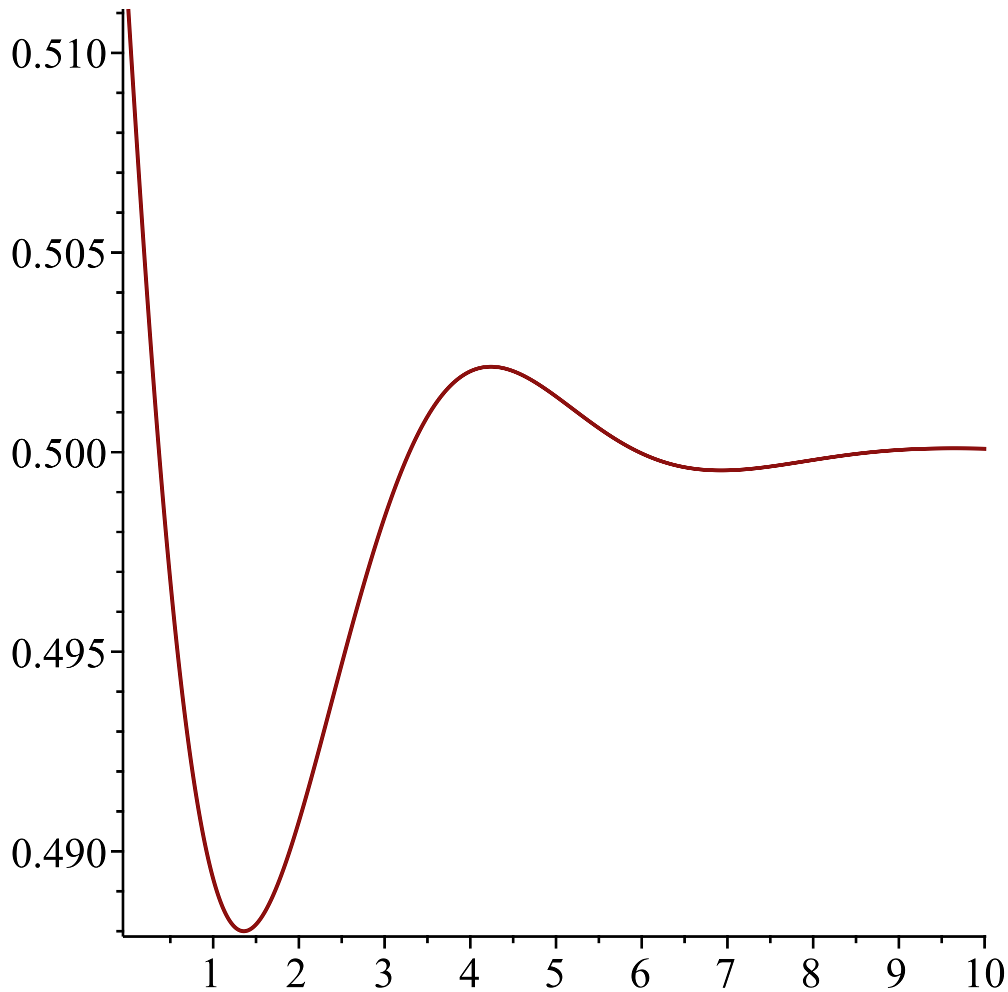
```
> EquP(F,[x,y])
```

$$\left\{\left[-\frac{1}{2}, 2\right], \left[\frac{1}{2}, \frac{1}{6}\right], \left[\frac{2}{5}, \frac{1}{5}\right]\right\} \quad (44)$$

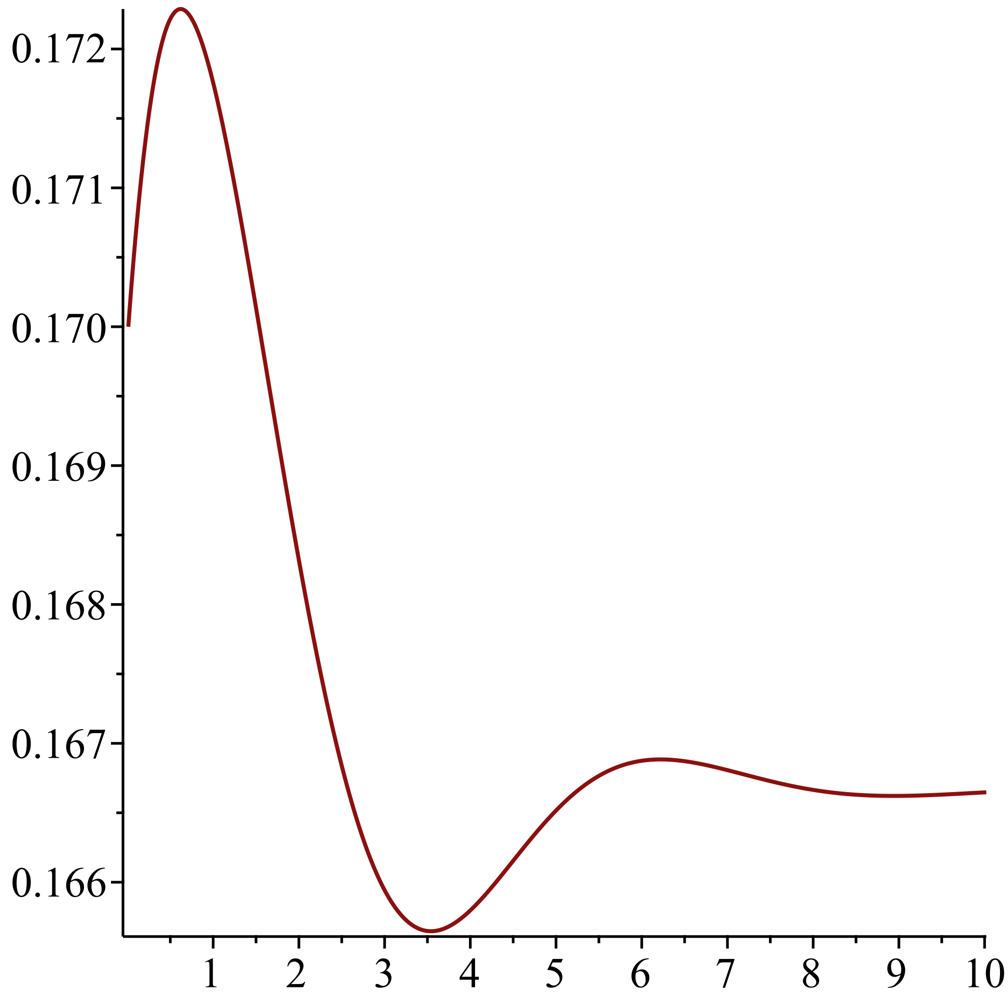
```
> SEquP(F,[x,y])
```

$$\{[0.5000000000, 0.1666666667]\} \quad (45)$$

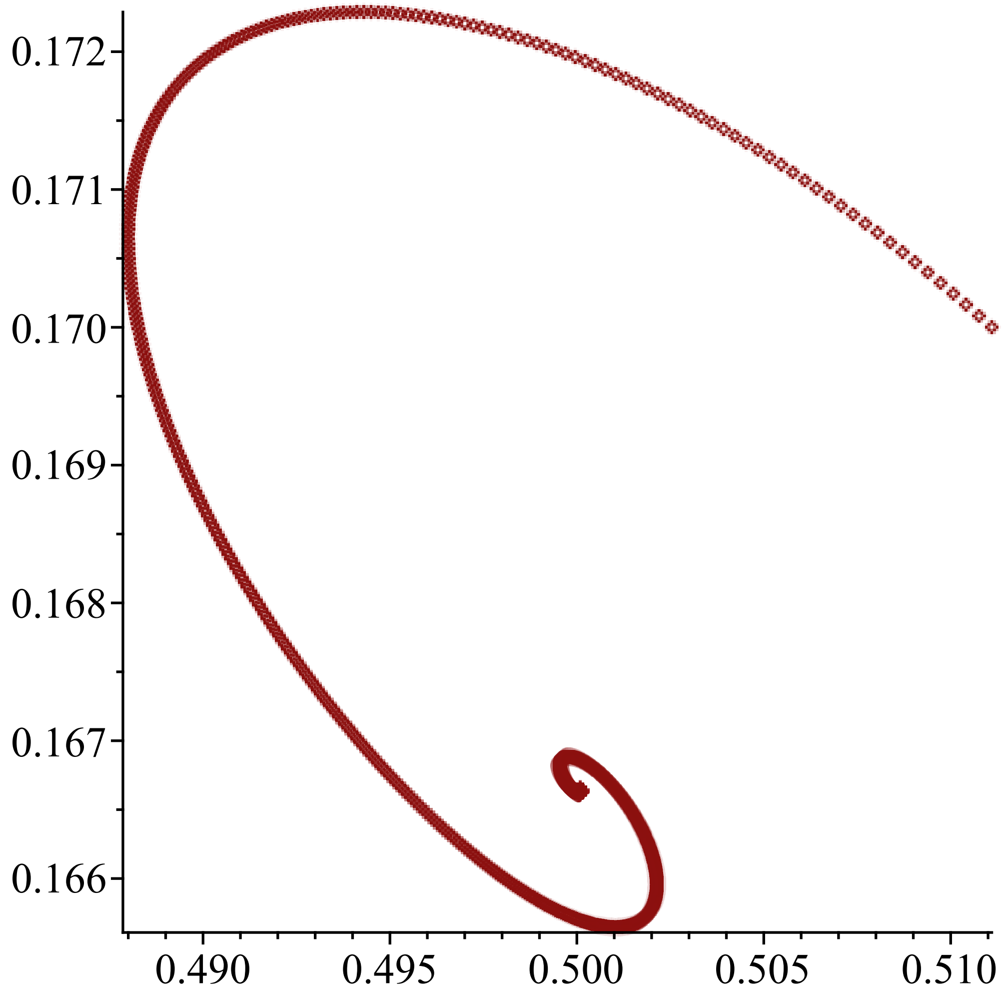
```
> TimeSeries(F,[x,y],[.5111,.17],.01,10,1)
```




```
> TimeSeries(F,[x,y],[.5111,.17],.01,10,2)
```




```
> PhaseDiag(F,[x,y],[.5111,.17],.01,10)
```




```
> F:=RandNice([x,y],3)
```

$$F := [(2-3x-2y)(3-2x-y), (1-x-3y)(1-3x-2y)] \quad (46)$$

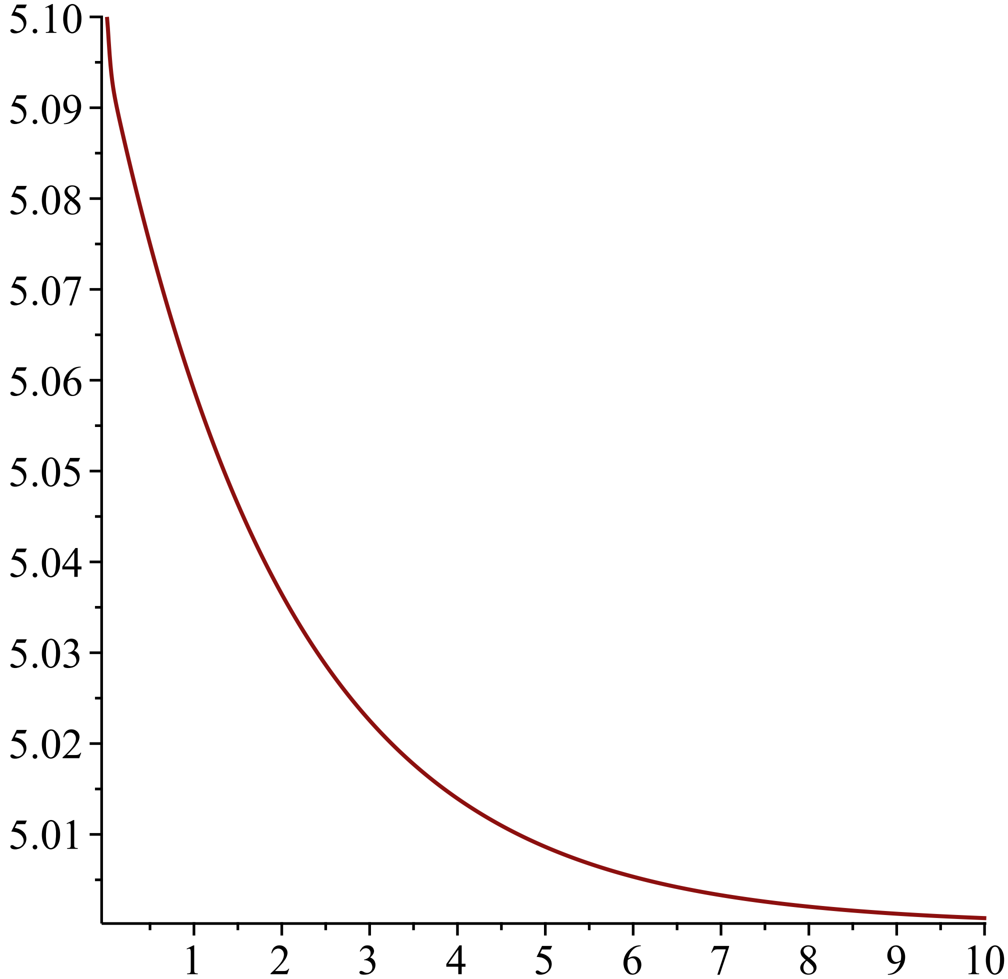
```
> EquP(F,[x,y])
```

$$\left\{[5, -7], \left[\frac{4}{7}, \frac{1}{7}\right], \left[\frac{8}{5}, -\frac{1}{5}\right]\right\} \quad (47)$$

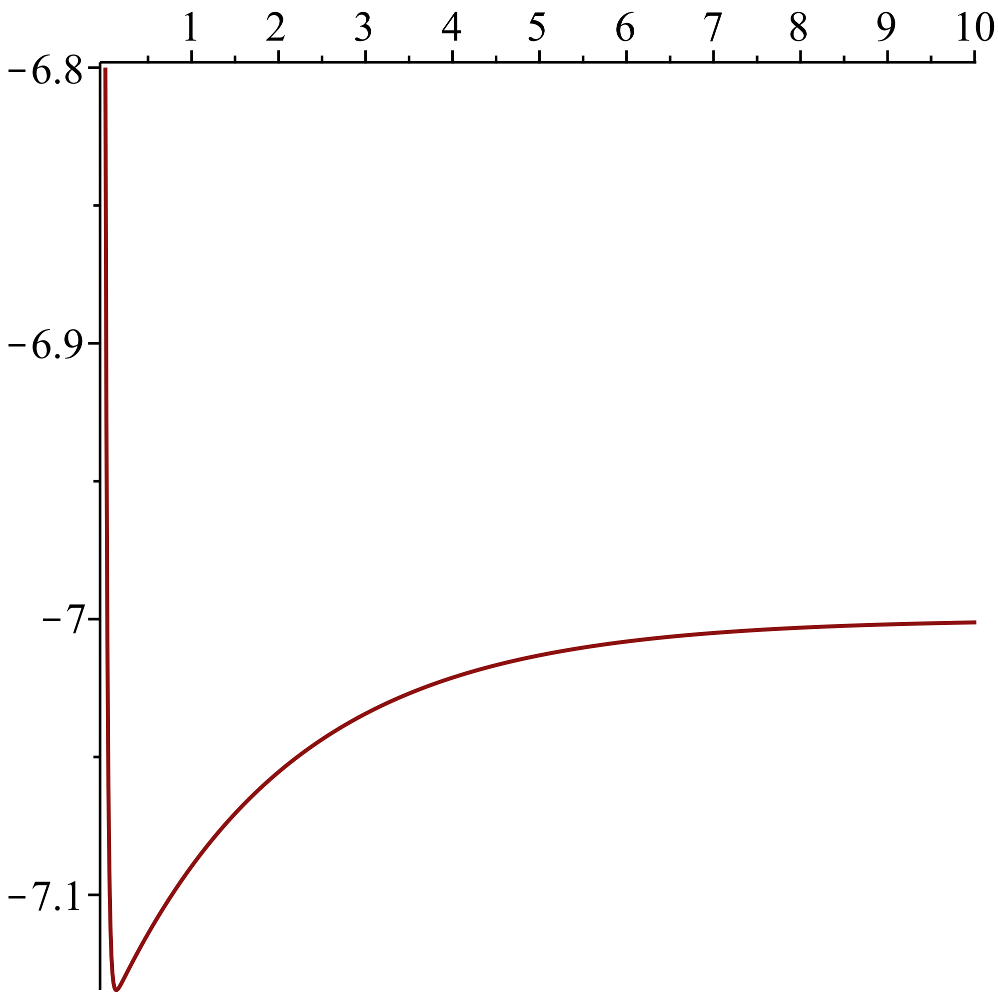
```
> SEquP(F,[x,y])
```

$$\{[5., -7.]\} \quad (48)$$

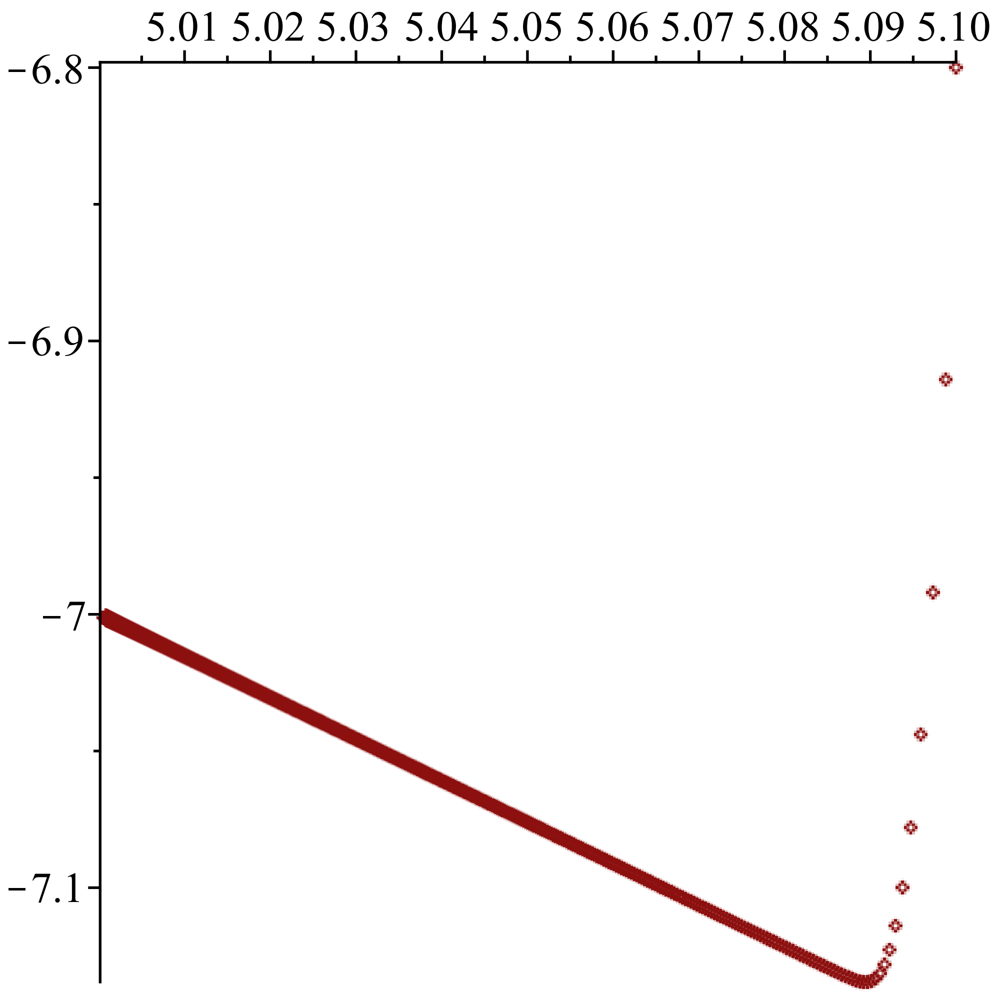
```
> TimeSeries(F,[x,y],[5.1,-6.8],.01,10,1)
```




```
> TimeSeries(F,[x,y],[5.1,-6.8],.01,10,2)
```




```
> PhaseDiag(F,[x,y],[5.1,-6.8],.01,10)
```




```
> F:=RandNice([x,y],3)
```

$$F := [(3-x-2y)(1-3x-y), (3-x-y)(2-2x-3y)] \quad (49)$$

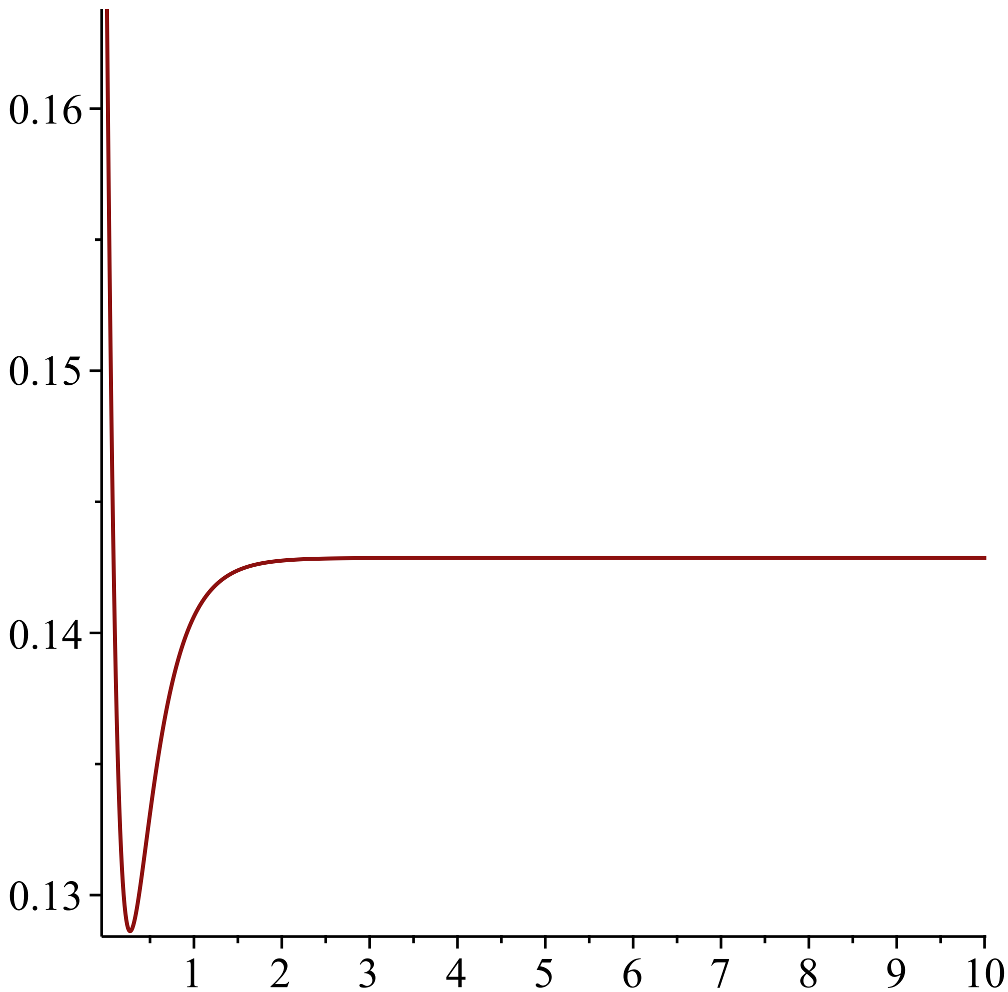
```
> EquP(F,[x,y])
```

$$\left\{[-5,4], [-1,4], [3,0], \left[\frac{1}{7}, \frac{4}{7}\right]\right\} \quad (50)$$

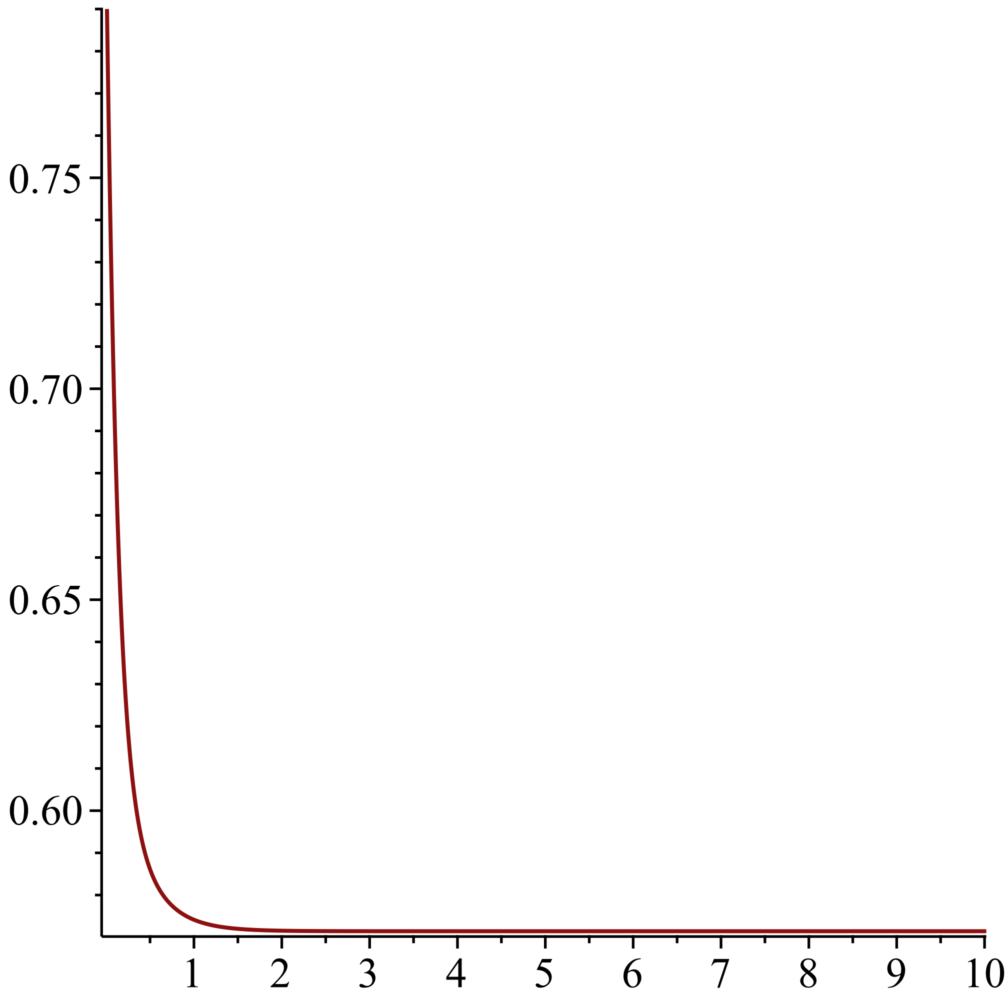
```
> SEquP(F,[x,y])
```

$$\{[0.1428571429, 0.5714285714]\} \quad (51)$$

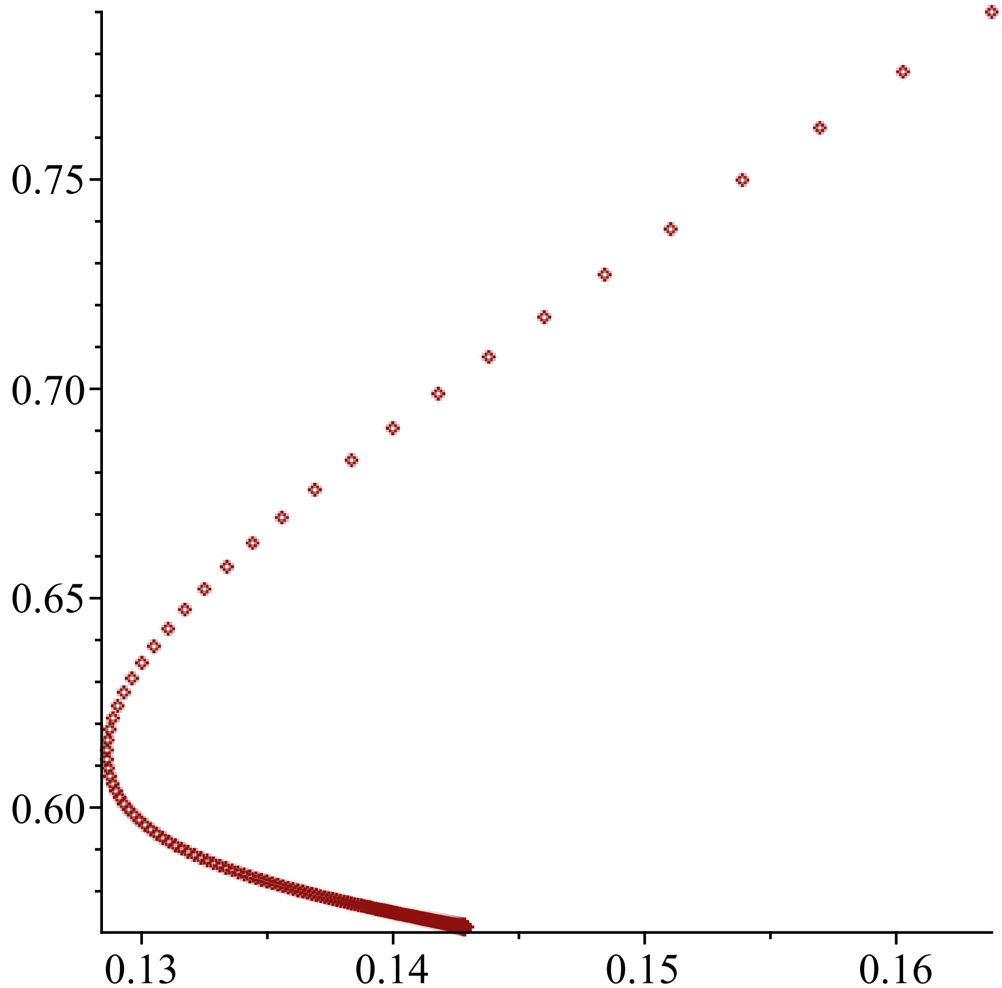
```
> TimeSeries(F,[x,y],[.1638,.790],.01,10,1)
```




```
> TimeSeries(F,[x,y],[.1638,.790],.01,10,2)
```




```
> PhaseDiag(F,[x,y],[.1638,.790],.01,10)
```




```
> #Horizontal asymptotes coincide with those predicted the SEquP
  procedure
```

```
> #Question 3
```

```
> Help(Orbk)
```

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, try:

*Orbk(1,z,5/2*z[1]*(1-z[1]),[0.5],1000,1010);*

To get the Fibonacci sequence, type:

Orbk(2,z,z[1]+z[2],[1,1],1000,1010);

To get the part of the orbit between n=1000 and n=1010, of the 3rd order recurrence given in Eq. (4) of the Ladas-Amleh paper

https://sites.math.rutgers.edu/~zeilberg/Bio21/AmlehLadas.pdf

with initial conditions x(0)=1, x(1)=3, x(2)=5, Type:

Orbk(3,z,z[2]/(z[2] +z[3]),[1.,3.,5.],1000,1010);

To get the part of the orbit between n=1000 and n=1010, of the 3rd order recurrence given in Eq. (5) of the Ladas-Amleh paper

with initial conditions x(0)=1, x(1)=3, x(2)=5, Type:

Orbk(3,z,(z[1] +z[3])/z[2],[1.,3.,5.],1000,1010);

To get the part of the orbit between n=1000 and n=1010, of the 3rd order recurrence given in Eq. (6) of the Ladas-Amleh paper

with initial conditions x(0)=1, x(1)=3, x(2)=5, Type:

Orbk(3,z,(1 +z[3])/z[1],[1.,3.,5.],1000,1010);

To get the part of the orbit between n=1000 and n=1010, of the 3rd order recurrence given in Eq. (7) of the Ladas-Amleh paper

with initial conditions x(0)=1, x(1)=3, x(2)=5, Type:

Orbk(3,z,(1 +z[1])/(z[2] +z[3]),[1.,3.,5.],1000,1010); **(52)**

```
> x(n)=(3+x(n-2)+x(n-3)+x(n-4))/(1+x(n-1)+x(n-3))
```

$$x(n) = \frac{3 + x(n-2) + x(n-3) + x(n-4)}{1 + x(n-1) + x(n-3)} \quad (53)$$

```
> #replace x(n-1) with z[1], x(n-2) with z[2], x(n-3) with z[3], x
  (n-4) with z[4]
```

```
> F:=(3+z[2]+z[3]+z[4])/(1+z[1]+z[3])
```

$$F := \frac{3 + z_2 + z_3 + z_4}{1 + z_1 + z_3} \quad (54)$$

```
> Orbk(4,z,F,[2.,6.,9.,11.],1000,1010)
```

[3.585161045, 1.067231329, 3.585161045, 1.067231329, 3.585161045, 1.067231329, 3.585161045, 1.067231329, 3.585161045, 1.067231329, 3.585161045] **(55)**

```
> Orbk(4,z,F,[3.,8.,11.,14.],1000,1010)
```

[4.107419924, 0.9851112531, 4.107419924, 0.9851112531, 4.107419924, 0.9851112531, 4.107419924, 0.9851112531, 4.107419924, 0.9851112531, 4.107419924] **(56)**

```
> Orbk(4,z,F,[6.,10.,13.,14.],1000,1010)
```

[3.708503743, 1.045425575, 3.708503743, 1.045425575, 3.708503743, 1.045425575, 3.708503743, 1.045425575, 3.708503743, 1.045425575, 3.708503743] **(57)**

```
> Orbk(4,z,F,[5.,13.,16.,20.],1000,1010)
```

(58)

[4.973544301, 0.8911887048, 4.973544301, 0.8911887048, 4.973544301, 0.8911887048, 4.973544301, 0.8911887048, 4.973544301, 0.8911887048, 4.973544301] **(58)**

```
> Orbk(4,z,F,[-1.39,2.,.343,5.],1000,1010)
```

[6.301659410, 0.8016378381, 6.301659410, 0.8016378381, 6.301659410, 0.8016378381, 6.301659410, 0.8016378381, 6.301659410, 0.8016378381, 6.301659410] **(59)**

```
> Help(ToSys)
```

ToSys(k,z,f): 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)), it gives the unerlying transormation, followed by the set of variables

Try:
ToSys(2,z,z[1]+z[2]); **(60)**

```
> G:=ToSys(4,z,F)
```

$$G := \left[\frac{3+z_2+z_3+z_4}{1+z_1+z_3}, z_1, z_2, z_3\right], [z_1, z_2, z_3, z_4] \quad \textbf{(61)}$$

```
> Help(SFP)
```

SFP(F,x): Given a transformation F in the list of variables finds all the STABLE fixed point of the transformation x->F(x), i.e. the set of solutions of
the system {x[1]=F[1], ..., x[k]=F[k]}} that are stable. Try:
*SFP([5/2*x*(1-x)],[x]]);*
*SFP([(1+x+y)/(2+3*x+y), (3+x+2*y)/(5+x+3*y)],[x,y]]);* **(62)**

```
> T:=SFP(G,z)[1]
```

$T := [1.822875656, 1.822875656, 1.822875656, 1.822875656]$ **(63)**

```
> Orbk(4,z,F,T,1000,1010)
```

[1.822875656, 1.822875655, 1.822875656, 1.822875655, 1.822875656, 1.822875655, 1.822875656, 1.822875655, 1.822875656, 1.822875655, 1.822875656] **(64)**