Lucky Lao

DSC-VI : Practical-01

Family of Solutions: First Order Differential Equations

We'll plot the family of solutions of the following first order differential equations:

1 y' = x² where y(0)=k

1.1 Using the pre-defined function 'ode2()' (works for an O.D.E. of order upto 2)

```
-- ratprint : false $            /* suppresses error messages */
>  kill ( all ) $                /* clear all user-defined variables */
   de : ' diff ( y , x ) = x ^ 2 ;       /* the eqn. is y' = x^2 */
   sol : ode2 ( de , y , x ) ;       /* `sol` is assigned the General soln. of `de`*/
   sol1 : ic1 ( sol , x = 0 , y = k ) ;  /* `sol1` is a particular solution, w/ def. constt. %c being replaced by
   `k` */
   v1 : ev ( sol1 , k = − 2 ) ;       /* random values are given to `k` */
   v2 : ev ( sol1 , k = − 1 ) ;
   v3 : ev ( sol1 , k = 1 ) ;
   v4 : ev ( sol1 , k = 2 ) ;

   /* To plot the graphs */
   wxplot2d ( [ rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) ] ,
        [ x , − 3 , 3 ] ,
        [ style , [ lines , 1 ] , [ lines , 2 ] , [ lines , 3 ] , [ lines , 4 ] ] ) $
```

$$\text{(de)}\ \frac{d}{dx}y = x^2$$

$$\text{(sol)}\ y = \frac{x^3}{3} + \%c$$

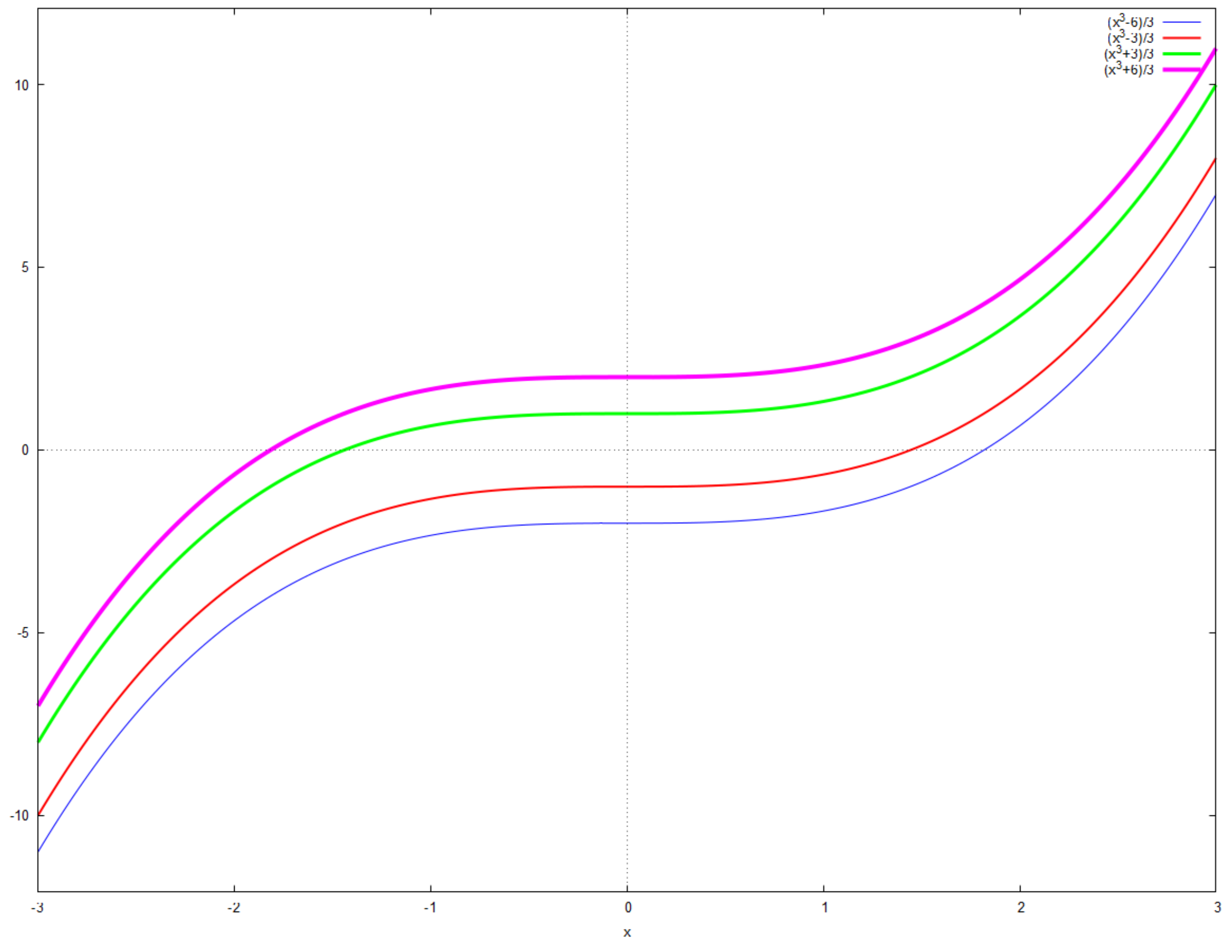
$$\text{(sol1)}\ y = \frac{x^3 + 3k}{3}$$

$$\text{(v1)}\ y = \frac{x^3 - 6}{3}$$

$$\text{(v2)}\ y = \frac{x^3 - 3}{3}$$

$$\text{(v3)}\ y = \frac{x^3 + 3}{3}$$

$$\text{(v4)}\ y = \frac{x^3 + 6}{3}$$



1.2 Using the pre-defined function 'desolve()' (works for an O.D.E. of any order)

```
--> ratprint : false $
    kill ( all ) $                    /* clear all user-defined variables */
    de : diff ( y ( x ) , x ) = x ^ 2 ;       /* y is explicitly written as a function of x */
    sol : desolve ( de , y ( x ) ) ;        /* doesn't give constt.s explicitly but their values */
    sol1 : ev ( sol , y ( 0 ) = k ) ;
    v1 : ev ( sol1 , k = − 2 ) ;
    v2 : ev ( sol1 , k = − 1 ) ;
    v3 : ev ( sol1 , k = 1 ) ;
    v4 : ev ( sol1 , k = 2 ) ;

    /* To plot the graphs */
    wxplot2d ( [ rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) ] ,
        [ x , − 3 , 3 ] ,
        [ style , [ lines , 1 ] , [ lines , 2 ] , [ lines , 3 ] , [ lines , 4 ] ] ) $
```

$$\text{(de)}\ \frac{d}{dx}\mathrm{y}(x)=x^2$$

$$\text{(sol)}\ \mathrm{y}(x)=\frac{x^3}{3}+\mathrm{y}(0)$$

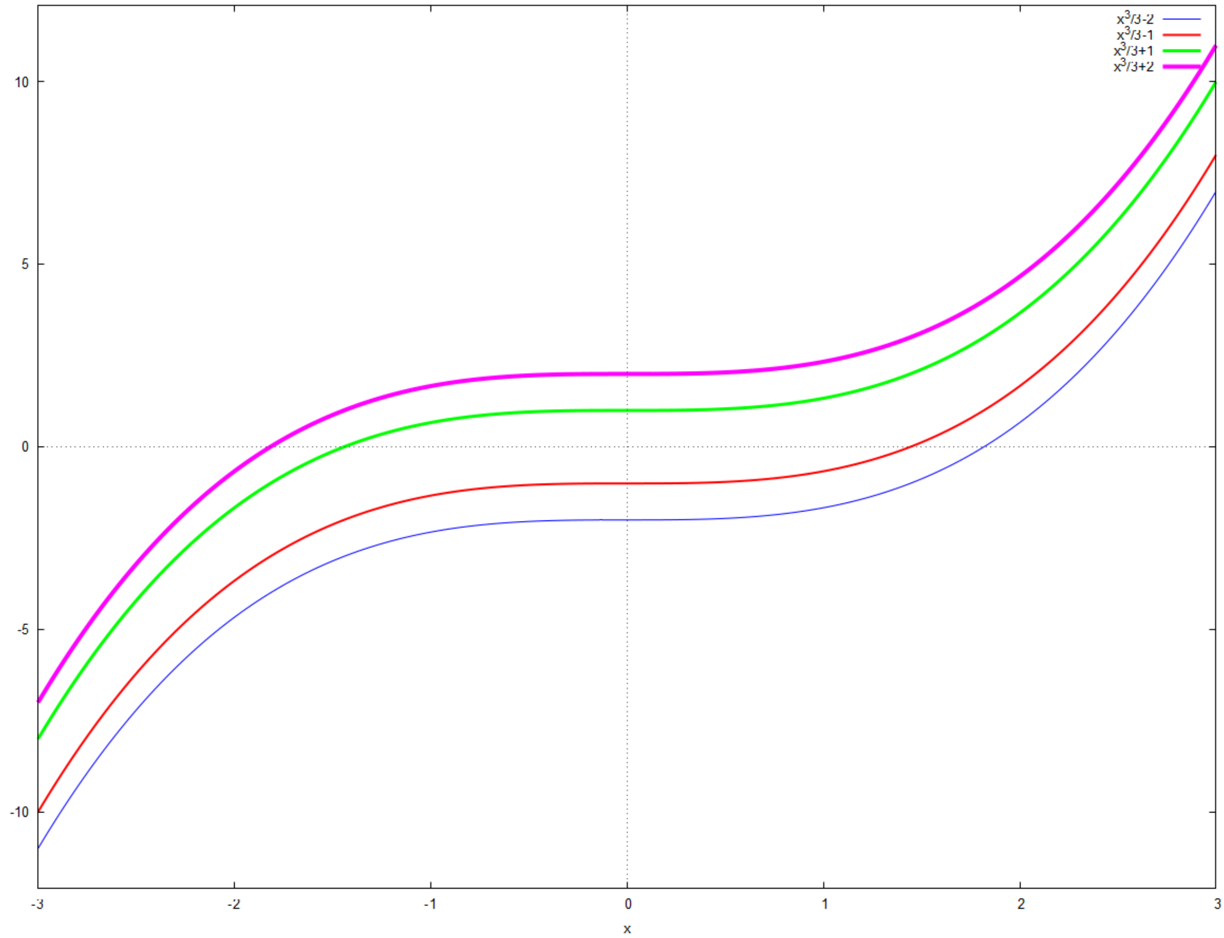
$$\text{(sol 1)}\ \mathrm{y}(x)=\frac{x^3}{3}+k$$

$$\text{(v1)}\ \mathrm{y}(x)=\frac{x^3}{3}-2$$

$$\text{(v2)}\ \mathrm{y}(x)=\frac{x^3}{3}-1$$

$$(\text{v3})\ \mathrm{y}(x) = \frac{x^3}{3} + 1$$

$$(\text{v4})\ \mathrm{y}(x) = \frac{x^3}{3} + 2$$



2 y' = 9.8 - 0.196y

2.1 Using 'ode2()'

```
--> ratprint : false $
    kill ( all ) $
    de : ' diff ( y , x ) = 9 . 8 − 0 . 196 · y ;
    gsol : ode2 ( de , y , x ) ;
    psol : ic1 ( gsol , x = 0 , y = k ) ;
    v0 : ev ( psol , k = 0 ) ;
    v1 : ev ( psol , k = 1 ) ;
    v2 : ev ( psol , k = 2 ) ;
    v3 : ev ( psol , k = − 1 ) ;
    v4 : ev ( psol , k = − 2 ) ;
    wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) ] ,
        [ x , − 1 , 1 ] ,
        [ style , [ lines , 1 ] , [ lines , 2 ] , [ lines , 3 ] , [ lines , 4 ] , [ lines , 5 ] ] ) $
```

$$(\text{de})\ \frac{d}{dx}y = 9.8 - 0.196y$$

$$(\text{gsol})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} + \%c\right)$$

$$(\mathrm{psol})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} + k - 50\right)$$

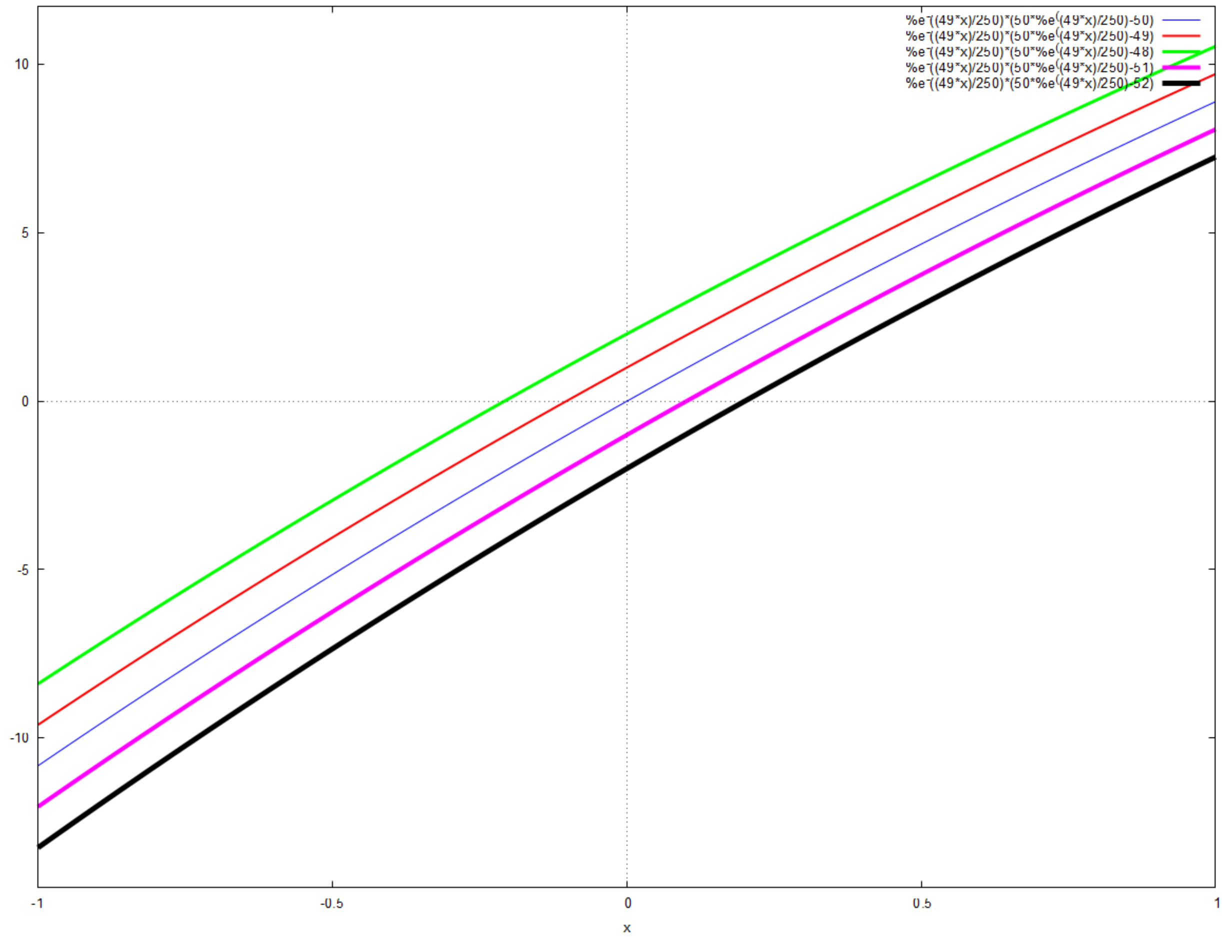
$$(\mathrm{v0})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} - 50\right)$$

$$(\mathrm{v1})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} - 49\right)$$

$$(\mathrm{v2})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} - 48\right)$$

$$(\mathrm{v3})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} - 51\right)$$

$$(\mathrm{v4})\ y = \%e^{-\left(\frac{49x}{250}\right)}\left(50\%e^{\frac{49x}{250}} - 52\right)$$



2.2 Using 'desolve()'

```
--> ratprint : false $
    kill ( all ) $
    de : diff ( y ( x ) , x ) = 9 . 8 − 0 . 196 · y ( x ) ;
    gsol : desolve ( de , y ( x ) ) ;
    psol : ev ( gsol , y ( 0 ) = k ) ;
    v0 : ev ( psol , k = 0 ) ;
    v1 : ev ( psol , k = − 4 ) ;
    v2 : ev ( psol , k = − 2 ) ;
    v3 : ev ( psol , k = 2 ) ;
    v4 : ev ( psol , k = 4 ) ;
    wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) ] ,
```

[x , − 3 , 3] ,
[style , [lines , 1] , [lines , 2] , [lines , 3] , [lines , 4] , [lines , 5]]) $

$$\text{(de)}\ \frac{d}{dx}\,\mathrm{y}(x)=9.8\text{ - }0.196\,\mathrm{y}(x)$$

$$\text{(gsol)}\ \mathrm{y}(x)=\frac{(250\,\mathrm{y}(0)\text{ - }12500)\%e^{-\left(\frac{49x}{250}\right)}}{250}+50$$

$$\text{(psol)}\ \mathrm{y}(x)=\frac{(250k\text{ - }12500)\%e^{-\left(\frac{49x}{250}\right)}}{250}+50$$

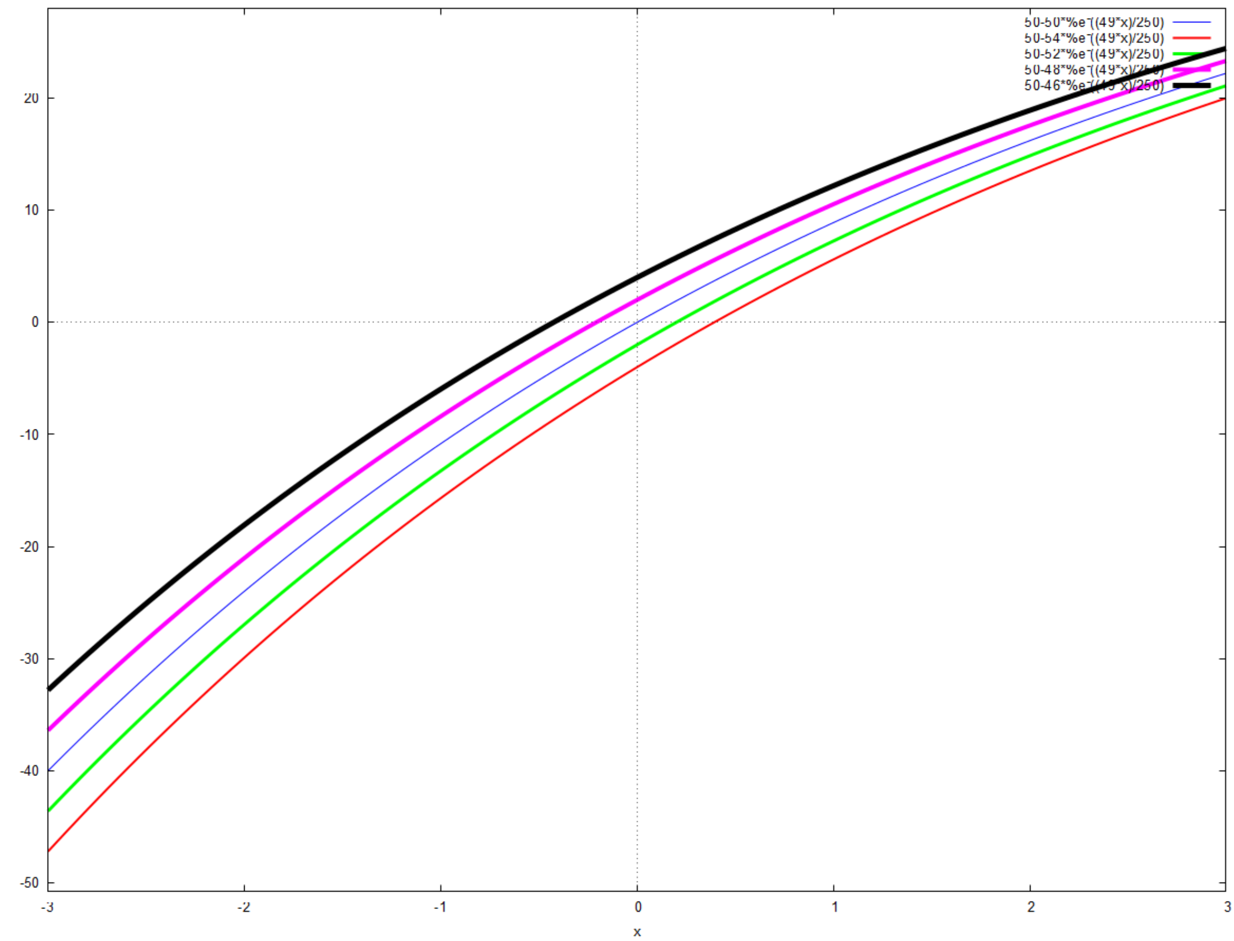
$$\text{(v0)}\ \mathrm{y}(x)=50\text{ - }50\%e^{-\left(\frac{49x}{250}\right)}$$

$$\text{(v1)}\ \mathrm{y}(x)=50\text{ - }54\%e^{-\left(\frac{49x}{250}\right)}$$

$$\text{(v2)}\ \mathrm{y}(x)=50\text{ - }52\%e^{-\left(\frac{49x}{250}\right)}$$

$$\text{(v3)}\ \mathrm{y}(x)=50\text{ - }48\%e^{-\left(\frac{49x}{250}\right)}$$

$$\text{(v4)}\ \mathrm{y}(x)=50\text{ - }46\%e^{-\left(\frac{49x}{250}\right)}$$



3 y'cos(x) + ysin(x) = 2cos³(x)sin(x)-1

```
--> ratprint : false $
    kill ( all ) $
    de : ' diff ( y , x ) · cos ( x ) + y · sin ( x ) = 2 · ( cos ( x ) ) ^ 3 · sin ( x ) − 1 ;
    gsol : ode2 ( de , y , x ) ;
```

```
psol : ic1 ( gsol , x = 0 , y = k ) ;
v0 : ev ( psol , k = 0 ) ;
v1 : ev ( psol , k = − 1 ) ;
v2 : ev ( psol , k = − 2 ) ;
v3 : ev ( psol , k = 1 ) ;
v4 : ev ( psol , k = 2 ) ;
wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) ] ,
    [ x , − 1 , 1 ] ,
    [ style , [ lines , 1 ] , [ lines , 2 ] , [ lines , 3 ] , [ lines , 4 ] , [ lines , 5 ] ] ) $
```

$$\text{(de)}\ \cos(x)\left(\frac{d}{dx}y\right)+\sin(x)y=2\cos(x)^3\sin(x)-1$$

$$\text{(gsol)}\ y=\cos(x)\left(-\left(\frac{1}{\tan(x)^2+1}\right)-\tan(x)+\%c\right)$$

$$\text{(psol)}\ y=-\left(\frac{\cos(x)\tan(x)^3+(-k-1)\cos(x)\tan(x)^2+\cos(x)\tan(x)-k\cos(x)}{\tan(x)^2+1}\right)$$

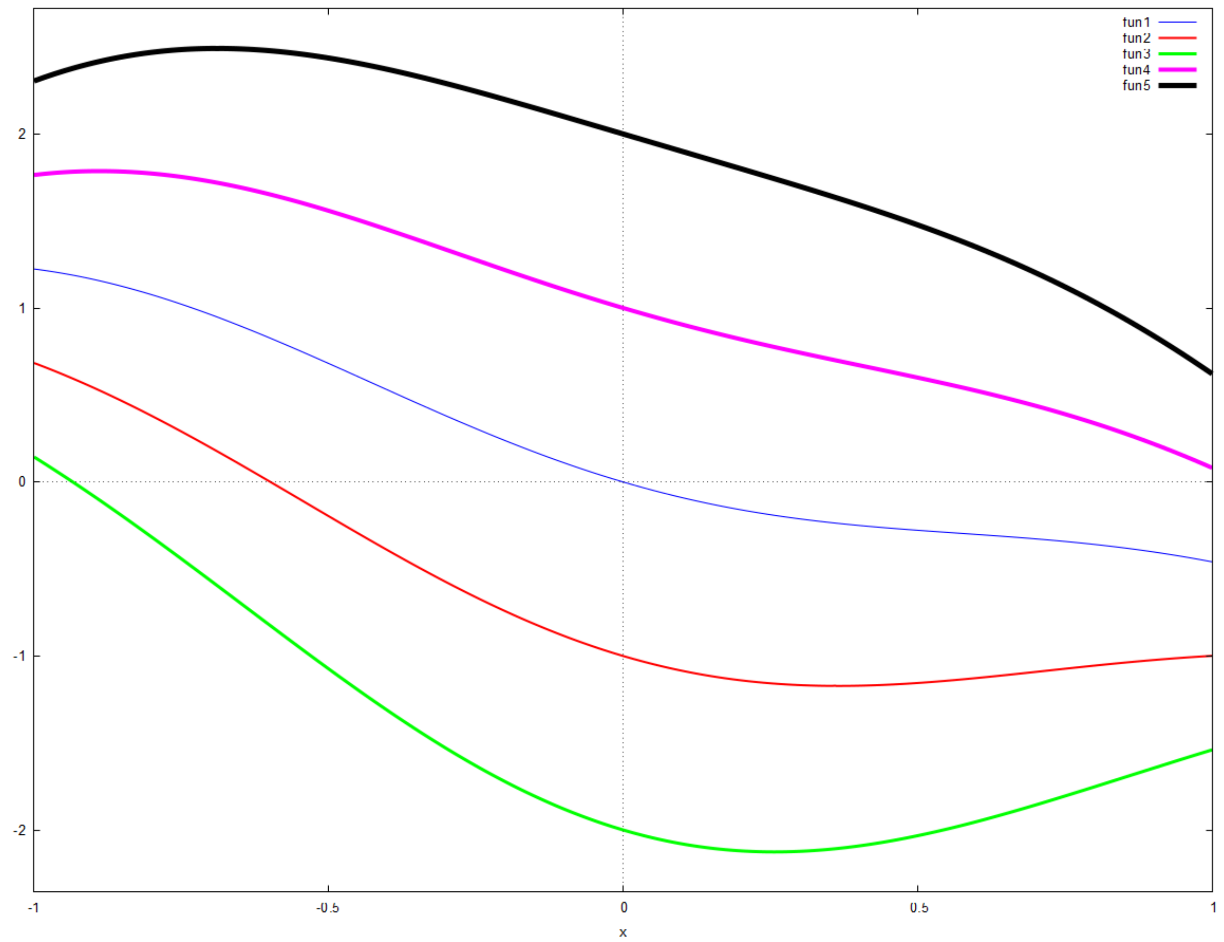
$$\text{(v0)}\ y=-\left(\frac{\cos(x)\tan(x)^3-\cos(x)\tan(x)^2+\cos(x)\tan(x)}{\tan(x)^2+1}\right)$$

$$\text{(v1)}\ y=-\left(\frac{\cos(x)\tan(x)^3+\cos(x)\tan(x)+\cos(x)}{\tan(x)^2+1}\right)$$

$$\text{(v2)}\ y=-\left(\frac{\cos(x)\tan(x)^3+\cos(x)\tan(x)^2+\cos(x)\tan(x)+2\cos(x)}{\tan(x)^2+1}\right)$$

$$\text{(v3)}\ y=-\left(\frac{\cos(x)\tan(x)^3-2\cos(x)\tan(x)^2+\cos(x)\tan(x)-\cos(x)}{\tan(x)^2+1}\right)$$

$$\text{(v4)}\ y=-\left(\frac{\cos(x)\tan(x)^3-3\cos(x)\tan(x)^2+\cos(x)\tan(x)-2\cos(x)}{\tan(x)^2+1}\right)$$



Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-02

Family of Solutions: Second Order Differential Equations

We'll now plot the family of solutions of the following second order differential equations:

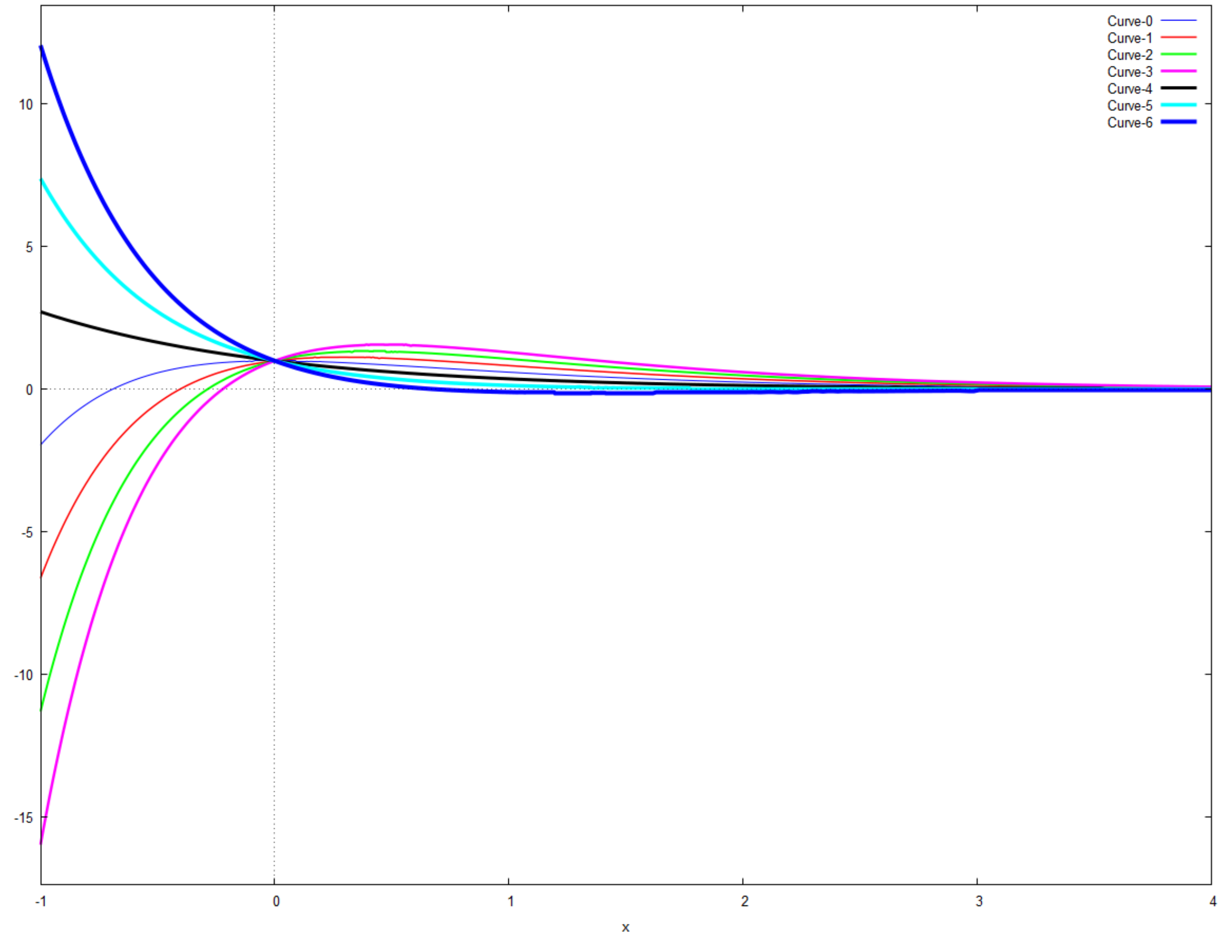
1 y'' + 3y' + 2y = 0 where y'(0)=b, y(0)=1, varying between -3 and 3

```
-- /* We'll use 'ode2()' */
>  ratprint : false $

   de : ' diff ( y , x , 2 ) + 3 · ' diff ( y , x ) + 2 · y = 0 ;
   gsol : ode2 ( de , y , x ) ; /* general soln. */
   psol : ic2 ( gsol , x = 0 , y = 1 , ' diff ( y , x ) = b ) ;    /* particular soln. */

   /* Fixing values for 'b' */
   v0 : ev ( psol , b = 0 ) $ /* suppress o/p */
   v1 : ev ( psol , b = 1 ) $
   v2 : ev ( psol , b = 2 ) $
   v3 : ev ( psol , b = 3 ) $
   v4 : ev ( psol , b = − 1 ) $
   v5 : ev ( psol , b = − 2 ) $
   v6 : ev ( psol , b = − 3 ) $

   /* Using 'wxplot2d()' to plot the family of solutions */
   wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) , rhs ( v5 ) , rhs ( v6 ) ] ,
        [ x , − 1 , 4 ] ,
        [ style , [ lines , 1 ] , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] , [ lines , 3 ] , [ lines , 3 . 5 ] , [
   lines , 4 ] ] ,
        [ legend , "Curve-0" , "Curve-1" , "Curve-2" , "Curve-3" , "Curve-4" , "Curve-5" , "Curve-6" ,
   "Curve-7" ] ) ;
```

$$\frac{d^2}{dx^2}y + 3\left(\frac{d}{dx}y\right) + 2y = 0$$

$$y = \%k1\%e^{-x} + \%k2\%e^{-2x}$$

$$y = (b+2)\%e^{-x} + (-b-1)\%e^{-2x}$$



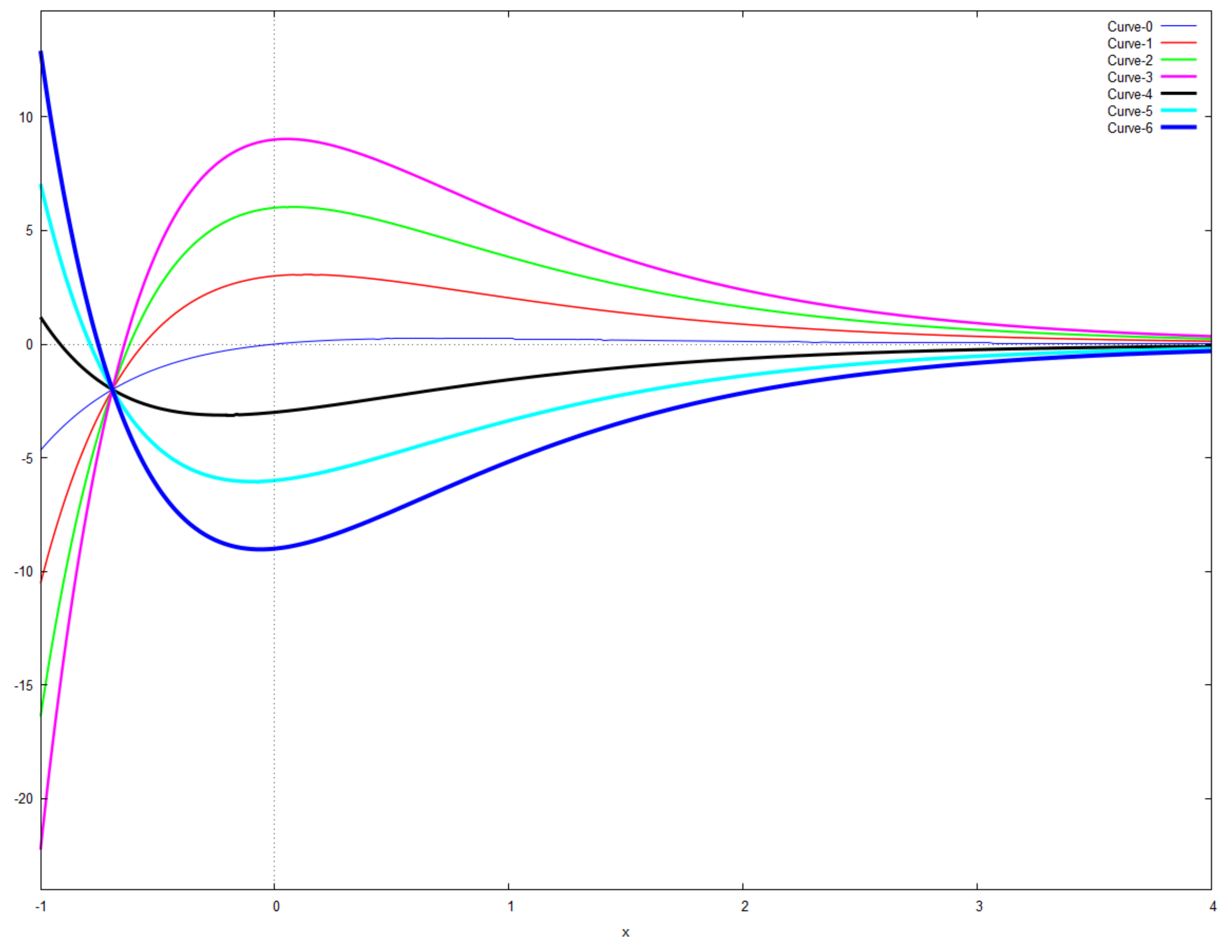
2 y'' + 3y' + 2y = 0 where y(0)=a, y'(0)=1

```
-- /* We'll use 'ode2()' */
>  kill ( all ) $

   de : ' diff ( y , x , 2 ) + 3 · ' diff ( y , x ) + 2 · y = 0 ;
   gsol : ode2 ( de , y , x ) ; /* general soln. */
   psol : ic2 ( gsol , x = 0 , y = a , ' diff ( y , x ) = 1 ) ;    /* particular soln. */

   /* Fixing values for 'a' */
   v0 : ev ( psol , a = 0 ) $ /* suppress o/p */
   v1 : ev ( psol , a = 3 ) $
   v2 : ev ( psol , a = 6 ) $
   v3 : ev ( psol , a = 9 ) $
   v4 : ev ( psol , a = − 3 ) $
   v5 : ev ( psol , a = − 6 ) $
   v6 : ev ( psol , a = − 9 ) $

   /* Using 'wxplot2d()' to plot the family of solutions */
   wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) , rhs ( v5 ) , rhs ( v6 ) ] ,
         [ x , − 1 , 4 ] ,
         [ style , [ lines , 1 ] , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] , [ lines , 3 ] , [ lines , 3 . 5 ] , [
   lines , 4 ] ] ,
         [ legend , "Curve-0" , "Curve-1" , "Curve-2" , "Curve-3" , "Curve-4" , "Curve-5" , "Curve-6" ,
   "Curve-7" ] ) ;
```

$$\frac{d^2}{dx^2} y + 3 \left(\frac{d}{dx} y \right) + 2y = 0$$

$$y = \%k1\%e^{-x} + \%k2\%e^{-2x}$$

$$y = (2a+1)\%e^{-x} + (-a-1)\%e^{-2x}$$



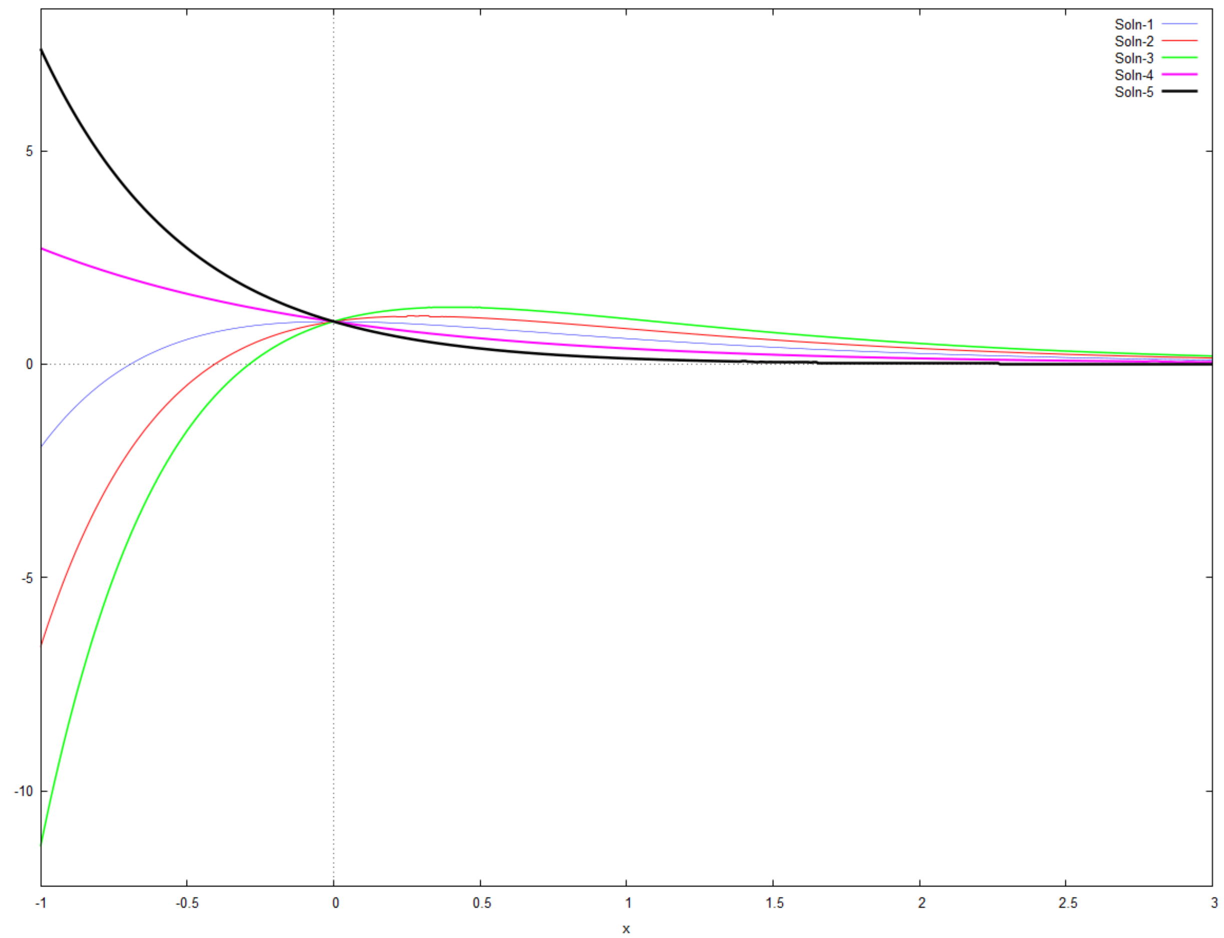
3 Solving (1) using desolve()
y'' + 3y' +2y = 0 where y'(0)=b, y(0)=1, varying between -3 and 3

```
--> kill ( all ) $

    de : diff ( y ( x ) , x , 2 ) + 3 · diff ( y ( x ) , x ) + 2 · y ( x ) = 0 ;
    gsol : desolve ( de , y ( x ) ) ;    /* general soln. */
    psol : ev ( gsol , y ( 0 ) = 1 , diff ( y ( x ) , x ) = b ) ; /* particular soln. */

    /* Fixing values for 'b' */
    v0 : ev ( psol , b = 0 ) $
    v1 : ev ( psol , b = 1 ) $
    v2 : ev ( psol , b = 2 ) $
    v3 : ev ( psol , b = − 1 ) $
    v4 : ev ( psol , b = − 2 ) $

    /* Using 'wxplot2d()' to plot the family of solutions */
    wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) ] ,
        [ x , − 1 , 3 ] ,
        [ style , [ lines , 0 . 5 ] , [ lines , 1 ] , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] ] ,
        [ legend , "Soln-1" , "Soln-2" , "Soln-3" , "Soln-4" , "Soln-5" ] ) $
```

$$\frac{d^2}{dx^2}\mathrm{y}(x)+3\left(\frac{d}{dx}\mathrm{y}(x)\right)+2\,\mathrm{y}(x)=0$$

$$\mathrm{y}(x)=\%e^{-x}\left(\left.\frac{d}{dx}\mathrm{y}(x)\right|_{x=0}+2\,\mathrm{y}(0)\right)+\%e^{-2x}\left(-\left.\frac{d}{dx}\mathrm{y}(x)\right|_{x=0}-\mathrm{y}(0)\right)$$

$$\mathrm{y}(x)=(b+2)\%e^{-x}+(-b-1)\%e^{-2x}$$



4 y'' + y' - 6y = 0 (No initial conditions given)

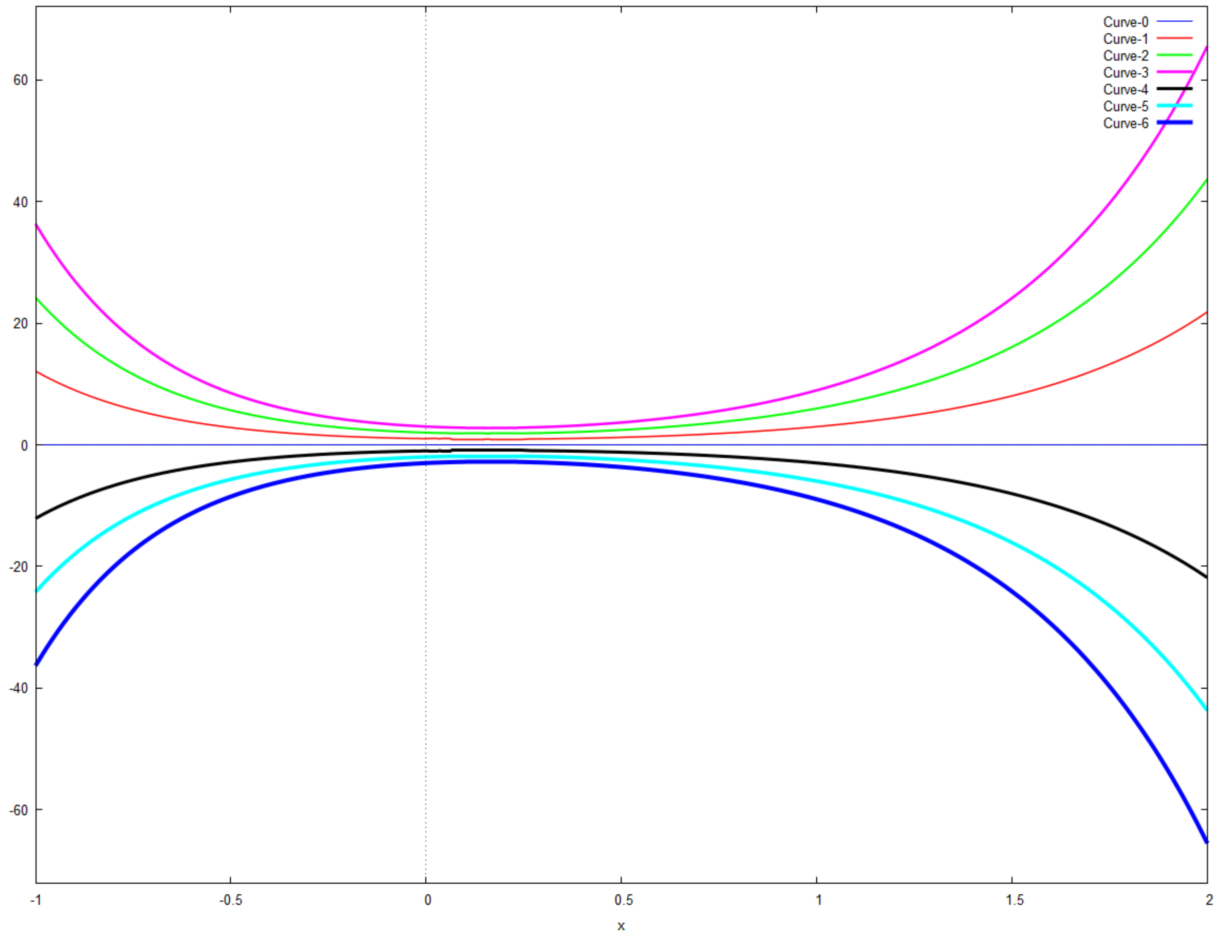
```
-- /* We'll use 'ode2()' */
> kill ( all ) $

de : ' diff ( y , x , 2 ) + ' diff ( y , x ) − 6 · y = 0 ;
gsol : ode2 ( de , y , x ) ; /* general soln. */
psol : ic2 ( gsol , x = 0 , y = c , ' diff ( y , x ) = k ) ;   /* particular soln. */

/* Fixing values for 'c' and 'k' */
v0 : ev ( psol , c = 0 , k = 0 ) $
v1 : ev ( psol , c = 1 , k = − 1 ) $
v2 : ev ( psol , c = 2 , k = − 2 ) $
v3 : ev ( psol , c = 3 , k = − 3 ) $
v4 : ev ( psol , c = − 1 , k = 1 ) $
v5 : ev ( psol , c = − 2 , k = 2 ) $
v6 : ev ( psol , c = − 3 , k = 3 ) $

/* Using wxplot2d() to plot the family of solutions */
wxplot2d ( [ rhs ( v0 ) , rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) , rhs ( v5 ) , rhs ( v6 ) ] ,
     [ x , − 1 , 2 ] ,
```

[style , [lines , 1] , [lines , 1 . 5] , [lines , 2] , [lines , 2 . 5] , [lines , 3] , [lines , 3 . 5] , [lines , 4]] ,
[legend , "Curve-0" , "Curve-1" , "Curve-2" , "Curve-3" , "Curve-4" , "Curve-5" , "Curve-6" , "Curve-7"]) ;

$$\frac{d^2}{dx^2}y+\frac{d}{dx}y-6y=0$$

$$y=\%\mathrm{k1}\%e^{2x}+\%\mathrm{k2}\%e^{-3x}$$

$$y=\frac{(k+3c)\%e^{2x}}{5}+\frac{(2c-k)\%e^{-3x}}{5}$$



Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

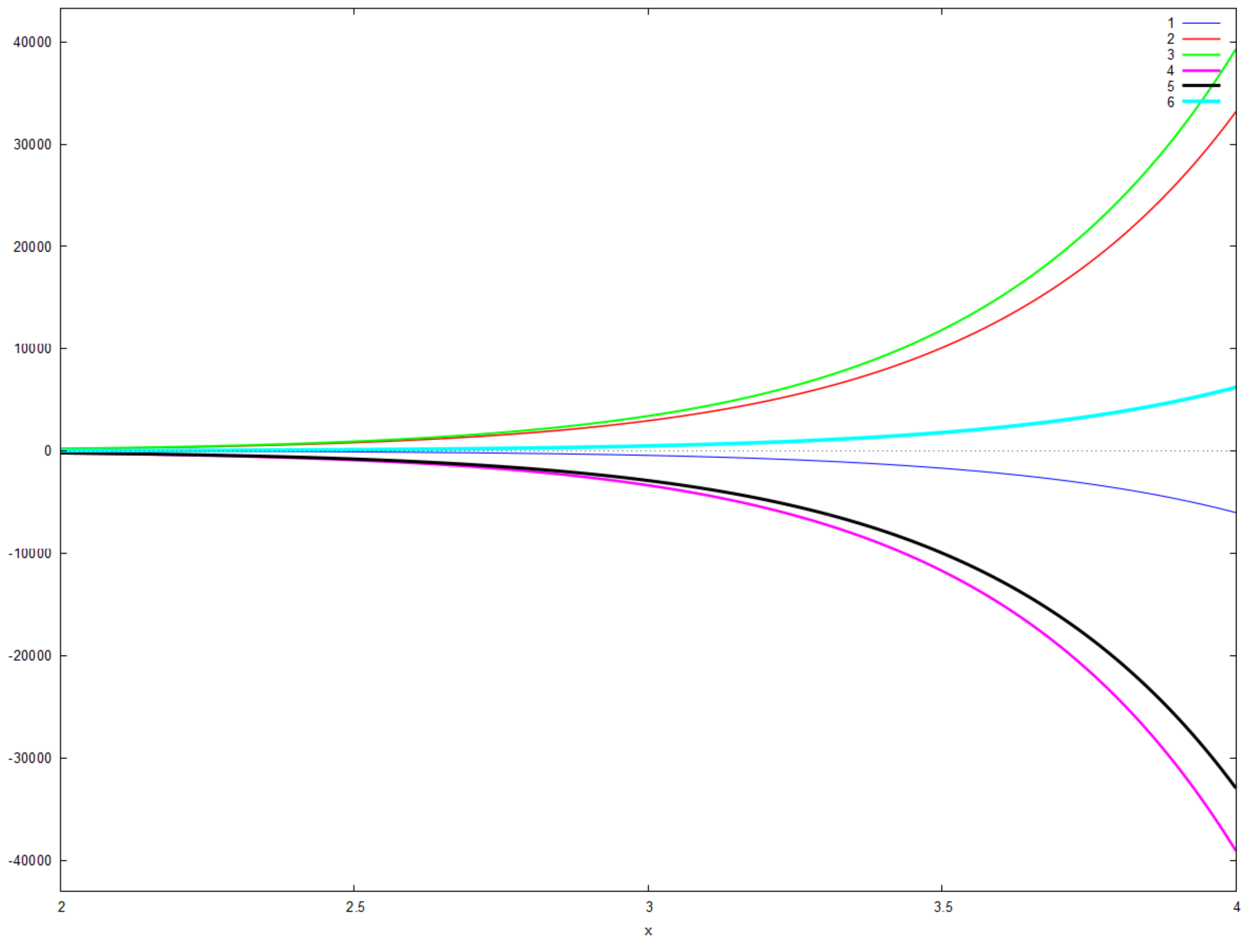
DSC-VI : Practical-03

Family of Solutions: Third Order Differential Equations

We'll now plot the family of solutions of the following third order differential equations:

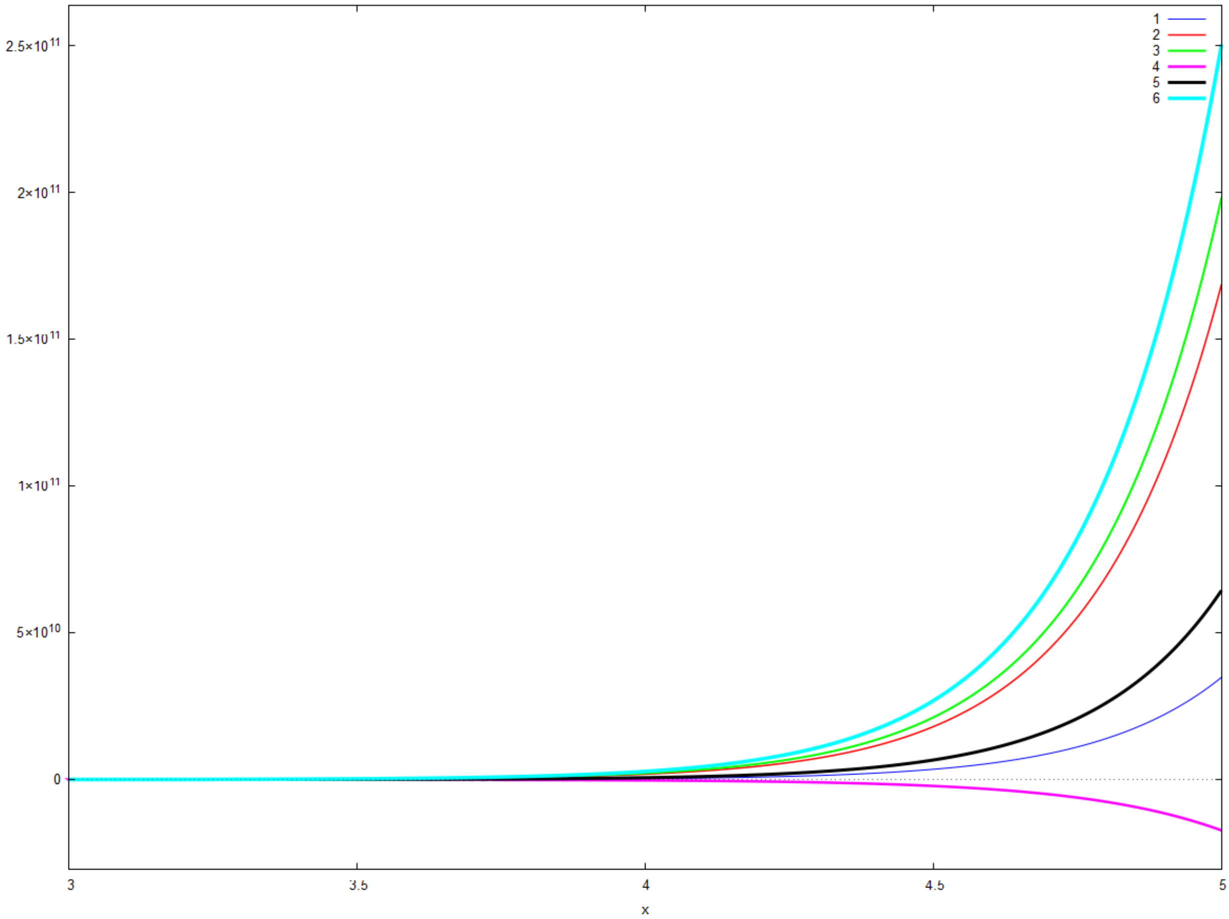
1 y''' - 5y'' + 8y' - 4y = 0

```
--> kill ( all ) $
    de : diff ( y ( x ) , x , 3 ) − 5 · diff ( y ( x ) , x , 2 ) + 8 · diff ( y ( x ) , x ) − 4 · y ( x ) = 0 $
    gsol : desolve ( de , y ( x ) ) $
    psol : ev ( gsol , y ( 0 ) = c1 , diff ( y ( x ) , x ) = c2 , diff ( y ( x ) , x , 2 ) = c3 ) $
    s1 : ev ( psol , c1 = 1 , c2 = 2 , c3 = 3 ) $
    s2 : ev ( psol , c1 = 2 , c2 = 1 , c3 = 3 ) $
    s3 : ev ( psol , c1 = 3 , c2 = 1 , c3 = 2 ) $
    s4 : ev ( psol , c1 = 1 , c2 = 3 , c3 = 2 ) $
    s5 : ev ( psol , c1 = 2 , c2 = 3 , c3 = 1 ) $
    s6 : ev ( psol , c1 = 3 , c2 = 2 , c3 = 1 ) $

    wxplot2d ( [ rhs ( s1 ) , rhs ( s2 ) , rhs ( s3 ) , rhs ( s4 ) , rhs ( s5 ) , rhs ( s6 ) ] ,
        [ x , 2 , 4 ] ,
        [ style , [ lines , 1 ] , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] , [ lines , 3 ] , [ lines , 3 . 5 ] ] ,
        [ legend , "1" , "2" , "3" , "4" , "5" , "6" ] ) $
```



2 y''' - 12y'' + 48y' - 64y = 12 - 32exp(-8x) + 2exp(4x)

```
--  kill ( all ) $
>   de : diff ( y ( x ) , x , 3 ) − 12 · diff ( y ( x ) , x , 2 ) + 48 · diff ( y ( x ) , x ) − 64 · y ( x ) = 12 − 32 · exp
    ( − 8 · x ) + 2 · exp ( 4 · x ) $
    gsol : desolve ( de , y ( x ) ) $
    psol : ev ( gsol , y ( 0 ) = k1 , diff ( y ( x ) , x ) = k2 , diff ( y ( x ) , x , 2 ) = k3 ) $
    s1 : ev ( psol , k1 = 1 , k2 = 2 , k3 = 3 ) $
    s2 : ev ( psol , k1 = 2 , k2 = 1 , k3 = 3 ) $
    s3 : ev ( psol , k1 = 3 , k2 = 2 , k3 = 1 ) $
    s4 : ev ( psol , k1 = 1 , k2 = 3 , k3 = 2 ) $
    s5 : ev ( psol , k1 = 2 , k2 = 3 , k3 = 1 ) $
    s6 : ev ( psol , k1 = 3 , k2 = 1 , k3 = 2 ) $

    wxplot2d ( [ rhs ( s1 ) , rhs ( s2 ) , rhs ( s3 ) , rhs ( s4 ) , rhs ( s5 ) , rhs ( s6 ) ] ,
        [ x , 3 , 5 ] ,
        [ style , [ lines , 1 ] , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] , [ lines , 3 ] , [ lines , 3 . 5 ] ] ,
        [ legend , "1" , "2" , "3" , "4" , "5" , "6" ] ) $
```



Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-04

Exponential Growth/Decay Model

x(t): population at time t
a: per capita death rate
b: per capita birth rate
initial condition: x(0) = x_0

1 Exponential Growth

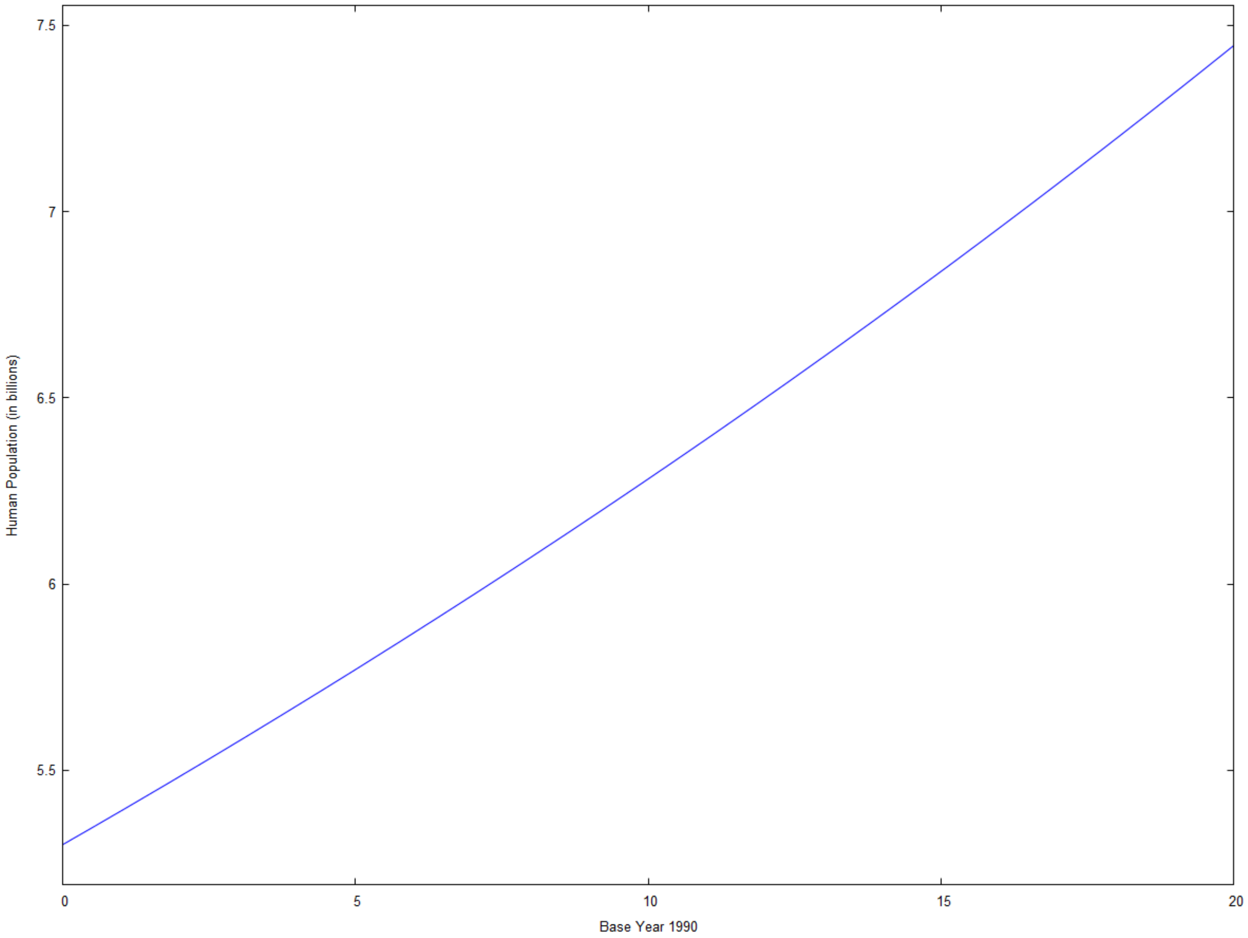
```
--> kill ( all ) $
    eqn : ' diff ( x , t ) = b · x − a · x ;
    sol : ode2 ( eqn , x , t ) ;
    fsol : ic1 ( sol , x = x_0 , t = 0 ) ;
    fsol1 : ev ( fsol , a = 0 . 010 , b = 0 . 027 , x_0 = 5 . 3 ) ;    /* b > a */
    wxplot2d ( rhs ( fsol1 ) ,
        [ t , 0 , 20 ] ,
        [ xlabel , "Base Year 1990" ] ,
        [ ylabel , "Human Population (in billions)" ] ) $
```

$$\frac{d}{dt}x = bx - ax$$

$$x = \%c\%e^{(b-a)t}$$

$$x = \%e^{bt-at}x_0$$

$$x = 5.3\%e^{0.017t}$$



2 Exponential Decay

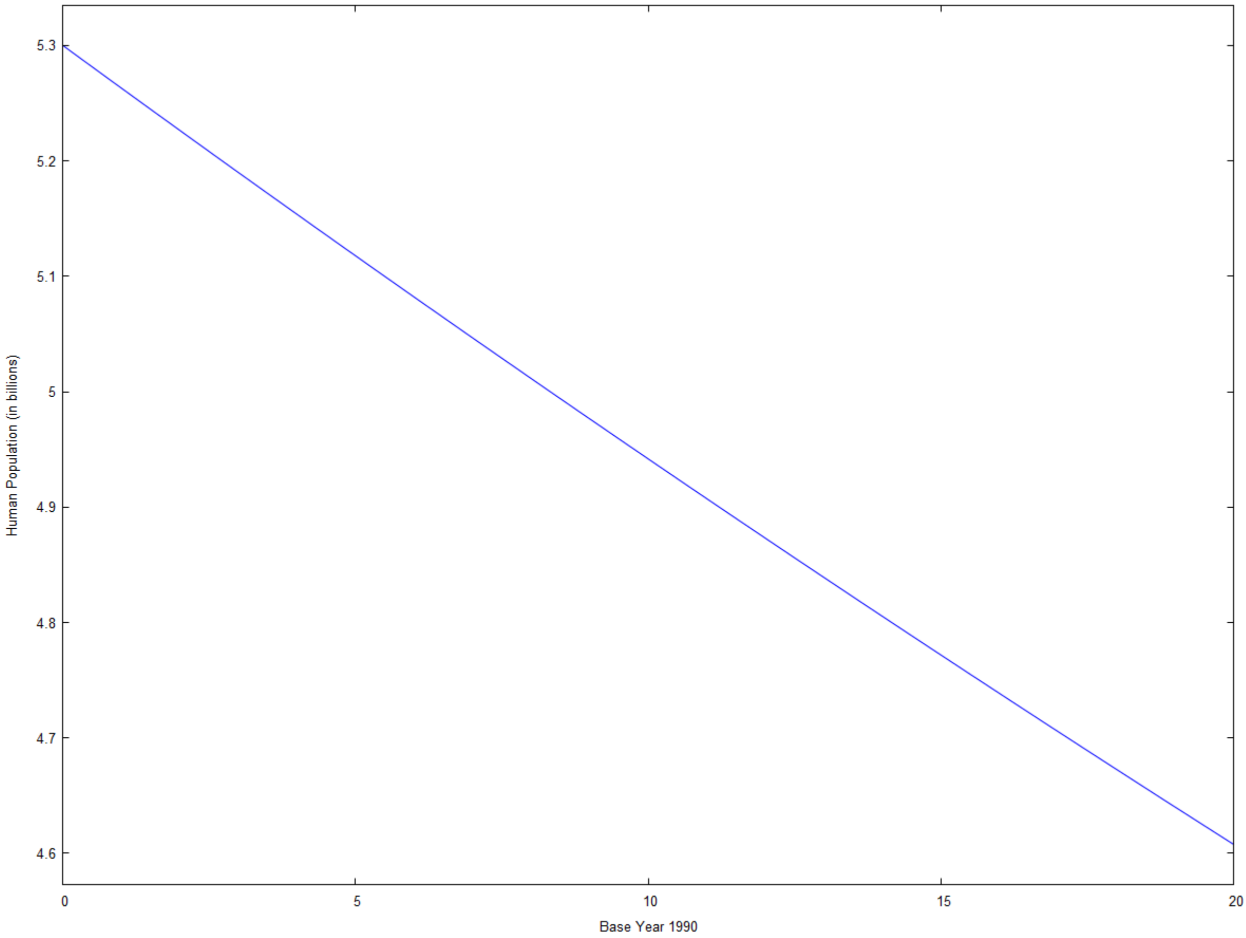
```
--> kill ( all ) $
    eqn : ' diff ( x , t ) = b · x − a · x ;
    sol : ode2 ( eqn , x , t ) ;
    fsol : ic1 ( sol , x = x_0 , t = 0 ) ;
    fsol1 : ev ( fsol , a = 0 . 027 , b = 0 . 020 , x_0 = 5 . 3 ) ;     /* b > a */
    wxplot2d ( rhs ( fsol1 ) ,
        [ t , 0 , 20 ] ,
        [ xlabel , "Base Year 1990" ] ,
        [ ylabel , "Human Population (in billions)" ] ) $
```

$$\frac{d}{dt}x = bx - ax$$

$$x = \%c\%e^{(b-a)t}$$

$$x = \%e^{bt-at}x_0$$

$$x = 5.3\%e^{-0.006999999999999999t}$$



Created with wxMaxima.

The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-05

Method of Variation of Parameters

1 Solve y'' - y = x

Sol: The general solution comprises of two parts: y = y_c + y_p

We have:

```
--> de : ' diff ( y , x , 2 ) − y = x ;     /* our d.e. */
    hp : lhs ( de ) = 0 $    /* homogeneous part */
    r : rhs ( de ) $        /* `r` i.e. non-homogeneous part */
```

$$\frac{d^2}{dx^2}y - y = x$$

1.1 Calculating y_c

```
--> y_c : rhs ( ode2 ( hp , y , x ) ) ;
    y_1 : exp ( x ) $
    y_2 : exp ( − x ) $
```

$$\%k1\%e^{x} + \%k2\%e^{-x}$$

1.2 Calculating y_p

```
--> A : matrix (
      [ y_1 , y_2 ] ,
      [ diff ( y_1 , x ) , diff ( y_2 , x ) ]
    ) ;
```

$$\begin{pmatrix} \%e^{x} & \%e^{-x} \\ \%e^{x} & -\%e^{-x} \end{pmatrix}$$

```
--> W : determinant ( A ) ;
```

$$-2$$

We now find u_1 and u_2

```
--> u_1 : integrate ( − y_2 · r / W , x ) ;
    u_2 : integrate ( y_1 · r / W , x ) ;
```

$$\frac{(-x-1)\%e^{-x}}{2}$$

$$-\frac{(x-1)\%e^{x}}{2}$$

Now, our y_p is:

```
--> y_p : ratsimp ( u_1 · y_1 + u_2 · y_2 ) ;    /* this will return a simplified expression */
```

$$-x$$

1.3 General Solution

The general solution (y=y_c+y_p):

```
--> 'y = y_c + y_p ;
```

$$y = \%\mathrm{k1}\%e^{x} + \%\mathrm{k2}\%e^{-x} - x$$

Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-06

Lake Pollution Model

1 Constant flow and constant pollution concentration inflow

c(t): concentration of pollutant in the lake at time t.
F: constant flow rate.
V: constant volume of the lake.
cin: constant concentration of pollutant in the flow entering the lake.
initial condition: c(0)=c0.(5 different initial conditions taken)

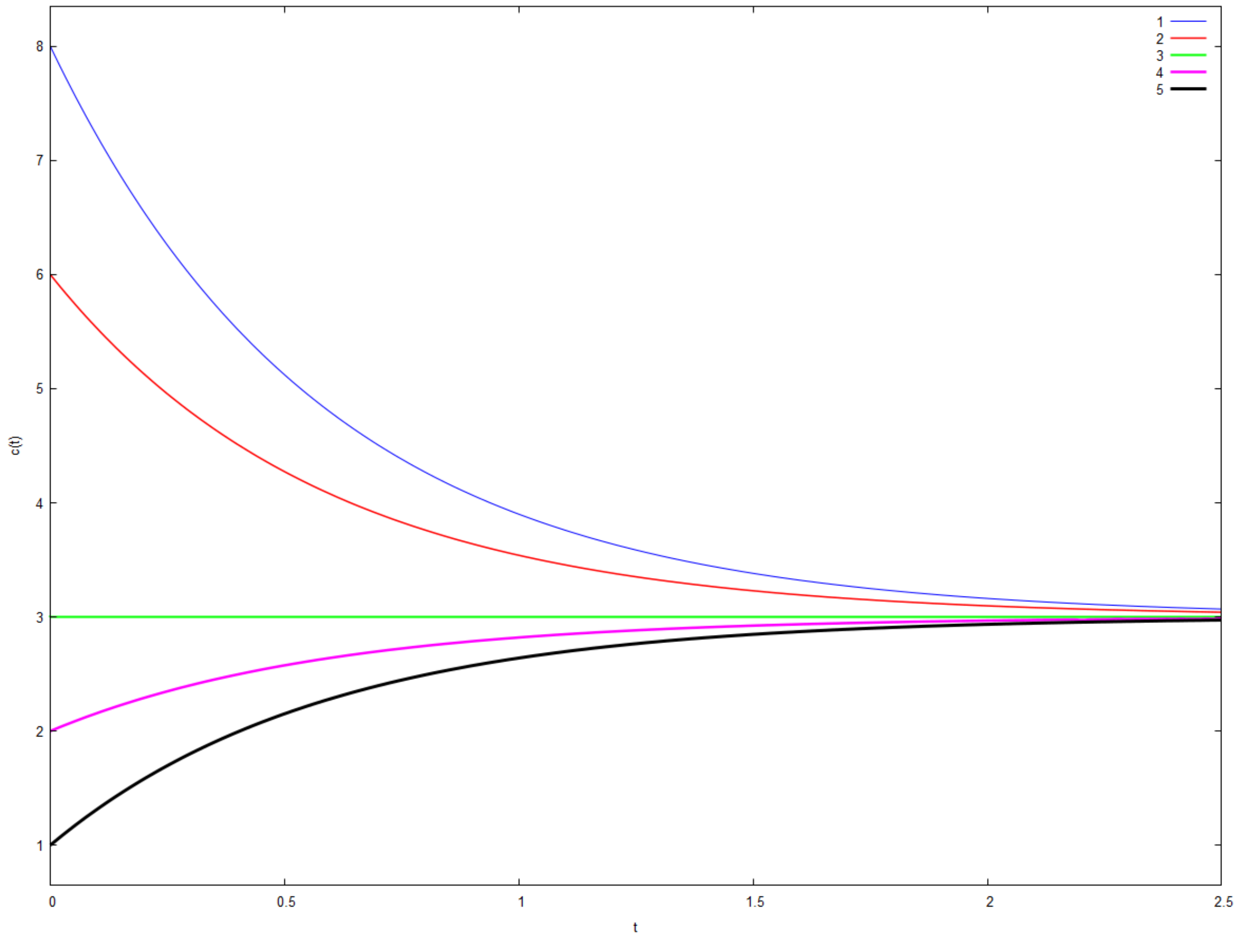
```
--> kill ( all ) $
    eqn1 : ' diff ( c , t ) = ( F / V ) · cin − ( F / V ) · c ;
    sol1 : ode2 ( eqn1 , c , t ) ;
    fsol1 : ic1 ( sol1 , c = c0 , t = 0 ) ;
    v : ev ( fsol1 , cin = 3 , V = 28 , F = 4 · 12 ) ;
    v1 : ev ( v , c0 = 8 ) $
    v2 : ev ( v , c0 = 6 ) $
    v3 : ev ( v , c0 = 3 ) $
    v4 : ev ( v , c0 = 2 ) $
    v5 : ev ( v , c0 = 1 ) $
    wxplot2d ( [ rhs ( v1 ) , rhs ( v2 ) , rhs ( v3 ) , rhs ( v4 ) , rhs ( v5 ) ] ,
        [ t , 0 , 2 . 5 ] ,
        [ legend , "1" , "2" , "3" , "4" , "5" ] ,
        [ style , [ lines , 1 ] , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] , [ lines , 3 ] ] ,
        [ ylabel , "c(t)" ] ) $
```

$$\frac{d}{dt}c = \frac{F\,\mathrm{cin}}{V} - \frac{Fc}{V}$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} + \%c\right)$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} - \mathrm{cin} + \mathrm{c0}\right)$$

$$c = \%e^{-\frac{12t}{7}}\left(3\%e^{\frac{12t}{7}} + \mathrm{c0} - 3\right)$$



2 Seasonal flow and constant pollution concentration inflow

c(t): concentration of pollutant in the lake at time t.
F: seasonal flow rate.
V: constant volume of the lake.
cin: constant concentration of pollutant in the flow entering the lake.
initial condition: c(0)=c0.

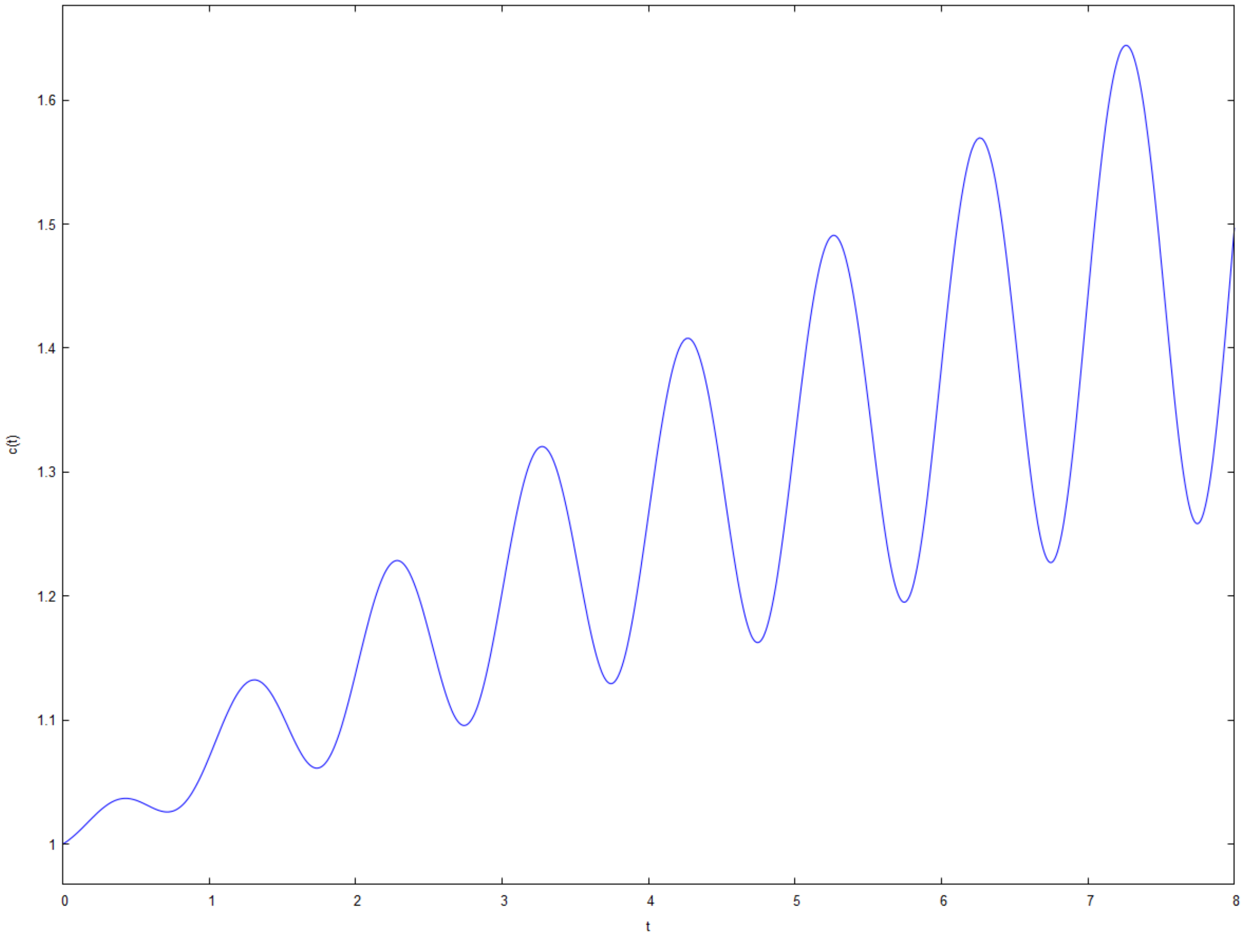
```
--> kill ( all ) $
    eqn1 : ' diff ( c , t ) = ( F / V ) · cin − ( F / V ) · c ;
    sol1 : ode2 ( eqn1 , c , t ) ;
    fsol1 : ic1 ( sol1 , c = c0 , t = 0 ) ;
    v : ev ( fsol1 , cin = 3 , V = 28 , F = 1 + 0 . 5 · sin ( 2 · π · t ) ) ;
    v1 : ev ( v , c0 = 1 ) $
    wxplot2d ( rhs ( v1 ) ,
        [ t , 0 , 8 ] ,
        [ legend , "" ] ,
        [ ylabel , "c(t)" ] ) $
```

$$\frac{d}{dt}c = \frac{F\,\mathrm{cin}}{V} - \frac{Fc}{V}$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} + \%\mathrm{c}\right)$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} - \mathrm{cin} + \mathrm{c0}\right)$$

$$c = \%e^{-\frac{t\,(0.5\sin(2\pi t)+1)}{28}}\left(3\%e^{\frac{t\,(0.5\sin(2\pi t)+1)}{28}} + \mathrm{c0} - 3\right)$$



3 Constant flow and seasonal pollution concentration inflow

c(t): concentration of pollutant in the lake at time t.
F: constant flow rate.
V: constant volume of the lake.
cin: seasonal concentration of pollutant in the flow entering the lake.
initial condition: c(0)=c0.

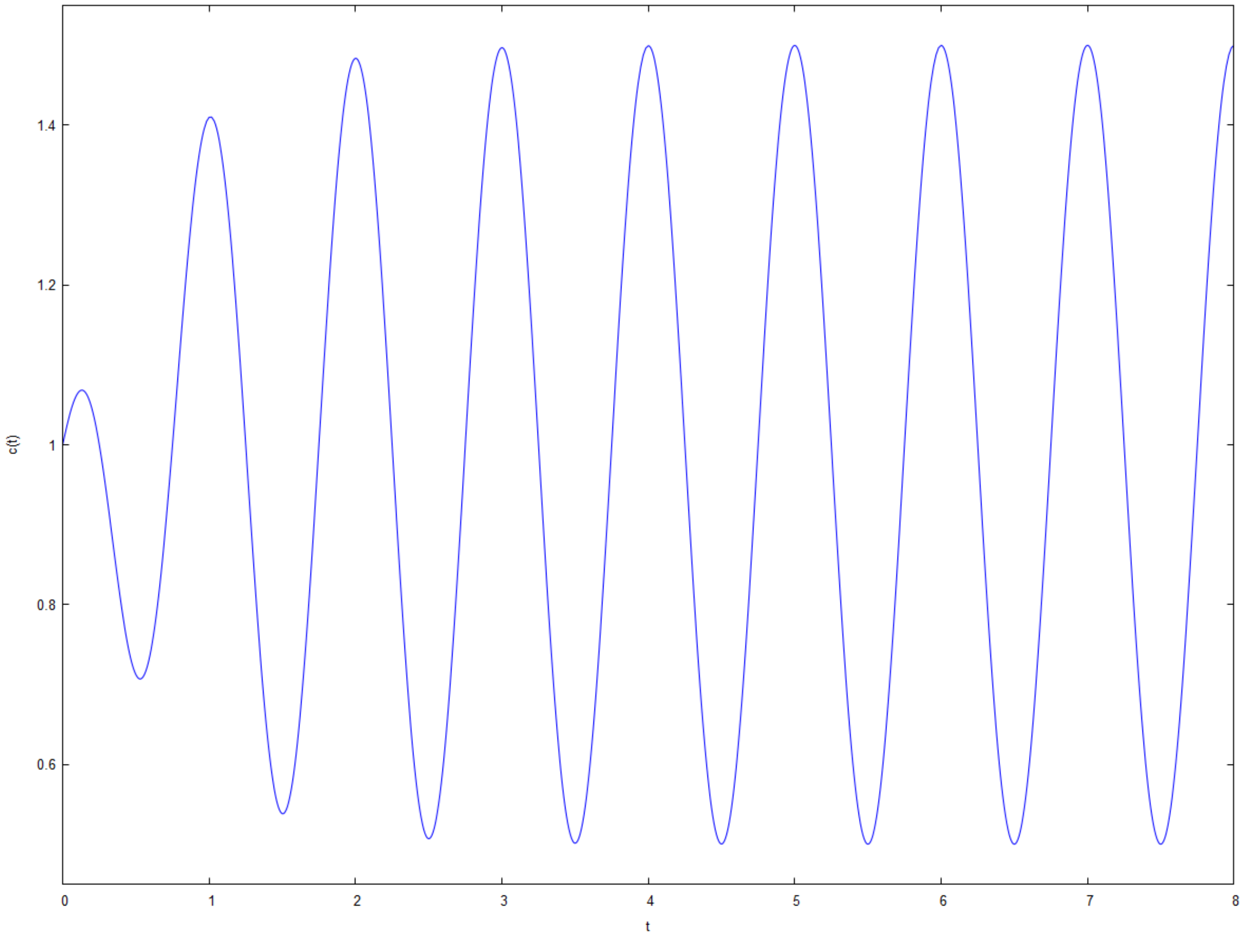
```
--> kill ( all ) $
    eqn1 : ' diff ( c , t ) = ( F / V ) · cin − ( F / V ) · c ;
    sol1 : ode2 ( eqn1 , c , t ) ;
    fsol1 : ic1 ( sol1 , c = c0 , t = 0 ) ;
    v : ev ( fsol1 , cin = 1 + 0 . 5 · cos ( 2 · π · t ) , V = 28 , F = 4 · 12 ) ;
    v1 : ev ( v , c0 = 1 ) $
    wxplot2d ( rhs ( v1 ) ,
          [ t , 0 , 8 ] ,
          [ legend , "" ] ,
          [ ylabel , "c(t)" ] ) $
```

$$\frac{d}{dt}c = \frac{F\,\mathrm{cin}}{V} - \frac{Fc}{V}$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} + \%\mathrm{c}\right)$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} - \mathrm{cin} + \mathrm{c0}\right)$$

$$c = \%e^{-\frac{12t}{7}}\left(-0.5\cos(2\pi t) + \%e^{\frac{12t}{7}}(0.5\cos(2\pi t) + 1) + \mathrm{c0} - 1\right)$$



4 Seasonal flow and seasonal pollution concentration inflow

c(t): concentration of pollutant in the lake at time t.
F: seasonal flow rate.
V: constant volume of the lake.
cin: seasonal concentration of pollutant in the flow entering the lake.
initial condition: c(0)=c0.

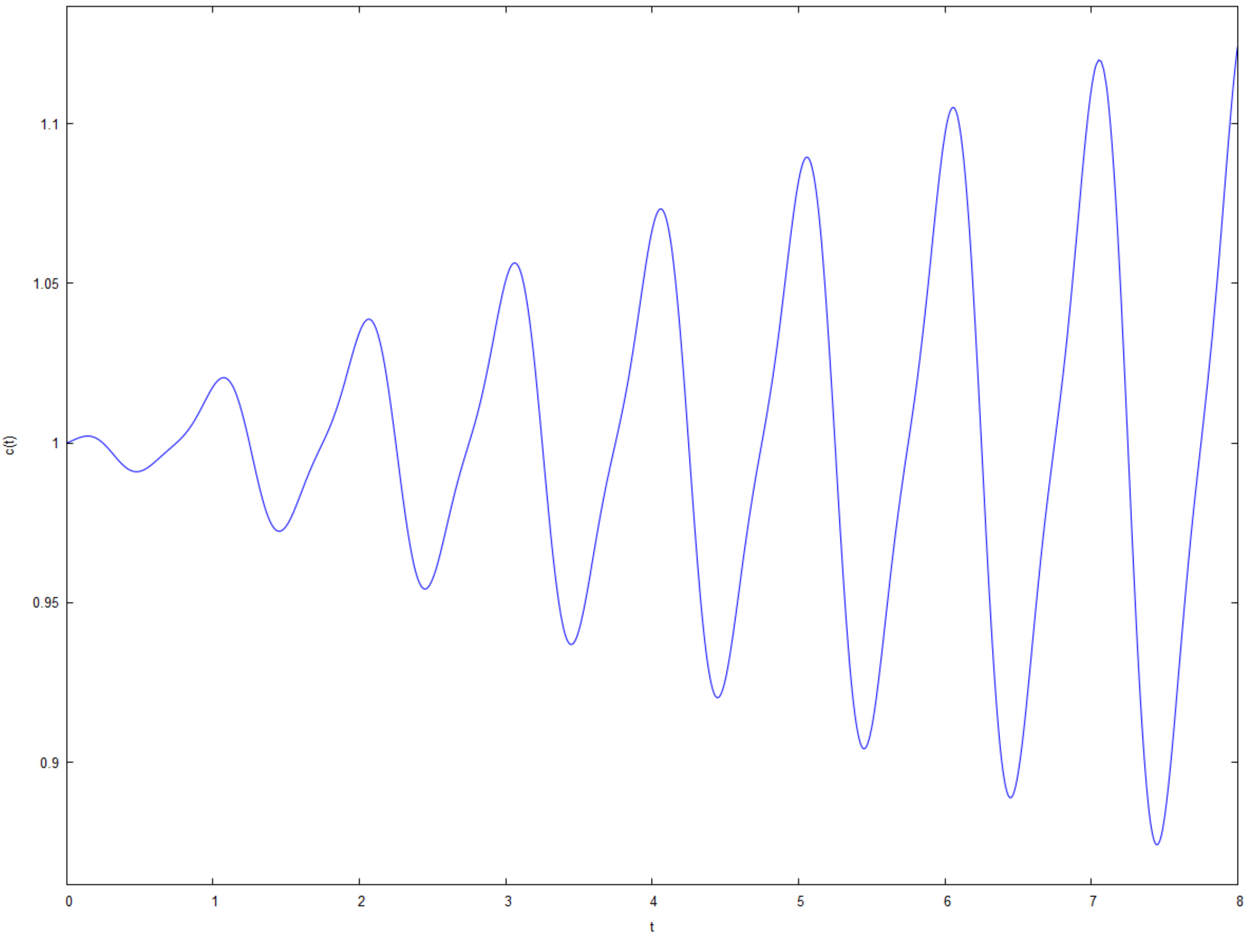
```
--> kill ( all ) $
    eqn1 : ' diff ( c , t ) = ( F / V ) · cin − ( F / V ) · c ;
    sol1 : ode2 ( eqn1 , c , t ) ;
    fsol1 : ic1 ( sol1 , c = c0 , t = 0 ) ;
    v : ev ( fsol1 , cin = 1 + 0 . 5 · cos ( 2 · π · t ) , V = 28 , F = 1 + 0 . 5 · sin ( 2 · π · t ) ) ;
    v1 : ev ( v , c0 = 1 ) $
    wxplot2d ( rhs ( v1 ) ,
        [ t , 0 , 8 ] ,
        [ legend , "" ] ,
        [ ylabel , "c(t)" ] ) $
```

$$\frac{d}{dt}c = \frac{F\,\mathrm{cin}}{V} - \frac{Fc}{V}$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} + \%c\right)$$

$$c = \%e^{-\frac{Ft}{V}}\left(\mathrm{cin}\%e^{\frac{Ft}{V}} - \mathrm{cin} + \mathrm{c0}\right)$$

$$c = \%e^{-\frac{t\,(0.5\sin(2\pi t)+1)}{28}}\left(-0.5\cos(2\pi t) + \%e^{\frac{t\,(0.5\sin(2\pi t)+1)}{28}}(0.5\cos(2\pi t)+1) + \mathrm{c0} - 1\right)$$



Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-07

Logistic Growth Model

x(t): population at time t.
a: per capita death rate.
b: per capita birth rate.
r: b-a is the reproduction rate.
K: carrying capacity.
initial condition: x(0)=x0.

```
--> r : 1 $ K : 1000 $
    eqn : ' diff ( x , t ) = r · x · ( 1 − x / K ) ;
    gs : ode2 ( eqn , x , t ) ;
    gs1 : logcontract ( gs ) ;
    gs2 : solve ( gs1 , x ) [ 1 ] ;
    ps : ic1 ( gs2 , t = 0 , x = x0 ) ;
    ps1 : ev ( ps , x0 = 10 ) $
    ps2 : ev ( ps , x0 = 200 ) $
    ps3 : ev ( ps , x0 = 500 ) $
    ps4 : ev ( ps , x0 = 800 ) $
    ps5 : ev ( ps , x0 = 1400 ) $
    wxplot2d ( [ rhs ( ps1 ) , rhs ( ps2 ) , rhs ( ps3 ) , rhs ( ps4 ) , rhs ( ps5 ) , K ] ,
        [ t , 0 , 10 ] ,
        [ legend , "x0=10" , "x0=200" , "x0=500" , "x0=800" , "x0=1400" , "K (=1000)" ] ,
        [ style , [ lines , 1 . 5 ] , [ lines , 2 ] , [ lines , 2 . 5 ] , [ lines , 3 ] , [ lines , 3 . 5 ] , [ lines , 1 ] ] ,
        [ xlabel , "t (in years)" ] , [ ylabel , "Population" ] ) $
```

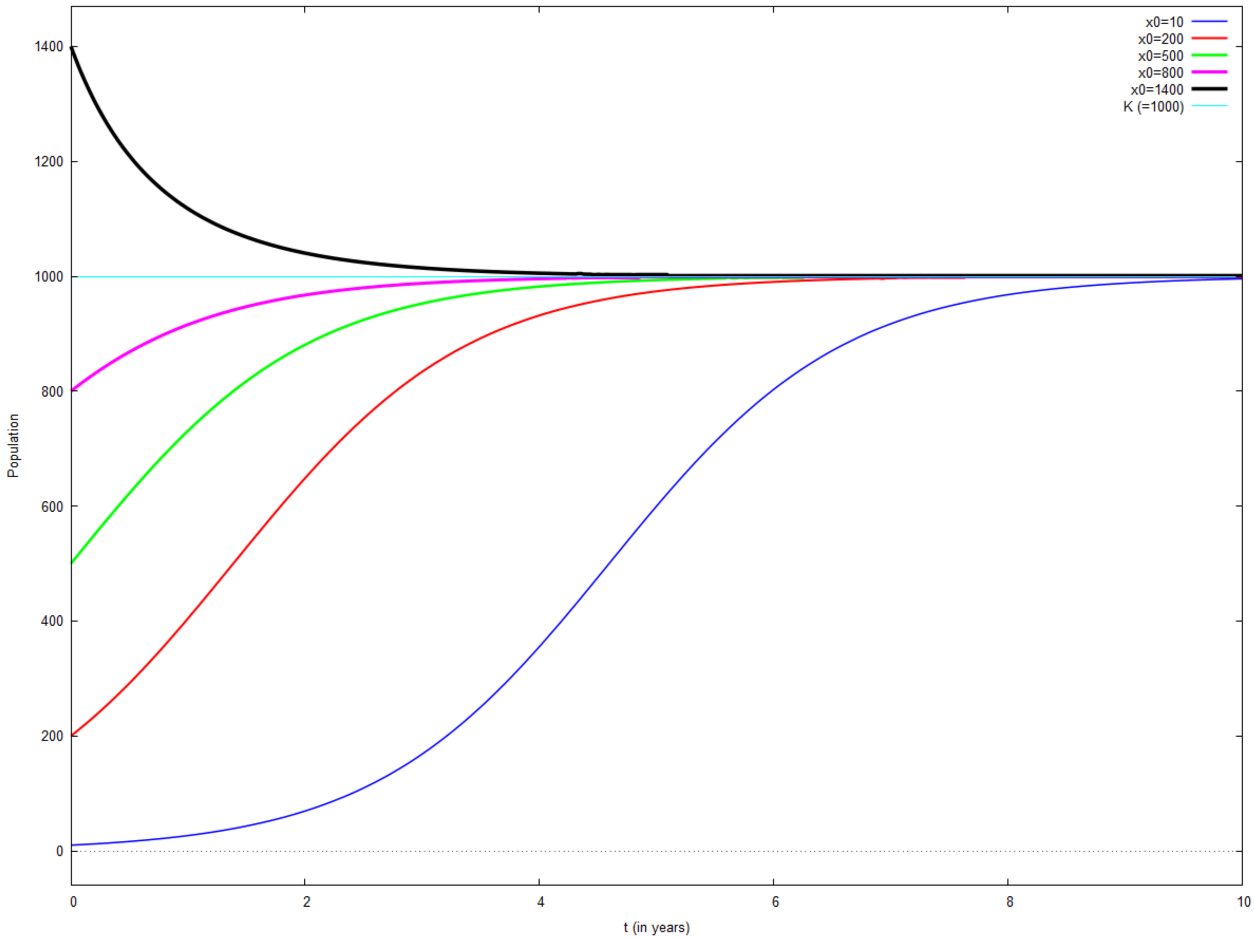
$$\frac{d}{dt}x = \left(1 - \frac{x}{1000}\right)x$$

$$\log(x) - \log(x - 1000) = t + \%c$$

$$\log\left(\frac{x}{x - 1000}\right) = t + \%c$$

$$x = \frac{1000\%e^{t+\%c}}{\%e^{t+\%c} - 1}$$

$$x = \frac{1000\%e^{t}\mathrm{x0}}{\left(\%e^{t} - 1\right)\mathrm{x0} + 1000}$$



Created with wxMaxima.

The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-08

Demonstration of the Runge-Kutta Method

1 Solve dy/dx = x, where at x=0, y=0

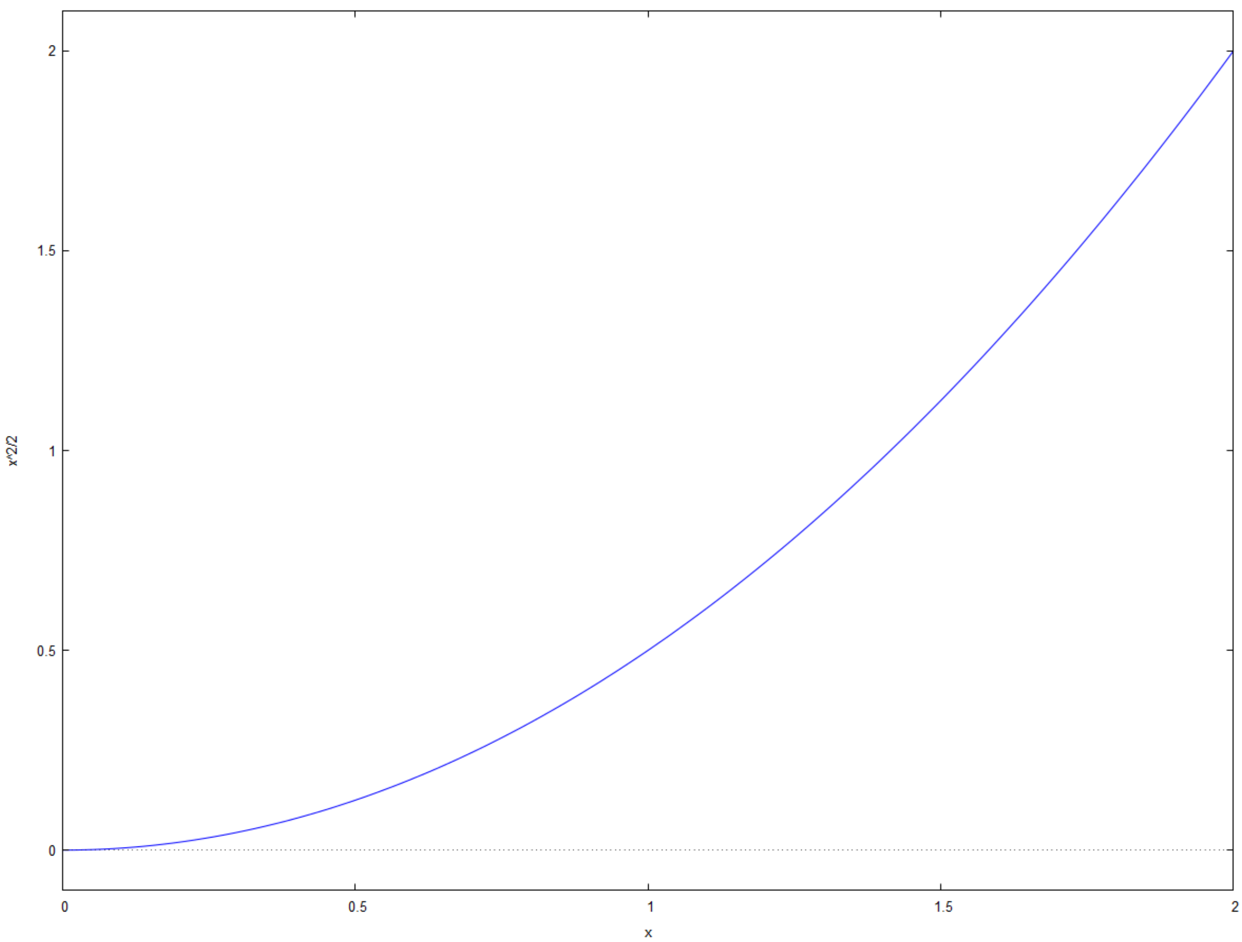
1.1 We solve the above differential equation using ode2():

```
--> eqn : ' diff ( y , x ) = x ;
    gs : ode2 ( eqn , y , x ) ;
    ps : ic1 ( gs , y = 0 , x = 0 ) ;
    wxplot2d ( rhs ( ps ) , [ x , 0 , 2 ] ) $
```

$$\frac{d}{dx} y = x$$

$$y = \frac{x^2}{2} + \%c$$

$$y = \frac{x^2}{2}$$



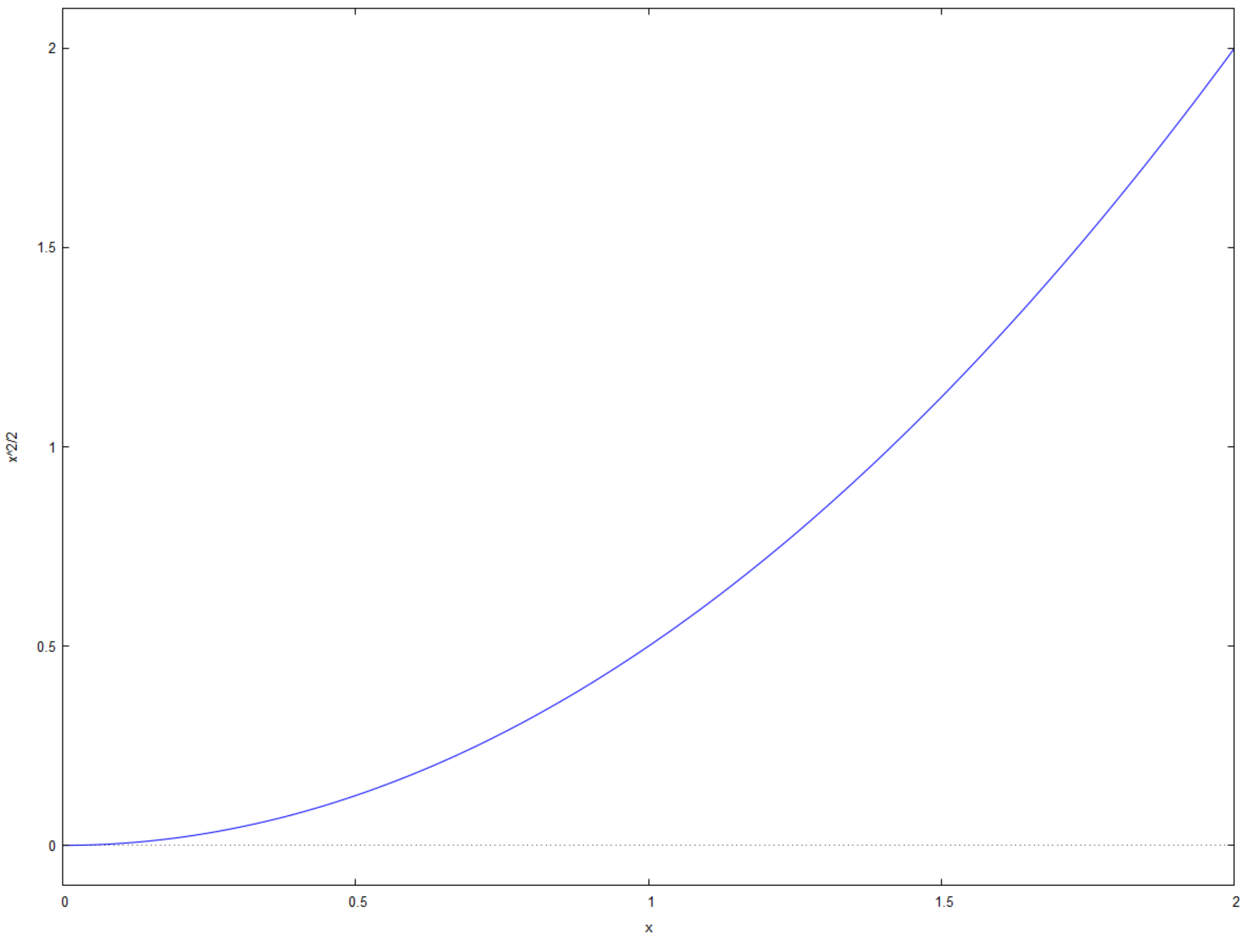
1.2 Now, we use desolve() to solve the same equation

```
--> kill ( all ) $
    eqn : diff ( y ( x ) , x ) = x ;
    gs : desolve ( eqn , y ( x ) ) ;
    ps : ev ( gs , y ( 0 ) = 0 ) ;
    wxplot2d ( [ rhs ( ps ) ] , [ x , 0 , 2 ] ) $
```

$$\frac{d}{dx}\mathrm{y}(x) = x$$

$$\mathrm{y}(x) = \frac{x^2}{2} + \mathrm{y}(0)$$

$$\mathrm{y}(x) = \frac{x^2}{2}$$

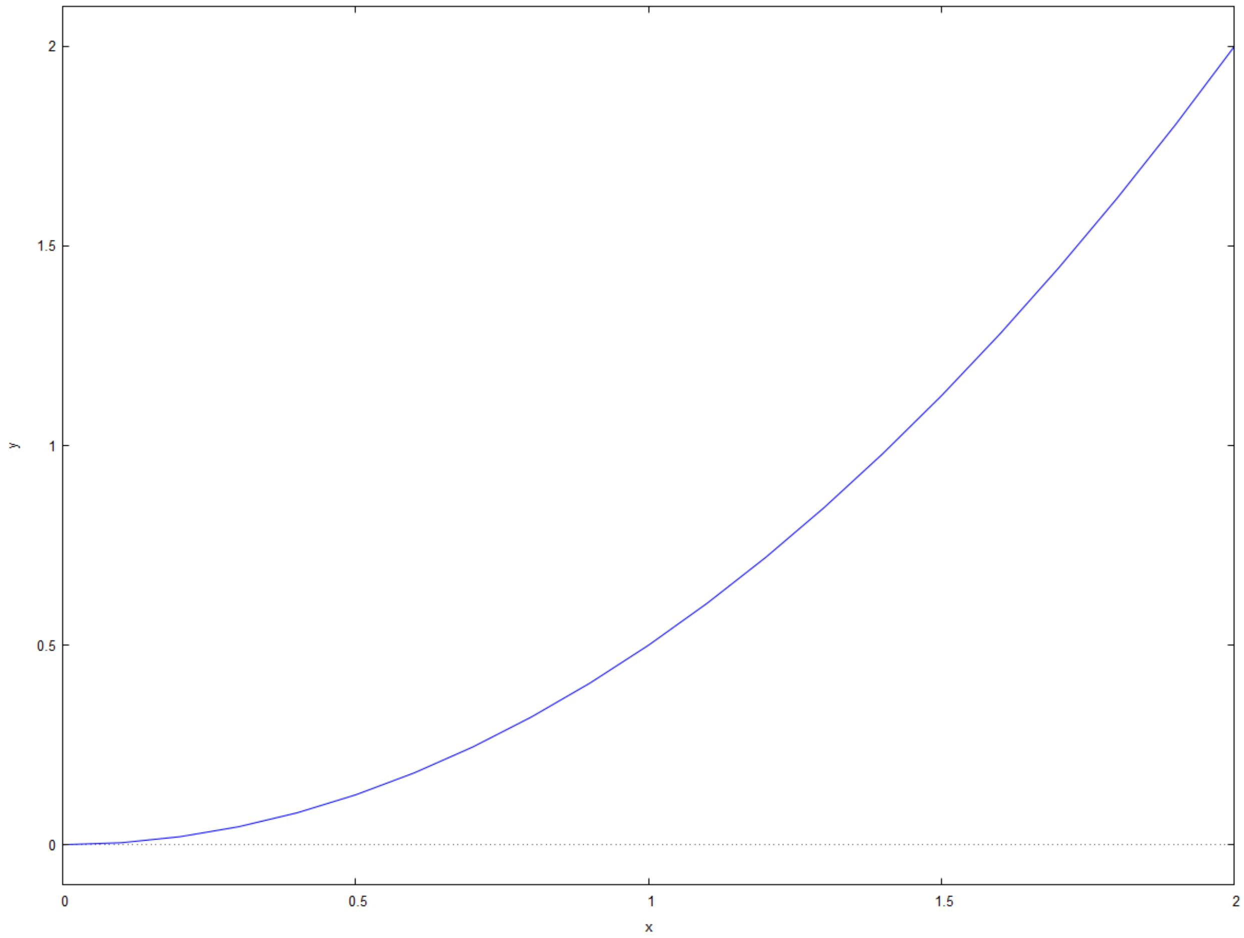


1.3 Finally, using the Runge-Kutta Method

Syntax:
rk([rhs(ODE_1),...,rhs(ODE_n)], [v1,...,vn], [init_1,...,init_n], domain)

```
--> kill ( all ) $
    eqn : ' diff ( y , x ) = x ;
    fsol : rk ( rhs ( eqn ) , [ y ] , [ 0 ] , [ x , 0 , 2 , 0 . 1 ] ) ;
    wxplot2d ( [ discrete , fsol ] ) $
```

$$\frac{d}{dx}y = x$$



2 We now have a coupled system of differential equations

dx/dt = 4 - x^2 - 4*y^2
dy/dt = y^2 - x^2 + 1

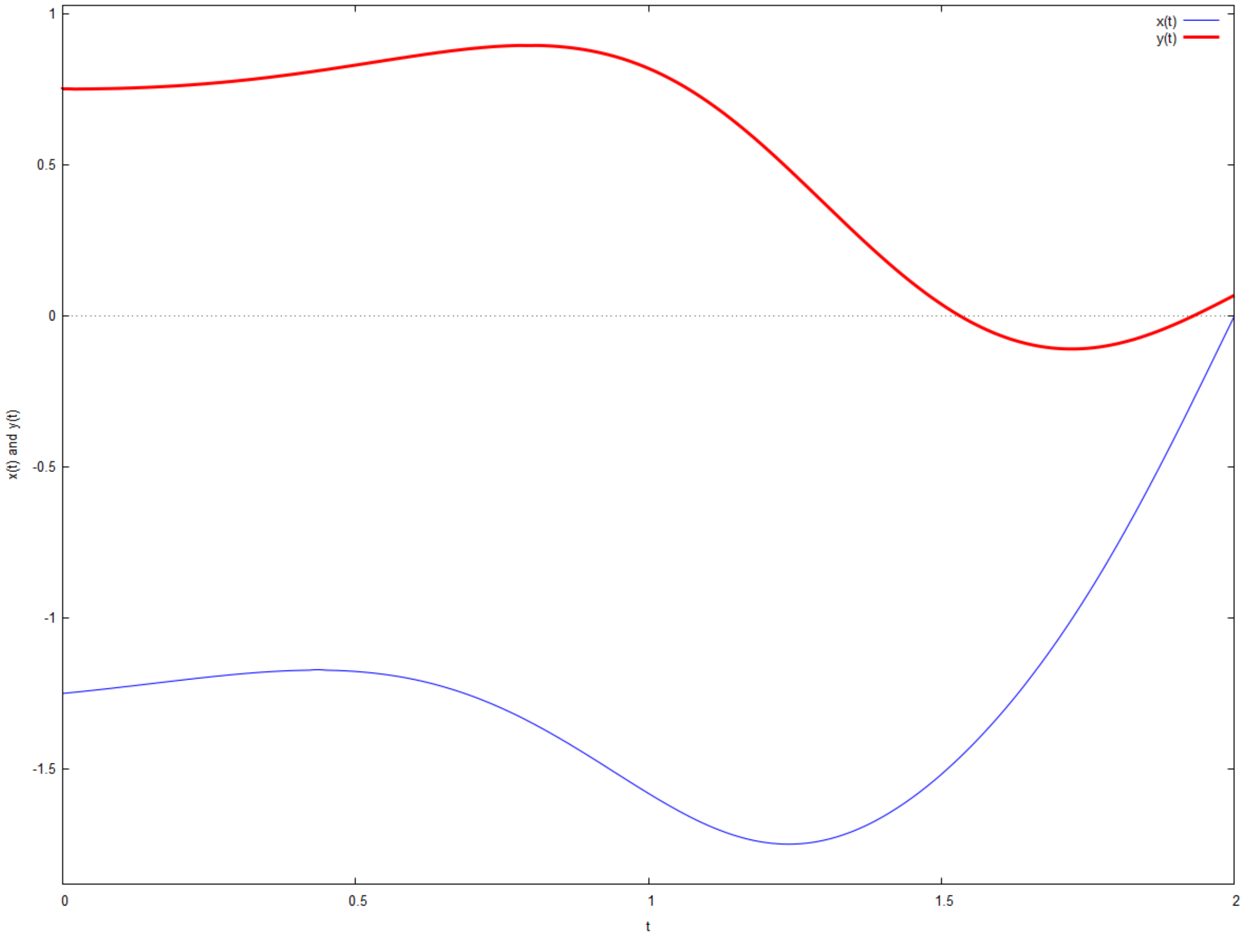
x(0) = -1.25, y(0) = 0.75

```
--> kill ( all ) $
    eqn1 : ' diff ( x , t ) = 4 − x ^ 2 − 4 · y ^ 2 ;
    eqn2 : ' diff ( y , t ) = y ^ 2 − x ^ 2 + 1 ;
    pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) ] , [ x , y ] , [ − 1 . 25 , 0 . 75 ] , [ t , 0 , 2 , 0 . 01 ] ) $
```

$$\frac{d}{dt}x = -4y^2 - x^2 + 4$$

$$\frac{d}{dt}y = y^2 - x^2 + 1$$

```
--> curve_x : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
    /*wxplot2d([discrete, curve_x])$*/
    curve_y : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
    wxplot2d ( [ [ discrete , curve_x ] , [ discrete , curve_y ] ] ,
        [ legend , "x(t)" , "y(t)" ] ,
        [ style , [ lines , 1 ] , [ lines , 3 ] ] ,
        [ xlabel , "t" ] ,
        [ ylabel , "x(t) and y(t)" ] ) $
```



Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-09

Predator-Prey Model

1 Basic Lotka-Volterra Model

x(t): Number of prey per unit area.
y(t): Number of predators per unit area.
Initial condition: x(0)=200, y(0)=80.
The constant b1,c1,c2,a2 are all positive.

```
--> b1 : 1 . 0 $ a2 : 0 . 5 $ c1 : 0 . 01 $ c2 : 0 . 005 $
    eqn1 : ' diff ( x , t ) = b1 · x − c1 · x · y ;
    eqn2 : ' diff ( y , t ) = c2 · x · y − a2 · y ;
    pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) ] , [ x , y ] , [ 200 , 80 ] , [ t , 0 , 20 , . 1 ] ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    x_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    y_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    wxplot2d ( [ [ discrete , x_pts ] , [ discrete , y_pts ] ] ,
        [ t , 0 , 20 ] , [ y , 0 , 250 ] ,
        [ style , [ lines , 2 ] , [ lines , 4 ] ] ,
        [ xlabel , "t (in years)" ] ,
        [ ylabel , "Population density" ] ,
        [ legend , "x(t): Preys" , "y(t): Predators" ] ) $
```

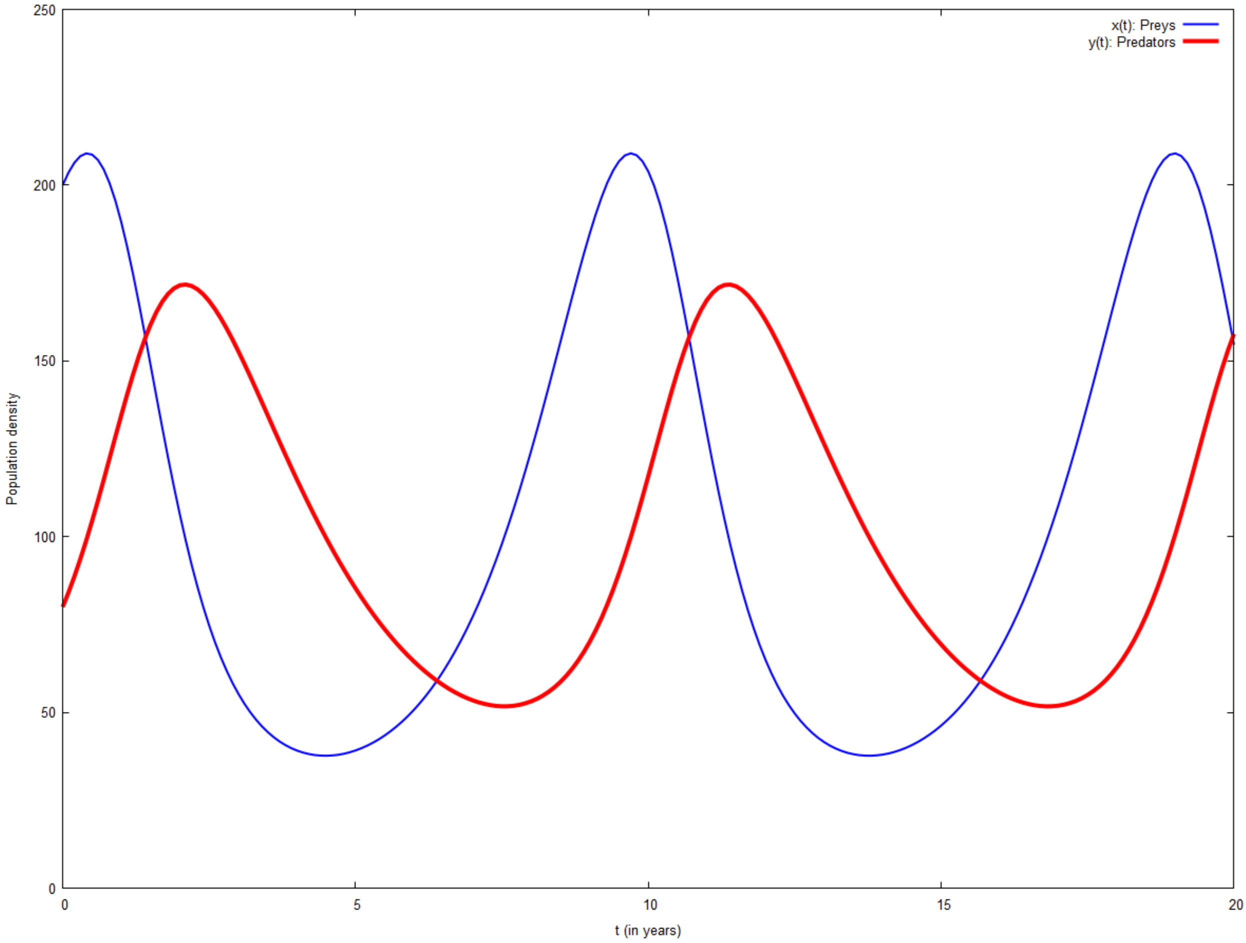
$$\frac{d}{dt}x = 1.0x - 0.01xy$$

$$\frac{d}{dt}y = 0.005xy - 0.5y$$

$$[[0.0 , 200.0 , 80.0] , [20.0 , 154.5921330991444 , 157.6845208218405] , 201]$$

$$[[0.0 , 200.0] , [20.0 , 154.5921330991444] , 201]$$

$$[[0.0 , 80.0] , [20.0 , 157.6845208218405] , 201]$$



2 Density-Dependent Growth

x(t): Number of preys per unit area.
y(t): Number of predators per unit area.
Initial conditions: x(0)=120, y(0)=40.
The constants b1,c1,c2,a2,K are all positive.

```
--> kill ( all ) $
    b1 : 1 . 0 $ a2 : 0 . 5 $ c1 : 0 . 01 $ c2 : 0 . 005 $ K : 1000 $
    eqn1 : ' diff ( x , t ) = b1 · x · ( 1 − x / K ) − c1 · x · y ;
    eqn2 : ' diff ( y , t ) = c2 · x · y − a2 · y ;
    pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) ] , [ x , y ] , [ 120 , 40 ] , [ t , 0 , 40 , . 1 ] ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    x_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    y_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    wxplot2d ( [ [ discrete , x_pts ] , [ discrete , y_pts ] ] ,
        [ t , 0 , 40 ] , [ y , 0 , 250 ] ,
        [ style , [ lines , 2 ] , [ lines , 4 ] ] ,
        [ xlabel , "t (in years)" ] ,
        [ ylabel , "Population density" ] ,
        [ legend , "x(t): Preys" , "y(t): Predators" ] ) $
```

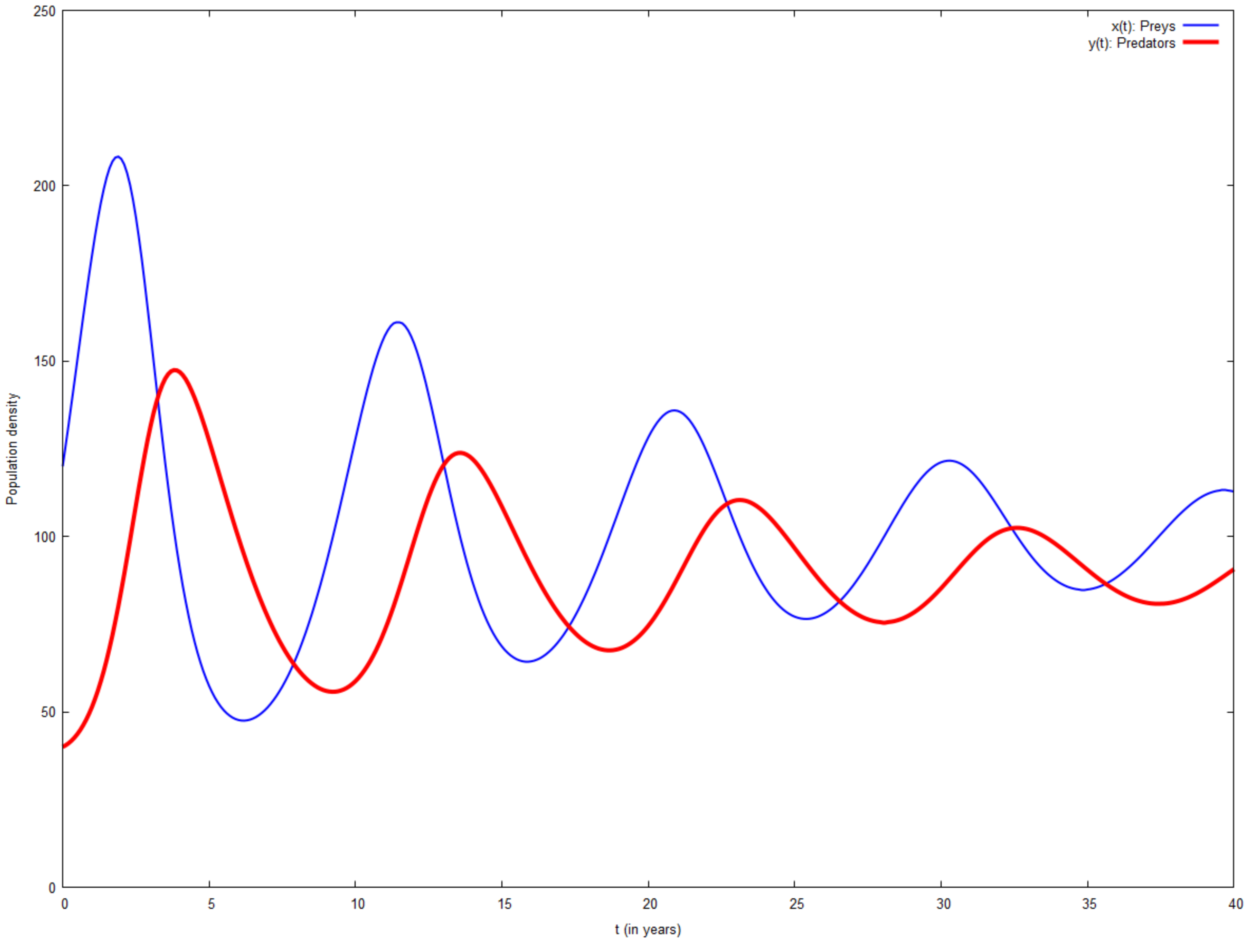
$$\frac{d}{dt}x = 1.0\left(1-\frac{x}{1000}\right)x - 0.01xy$$

$$\frac{d}{dt}y = 0.005xy - 0.5y$$

$$[[0.0\,,120.0\,,40.0]\,,[40.0\,,112.8076936975965\,,90.67006550194145]\,,401]$$

$$[[0.0\,,120.0]\,,[40.0\,,112.8076936975965]\,,401]$$

$$[[0.0\,,40.0]\,,[40.0\,,90.67006550194145]\,,401]$$



3 Effect of DDT

x(t): Number of preys per unit area.
y(t): Number of predators per unit area.
Initial conditions: x(0)=200, y(0)=80.
The constants b1,c1,c2,a2,p1,p2 are all positive.

```
--> kill ( all ) $
    b1 : 1 . 0 $ a2 : 0 . 5 $ c1 : 0 . 01 $ c2 : 0 . 005 $ p1 : 0 . 1 $ p2 : 0 . 1 $
    eqn1 : ' diff ( x , t ) = b1 · x − c1 · x · y − p1 · x ;
    eqn2 : ' diff ( y , t ) = c2 · x · y − a2 · y − p2 · y ;
    pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) ] , [ x , y ] , [ 200 , 80 ] , [ t , 0 , 20 , . 1 ] ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    x_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    y_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    wxplot2d ( [ [ discrete , x_pts ] , [ discrete , y_pts ] ] ,
        [ t , 0 , 20 ] , [ y , 0 , 250 ] ,
        [ style , [ lines , 2 ] , [ lines , 4 ] ] ,
        [ xlabel , "t (in years)" ] ,
        [ ylabel , "Population density" ] ,
        [ legend , "x(t): Preys" , "y(t): Predators" ] ) $
```

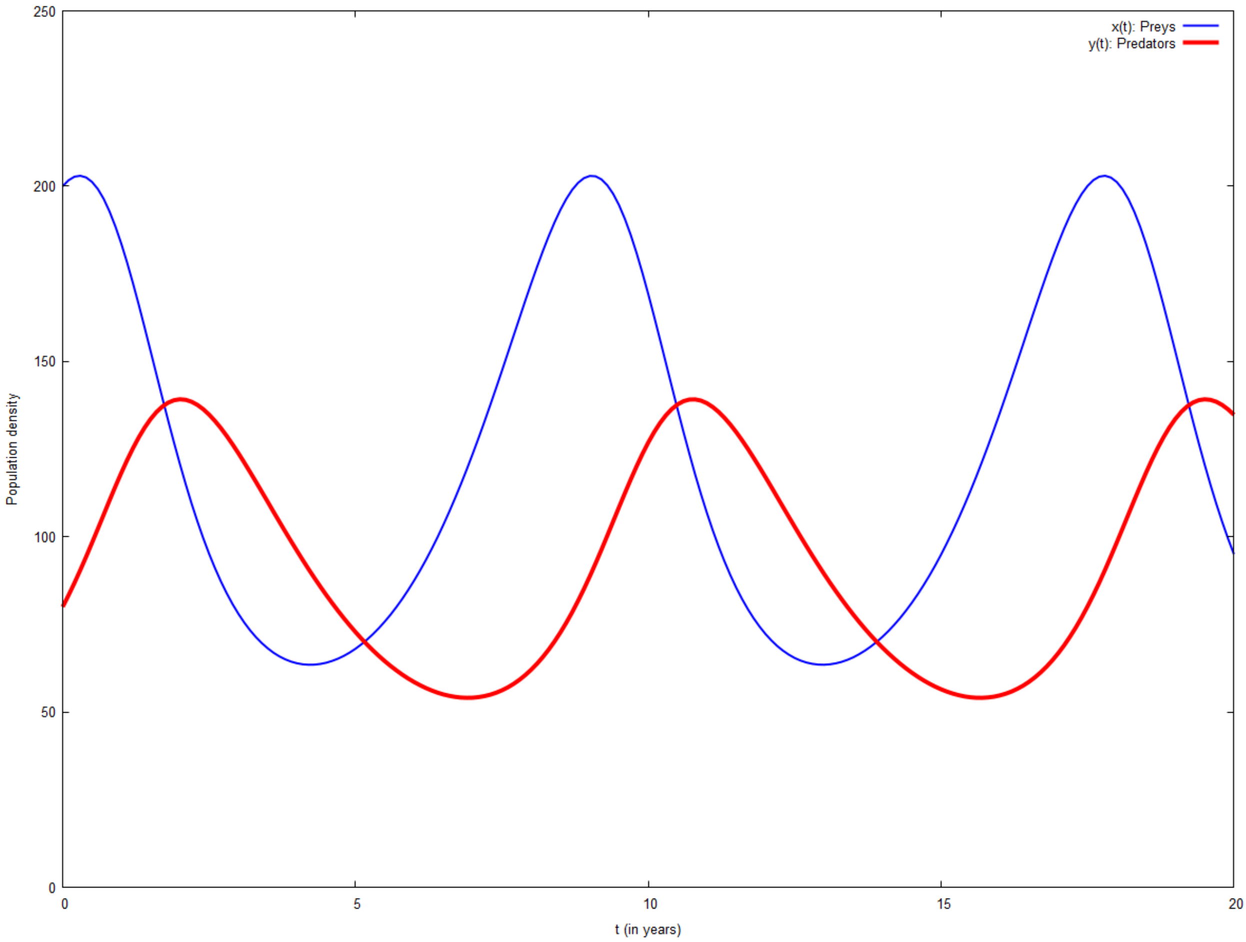
$$\frac{d}{dt}x = 0.9x - 0.01xy$$

$$\frac{d}{dt}y = 0.005xy - 0.6y$$

$$[[0.0\,,200.0\,,80.0]\,,[20.0\,,94.9961084048835\,,134.7757919917374]\,,201]$$

$$[[0.0\,,200.0]\,,[20.0\,,94.9961084048835]\,,201]$$

$$[[0.0\,,80.0]\,,[20.0\,,134.7757919917374]\,,201]$$



4 Two Prey and One Predator

x1(t): Number of prey1 per unit area.
x2(t): Number of prey2 per unit area.
y(t): Number of predators per unit area.
Initial condition: x1(0)=150, x2(0)=130, y(0)=80.
The constant b1,b2,c1,c2,c3,c4,a are all positive.

```
-- kill ( all ) $
>  b1 : 1 . 32 $ b2 : 1 . 3 $ a : 0 . 5 $ c1 : 0 . 01 $ c2 : 0 . 01 $ c3 : 0 . 003 $ c4 : 0 . 004 $
   eqn1 : ' diff ( x1 , t ) = b1 · x1 − c1 · x1 · y ;
   eqn2 : ' diff ( x2 , t ) = b2 · x2 − c2 · x2 · y ;
   eqn3 : ' diff ( y , t ) = c3 · x1 · y + c4 · x2 · y − a · y ;
   pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) , rhs ( eqn3 ) ] , [ x1 , x2 , y ] , [ 150 , 130 , 80 ] , [ t , 0 , 40 , . 1 ] )
   $
   [ % [ 1 ] , last ( % ) , length ( % ) ] ;
   x1_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
   [ % [ 1 ] , last ( % ) , length ( % ) ] ;
   x2_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
   [ % [ 1 ] , last ( % ) , length ( % ) ] ;
   y_pts : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 4 ] ] , i , 1 , length ( pts ) ) $
```

```
[ % [ 1 ] , last ( % ) , length ( % ) ] ;
wxplot2d ( [ [ discrete , x1_pts ] , [ discrete , x2_pts ] , [ discrete , y_pts ] ] ,
      [ t , 0 , 40 ] , [ y , 0 , 300 ] ,
      [ style , [ lines , 2 ] , [ lines , 3 ] , [ lines , 4 ] ] ,
      [ xlabel , "t (in years)" ] ,
      [ ylabel , "Population density" ] ,
      [ legend , "x1(t): Prey1" , "x2(t): Prey2" , "y(t): Predator" ] ) $
```

$$\frac{d}{dt}\mathrm{x1} = 1.32\mathrm{x1} - 0.01\mathrm{x1}y$$

$$\frac{d}{dt}\mathrm{x2} = 1.3\mathrm{x2} - 0.01\mathrm{x2}y$$

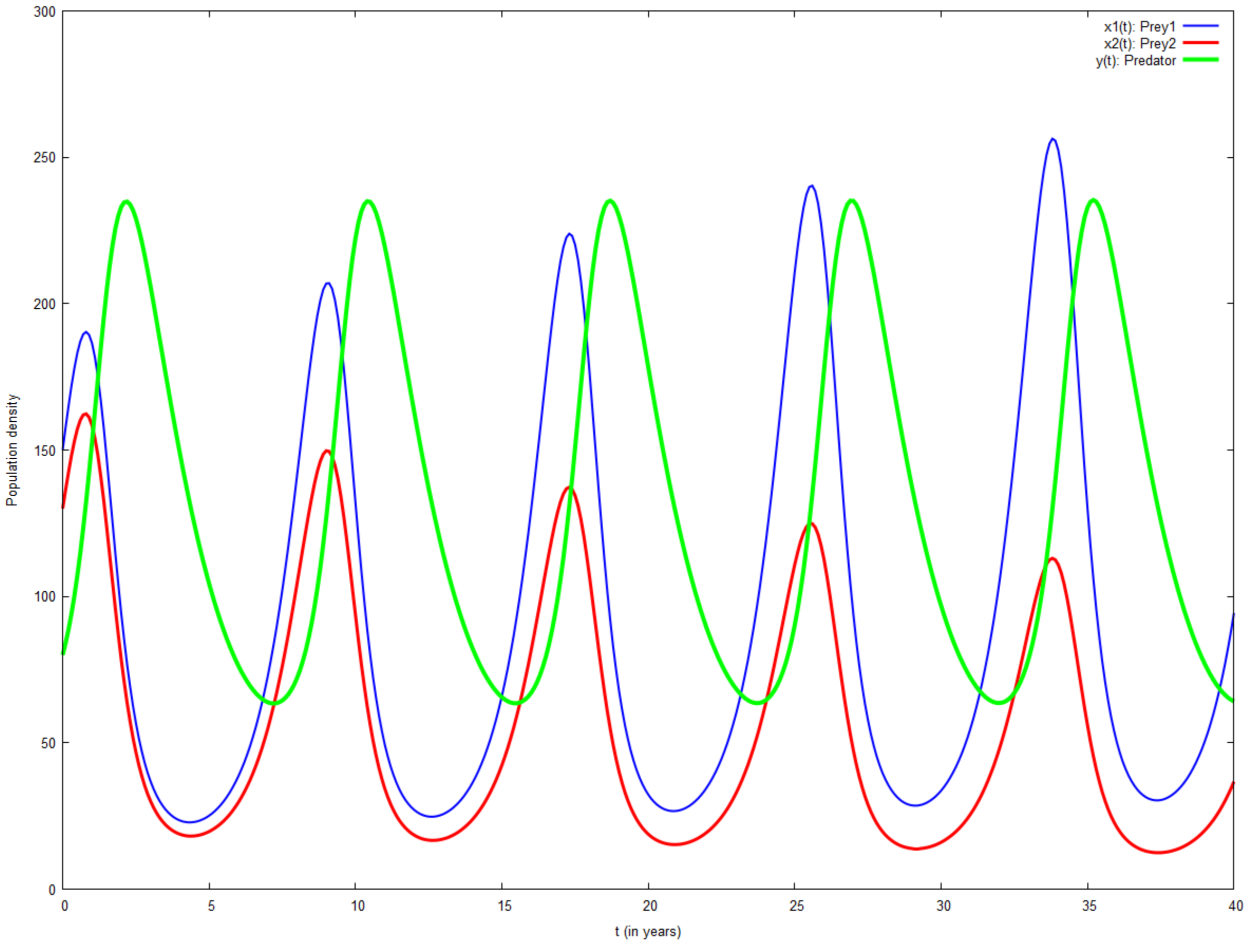
$$\frac{d}{dt}y = 0.004\mathrm{x2}y + 0.003\mathrm{x1}y - 0.5y$$

$$[[0.0, 150.0, 130.0, 80.0], [40.0, 94.32633644255458, 36.73244132377353, 64.13219452304699], 401]$$

$$[[0.0, 150.0], [40.0, 94.32633644255458], 401]$$

$$[[0.0, 130.0], [40.0, 36.73244132377353], 401]$$

$$[[0.0, 80.0], [40.0, 64.13219452304699], 401]$$



Created with wxMaxima.
The source of this Maxima session can be downloaded here.

Lucky Lao

DSC-VI : Practical-10

Epidemic Model for Influenza

1 Basic Epidemic Model

S(t): susceptibles at time t
I(t): infectives at time t
Initial condition: S(0)=762, I(0)=1.
The constants b,c are all positive

```
--> b : 2 . 18 · 10 ^ − 3 $ c : 0 . 44 $
    eqn1 : ' diff ( S , t ) = − b · S · I ;
    eqn2 : ' diff ( I , t ) = b · S · I − c · I ;
    pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) ] , [ S , I ] , [ 762 , 1 ] , [ t , 0 , 16 , 0 . 1 ] ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    susc : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    infec : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    wxplot2d ( [ [ discrete , susc ] , [ discrete , infec ] ] ,
        [ t , 0 , 16 ] , [ y , 0 , 800 ] ,
        [ style , [ lines , 2 ] , [ lines , 4 ] ] ,
        [ xlabel , "t (in days)" ] ,
        [ ylabel , "Susceptibles and Infectives Population" ] ,
        [ legend , "S(t): Susceptibles" , "I(t): Infectives" ] ) $
```

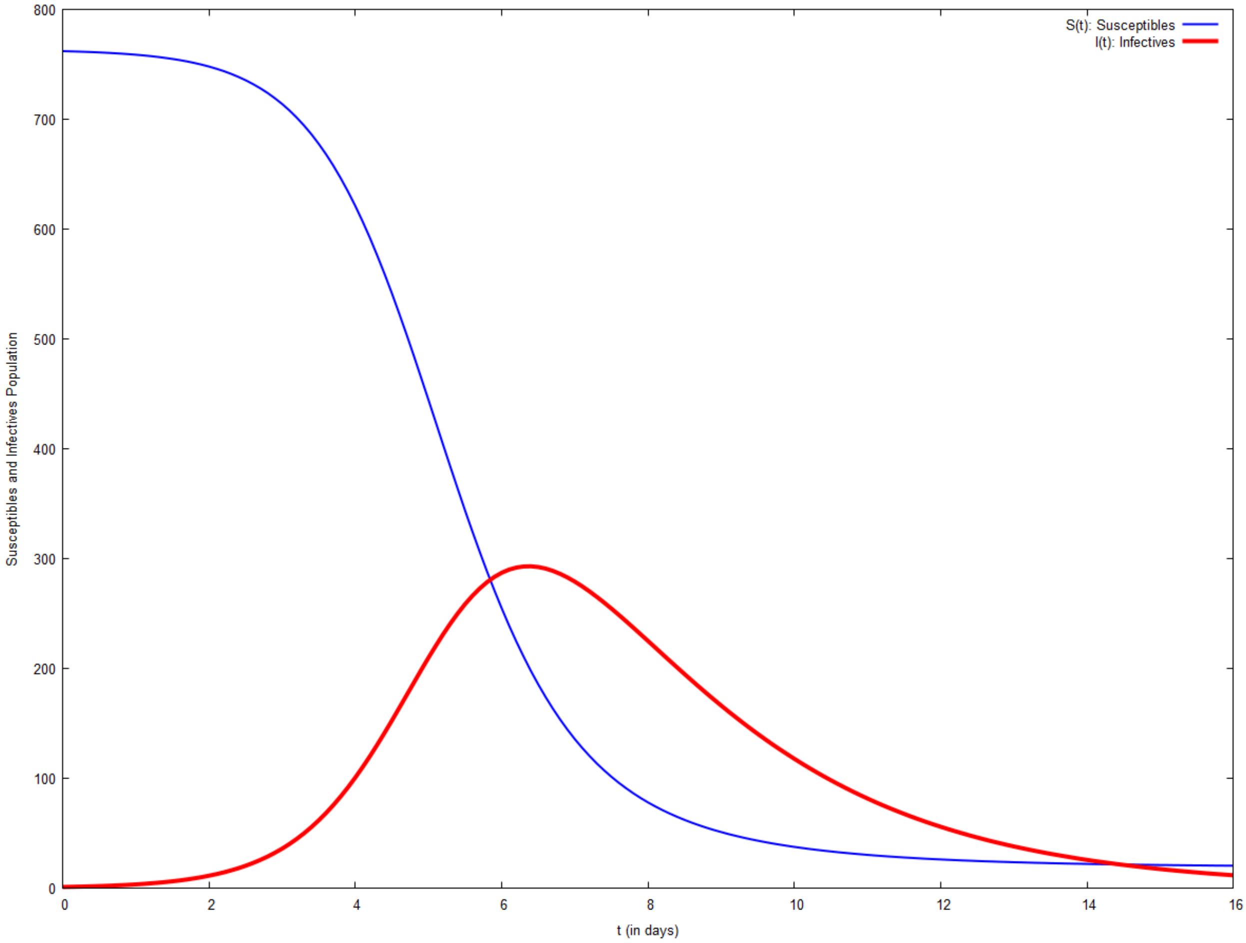
$$\frac{d}{dt}S = -0.00218 IS$$

$$\frac{d}{dt}I = 0.00218 IS - 0.44 I$$

$$[[0.0\,,762.0\,,1.0]\,,[16.0\,,20.37747256075321\,,11.67419228363819]\,,161]$$

$$[[0.0\,,762.0]\,,[16.0\,,20.37747256075321]\,,161]$$

$$[[0.0\,,1.0]\,,[16.0\,,11.67419228363819]\,,161]$$



2 Contagious for Life

S(t): susceptibles at time t
I(t): infectives at time t
Initial condition: S(0)=762, I(t)=1.
The constant b is positive.

```
--> b : 2 . 18 · 10 ^ − 3 $
    eqn1 : ' diff ( S , t ) = − b · S · I ;
    eqn2 : ' diff ( I , t ) = b · S · I ;
    pts : rk ( [ rhs ( eqn1 ) , rhs ( eqn2 ) ] , [ S , I ] , [ 762 , 1 ] , [ t , 0 , 16 , 0 . 1 ] ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    susc : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 2 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    infec : makelist ( [ pts [ i ] [ 1 ] , pts [ i ] [ 3 ] ] , i , 1 , length ( pts ) ) $
    [ % [ 1 ] , last ( % ) , length ( % ) ] ;
    wxplot2d ( [ [ discrete , susc ] , [ discrete , infec ] ] ,
        [ t , 0 , 16 ] , [ y , 0 , 800 ] ,
        [ style , [ lines , 2 ] , [ lines , 4 ] ] ,
        [ xlabel , "t (in days)" ] , [ ylabel , "Susceptibles and Infectives Population" ] ,
        [ legend , "S(t): Susceptibles" , "I(t): Infectives" ] ) $
```

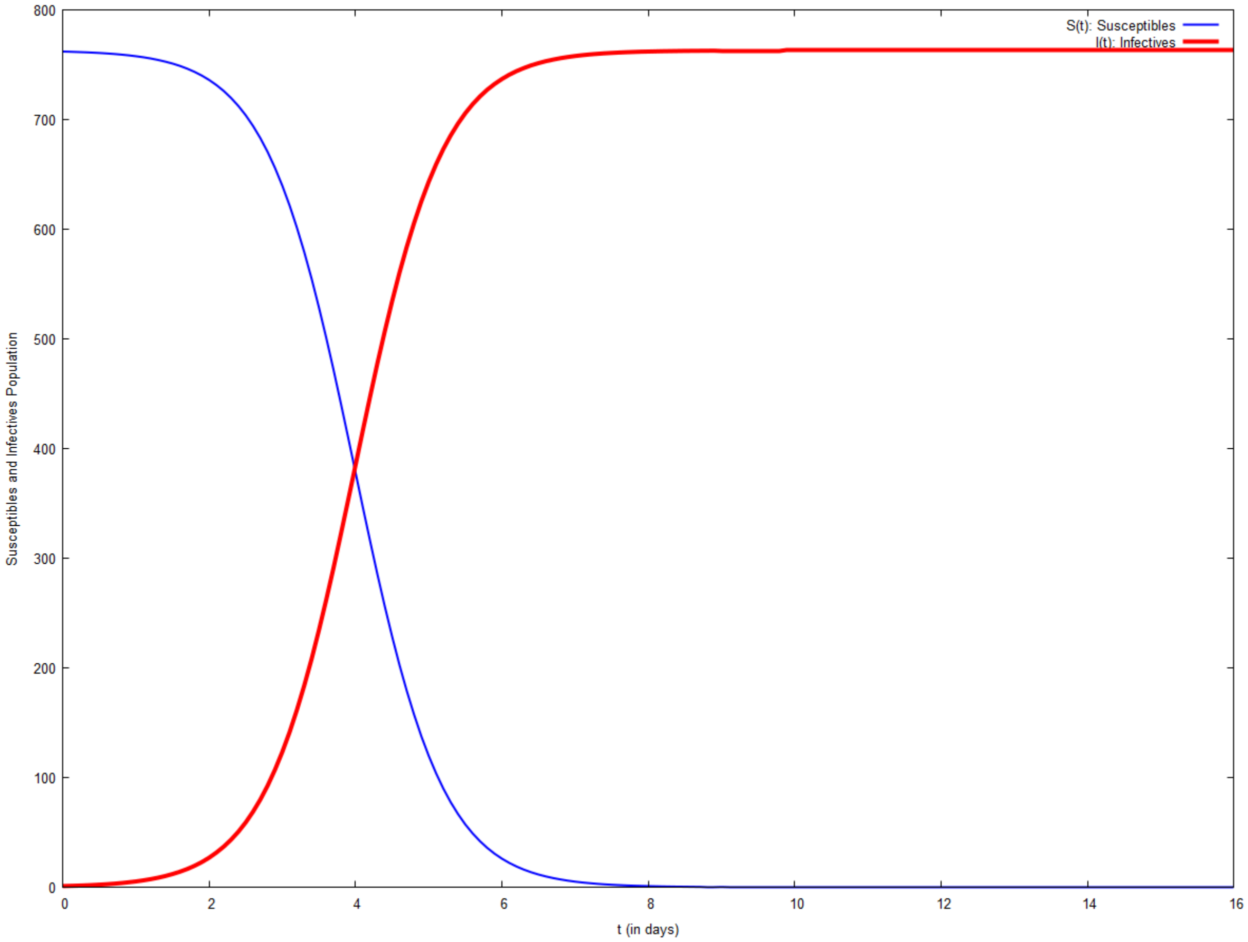
$$\frac{d}{dt}S = -0.00218IS$$

$$\frac{d}{dt}I = 0.00218IS$$

$$[[0.0\,, 762.0\,, 1.0]\,, [16.0\,, 1.60870825366987910^{-6}\,, 762.9999983912916]\,, 161]$$

$$[[0.0\,, 762.0]\,, [16.0\,, 1.60870825366987910^{-6}]\,, 161]$$

$$[[0.0\,,1.0]\,,[16.0\,,762.9999983912916]\,,161]$$



Created with wxMaxima.
The source of this Maxima session can be downloaded here.