NSAE: Classroom problem 1.1

Miquel Borrell

This document aims to solve the first Problem of the course. The statement is as follows:

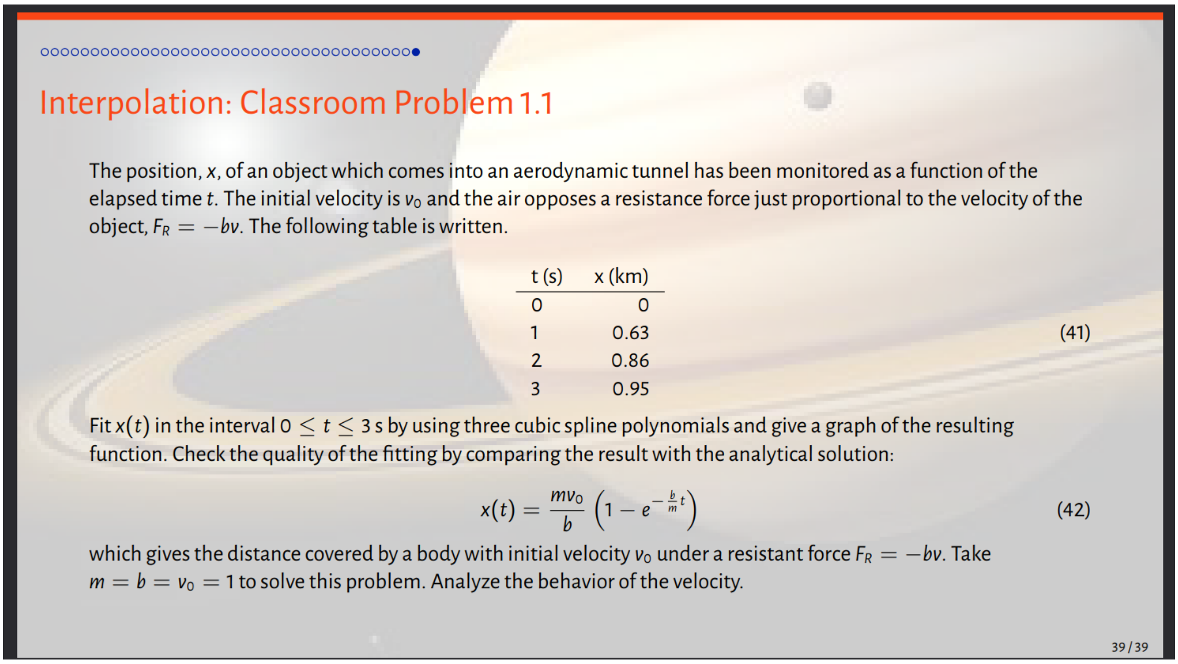

Figure 1: Problem NSAE 1.1

First we will discuss the analytical solution and it will be followed by an coded solution.

1 Analytical solution

Let's take a look to the data on a graph (Figure 2):

Following the procedure explained in class we can calculate the polynomial between t_0 and t_1 using expression 29 and 35. In this way we can find the coefficients of $p_{0,1}$:

$$P_{j,j+1}(t) = f_j + \left[\frac{f_{j+1} - f_j}{h_j} - \frac{h_j p''_{j+1}}{6} - \frac{h_j p''_j}{3}\right](t - t_j) + \frac{p''_j}{2}(t - t_j)^2 + \frac{p''_{j+1} - p''_j}{6h_j}(t - t_j)^3 \quad (1)$$

In our case $p_{0,1}$ would be calculated as:

$$p_{0,1} = 0 + \left[\frac{0.63 - 0}{1} - \frac{1 * p''_1}{6} - \frac{1 * p''_0}{3}\right](t - 0) + \frac{p''_0}{2}(t - 0)^2 + \frac{p''_1 - p''_0}{6 * 1}(t - 0)^3 \quad (2)$$

At this point we realize we need to calculate p''_0 and p''_1. But as we are working with natural splines $p''_0 = 0$. So in order to get p''_1 we will make use of expression 35 in the bibliography which states that:

$$2(h_0 + h_1)p''_1 + h_1 p''_2 = \frac{6(x_2 - x_1)}{h_1} - \frac{6(x_1 - x_0)}{h_0} \quad (3)$$

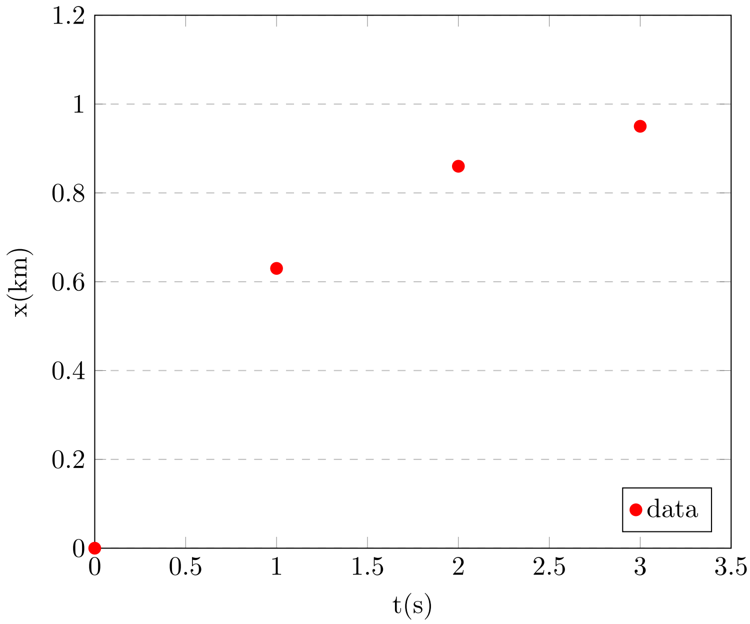

Figure 2: Data visualization

$$h_1 p_1'' + 2(h_1 + h_2)p_2'' + h_3 p_3'' = \frac{6(x_3 - x_2)}{h_2} - \frac{x_2 - x_1}{h_1} \tag{4}$$

Where $h_j = (t_{j+1} - t_j)$ in this case $h_0 = h_1 = h_2 = 1$. Resolving we have:

$$2(2)p_1'' + p_2'' = 6(0.86 - 0.63) - 6(0.63 - 0) \tag{5}$$

$$p_1'' + 4p_2'' + p_3'' = 6(0.95 - 0.86) - (0.86 - 0.63) \tag{6}$$

It results in:

$$p_1'' = -0.584$$

$$p_2'' = -0.064$$

This coupled with the natural splines condition means that we can now fully resolve equation 2 and discover the best fitting cubic spline between t_0 and t_1:

$$p_{0,1} = 0.7273t - 0.0973t^3$$

with coefficients being:

$$a_0 = 0.097\overline{3}$$

$$b_0 = 0$$

$$c_0 = 0.0727\overline{3}$$

$$d_0 = 0$$

Now let's look at the output of this polynomial in our case (Figure 3):

We can observe that the cubic spline fits perfectly between the interval $0 <= t <= 1$.

Now we could repeat the process for the other points until we have a set of 3 polynomials for each data intervals. However we could also automate this using Python and the package Scipy which provides a handy way of computing all the coefficients and providing the polynomials. We could follow the code below:

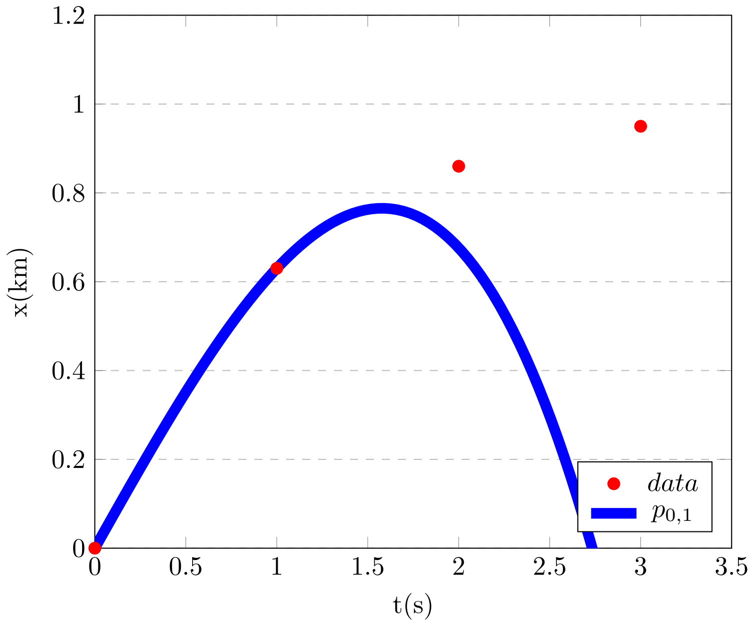

Figure 3: Data visualization

```
import numpy as np
import math
from scipy import interpolate
t=np.linspace(0,3,4,dtype=int)
x=np.array([0,0.63,0.86,0.95])

# calculate Cubic Spline using Scipy method
cs=interpolate.CubicSpline(t,xd,bc_type='natural')

#Print cs coeficients to construct polynomials (.T to transpose for easy visualitzation)
print(cs.c.T)
```

The output of this code will produce is something like this:

```
array([[-9.73333333e-02, -1.11022302e-16,  7.27333333e-01,
         0.00000000e+00],
       [ 8.66666667e-02, -2.92000000e-01,  4.35333333e-01,
         6.30000000e-01],
       [ 1.06666667e-02, -3.20000000e-02,  1.11333333e-01,
         8.60000000e-01]])
```

Which in turn can be interpreted as the following table:

	a	b	c	d
$p_{0,1}$	-9.73e-02	-1.11e-16	7.27e-01	0.00e+00
$p_{1,2}$	8.66e-02	-2.92e-01	4.35e-01	6.30e-01
$p_{2,3}$	1.06e-02	-3.20e-02	1.11e-01	8.60e-01

Which translates to the following cubic splines:

$$p_{0,1} = 0.7273t - 0.0973t^3$$

$$p_{1,2} = 0.0866(t-1)^3 - 0.292(t-1)^2 + 0.435(t-1) + 0.630$$

$$p_{2,3} = 0.0106(t-2)^3 - 0.032(t-2)^2 + 0.111(t-2) + 0.860$$

Visually it would be translated to the following graph (Figure 4):

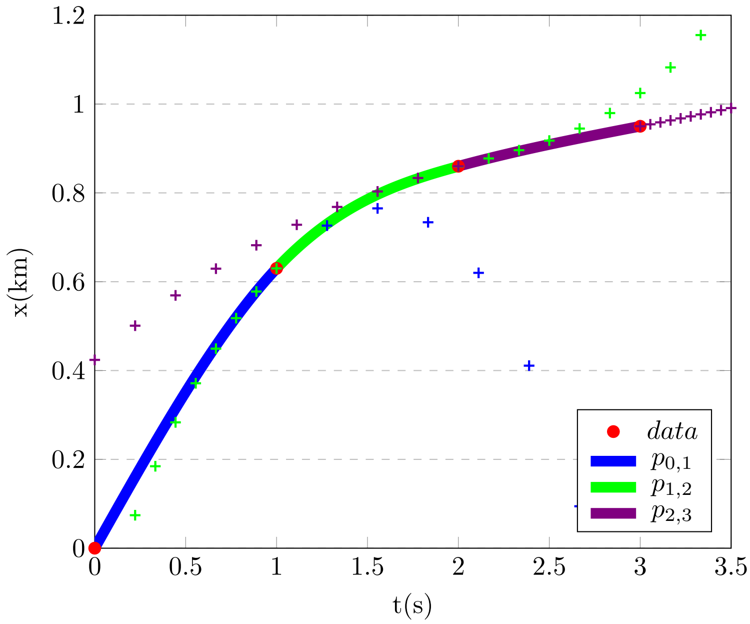

Figure 4: Representation of the splines

Furthermore we can compare the result with the analytical solution to grasp a sense of the error committed by this approximation as we can see in the figure 5:

Going one step further we can also take a look to the derivatives of our splines in figure 6

We can observe the continuity of the derivative up to the third derivation.

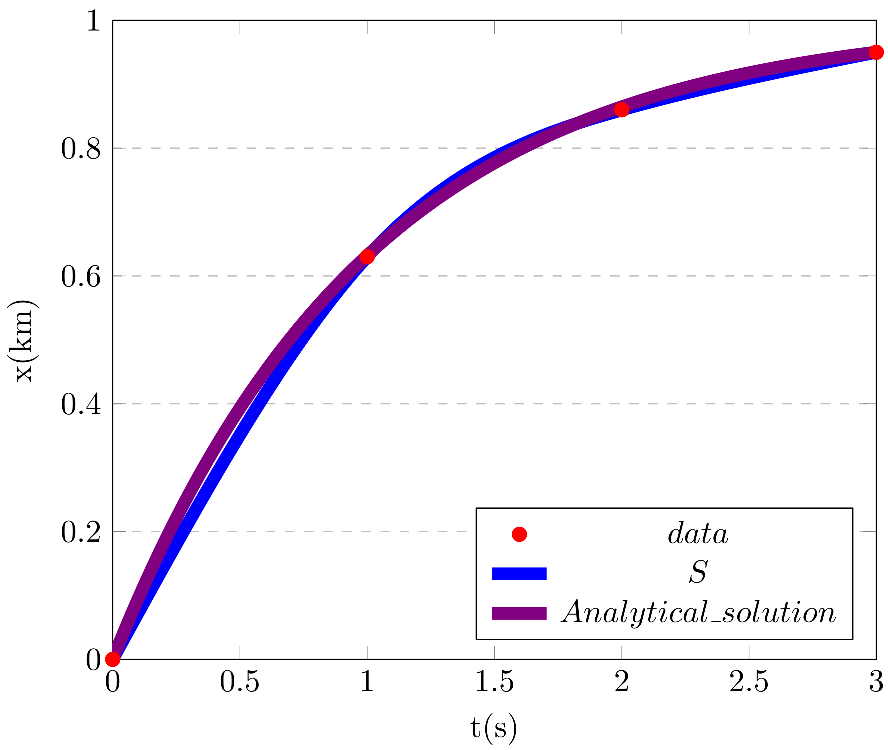

Figure 5: Comparison between analytical and approximation solutions

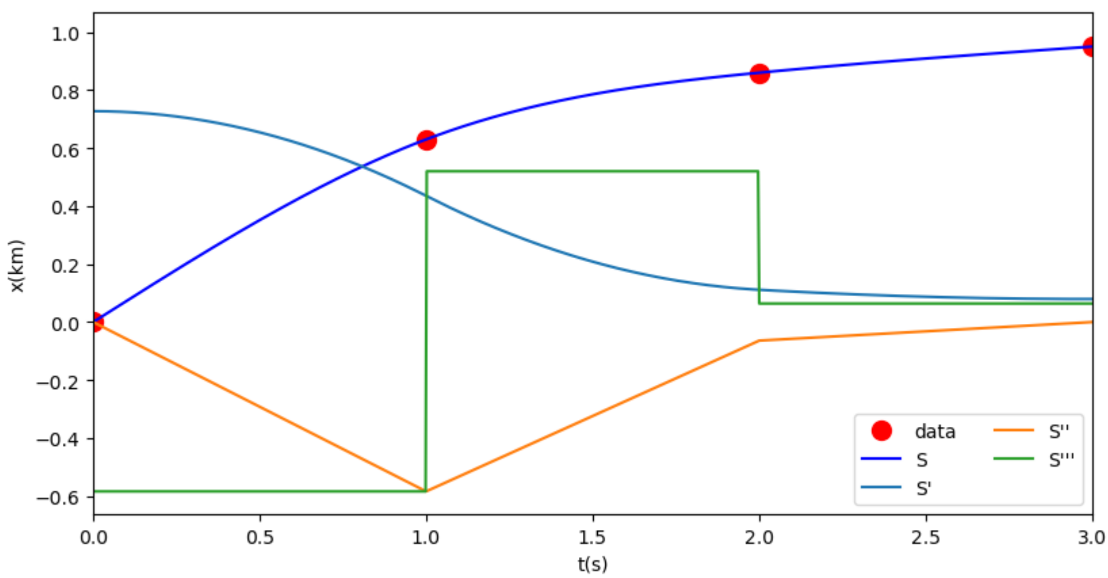

Figure 6: Spline derivatives