Rezolvarea ecuatiilor diferentiale in SageMath

Definirea si rezolvarea ecuatiilor diferentiale de ordinul I

Consideram urmatoarea ecuație diferențială

$$y'(x) = 2 \cdot y(x)$$

Aceasta poate fi definită in SageMath astfel:

In [1]:

```
reset()
x=var('x')
y=function('y')(x)
eqd=diff(y,x)==2*y
eqd
```

Out[1]:

```
diff(y(x), x) == 2*y(x)
```

In [2]:

```
show(eqd)
```

$$\frac{\partial}{\partial x} y(x) = 2\, y(x)$$

```
Comanda de rezolvare este urmatoarea:

desolve(equation, variable, ics = ..., ivar = ..., show_method = ..., contrib_ode = ...)

unde:

equation este ecuația diferențală. Egalitatea se scrie  ==;

variable este variabila dependentă, i.e., y ca y(x);

ics este optional si se foloseste atunci cand, pe langa ecuatie sunt si conditii
inițiale. Pentru ecuația diferențială de ordinul 1 se va scrie [x0,y0] iar pentru ecuația
diferențială de ordinul 2 se va scrie [x0,y0,x1,y1] sau [x0,y0,y0'];

ivar este optional; cu ajutorul acestei optiuni se poate specifica variabila
independenta, i.e., x in y(x). Aceasta trebuie specificata daca sunt mai multe variabile
independente sau parametri.

show_method este un boolean optional si este setat implicit false. Daca este setat true
atunci Sage resturneza o pereche de forma "[solution, method]", unde method este metoda
folosita pentru a determina soluția.

contrib_ode este un boolean optional setat implicit false. Daca este setat true, desolve
poate rezolva ecuații Clairaut, Lagrange, Riccati si altele. Determinarea metodei poate
dura, de aceea este setat implicit false.
```

In [3]:

```
desolve(eqd,y)
```

Out[3]:

```
_C*e^(2*x)
```

In [4]:

```
show(desolve(eqd,y))
```

$$Ce^{(2\,x)}$$

Dacă dorim să vedem ce metodă foloseste SageMath pentru determinarea solutiei vom folosi comanda:

In [5]:

```
desolve(eqd,y,show_method=True)
```

Out[5]:

```
[_C*e^(2*x), 'linear']
```

Nu intotdeauna *dsolve* returneaza solutia ecuatiei diferentiale in forma explicita. In astfel de situatii putem incerca sa rezolvam expresia solutiei in forma implicita in raport cu functia necunoscuta y. In situatia in care comanda *solve* returneaza expresia solutiei in forma explicita putem lucra cu aceasta, daca comanda *solve* nu reuseste explicitarea ui y atunci vom lucra in continuare cu forma implicita a solutiei.

De exemplu, pentru ecuatia

$$1 + y^2(x) + xy(x)y'(x) = 0$$

obtinem

In [6]:

```
eqd=(1+y^2)+x*y*diff(y,x)==0
sol=desolve(eqd,y)
show(sol)
```

$$-\frac{1}{2}\log\left(y(x)^2 + 1\right) = C + \log(x)$$

In [7]:

```
solve(sol,y)
```

Out[7]:

```
[y(x) == -sqrt(-x^2*e^(2*_C) + 1)*e^(-_C)/x, y(x) == sqrt(-x^2*e^(2*_C) + 1)
*e^(-_C)/x]
```

In [8]:

```
show(solve(sol,y))
```

$$\left[y\left(x\right) = -\frac{\sqrt{-x^{2} e^{\left(2 \, C\right)} + 1} e^{\left(-C\right)}}{x}, y\left(x\right) = \frac{\sqrt{-x^{2} e^{\left(2 \, C\right)} + 1} e^{\left(-C\right)}}{x}\right]$$

Reprezentarea grafică a soluțiilor obținute în formă explicită

Pentru a reprezenta grafic solutiile ecuatiilor diferentiale avem două variante.

Prima variantă este sa dăm o valoare (sau mai multe) constantei de integrare si apoi sa folosim *plot* pentru reprezentare grafica.

A doua variantă este sa definim solutia ca o funcție de doua variabile, una fiind x iar cealalta constanta de integrare $\_C$. Graficul se va obtine folosind comanda *plot* pentru o anumita valoare a constantei.

In [9]:

```
eqd=diff(y,x)==2*y
sol=desolve(eqd,y)
_C=var('_C')
sol.substitute(_C==1)
```

Out[9]:

```
e^(2*x)
```

In [10]:

```
show(sol.substitute(_C==1))
```

$$e^{\left(2 \, x\right)}$$

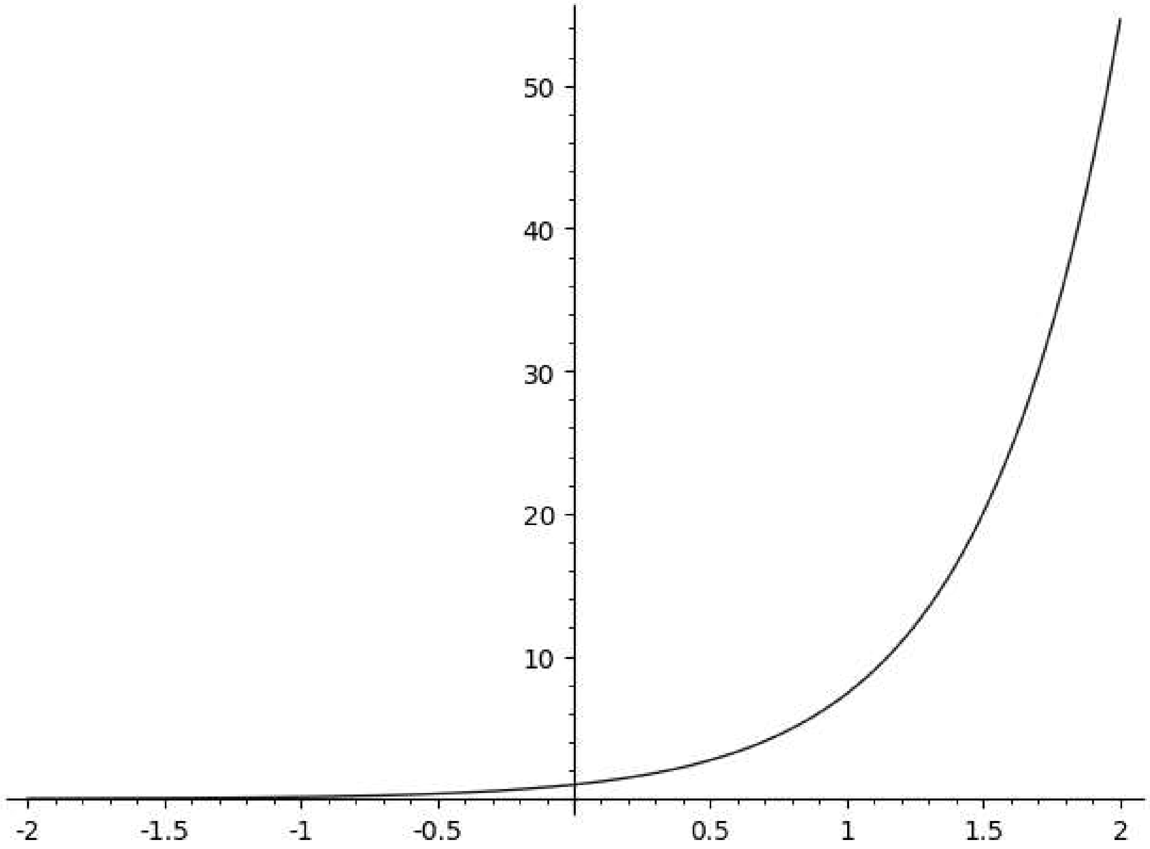
In [11]:

```
sol1=sol.substitute(_C==1)
```

