YOU_a4q5

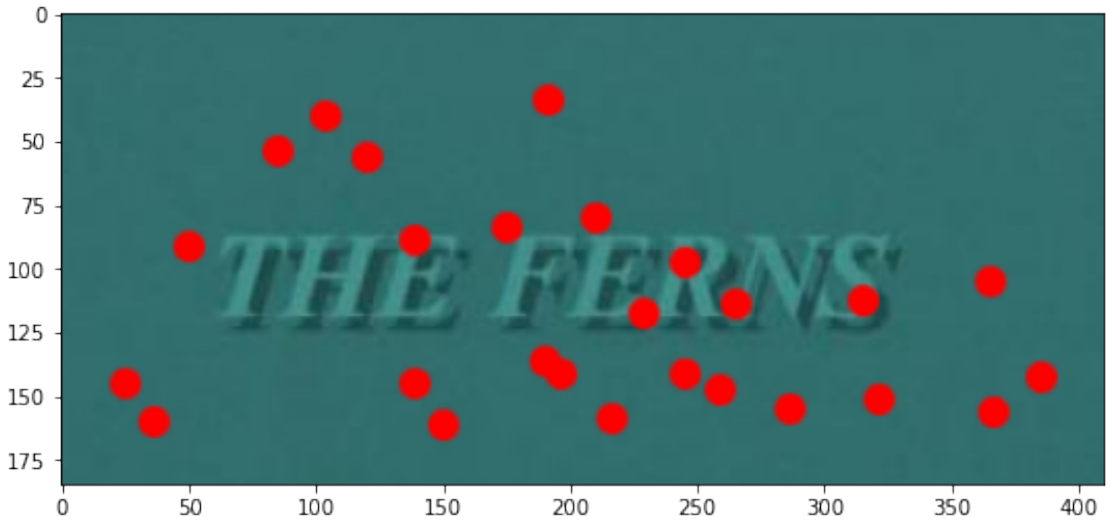
November 17, 2021

1 A4-Q5: Parametric Spline

[3]:
```
import numpy as np
from scipy.interpolate import make_interp_spline
import matplotlib.pyplot as plt
```

1.1 (a) Write your nickname and display it

[4]:
```
# Display nickname image
f = plt.imread('alex.png')
plt.figure(figsize=(9,5)); plt.imshow(f);
```



1.2 (b) Hardcode interpolation points

[46]:
```
# === YOUR CODE HERE ===
points = [[],[],[]]
points[0] = [
(29,140 ),
```

```
(40,153 ),
(84,56  ),
(102,43 ),
(117,58 ),
(134,140),
(144,154),
(181,131),
(199,80 ),
(182,37 ),
(167,83 ),
(187,136),
(205,152),
(231,136),
(250,111),
(231,96 ),
(216,114),
(244,142),
(269,149),
(296,110),
(343,150),
(360,137)
]
points[1] = [(52,90),(134,88)]
points[2] = [(301,145),(341,103)]
xlists = [[],[],[]]
ylists = [[],[],[]]
for i in range(3):
    for p in points[i]:
        xlists[i].append(p[0])
        ylists[i].append(-p[1])
```

1.3 (c) ParametricSpline

[50]:
```
def ParametricSpline(Sx,Sy):
    '''
     x_cs, y_cs, t = ParametricSpline(Sx,Sy)

       Takes an array of x- and y-values, and returns a parametric
       cubic spline in the form of two piecewise-cubic data structures
       (one for the x-component and one for the y-component), as well as
       the corresponding parameter values.

       The splines use natural boundary conditions.

       Input:
        Sx   array of x-values
        Sy   array of y-values
```

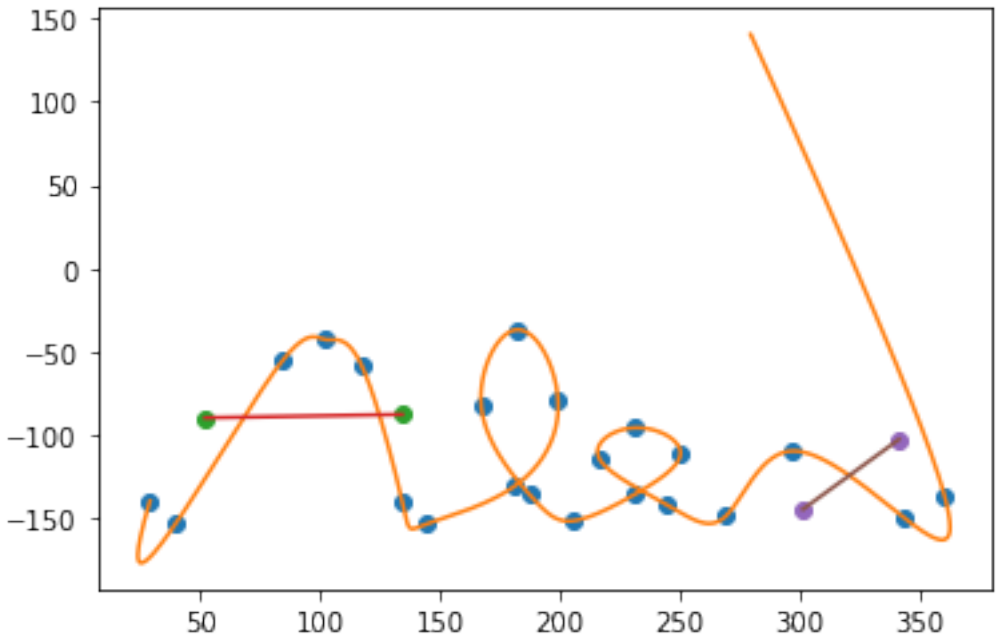
```
    Output:
     x_cs function that evaluates the cubic spline for x-component
     y_cs function that evaluates the cubic spline for y-component
     t is the array of parameter values use for the splines

    Note that x_cs(t) and y_cs(t) give Sx and Sy, respectively.
'''

# These lines are just placeholders... replace them
t = []
for i in range(len(Sx)):
    t.append(i)

spl = make_interp_spline(t, np.c_[Sx, Sy])
def x_cs(k):
    return spl(k).T
def y_cs(k):
    return spl(k).T

return x_cs, y_cs, t
```

1.4 (d) Find parametric splines for each segment

[51]:
```
# === YOUR CODE HERE ===
tt = np.linspace(0, 22, 1000)
x_cs, y_cs, t = ParametricSpline(xlists[0],ylists[0])
xx, yy = x_cs(tt)
```

1.5 (e) Plot the segments

[52]:
```
# === YOUR CODE HERE ===
plt.plot(xlists[0], ylists[0], 'o')
plt.plot(xx,yy,'-')
for i in range(1,3):
    plt.plot(xlists[i],ylists[i],'o')
    plt.plot(xlists[i],ylists[i],'-')
```


150
100
50
0
−50
−100
−150
50
100
150
200
250
300
350

[ ]:

[ ]: