

Mathematics for Robotics and Control - Assignment 5: Jacobians

Solve Exercise 1 of Assignment 6 using any Iterative method.

In [23]:

You solution here

```
def update_position(distances):
    # Determine robot position
    # ...
    # <your code here>
    #position of sensors
    %matplotlib inline
    import matplotlib.pyplot as plt
    from matplotlib import pyplot as plt

    import sympy as sp
    import numpy as np

    sp.init_printing()
    %matplotlib inline
    x,y,z = sp.symbols('x,y,z')
    #Create array of sensors.
    sensor1 = np.array([0,0,5.3])
    sensor2 = np.array([6,0,5.3])
    sensor3 = np.array([6,4.5,5.3])
    sensor4 = np.array([0,4.5,5.3])

    fig = plt.figure()
    for i in range(len(distances)):

        radius = distances[i]
        #Take values of r1,r2,r3 and r4 from distances file.

        r1 = radius[0]
        r2 = radius[1]
        r3 = radius[2]
        r4 = radius[3]

        #create matrix of all three sensor, sensor1, sensor2
        #and sensor3 with given values.
        G = sp.Matrix([[x**2+y**2+z**2 - 10.6*z + 28.09 - r1**2],
                       [x**2+y**2+z**2 - 12*x-10.6*z + 64.09 - r2**2],
                       [x**2+y**2+z**2-12*x-9*y-10.6*z+84.34-r3**2]])

        #Create matrix of unknown variables.
        unknown = sp.Matrix([x,y,z])
        #Compute jacobian of unknown .
        jacobian = G.jacobian(unknown)

        #Take initial values of x, y and z to compute x0,y0, and z0
        x_initial = 6
        y_initial = 4.5
        z_initial = 5.3

        #Create equation of sphere for x0, y0 and z0

        x0 = sp.sqrt((r1**2 - (y_initial**2) - (z_initial**2)+
                      (10.6*z_initial) - 28.09))
        y0 = sp.sqrt((r2**2 - (x_initial**2) - (z_initial**2) +
                      (12*x_initial) + (10.6*z_initial) - 64.09))
        z0 = sp.sqrt((r3**2 - (x_initial**2) -(y_initial**2) +
                      (10.6*z_initial)+
                      (12*x_initial) + (9*y_initial) - 84.34))
```

```

#For loop to run the number of iteration to converge x, y and z
for i in range(20):

    G_new = G.subs({x:x0,y:y0,z:z0})
    jacobian_new = jacobian.subs({x:x0,y:y0,z:z0})

    x_old = sp.Matrix([x0,y0,z0])
    #Compute d with product of -(jacobian inverse and G)
    d = jacobian_new.inv() * (-G_new)
    #compute new value of x.
    x_new = x_old + d
    #Take x,y and z from x_new
    x0 = x_new[0]
    y0 = x_new[1]
    z0 = x_new[2]

#Plot converged x0 and y0
plt.scatter(x0,y0)

```

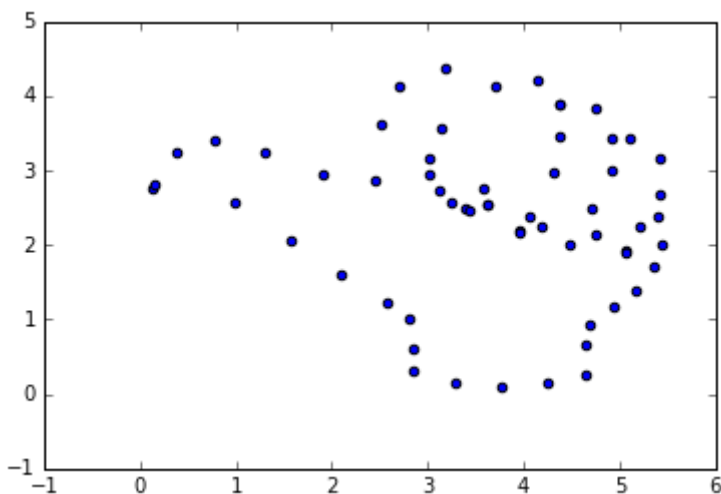
In [24]:

```

import sympy as sp
import numpy as np
distances = np.load("./distances.npy")

update_position(distances)

```



Mathematics for Robotics and Control - Assignment 6: Ordinary Differential Equations

For this assignment, you will be working with symbolic mathematics using the SymPy package for Python. In order to be able to work with the package efficiently, please familiarize yourself with SymPy by working through the tutorial:

[SymPy Tutorial \(http://docs.sympy.org/dev/tutorial/tutorial.en.html#tutorial\)](http://docs.sympy.org/dev/tutorial/tutorial.en.html#tutorial)

Also, make sure you know how SymPy differs from other CAS like Maple or Mathematica, which you may be familiar with. This information is detailed in the following document:

[SymPy Gotchas \(http://docs.sympy.org/dev/gotchas.html\)](http://docs.sympy.org/dev/gotchas.html)

There is also an FAQ:

[SymPy FAQ \(https://github.com/sympy/sympy/wiki/Faq\)](https://github.com/sympy/sympy/wiki/Faq)

Also, be sure to read the documentation/help on SymPy's ODE module. Simply evaluate the code in the next cell to obtain said documentation. Note that using `?` allows you to obtain help on any function, module etc. when used in IPython.

In [4]:

```
import sympy
?sympy.ode
```

Let us now start the assignment by examining some differential equations. First, have a look at the following quick example below to see how to enter code into the IPython notebook and how to work with SymPy.

In [2]:

```
# Enable pretty printing
sympy.init_printing(use_latex=True)
```

In [3]:

```
# Declare a symbolic variable x. Note that the x on the left side of the assignment
# while the "x" on the right hand side denotes the symbol name. It makes a lot of s
from sympy import *
x = Symbol("x")
# y is a function of x, i.e. y(x)
y = Function("y")(x)
# We use y_ to denote y'(x), i.e. the first derivative of y w.r.t. x
y_ = Derivative(y, x)
# ...an alternative way of writing this would be:
y_ = y.diff(x)

# Now let's examine a differential equation
#
#
#           x      d
# y(x) · sin(x) + e  = —(y(x))
#                      dx
#
eq1 = y * sin(x) + exp(x) - y_

# We now determine a solution for this differential equation
dsolve(eq1, y)
```

Out[3]:

$$\int (y(x) \sin(x) + e^x) e^{\cos(x)} dx = C_1$$

In []:

For your first assignment, solve the following three ODEs *by hand* and verify your results via SymPy. Please include the steps of your solutions in the input cell below using TeX syntax. You can find an introduction to mathematical expressions in LaTeX [here \(http://en.wikibooks.org/wiki/LaTeX/Mathematics\)](http://en.wikibooks.org/wiki/LaTeX/Mathematics). In order for the IPython notebook to evaluate your mathematical expressions, enclose them in \$, i.e.

$$\sum_{i=1}^{10} t_i$$

will display the following expression: $\sum_{i=1}^{10} t_i$

Assignment 7.1: Solve the following ordinary differential equations by hand and verify your results using SymPy.

Equation 1.1: $y' = 5 \cdot y$

$$y' = 5 \cdot y$$

Using separation of variables

$$\int \frac{1}{y} \cdot dy = 5 \cdot dt$$

Taking integration of above equation

$$\ln y = 5t + C$$

Multiplying exponential function on both sides.

$$y = e^{5t+C}$$

or it can be written as :

$$y = Ke^{5t}$$

Insert your solution for Equation 1.1 here

In [8]:

```
# Insert code to verify your solution for Equation 1.1 here and evaluate it
import sympy as sp
sp.init_printing()
t = sp.Symbol("t")
y = sp.Function("y")(t)
equation = sp.Eq((5*y), y.diff(t))
equation
sp.dsolve(equation)
```

Out[8]:

$$y(t) = C_1 e^{5t}$$

Equation 1.2: $\frac{dy}{dx} = -2 \cdot x \cdot y$

$$\frac{dy}{dx} = -2 \cdot x \cdot y$$

Using separation of variables

$$\int \frac{1}{y} \cdot dy = -2x \cdot dx$$

Taking integration of above equation

$$\ln y = -x^2 + C$$

Multiplying exponential function on both sides.

$$y = e^{-x^2+C}$$

or it can be written as :

$$y = Ke^{-x^2}$$

Insert your solution for Equation 1.2 here

In [6]:

```
# Insert code to verify your solution for Equation 1.2 here and evaluate it
import sympy as sp
sp.init_printing()
x = sp.Symbol("x")
y = sp.Function("y")(x)
equation = sp.Eq((-2*x*y), y.diff(x))
equation
sp.solve(equation)
```

Out[6]:

$$y(x) = C_1 e^{-x^2}$$

Equation 1.3: $\frac{dy}{dt} = y^2$

$$\frac{dy}{dt} = y^2$$

Using separation of variables

$$\int \frac{1}{y^2} \cdot dy = 1 \cdot dt$$

Taking integration of above equation

$$\frac{-1}{y} = t + C$$

separting y on one side.