In [12]:

```
plot(sol1,x,-2,2)
```

Out[12]:

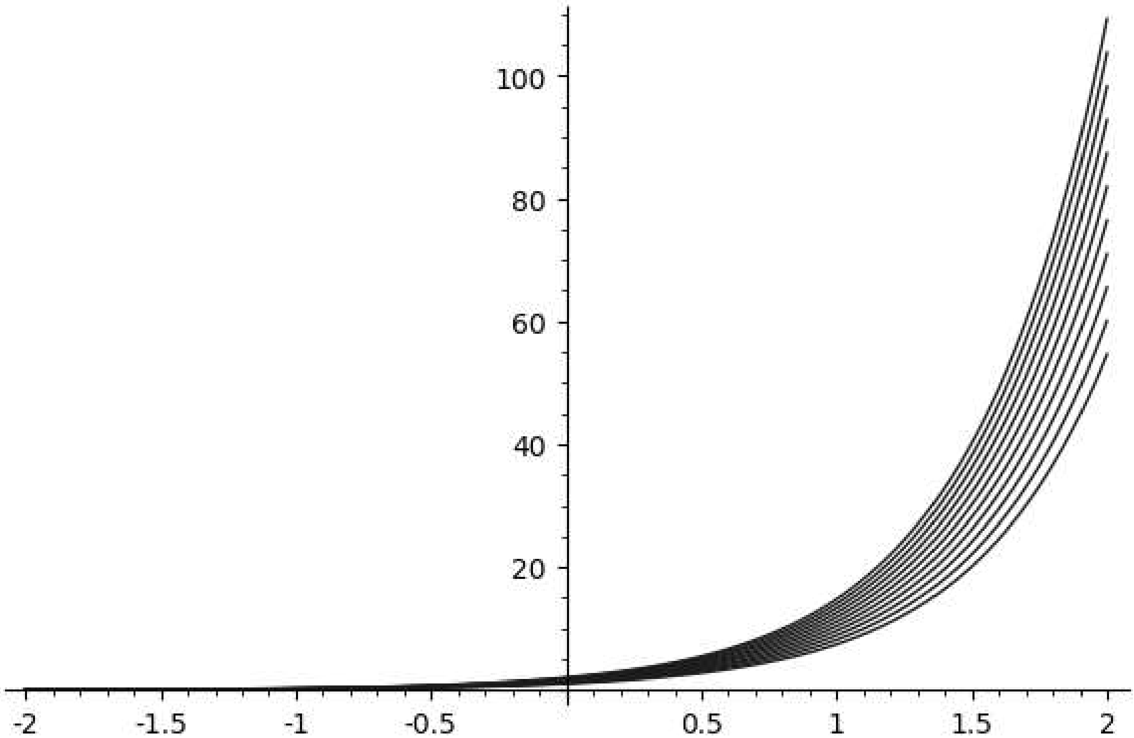


Dacă dorim să reprezentam mai multe solutii pentru diverse valori ale constantei de integrare putem folosi comanda *for*. De exemplu să reprezentăm grafic soluțiile corespunzătoare constantelor

$\_C = 1, 1.1, 1.2, \ldots, 2$

In [13]:

```
sol=desolve(eqd,y)
sol1=sol.substitute(_C==1)
g=plot(sol1,x,-2,2)
for i in [11..20]:
    sol1=sol.substitute(_C==i/10)
    g=g+plot(sol1,x,-2,2)
show(g)
```



A doua metoda este sa definim solutia ca o functie de doua variabile, x si $\_C$

In [14]:

```
sol(x,_C)=desolve(eqd,y)
sol(x,_C)
```

Out[14]:

_C*e^(2*x)

In [15]:

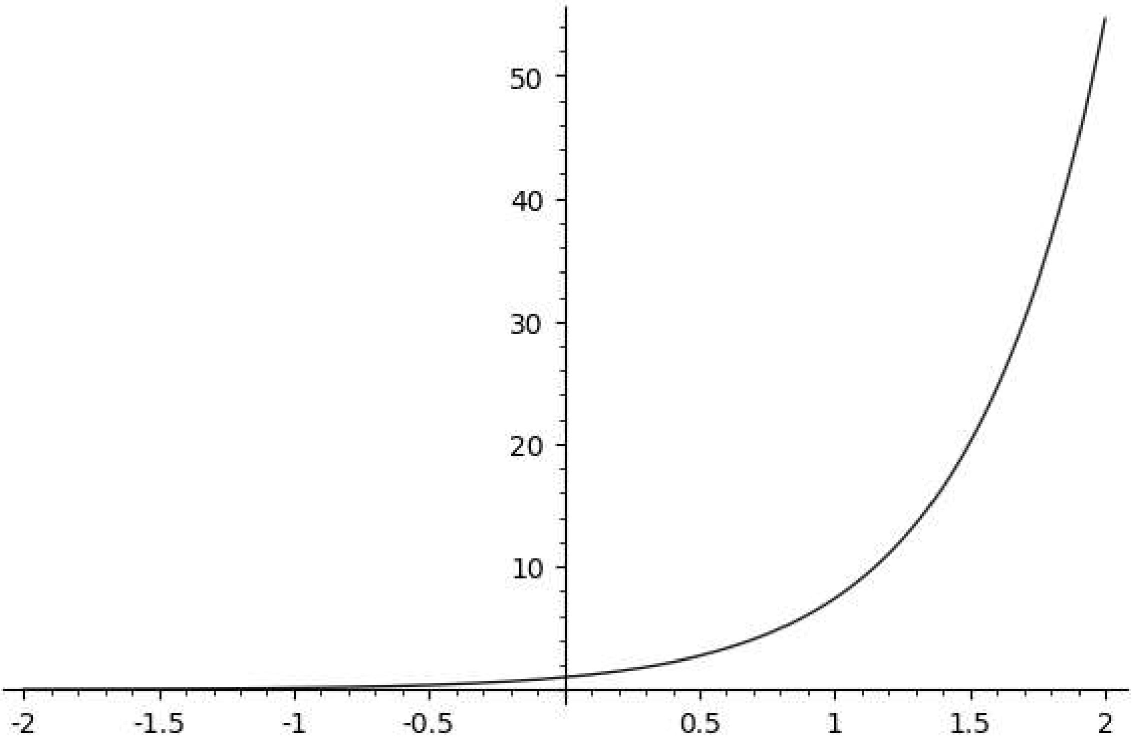
```
show(sol(x,_C))
```

$Ce^{(2\,x)}$

In [16]:

```
plot(sol(x,1),x,-2,2)
```

Out[16]:



Graficele solutiilor corespunzatoare constantelor de integrare

$\_C = 1, 1.1, 1.2, \ldots, 2$

le putem obtine si definind o listă a solutiilor corespunzătoare si apoi folosind comanda *plot*.

In [17]:

```
list_f=[sol(x,i/10) for i in [10..20]]
list_f
```

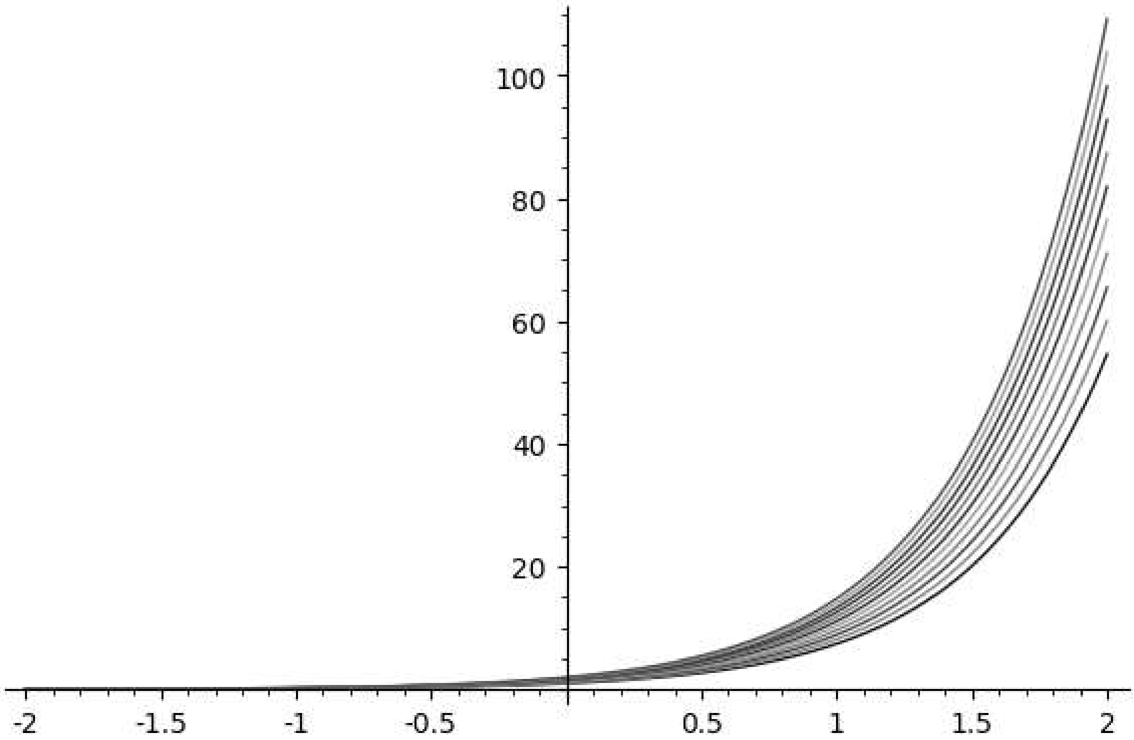
Out[17]:

```
[e^(2*x),
 11/10*e^(2*x),
 6/5*e^(2*x),
 13/10*e^(2*x),
 7/5*e^(2*x),
 3/2*e^(2*x),
 8/5*e^(2*x),
 17/10*e^(2*x),
 9/5*e^(2*x),
 19/10*e^(2*x),
 2*e^(2*x)]
```

In [18]:

```
plot(list_f,x,-2,2)
```

Out[18]:



Reprezentarea grafică a soluțiilor in formă implicită

In cazul in care soluția este obținută în formă implicită se va folosi comanda implicit_plot pentru reprezentarea grafică a acesteia. De exemplu, fie ecuația diferențială

$$(3y^2(x) + e^x)y'(x) + e^x(y(x) + 1) + cos(x) = 0$$

In [19]:

```
x=var('x')
y=function('y')(x)
eqd=(3*y^2+exp(x))*diff(y,x)+exp(x)*(y+1)+cos(x)==0
desolve(eqd,y)
```

Out[19]:

```
y(x)^3 + e^x*y(x) + e^x + sin(x) == _C
```

Să observăm că soluția în formă implicită apare $y(x)$. Prima data vom rescrie solutia in forma implicita din partea stanga, inlocuind $y(x)$ cu variabila yy.

In [20]:

```
sol=desolve(eqd,y)
sol
```

Out[20]:

```
y(x)^3 + e^x*y(x) + e^x + sin(x) == _C
```

In [21]:

```
yy=var('yy')
sol.substitute(y(x)==yy)
```

Out[21]:

```
yy^3 + yy*e^x + e^x + sin(x) == _C
```

In [22]:

```
_C=var('_C')
f(x,yy,_C)=sol.substitute(y(x)==yy)
f(x,yy,_C)
```

Out[22]:

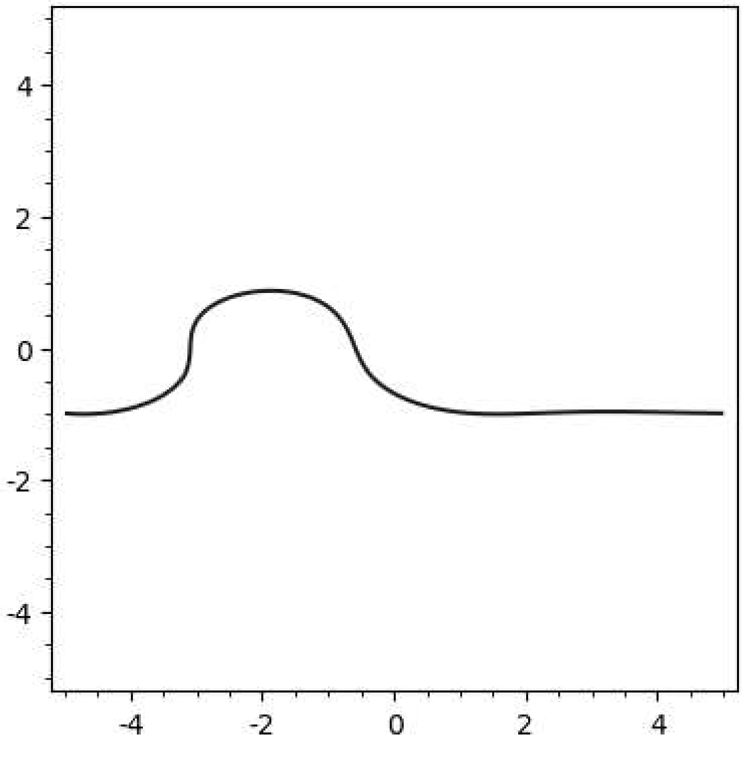
```
yy^3 + yy*e^x + e^x + sin(x) == _C
```

Pentru valoarea constantei $\_C = 0$ vom obține următorul grafic.

In [23]:

```
implicit_plot(f(x,yy,0),(x,-5,5),(yy,-5,5))
```

Out[23]:

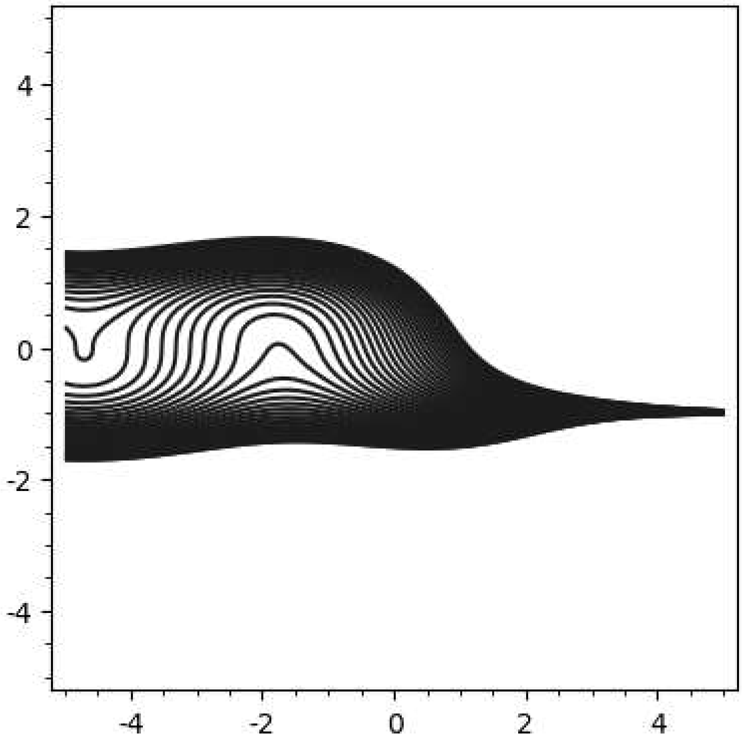


Daca dorim sa reprezentăm mai mult soluții vom atribui fiecarui grafic o variabilă și le reprezentăm folosind suma (+) si comanda *show*.

De exemplu, pentru valorile $c = -4, -19/5, \ldots, -1/5, 0, 1/5, 2/5, \ldots, 4$ folosim *for* pentru a reprezenta graficele.

In [24]:

```
g=implicit_plot(f(x,yy,-4),(x,-5,5),(yy,-5,5))
for i in [-20..20]:
    g1=implicit_plot(f(x,yy,i/5),(x,-5,5),(yy,-5,5))
    g=g+g1
show(g)
```



Definirea si rezolvarea ecuațiilor diferențiale de ordinul 2. Reprezentarea grafică a soluțiilor.

Definirea unei ecuații diferențiale de ordinul 2 se face asemanator cu definirea celei de ordinul 1, in plus va aparea derivata a de ordinul 2 y''.

Consideram urmatoarea ecuație diferențială de ordinul 2:

$$y'' + 3y' + 2y = 1 + x^2$$

In [25]:

```
x=var('x')
y=function('y')(x)
eqd=diff(y,x,2)+3*diff(y,x)+2*y==1+x^2
desolve(eqd,y)
```

Out[25]:

```
1/2*x^2 + _K1*e^(-x) + _K2*e^(-2*x) - 3/2*x + 9/4
```

In [26]:

```
show(desolve(eqd,y))
```

$$\frac{1}{2}x^2 + K_1 e^{(-x)} + K_2 e^{(-2\,x)} - \frac{3}{2}x + \frac{9}{4}$$

Observam ca soluția depinde de doua constante. Pentru reprezentarea grafica a solutiei va trebui sa dam valori ambelor constante. Ca și în cazul ecuațiilor de ordinul 1, se pot folosi două metode pentru reprezentarea grafică. Prima metoda este să atribuim valori constantelor, a doua metoda este sa construim solutia ca functie ce depinde de trei variabile x, $\_K1$, $\_K2$.

In [27]:

```
sol=desolve(eqd,y)
sol
```

Out[27]:

```
1/2*x^2 + _K1*e^(-x) + _K2*e^(-2*x) - 3/2*x + 9/4
```

In [28]:

```
_K1,_K2=var('_K1,_K2')
sol.substitute(_K1==1,_K2==3)
```

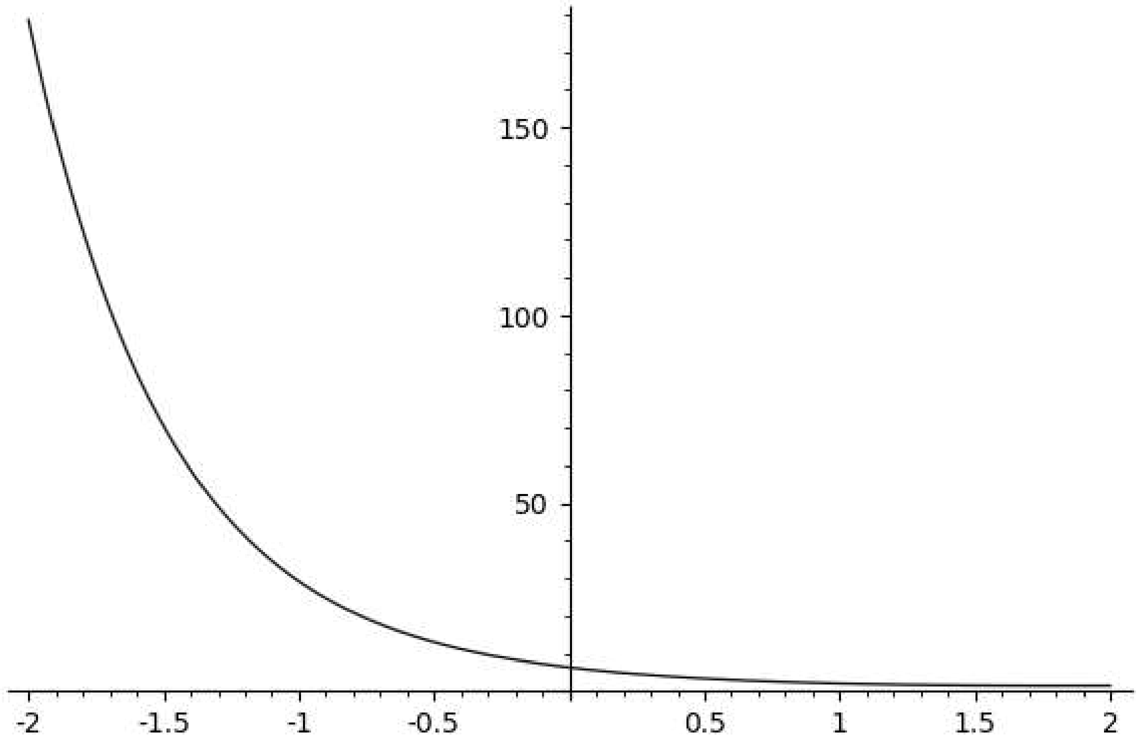
Out[28]:

```
1/2*x^2 - 3/2*x + e^(-x) + 3*e^(-2*x) + 9/4
```

In [29]:

```
sol1=sol.substitute(_K1==1,_K2==3)
plot(sol1,x,-2,2)
```

Out[29]:

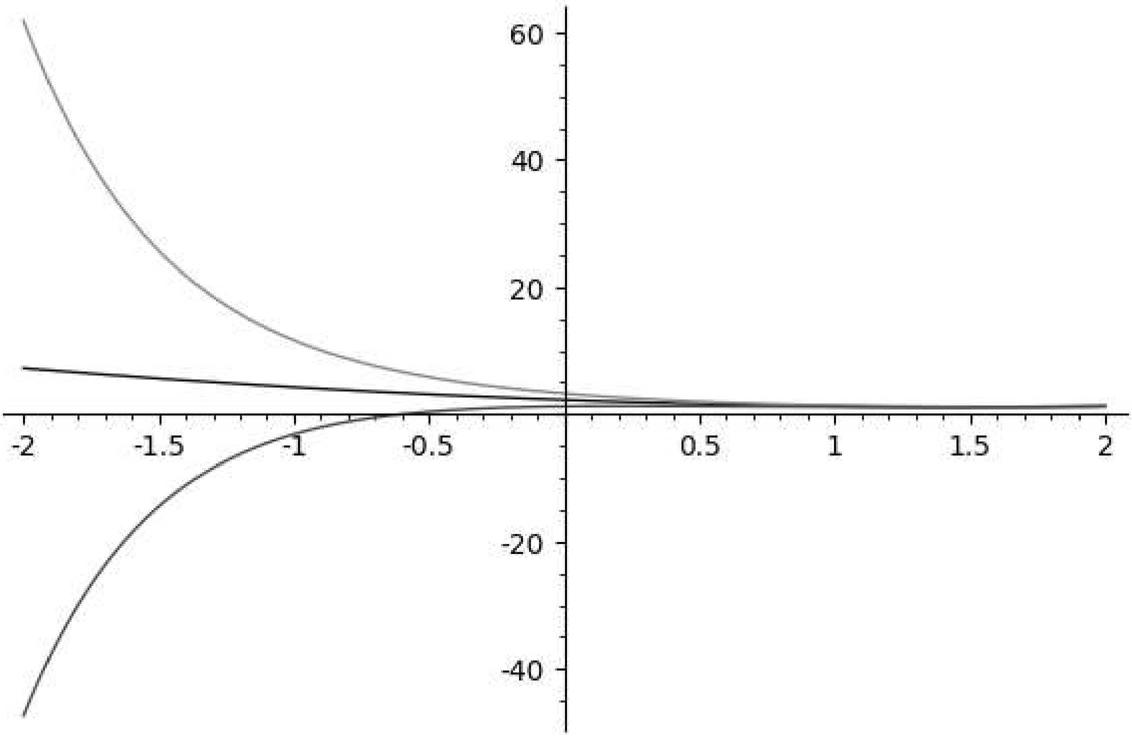


Dacă dorim să comparam diferite solutii corespunzatoare diferitelor valori ale constantelor _K1 și _K2 vom folosi

In [30]:

```
sol1=sol.substitute(_K1==0,_K2==0)
sol2=sol.substitute(_K1==0,_K2==1)
sol3=sol.substitute(_K1==0,_K2==-1)
plot([sol1,sol2,sol3],x,-2,2)
```

Out[30]:



Putem obtine acelasi lucru definind solutia ca funcție de trei variabile $x, \_K1, \_K2$

In [31]:

```
_K1,_K2=var('_K1,_K2')
sol(x,_K1,_K2)=desolve(eqd,y)
sol(x,_K1,_K2)
```

Out[31]:

```
1/2*x^2 + _K1*e^(-x) + _K2*e^(-2*x) - 3/2*x + 9/4
```

In [32]:

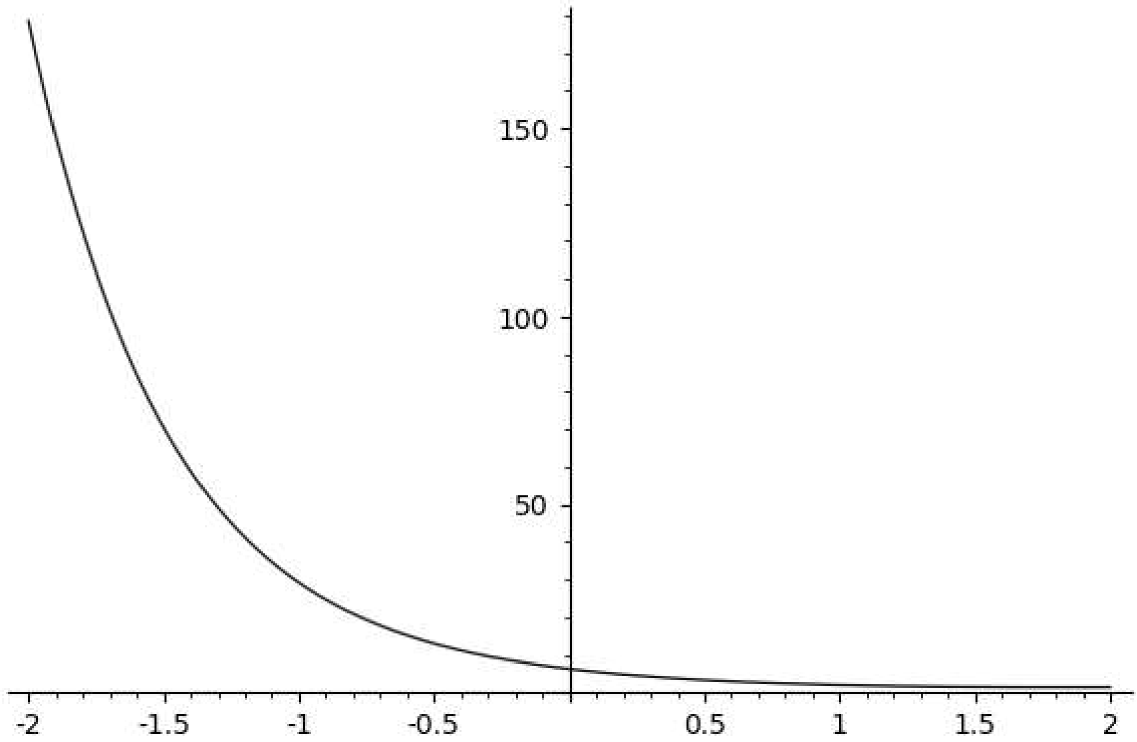
```
show(sol)
```

$$(x, K_1, K_2) \mapsto \frac{1}{2}x^2 + K_1 e^{(-x)} + K_2 e^{(-2x)} - \frac{3}{2}x + \frac{9}{4}$$

In [33]:

```
plot(sol(x,1,3),x,-2,2)
```

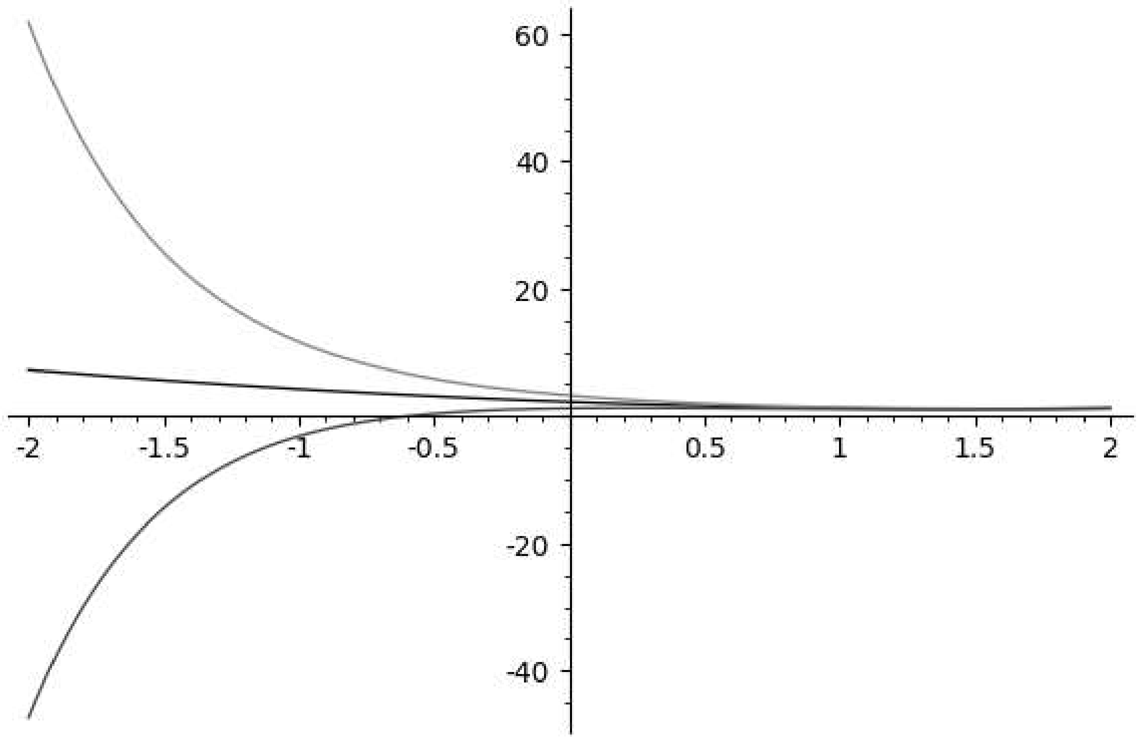
Out[33]:



In [34]:

```
plot([sol(x,0,0),sol(x,0,1),sol(x,0,-1)],x,-2,2)
```

Out[34]:



Rezolvarea problemelor cu valori initiale (Probleme Cauchy). Reprezentarea grafica a solutiilor.

Problema cu valori initiale pentru ecuatia diferentiala de ordinul 1.

```
Comanda de rezolvarea a unei probleme cu valori inițiale este:

desolve(equation, variable, ics = ...)

unde:

ics este variabila corespunzatoare conditiei initiale.

Pentru ecuatia de ordinul 1 vom scrie ics=[x0,y0] pentru conditia y(x0)=y0
```

De exemplu, pentru problema Cauchy

$$\begin{cases} y' & = 2y \\ y(0) & = 1 \end{cases}$$

avem

In [35]:

```
x=var('x')
y=function('y')(x)
eqd=diff(y,x)==2*y
desolve(eqd,y,ics=[0,1])
```

Out[35]:

```
e^(2*x)
```

In [36]:

```
sol=desolve(eqd,y,ics=[0,1])
sol
```

Out[36]:

```
e^(2*x)
```

In [37]:

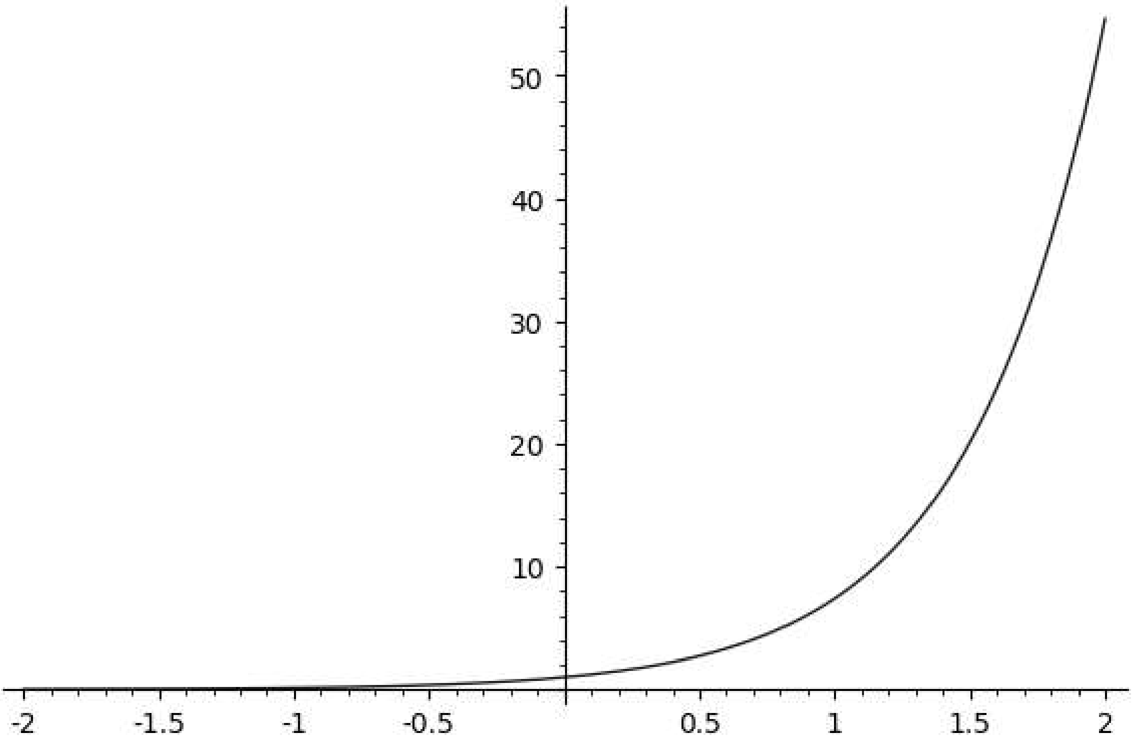
```
show(sol)
```

$e^{(2\,x)}$

In [38]:

```
plot(sol,x,-2,2)
```

Out[38]:



In cazul in care ecuatia diferentiala depinde de un parametru este interesant sa studiem dependenta solutiei de acel parametru. De exemplu, fie problema

$$\begin{cases} y' & = ky \\ y(0) & = 1 \end{cases}$$

```
Observatie: Daca ecuatia diferentiala depinde de un parametru, in comanda desolve va
trebui specificata lista variabilelor, i.e, daca dorim sa gasim solutia y=y(x) vom folosi
astfel comanda  desolve

desolve(eqd, [y,x])
```

In [39]:

```
x,k=var('x,k')
y=function('y')(x)
eqd=diff(y,x)==k*y
desolve(eqd,[y,x],ics=[0,1])
```

Out[39]:

e^(k*x)

In [40]:

```
sol(x,k)=desolve(eqd,[y,x],ics=[0,1])
sol(x,k)
```

Out[40]:

e^(k*x)

In [41]:

```
show(sol)
```

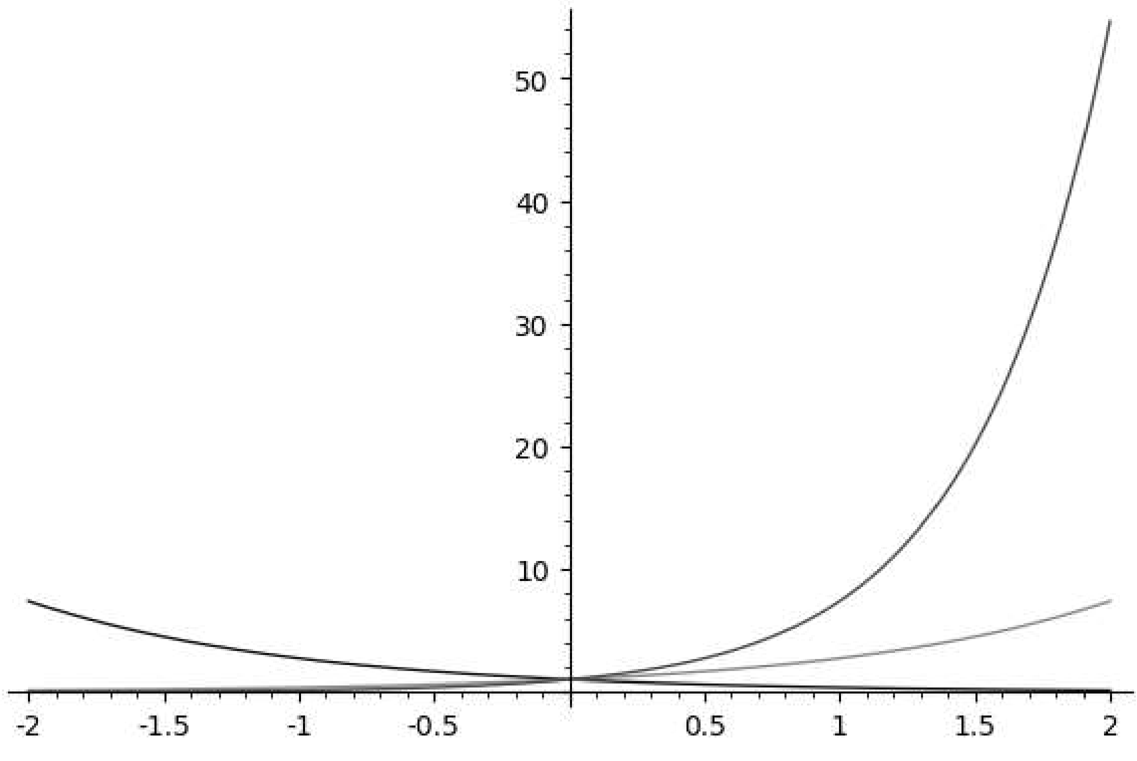
$(x, k) \mapsto e^{(kx)}$

Pentru a studia dependenta solutiei problemei Cauchy in raport cu k, vom reprezenta grafic cateva solutii pentru diferite valori ale lui k.

In [42]:

```
plot([sol(x,-1),sol(x,1),sol(x,2)],x,-2,2)
```

Out[42]:



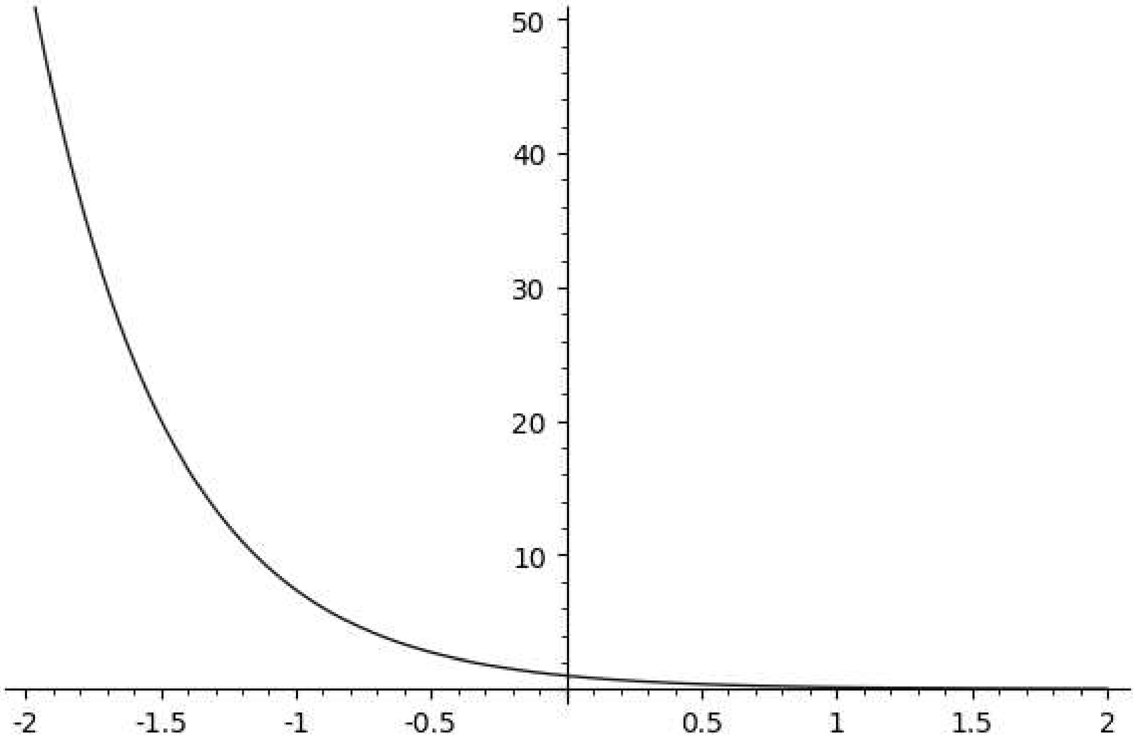
Se poate folosi comanda *animate* pentru a vizualiza modul in care se modifica solutia atunci cand k se

Se poate folosi comanda *animate* pentru a vizualiza modul in care se modifica solutia atunci cand k se modifica:

In [43]:

```
sols=[plot(sol(x,k/20),x,-2,2,ymin=0,ymax=50) for k in [-40..40]]
animate(sols)
```

Out[43]:



Problema Cauchy pentru ecuatia diferentiala de ordinul 2

```
Comanda pentru determinarea solutiei problemei Cauchy atasata unei ecuatii diferentiale
de ordinul 2 este

desolve(equation, variable, ics = ...)

unde:

ics=[x0,y0,y1] pentru conditiile initiale

y(x0)=y0

y'(x0)=y1
```

In cazul problemei Cauchy

$$\begin{cases} y'' + 3y' + 2y = 1 + x^2 \\ y(0) = 1 \\ y'(0) = 1 \end{cases}$$

avem

In [44]:

```
x=var('x')
y=function('y')(x)
eqd=diff(y,x,2)+3*diff(y,x)+2*y==1+x^2
desolve(eqd,y,ics=[0,1,1])
```

Out[44]:

```
1/2*x^2 - 3/2*x - 5/4*e^(-2*x) + 9/4
```

In [45]:

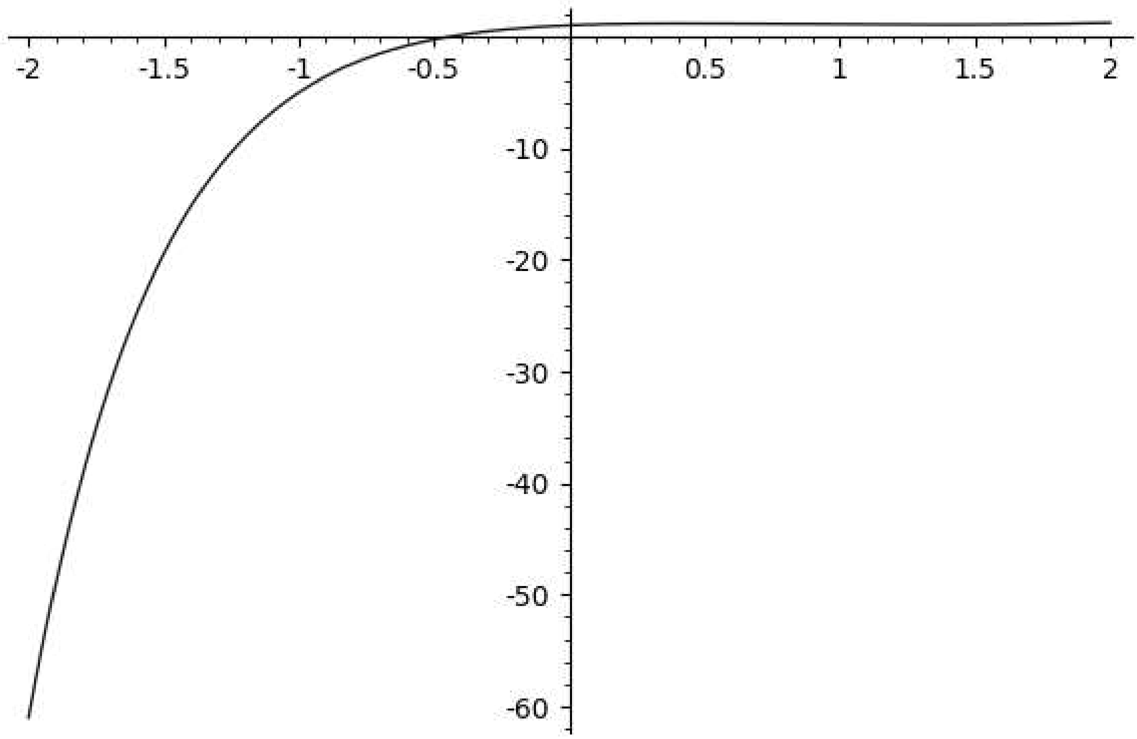
```
sol=desolve(eqd,y,ics=[0,1,1])
show(sol)
```

$$\frac{1}{2}\,x^2 - \frac{3}{2}\,x - \frac{5}{4}\,e^{(-2\,x)} + \frac{9}{4}$$

In [46]:

```
plot(sol,x,-2,2)
```

Out[46]:



In [ ]: