Mathematical Economics

Alpha Chiang

Chapter 19

Simultaneous Differential Equations

In [2]:

```
from sympy import *

t = symbols('t')
x = Function('x')
y = Function('y')
dydt =  61 - x(t) - 4*y(t)
eqs = [
        Eq(x(t).diff(t) + 2*dydt + 2*x(t)+ 5*y(t) -77,0),
        Eq(y(t).diff(t) +x(t) + 4*y(t) -61,0)
    ]

pprint(eqs[0])
pprint(eqs[1])
```

```
           d
-3⋅y(t) + ──(x(t)) + 45 = 0
          dt
                 d
x(t) + 4⋅y(t) + ──(y(t)) - 61 = 0
                dt
```

In [3]:

```
ics = {x(0): 6, y(0): 12}
DD = dsolve(eqs, [x(t), y(t)],ics = ics)
print(DD)
```

```
[Eq(x(t), 1 + 3*exp(-t) + 2*exp(-3*t)), Eq(y(t), 15 - exp(-t) - 2*exp(-3*t))]
```

Using https://www.researchgate.net/profile/Stephen-Mason-8

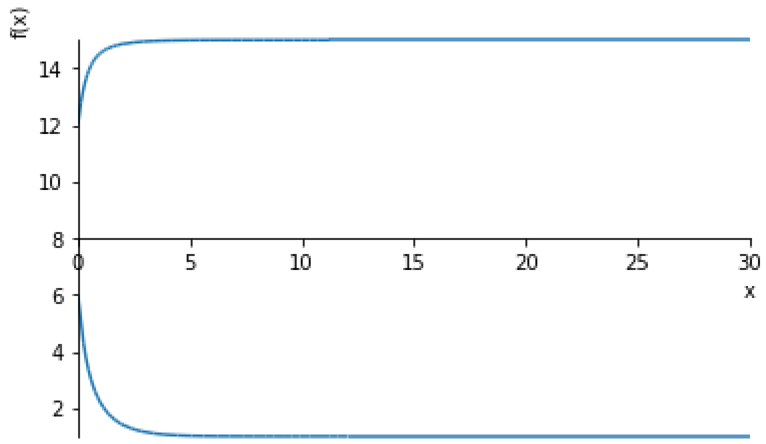
In [8]:
```python
import numpy as np
import matplotlib.pyplot as plt
from sympy import init_printing
init_printing()

from sympy import Function, Indexed, Tuple, sqrt, dsolve, solve, Eq, Derivative, sin, cos, symbols
from sympy.abc import k, t
from sympy import solve, Poly, Eq, Function, exp

from sympy import Indexed, IndexedBase, Tuple, sqrt
from IPython.display import display
from sympy import *
from sympy.abc import *
from sympy.plotting import plot
init_printing()
t, C1, C2 = symbols("t C1 C2")
x, y = symbols("x y", cls = Function, Function = True)


dydt =  61 - x(t) - 4*y(t)
eqs = [
        Eq(x(t).diff(t) + 2*dydt + 2*x(t)+ 5*y(t) -77,0),
        Eq(y(t).diff(t) +x(t) + 4*y(t) -61,0)
    ]
ics = {x(0): 6, y(0): 12}
soln = dsolve(eqs, [x(t), y(t)],ics = ics)

constants = solve((soln[0].subs(t,0).subs(x(0),1), soln[1].subs(t,0).subs(y(0),2)),{C1,C2})

xsoln = expand(soln[0].rhs.subs(constants))
display(xsoln)
print(xsoln)
ysoln = soln[1].rhs.subs(constants)
display(ysoln)
print(ysoln)

plot((xsoln, (t, 0, 30)), (ysoln, (t, 0, 30)))
```

$$1 + 3e^{-t} + 2e^{-3t}$$

```
1 + 3*exp(-t) + 2*exp(-3*t)
```

$$15 - e^{-t} - 2e^{-3t}$$

```
15 - exp(-t) - 2*exp(-3*t)
```



Out[8]: <sympy.plotting.plot.Plot at 0x7f74d0f35510>

EXERCISE 19.2 --Q/4/a--

In [9]:

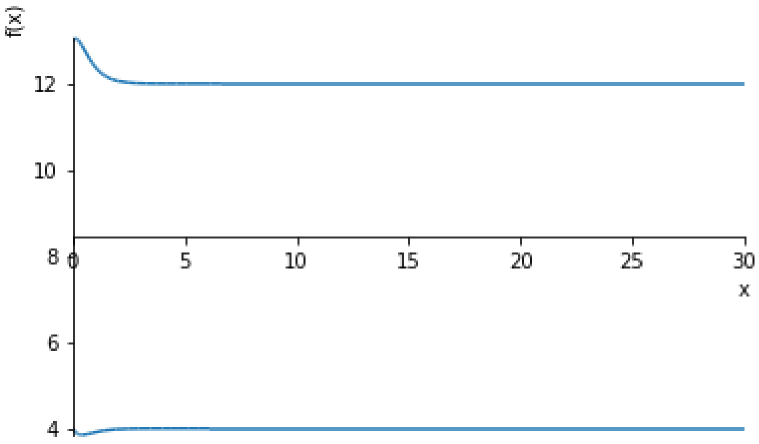
```
t, C1, C2 = symbols("t C1 C2")
x, y = symbols("x y", cls = Function, Function = True)


eqs = [
        Eq(x(t).diff(t) - x(t) - 12*y(t) + 60 ,0),
        Eq(y(t).diff(t) +x(t) + 6*y(t) - 36 ,0)
    ]
ics = {x(0): 13, y(0): 4}
soln = dsolve(eqs, [x(t), y(t)],ics = ics)

constants = solve((soln[0].subs(t,0).subs(x(0),1), soln[1].subs(t,0).subs(y(0),2)),{C1,C2})

xsoln = expand(soln[0].rhs.subs(constants))
display(xsoln)
print(xsoln)
ysoln = soln[1].rhs.subs(constants)
display(ysoln)
print(ysoln)
plot((xsoln, (t, 0, 30)), (ysoln, (t, 0, 30)))
```

$$12 + 4e^{-2t} - 3e^{-3t}$$

12 + 4*exp(-2*t) - 3*exp(-3*t)

$4 - e^{-2t} + e^{-3t}$

4 - exp(-2*t) + exp(-3*t)



Out[9]: <sympy.plotting.plot.Plot at 0x7f74d04f9710>

EXERCISE 19.2 --Q/4/b--

In [10]:

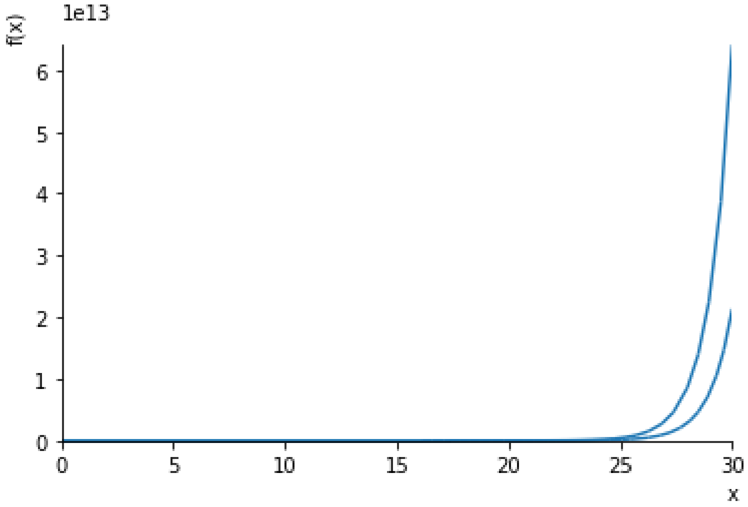
```
t, C1, C2 = symbols("t C1 C2")
x, y = symbols("x y", cls = Function, Function = True)


eqs = [
        Eq(x(t).diff(t) - 2*x(t) + 3*y(t) - 10 ,0),
        Eq(y(t).diff(t) - x(t) + 2*y(t) - 9 ,0)
     ]
ics = {x(0): 8, y(0): 5}
soln = dsolve(eqs, [x(t), y(t)],ics = ics)

constants = solve((soln[0].subs(t,0).subs(x(0),1), soln[1].subs(t,0).subs(y(0),2)),{C1,C2})

xsoln = expand(soln[0].rhs.subs(constants))
display(xsoln)
print(xsoln)
ysoln = soln[1].rhs.subs(constants)
display(ysoln)
print(ysoln)
plot((xsoln, (t, 0, 30)), (ysoln, (t, 0, 30)))
```

$$6e^t + 7 - 5e^{-t}$$

```
6*exp(t) + 7 - 5*exp(-t)
```

$$2e^t + 8 - 5e^{-t}$$

```
2*exp(t) + 8 - 5*exp(-t)
```



Out[10]:

```
<sympy.plotting.plot.Plot at 0x7f74d0386350>
```

Simultaneous Difference Equations

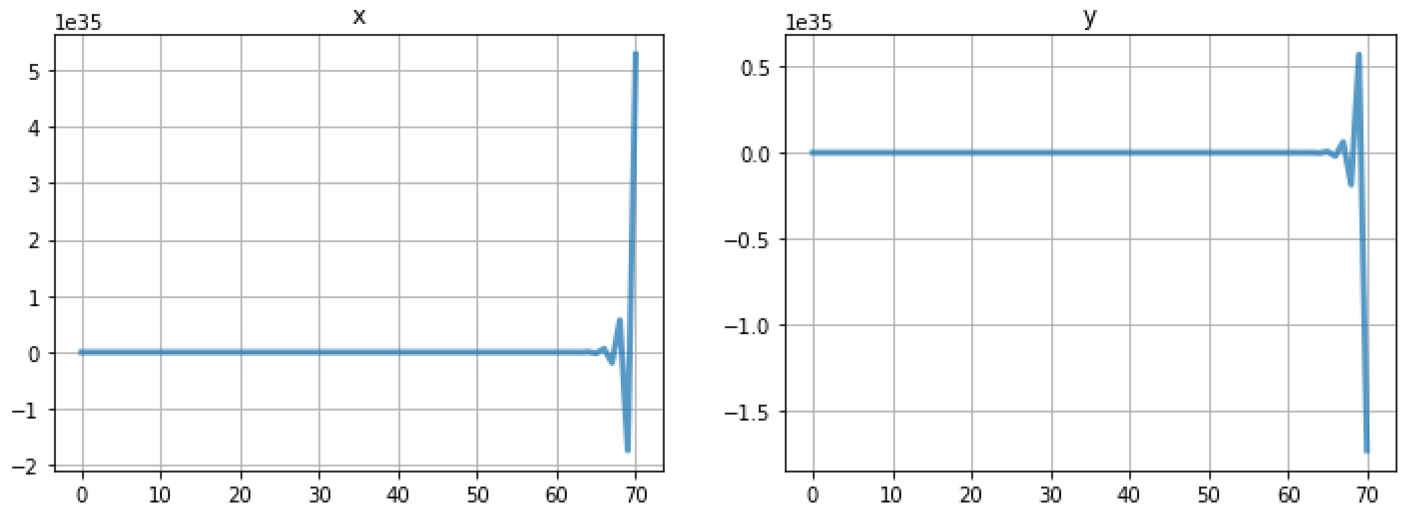
In [11]:

```
from sympy import Symbol, dsolve, Function, Derivative, Eq

from sympy import Function, rsolve
from sympy.abc import t,c
y = Function("y");
y0 = Symbol("y_0")
a1 = Symbol("a_1")
a2 = Symbol("a_2")

f = y(t+1) + 6*y(t) + 9*y(t-1) - 4
sol = rsolve(f, y(t), {y(0):1});
sol
```

Out[11]:

$$(-3)^t \left(C_1 t + \frac{3}{4}\right) + \frac{1}{4}$$

```
In [12]: T = 71
x = np.zeros(T)
x[0] = 1

y = np.zeros(T)

y[0] = 1

for t in range(T-1):
    x[t+1] = -6*x[t] - 9*y[t] + 4
    y[t+1] = x[t]



fig = plt.figure(figsize=(12,4))

ax = fig.add_subplot(1,2,1)
ax.plot(x,lw=3,alpha=0.75)
ax.set_title('x')
ax.grid()

ax = fig.add_subplot(1,2,2)
ax.plot(y,lw=3,alpha=0.75)
ax.set_title('y')
ax.grid()
```



The Inflation-Unemployment Model Once More

In [13]:

```python
from sympy import *

C1, C2 = symbols("C1 C2")
k,j, g, beta,alpha,T,mu = symbols("k j g \\beta \\alpha T \\mu")
t = symbols('t')
x = Function('x')
y = Function('y')

eqs = [
        Eq(x(t).diff(t) - j*(1-g)*x(t) + (j*beta)*y(t)- j*(alpha-T) ,0),
        Eq(y(t).diff(t) + k*g*x(t) + k*beta*y(t) - k*(alpha-T-mu) ,0)
    ]

pprint(eqs[0])
pprint(eqs[1])
```

```
                                                  d
\beta⋅j⋅y(t) - j⋅(1 - g)⋅x(t) - j⋅(-T + \alpha) + ──(x(t)) = 0
                                                  dt
                                                  d
\beta⋅k⋅y(t) + g⋅k⋅x(t) - k⋅(-T + \alpha - \mu) + ──(y(t)) = 0
                                                  dt
```

In [14]:

```python
DD = dsolve(eqs, [x(t), y(t)])
```

In [15]:

```python
constants = solve((DD[0].subs(t,0).subs(x(0),1), DD[1].subs(t,0).subs(y(0),2)),{C1,C2})

xsoln = expand(DD[0].rhs.subs(constants))
ysoln = DD[1].rhs.subs(constants)
```

In [16]:

```python
C1, C2 = symbols("C1 C2")
k,j, g, beta,alpha,T,mu = symbols("k j g \\beta \\alpha T \\mu")
t = symbols('t')
x = Function('x')
y = Function('y')

eqs = [
        Eq(x(t).diff(t) - 3/4*(1-1)*x(t) + (3/4*3)*y(t)- 3/4*(1/6) ,0),
        Eq(y(t).diff(t) + 1/2*1*x(t) + 1/2*3*y(t) - 1/2*(1/6-mu) ,0)
    ]
```

```
pprint(eqs[0])
pprint(eqs[1])
```

```
                d
2.25⋅y(t) + ──(x(t)) - 0.125 = 0
                dt
                                         d
0.5⋅\mu + 0.5⋅x(t) + 1.5⋅y(t) + ──(y(t)) - 0.0833333333333333 = 0
                                         dt
```

In [17]:

```
DD = dsolve(eqs, [x(t), y(t)])
```

In [18]:

```
constants = solve((DD[0].subs(t,0).subs(x(0),1), DD[1].subs(t,0).subs(y(0),2)),{C1,C2})

xsoln = expand(DD[0].rhs.subs(constants))
ysoln = DD[1].rhs.subs(constants)
```

Furkan zengin

In [ ]:

```
```