Mathematical Economics

Alpha Chiang

Chapter 16

Higher-Order Differential Equations

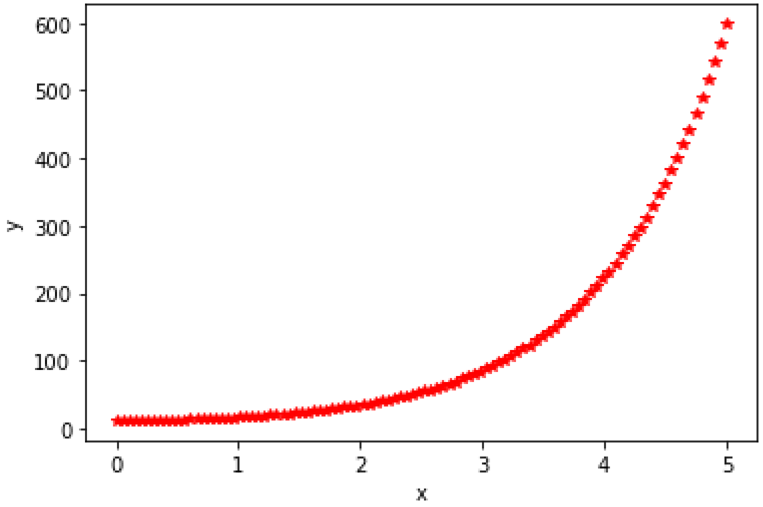
In [2]:
```
from matplotlib import pyplot as plt
from scipy.integrate import odeint
import numpy as np
```

Example 1

In [3]:
```
def f(y,x):
    return (y[1], - y[1] + 2 * y[0] - 10)

y0 = [12,-2]
xs = np.linspace(0,5,100)
sol = odeint(f, y0, xs)
ys = sol[:,0]
ys2 = sol[:,1]
```

In [4]:
```
plt.plot(xs, ys,"r*")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



In [5]:
```python
from sympy import Symbol, dsolve, Function, Derivative, Eq
y = Function("y")
t = Symbol('t')
dy2 = Derivative(y(t), t,2)
dy1 = Derivative(y(t),t)
eq1 = Eq(dy2 + dy1 - 2*y(t), -10)
eq1
```

Out[5]:

$$-2y(t) + \frac{d}{dt}y(t) + \frac{d^2}{dt^2}y(t) = -10$$

In [6]:
```python
sol1 = dsolve(eq1, y(t))
sol1
```

Out[6]:

$$y(t) = C_1e^{-2t} + C_2e^t + 5$$

Example 5

In [7]:
```python
y = Function("y")
t = Symbol('t')
dy2 = Derivative(y(t), t,2)
dy1 = 6 * Derivative(y(t),t)
eq1 = Eq(dy2 + dy1 +9 * y(t), 27)
eq1
```

Out[7]:

$$9y(t) + 6\frac{d}{dt}y(t) + \frac{d^2}{dt^2}y(t) = 27$$

In [8]:

```
sol2 = dsolve(eq1, y(t))
sol2
```

Out[8]:

$$y(t) = (C_1 + C_2 t)\, e^{-3t} + 3$$

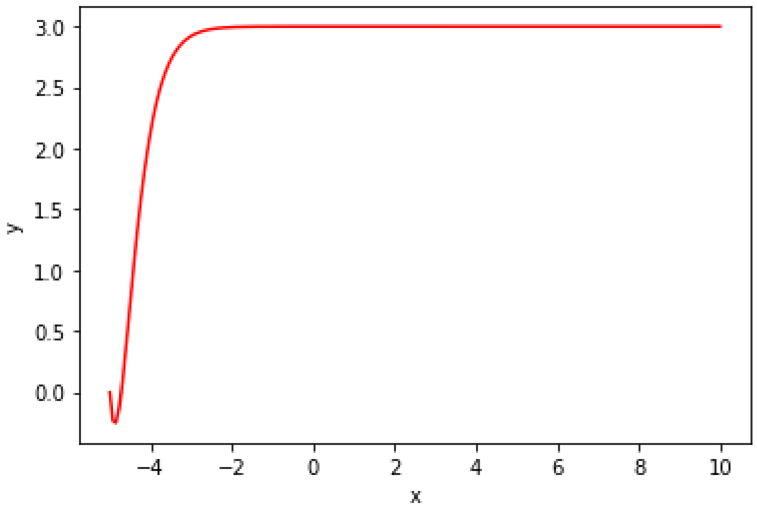
In [9]:

```
def f(y,x):
    return (y[1], - 6 * y[1] - 9 * y[0] + 27)

y0 = [0,-5]
xs = np.linspace(-5,10,200)
sol = odeint(f, y0, xs)
ys = sol[:,0]
ys2 = sol[:,1]
```

In [10]:

```
plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



Complex roots

In [11]:

```
y = Function("y")
```

```
t = Symbol('t')
dy2 = Derivative(y(t), t,2)
dy1 = 2 * Derivative(y(t),t)
eq1 = Eq(dy2 + dy1 + 17 * y(t), 34)
eq1
```

Out[11]:

$$17y(t) + 2\frac{d}{dt}y(t) + \frac{d^2}{dt^2}y(t) = 34$$

In [12]:

```
sol2 = dsolve(eq1, y(t))
sol2
```

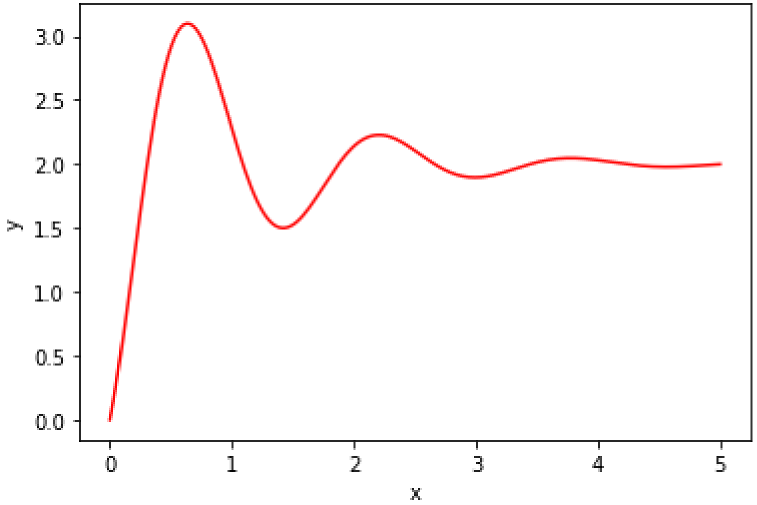
Out[12]:

$$y(t) = (C_1 \sin(4t) + C_2 \cos(4t))\, e^{-t} + 2$$

In [14]:

```
def f(y,x):
    return (y[1], - 2 * y[1] - 17 * y[0] + 34)

y0 = [0,5]
xs = np.linspace(0,5,200)
sol = odeint(f, y0, xs)
ys = sol[:,0]
plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



In [15]:
```python
y = Function("y")
t = Symbol('t')
dy2 = Derivative(y(t), t,2)
dy1 = -4 * Derivative(y(t),t)
eq1 = Eq(dy2 + dy1 + 8 * y(t), 0)
eq1
```

Out[15]:

$$8y(t) - 4\frac{d}{dt}y(t) + \frac{d^2}{dt^2}y(t) = 0$$

In [16]:
```python
sol2 = dsolve(eq1, y(t))
sol2
```

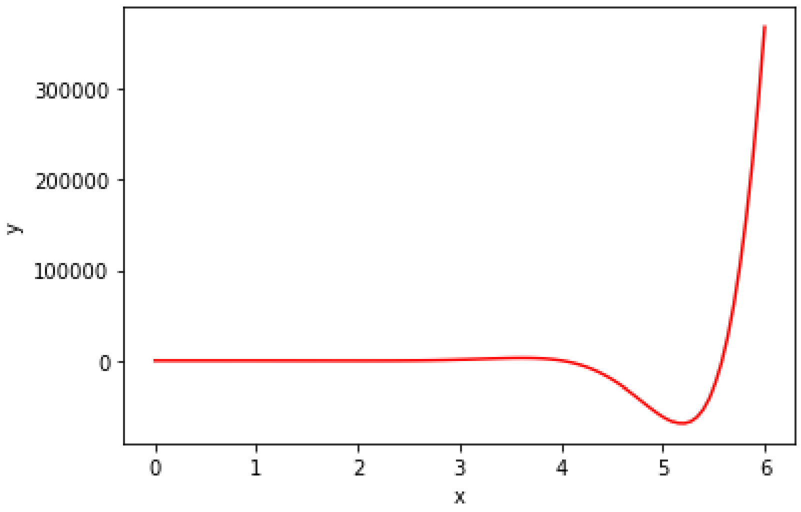
Out[16]:

$$y(t) = \left(C_1 \sin\left(2t\right) + C_2 \cos\left(2t\right)\right) e^{2t}$$

In [17]:
```python
def f(y,x):
    return (y[1], +4 * y[1] -8 * y[0] + 0)

y0 = [3,7]
xs = np.linspace(0,6,100)
sol = odeint(f, y0, xs)
ys = sol[:,0]

plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



EXERCISE 16.3

In [18]:
```
y = Function("y")
t = Symbol('t')
dy2 = 2 * Derivative(y(t), t,2)
dy1 = -12 * Derivative(y(t),t)
eq1 = Eq(dy2 + dy1 + 20 * y(t), 40)
eq1
```

Out[18]:

$$20y(t) - 12\frac{d}{dt}y(t) + 2\frac{d^2}{dt^2}y(t) = 40$$

In [19]:
```
sol2 = dsolve(eq1, y(t))
sol2
```

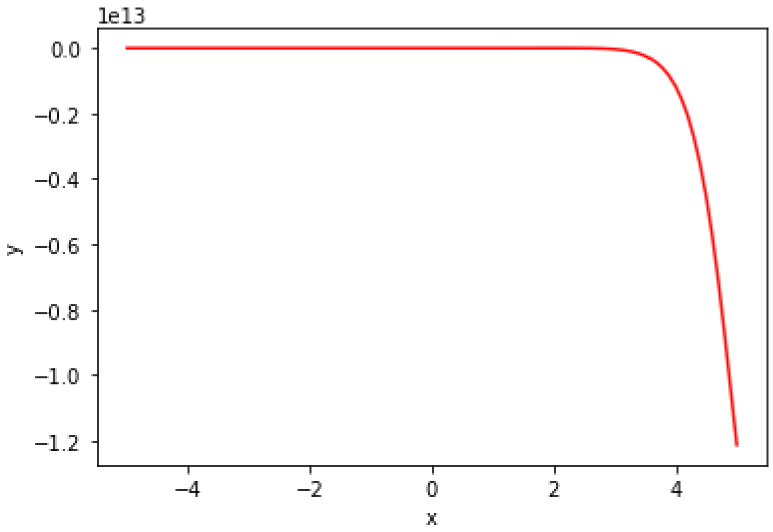
Out[19]:

$$y(t) = (C_1 \sin(t) + C_2 \cos(t))\, e^{3t} + 2$$

In [20]:
```
def f(y,x):
    return (y[1], + 6 * y[1] -10 * y[0] + 20)

y0 = [4,5]
xs = np.linspace(-5,5,100)
sol = odeint(f, y0, xs)
ys = sol[:,0]
```

```
plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



A Market Model with Price Expectations

Example 1

In [21]:
```
P = Function("P")
Q = Symbol('Q')
t = Symbol("t")
dy2 = Derivative(P(t), t,2)
dy1 = -4 * Derivative(P(t), t)
eq1 = Eq(dy2 + dy1 - 12 * P(t), -48)
eq1
```

Out[21]:

$$-12P(t) - 4\frac{d}{dt}P(t) + \frac{d^2}{dt^2}P(t) = -48$$

In [22]:
```
sol2 = dsolve(eq1, P(t))
sol2
```

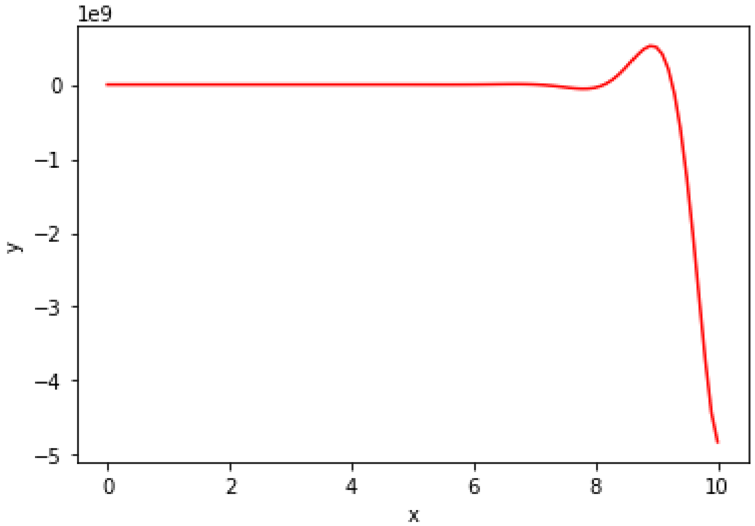
Out[22]:

$$P(t) = C_1 e^{-2t} + C_2 e^{6t} + 4$$

In [23]:
```python
def f(y,x):
    return (y[1], + 4*y[1] -12*y[0] - 48)

y0 = [6,4]
xs = np.linspace(0,10,100)
sol = odeint(f, y0, xs)
ys = sol[:,0]

plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



Example 2

In [24]:
```python
P = Function("P")
Q = Symbol('Q')
t = Symbol("t")
dy2 = Derivative(P(t), t,2)
dy1 = 2 * Derivative(P(t), t)
eq1 = Eq(dy2 + dy1 +5* P(t), 45)
eq1
```

Out[24]:

$$5P(t) + 2\frac{d}{dt}P(t) + \frac{d^2}{dt^2}P(t) = 45$$

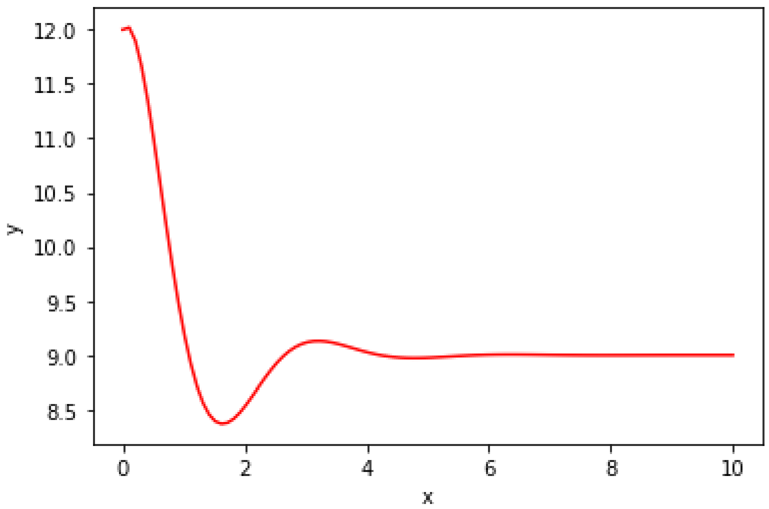
In [25]:
```
sol2 = dsolve(eq1, P(t))
sol2
```

Out[25]: $P(t) = (C_1 \sin(2t) + C_2 \cos(2t))\, e^{-t} + 9$

In [26]:
```
def f(y,x):
    return (y[1], -2*y[1] -5*y[0] + 45)

y0 = [12,1]
xs = np.linspace(0,10,100)
sol = odeint(f, y0, xs)
ys = sol[:,0]

plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



EXERCISE 16.4 Q/3

In [27]:
```
from sympy import symbols, Eq, solve
P = Function("P")
Q = Symbol('Q')
Q_d = Symbol("Q_d")
Q_s = Symbol("Q_s")
t = Symbol("t")
```

```
dy2 = 3 * Derivative(P(t), t,2)
dy1 = Derivative(P(t), t)
eq1 = Eq(dy2 + dy1 - P(t) + 9,Q_d)
display(eq1)


dy2_ = 5 * Derivative(P(t), t,2)
dy1_ = -Derivative(P(t), t)
eq2 = Eq(dy2_ + dy1_ +4* P(t) -1 ,Q_s)
display(eq2)
```

$$-P(t) + \frac{d}{dt}P(t) + 3\frac{d^2}{dt^2}P(t) + 9 = Q_d$$

$$4P(t) - \frac{d}{dt}P(t) + 5\frac{d^2}{dt^2}P(t) - 1 = Q_s$$

In [28]:

```
dy3 = 2 * Derivative(P(t), t,2)
dy2 = -2* Derivative(P(t), t)
eq3 = Eq(dy3 + dy2 +5* P(t),10)
display(eq3)
```

$$5P(t) - 2\frac{d}{dt}P(t) + 2\frac{d^2}{dt^2}P(t) = 10$$

In [29]:

```
eq1.lhs - eq2.lhs
```

Out[29]:

$$-5P(t) + 2\frac{d}{dt}P(t) - 2\frac{d^2}{dt^2}P(t) + 10$$

In [30]:

```
sol3 = dsolve(eq3, P(t))
sol3
```

Out[30]:

$$P(t) = \left(C_1 \sin\left(\frac{3t}{2}\right) + C_2 \cos\left(\frac{3t}{2}\right)\right) e^{\frac{t}{2}} + 2$$

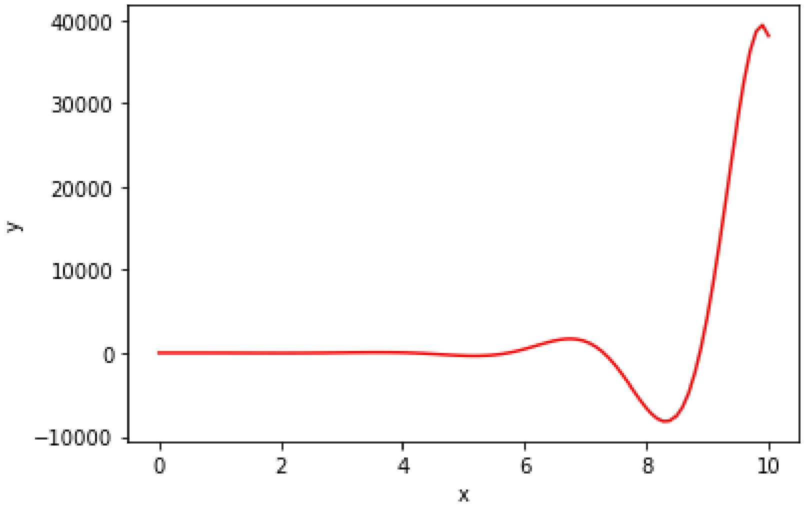
In [31]:

```
def f(y,x):
    return (y[1], +2*y[1] -5*y[0] + 10)

y0 = [4,4]
xs = np.linspace(0,10,100)
```

```
sol = odeint(f, y0, xs)
ys = sol[:,0]

plt.plot(xs, ys,"r")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



16.5 The Interaction of Inflation and Unemployment

In [32]:

```
P = Function("P")
pi = Function("\u03C0")
beta = Symbol("\\beta")
k = Symbol("k")
j = Symbol("j")
g = Symbol("g")
t = Symbol("t")
m = Symbol("m")
dy2 = Derivative(pi(t), t,2)
dy1 = (beta * k + j*(1-g)) * Derivative(pi(t), t)
eq4 = Eq(dy2 + dy1 - (j*beta*k)*pi(t) ,(j*beta*k*m))
eq4
```

Out[32]:

$$-\beta jk\pi(t) + (\beta k + j\,(1-g))\,\frac{d}{dt}\pi(t) + \frac{d^2}{dt^2}\pi(t) = \beta jkm$$

In [33]:
```
sol4 = dsolve(eq4, pi(t))
sol4
```

Out[33]:

$$\pi(t) = C_1 e^{\frac{t\left(-\beta k+gj-j-\sqrt{\beta^2k^2-2\beta gjk+6\beta jk+g^2j^2-2gj^2+j^2}\right)}{2}} + C_2 e^{\frac{t\left(-\beta k+gj-j+\sqrt{\beta^2k^2-2\beta gjk+6\beta jk+g^2j^2-2gj^2+j^2}\right)}{2}} - m$$

16.6 Differential Equations with a Variable Term

In [34]:
```
y = Function("y")
t = Symbol('t')
dy2 = Derivative(y(t), t,2)
dy1 = 5 * Derivative(y(t),t)
eq1 = Eq(dy2 + dy1 + 3 * y(t), 6*t**2 - t - 1)
eq1
```

Out[34]:

$$3y(t) + 5\frac{d}{dt}y(t) + \frac{d^2}{dt^2}y(t) = 6t^2 - t - 1$$

In [35]:
```
sol1 = dsolve(eq1, y(t))
sol1
```

Out[35]:

$$y(t) = C_1 e^{\frac{t\left(-5-\sqrt{13}\right)}{2}} + C_2 e^{\frac{t\left(-5+\sqrt{13}\right)}{2}} + 2t^2 - 7t + 10$$

Furkan Zengin