```
> # Runge-Kutta method  for dy/dx=-y, with  x0:=0.0; y0:=2.0;
  # on interval [0,1]
> restart:Digits:=5:  with(plots):with(linalg):
    f:=(x,y)->-y; x0:=0.0; y0:=2.0;  # f(x,y) in DE dy/dx=-y and
                 # initial condition ->  exact solution is y=2*exp
  (-x)
  n:=5; xn:=1.0; h:=(xn-x0)/n;     #  make  n=5
      xx:=vector(n+1):  yy:=vector(n+1):  xx[1]:=x0:  yy[1]:=y0:
  for i from 1 to n do
    x:=xx[i]; y:=yy[i];
    k1:=f(x,y); k2:=f(x+0.5*h,y+0.5*h*k1); k3:=f(x+0.5*h,y+0.5*h*
  k2);
     k4:=f(x+h,y+h*k3); xx[i+1]:=x+h; yy[i+1]:=y+h/6*(k1+2*k2+2*
  k3+k4);
  od:
  # now plot computed and exact solns
    points:= {seq([xx[i],yy[i]],i=1..n+1) }:
    pointplot(points,symbol=asterisk,
                title="Approximate soln computed by Runge-Kutta
  method");
   display(plot(2*exp(-t),t=0..1),pointplot(points),
                                  title="Exact and computed solns" )
  ;
```

$$f := (x, y) \rightarrow -y$$

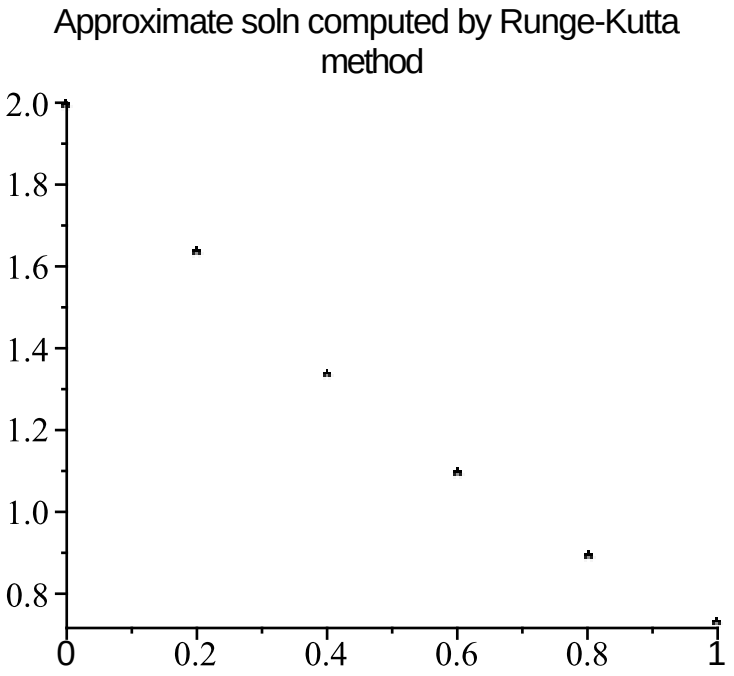
$$x0 := 0.$$

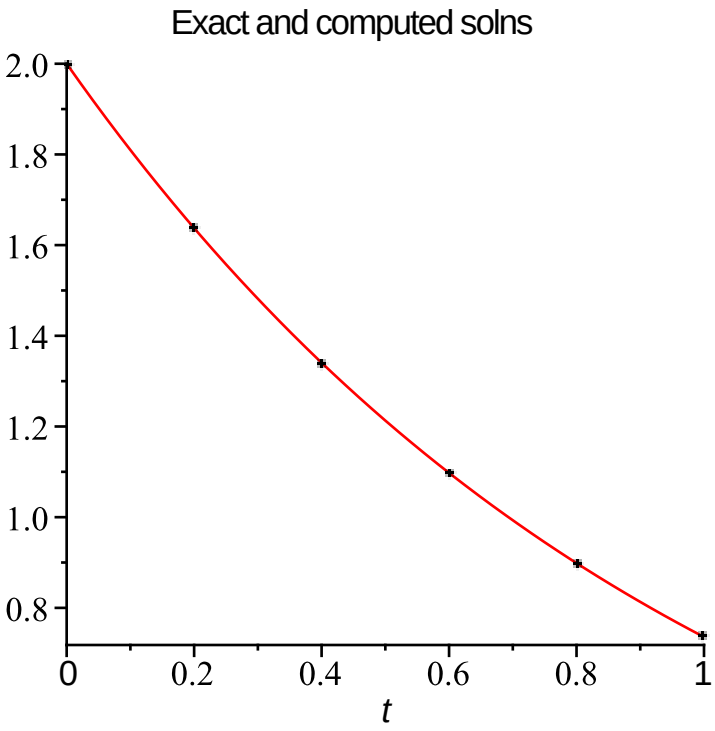
$$y0 := 2.0$$

$$n := 5$$

$$xn := 1.0$$

$$h := 0.20000$$


Approximate soln computed by Runge-Kutta method
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0
0.2
0.4
0.6
0.8
1


Exact and computed solns
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0
0.2
0.4
0.6
0.8
1
t