MicroProjectOne

March 12, 2019

1 Microproject 1

1.1 python code

1.1.1 import libraries

```
In [12]: import math
         import matplotlib.pyplot as plt
         import numpy as np
         from scipy.linalg import expm
         from scipy.interpolate import BSpline, make_interp_spline
```

1.1.2 create input data and settings

```
In [13]: #project settings
         #first point
         ybegin=0
         #how many points
         ycount = 10
         #step per point
         unitPrecision = 0.1
         #how many point generated per exising point (1 = raw values)
         smoothenFactor = 5
```

1.1.3 create input data

```
In [14]: #set matrix details and matrix infos
         #matrixCols = 3
         #matrixRows = 3
         #matrix = np.array([[-3, 3, -3],[5, -5, 5], [-9, 9, -9]])
         matrixCols = 2
         matrixRows = 2
         matrix = np.array([[-3, 3],[5, -5]])
```

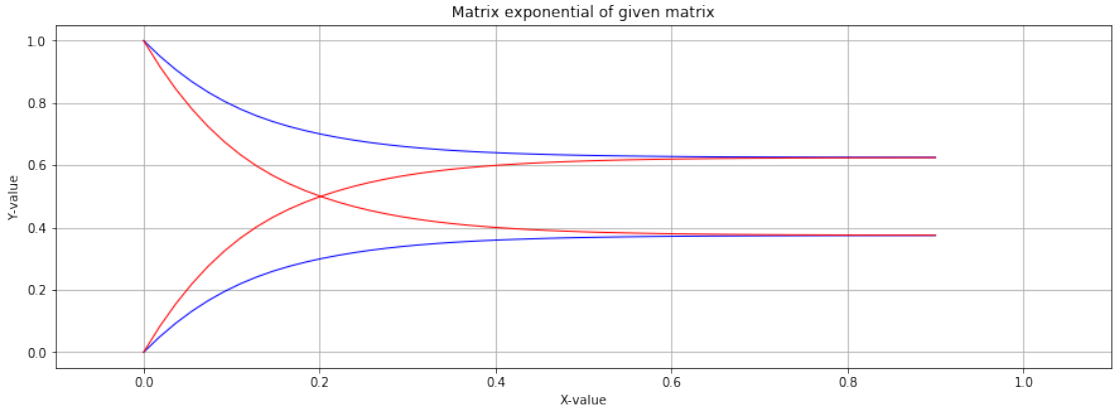
1.1.4 create data

```
In [15]: #extract data and store in data array
         data = [[] for _ in range(matrixCols*matrixRows)]
```

```
        for i in range(ybegin, ybegin+ycount):
            matrixi = expm(i*unitPrecision * matrix)
            for y in range(matrixRows):
                for x in range(matrixCols):
                    data[y*matrixCols + x].append(matrixi[y][x])
```

1.2 creating graph

In [17]:
```
        #creatin x values with smoothenfactor
        abscise = [i*unitPrecision for i in range(ybegin, ybegin+ycount)]
        abscise_smooth = np.linspace(abscise[0], abscise[ycount-1], ycount*smoothenFactor)

        #adding data lines after being smoothed
        colorArr=["blue", "red", "green", "purple" "brown", "yellow"]
        for i in range(matrixCols*matrixRows):
            #color=colorArr[i%matrixCols]
            color=colorArr[i//matrixCols]
            bsplineObj = make_interp_spline(abscise, data[i], k=3)
            data_smooth = bsplineObj(abscise_smooth)
            plt.plot(abscise_smooth, data_smooth, linewidth=1, color=color)
            #plt.plot(abscise, data[i], linewidth=1, color=color)

        #graph settigns
        plt.title('Matrix exponential of given matrix')
        plt.ylabel('Y-value')
        plt.xlabel('X-value')

        plt.grid(True)
        plt.xlim(-1*unitPrecision,(ycount*unitPrecision)+1*unitPrecision)
        #plt.ylim(0.0,17.0)
        #plt.axis([-1, 10, -1, 17.0])

        #attempt to stretch the graph so it is easier to read
        #(if graph small, re-execute this cell)
        fig_size = plt.rcParams["figure.figsize"]
        fig_size[0] = 15
        fig_size[1] = 5
        plt.rcParams["figure.figsize"] = fig_size

        #shows the graph
        plt.show()
```



/newpage ## R code

In [3]:
```
library(Matrix)
library(expm)
```

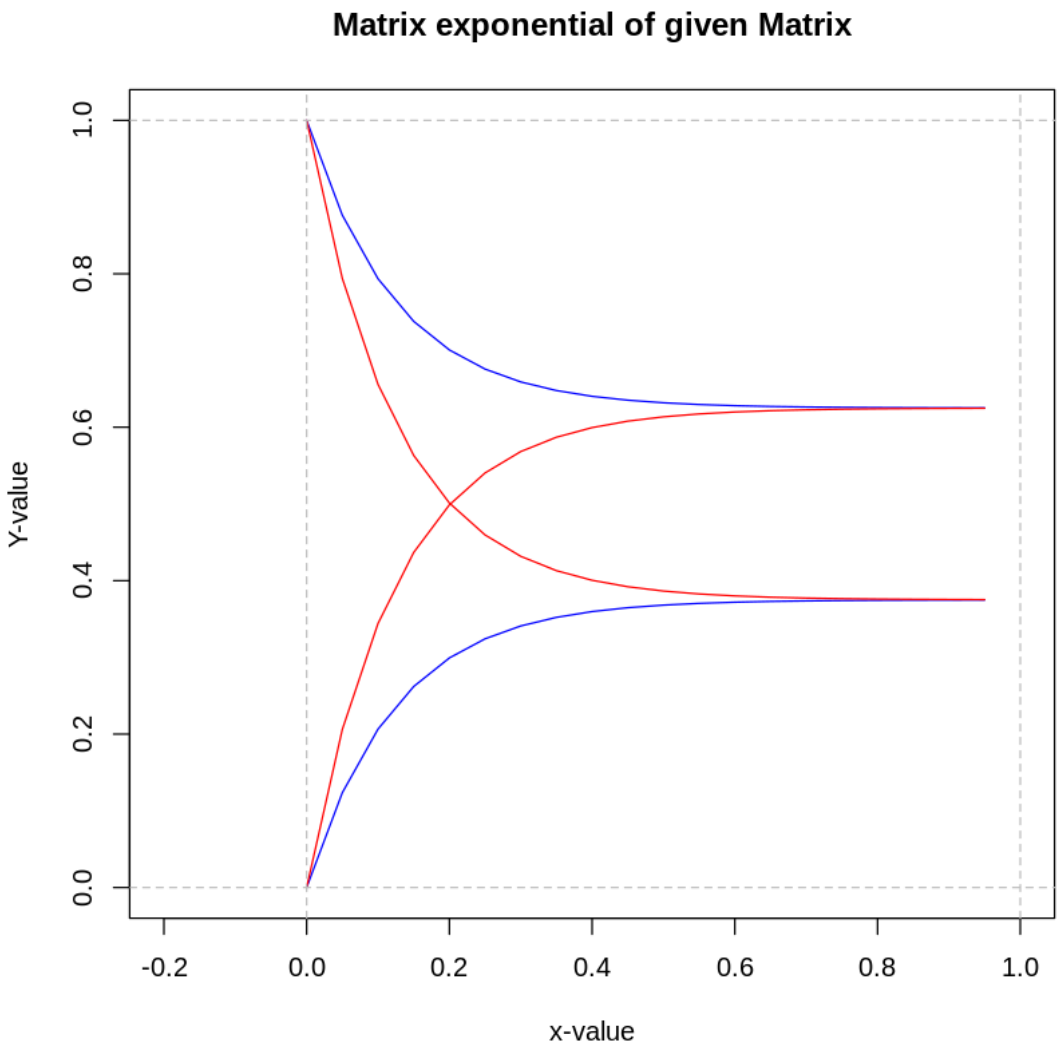
In [4]:
```
#how to make it work:
#install.packages("expm", "Matrix")
#cant write on conda library thus create personal library
#wait till installation completed and compy path to personal library
#paste personal library in lib.loc below
#library(expm, lib.loc = "/home/dominique/R/x86_64-pc-linux-gnu-library/3.4")
```

In [5]:
```
nrow <- 2
ncol <- 2
mdat <- matrix(c(-3, 5, 3, -5), nrow = nrow, ncol = ncol)
```

In [6]:
```
start <- 0
count <- 20
step <- 0.05

data <- list()
for(i in 1:(nrow*ncol))
{
    data[[i]] <- vector(mode="numeric", length=count)
}

for(i in start:(start+count-1))
{
    yindex <- i - start + 1
    xindex <- 0
    tempMatrix = Matrix::expm(mdat*(step*i))
    for(y in 1:nrow)
    {
        for(x in 1:ncol)
```

```
            {
                xindex <- xindex + 1
                data[[xindex]][yindex] <- tempMatrix[y,x]
            }
        }
    }
```

In [7]:
```
colors <- c("blue","red")
x <- seq(0,count*step - step, step)
plot(x, data[[1]], type ="l", main = "Matrix exponential of given Matrix",
     col = colors[1],
     xlab = "x-value", ylab = "Y-value",
     xlim = c(-0.2, count*step), ylim = c(0, 1))
for(i in (2:(nrow*ncol)))
    lines(x, data[[i]], col=colors[i%/%(ncol+1)+1])
abline(v = 0:130,  lty = 2, col = "grey")
abline(h = 0:3,  lty = 2, col = "grey")
options(repr.plot.width=14, repr.plot.height=9)
```


Matrix exponential of given Matrix
Y-value
x-value
0.0
0.2
0.4
0.6
0.8
1.0
-0.2
0.0
0.2
0.4
0.6
0.8
1.0