$$y = \frac{-1}{t+C}$$

or it can be written as :

$$y = \frac{1}{C-t}$$

In [10]:

```
# Insert code to verify your solution for Equation 1.3 here and evaluate it
import sympy as sp
sp.init_printing()
t = sp.Symbol("t")
y = sp.Function("y")(t)
equation = sp.Eq((y**2), y.diff(t))
equation
sp.dsolve(equation)
```

Out[10]:

$$y(t) = \frac{1}{C_1 - t}$$

Assignment 7.2: Determine if the following equations are linear and insert your solutions below.

Note: simply stating that an equation is linear or not is not sufficient, please provide arguments/proof as to why this is the case.

1. $y' = \sin(x) \cdot y + e^x$
2. $y' + x \cdot y = e^x \cdot y$
3. $y' + \frac{x}{y} = 0$
4. $x \cdot y' + y = \sqrt{y}$

#Conditions for linearity are :

- *i. The dependent variable y and its derivatives are of first degree; the power of each y is 1.
- *ii. Each coefficient depends only on the independent variable x .

or if equation is represented according to standard linear equation i.e.

$$\frac{dy}{dx} + P(x) \cdot y = Q(x)$$

*1. $y' = \sin(x) \cdot y + e^x$

This equation is linear, since it can be represented by standard by standard linear equation. Furthermore, power of each y is 1 and each co-efficient depends only on x .

*2. $y' + x \cdot y = e^x \cdot y$

This equation is also linear because power of each y is 1 and each co-efficient depends only on independent variable x .

*3. $y' + \frac{x}{y} = 0$

This equation is non-linear because power of y is -1.

*4. $x \cdot y' + y = \sqrt{y}$

This equation is non-linear, because y contains square root which shows power is not 1.

Assignment 7.3: Solve the following ordinary differential equations and specify an integrating factor. Insert your solutions below.

Wikipedia: Integrating Factor (http://en.wikipedia.org/wiki/Integrating_factor)

Tutorial on how to use the integrating factor method to solve ODEs

(<http://www.cse.salford.ac.uk/profiles/gsmcdonald/H-Tutorials/ordinary-differential-equations-integrating-factor.pdf>) (PDF)

$$1. y' = -3 \cdot y + 6$$

$$2. y' + \frac{4}{x} \cdot y = x^4$$

1. Integral of $y' = -3 \cdot y + 6$:

$$y' + 3 \cdot y = 6 \dots \text{eq 1}$$

$$\text{Integrating factor} = I = e^{\int 3 \cdot dx}$$

$$\text{Then, taking integration : } \int 3 \cdot dx = 3 \cdot x$$

$$\text{Now, substituting into integration factor } I = e^{3 \cdot x}$$

Now, Multiplying integrating factor with original equation (eq1) :

$$e^{3x} y' + 3e^{3x} \cdot y = 6e^{3x}$$

Taking integration of above equation :

$$e^{3x} y = 2e^{3x} + C$$

So, general equation is :

$$y = \frac{2e^{3x} + C}{e^{3x}}$$

$$2. y' + \frac{4}{x} \cdot y = x^4:$$

$$y' + \frac{4}{x} \cdot y = x^4 \dots \text{eq 1}$$

$$\text{Integrating factor} = I = e^{\int \frac{4}{x} \cdot dx}$$

$$\text{Then, taking integration : } \int \frac{4}{x} \cdot dx = 4 \cdot \ln x = \ln x^4$$

$$\text{Now, substituting into integration factor } I = e^{\ln x^4} = x^4$$

Now, Multiplying integrating factor with original equation (eq1) :

$$x^4 y' + x^4 \frac{4}{x} \cdot y = x^8$$

Taking integration of above equation :

$$x^4 y = \frac{x^9}{9} + C$$

So, general equation is :

$$y = \frac{x^5}{9} + \frac{C}{x^4}$$

Assignment 7.4: Solve the following differential equations

Remember that a second order differential equation $y'' + a_1 \cdot y' + a_0 \cdot y = 0$ with constants $a_{0,1}$ corresponds to the characteristic equation $\lambda^2 + a_1 \cdot \lambda + a_0 = 0$ which can be factored into $(\lambda - \lambda_1) \cdot (\lambda - \lambda_2) = 0$. This is useful since the general solution of $y'' + a_1 \cdot y' + a_0 \cdot y = 0$ can be obtained directly by determining the roots λ_1 and λ_2 if the coefficients are constant and the differential equation is linear. Determine the roots of the following differential equations and specify their solutions. See the Wikipedia article on the [characteristic equation](http://en.wikipedia.org/wiki/Characteristic_equation_(calculus)) ([http://en.wikipedia.org/wiki/Characteristic_equation_\(calculus\)](http://en.wikipedia.org/wiki/Characteristic_equation_(calculus))) for the three distinctive cases that determine how to obtain the solution based on what the roots are.

Insert your solutions below and include code to proof your solutions are correct. Hint: use the `sympy.ode.checkodesol` function to validate your results.

1. $y'' - y' - 2 \cdot y = 0$
2. $y'' - 3 \cdot y' + 4 \cdot y = 0$
3. $y'' + 4 \cdot y = 0$

$$1. y'' - y' - 2 \cdot y = 0$$

The characteristics equation of above equation can be written as :

$$\lambda^2 - \lambda - 2 = 0$$

Taking factors :

$$\lambda^2 - 2 \cdot \lambda + \lambda - 2 = 0$$

Taking common :

$$\lambda(\lambda - 2) + 1(\lambda - 2) = 0$$

So, roots are :

$$\lambda = -1, 2$$

Since roots are real and unequal so general equation is :

$$y = c_1 e^{\lambda_1 \cdot x} + c_2 e^{\lambda_2 \cdot x}$$

Substituting λ in general equation we get :

$$y = c_1 e^{-x} + c_2 e^{-x}$$

In [42]:

```
# Include code here to show that your solutions are correct.
from sympy import *
sp.init_printing()

x,c1,c2 = symbols("x,c1,c2")
y = sp.Function("y")(x)
solution = Eq(c1*exp((-x))+c2*exp(-x)-y)
solution
checkodesol((y.diff(x).diff(x)-y.diff(x) - 2*y), solution)
```

Out[42]:

(True, 0)

$$*2. y'' - 3 \cdot y' + 4 \cdot y = 0$$

The characteristics equation of above equation can be written as :

$$\lambda^2 - 3\lambda - 4 = 0$$

Taking factors by using Quadratic formula:

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

substituting values of a = 1, b = -3, and c = 4 in formula

$$\lambda = \frac{3 \pm \sqrt{3^2 - 4 \cdot 1 \cdot 4}}{2}$$

by solving further .

$$\lambda = \frac{3 \pm \sqrt{7}i}{2}$$

So, roots are :

$$\lambda = \frac{3 + \sqrt{7}i}{2}, \frac{3 - \sqrt{7}i}{2}$$

Since roots are complex, alpha is real part and Beta is imaginary part so general equation is :

$$y = e^{\alpha x} (c_1 \cos \beta x + c_2 \sin \beta x)$$

Substituting λ in general equation we get :

$$y = e^{\frac{3}{2}x} (c_1 \cos \frac{\sqrt{7}i}{2}x + c_2 \sin \frac{\sqrt{7}i}{2}x)$$

In [67]:

```
# Include code here to show that your solutions are correct.
from sympy import *
sp.init_printing()

x,c1,c2 = symbols("x,c1,c2")
y = sp.Function("y")(x)
#solution = Eq(c1*exp((-x))+c2*exp(-x)-y)

solution = sp.Eq(exp((3*x/2))*(c1*sp.cos((sp.sqrt(7)/2)*x)
                        + c2*sp.sin((sp.sqrt(7)/2)*x))-y)

checkodesol((y.diff(x).diff(x) - 3*(y.diff(x)) + 4*y),solution)
```

Out[67]:

(True, 0)

$$*3. y'' + 4 \cdot y = 0$$

The characteristics equation of above equation can be written as :

$$\lambda^2 + 4 = 0$$

Taking factors:

$$\lambda = \pm 2i$$

Since roots are complex, alpha is real part and Beta is imaginary part so general equation is :

$$y = e^{\alpha x}(c_1 \cos \beta x + c_2 \sin \beta x)$$

Substituting λ in general equation we get :

$$y = (c_1 \cos 2x + c_2 \sin 2x)$$

In [70]:

```
from sympy import *
sp.init_printing()

x,c1,c2 = symbols("x,c1,c2")
y = sp.Function("y")(x)
#solution = Eq(c1*exp((-x))+c2*exp(-x)-y)

solution = sp.Eq((c1*sp.cos(2*x) + c2*sp.sin(2*x))-y)

checkodesol((y.diff(x).diff(x)+ 4*y),solution)
```

Out[70]:

(True, 0)

In []:

