CAGD - Homework 1

Josefine Stål & Erik Ackzell

September 13, 2016

Task 1

In this task we want to find the affine map $\Phi : \mathbb{E}^2 \to \mathbb{E}^2$ such that

$$(0,0) \overset{\Phi}{\mapsto} (1,0)$$
$$(1,0) \overset{\Phi}{\mapsto} (1,-1)$$
$$(1,1) \overset{\Phi}{\mapsto} (3,-3)$$
$$(0,1) \overset{\Phi}{\mapsto} (3,-2)$$

We look for Φ on the form $\Phi(x) = Ax + v$, $A \in \mathbb{R}^{2\times 2}$, $v \in \mathbb{R}^2$. To determine A and v we solve a series of equations.
Set

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

We now solve a series of equations to determine $a_{11}, a_{12}, a_{21}, a_{22}, v_1, v_2$.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow v_1 = 1 \tag{1}$$

Insert $v_1 = 1$ and continue

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow \begin{cases} a_{11} = 0 \\ a_{21} = -1 - v_2 \end{cases} \tag{2}$$

Insert $a_{11} = 0$, $a_{21} = -1 - v_2$ and continue

$$\begin{bmatrix} 0 & a_{12} \\ -1 - v_2 & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \end{bmatrix} \Rightarrow \begin{cases} a_{12} = 2 \\ a_{22} = -2 \end{cases} \tag{3}$$

Insert $a_{12} = 2$, $a_{22} = -2$ and continue

$$\begin{bmatrix} 0 & 2 \\ -1 - v_2 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix} \Rightarrow v_2 = 0 \tag{4}$$

Thus,

$$(1), (2), (3), (4) \Rightarrow A = \begin{bmatrix} 0 & 2 \\ -1 & -2 \end{bmatrix}, \quad v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

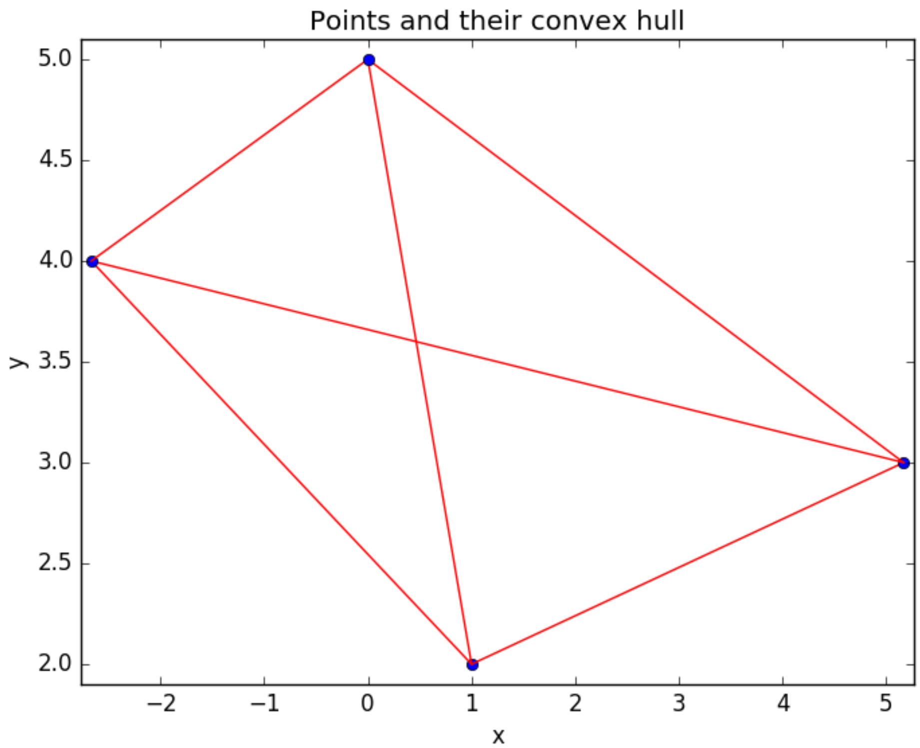
and we are done. $\square$

Task 2

By connecting all the points in a given set of points in the plane, the convex hull of a set points can be shown. In the figure below, the convex hull of the set of points

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \begin{pmatrix} \frac{31}{6} \\ 3 \end{pmatrix} \quad \begin{pmatrix} -\frac{8}{3} \\ 4 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

can be seen. The convex hull consists of all the points contained within the outermost lines.



Task 3

We want to show that linear interpolation $\Phi : \mathbb{R} \to \mathbb{E}^2$ is an affine map.
Proof: Let p, q be arbitrary points in $\mathbb{E}^2$. Then

$$\Phi(t) = (1-t)p + tq, \quad t \in \mathbb{R}$$

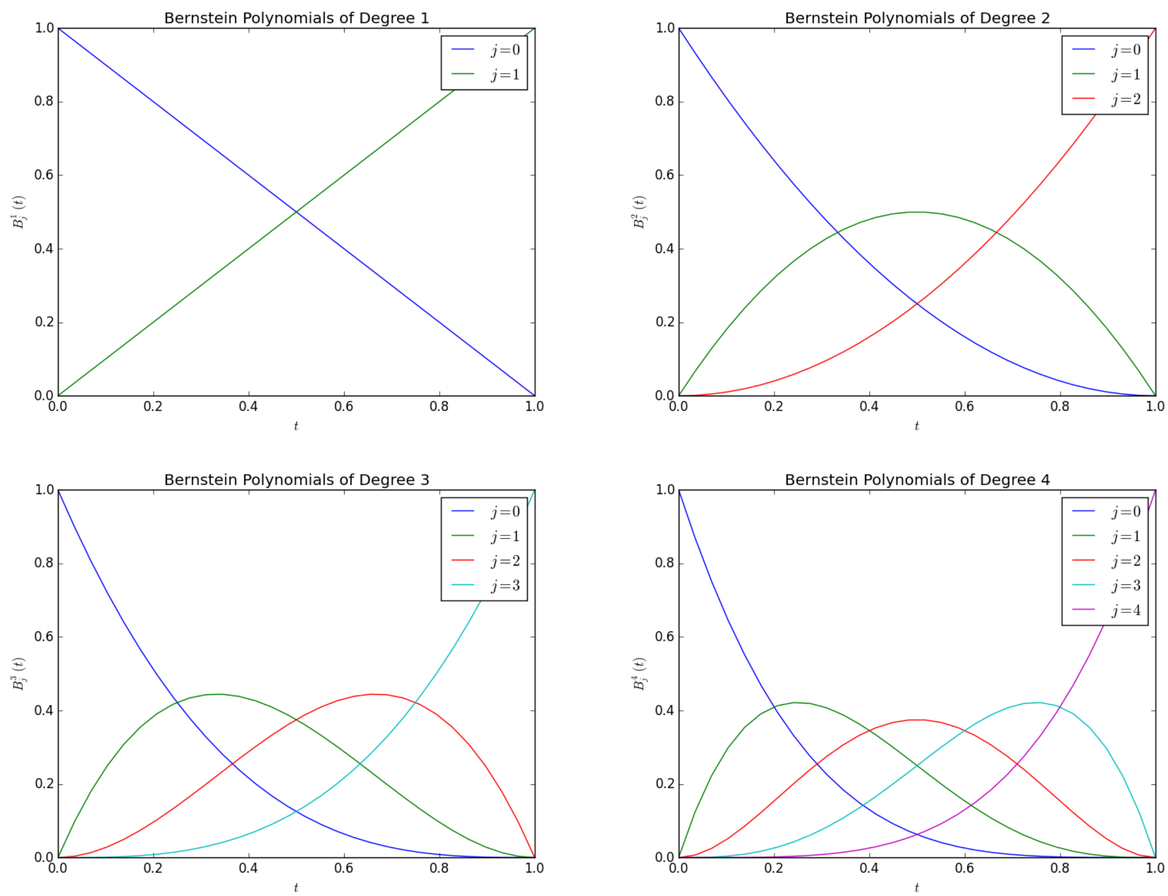
is the linear interpolation of p and q. Let $x \in \mathbb{R}$ be a barycentric combination, i.e. $x = \sum_{i=1}^n \alpha_i x_i$ with $x_i, \alpha_i \in \mathbb{R}$ and $\sum_{i=1}^n \alpha_i = 1$. Then

$$\begin{aligned}
\Phi(x) &= \Phi(\sum_{i=1}^n \alpha_i x_i) \\
&= (1 - \sum_{i=1}^n \alpha_i x_i)p + (\sum_{i=1}^n \alpha_i x_i)p \\
&\overset{(i)}{=} (\sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i x_i)p + \sum_{i=1}^n \alpha_i x_i p \\
&= \sum_{i=1}^n \alpha_i[(1-x_i)p + x_i q] \\
&= \sum_{i=1}^n \alpha_i \Phi(x_i)
\end{aligned}$$

where (i) follows from $\sum_{i=1}^n \alpha_i = 1$. This is what we wanted to show. $\square$

Task 4

By constructing a code that evaluates the Bernstein Polynomials in a given value $t = t0$, we could easily plot the Bernstein Polynomials of degree 1 to 4. See below.



Appendix I

Code from task 2.

```
import pylab


def show_hull(points):
    """
    This function takes a set of points in the plane and plots all line
    segments connecting the points.

    The boundary of the convex hull is the outermost line segments.
    """
    # Separating the x and y values
    xvals = [pt[0] for pt in points]
    yvals = [pt[1] for pt in points]
    # Nested loop to connect all the points
    for i in range(len(points) - 1):
        # Plotting all the points
        pylab.plot(xvals[i], yvals[i], 'bo')
        for j in range(i + 1, len(points)):
            pylab.plot([xvals[i], xvals[j]], [yvals[i], yvals[j]], 'r')
    pylab.plot(xvals[len(points) - 1], yvals[len(points) - 1], 'bo')
    # Padding the plot
    padding = 0.1
    pylab.xlim([min(xvals) - padding, max(xvals) + padding])
    pylab.ylim([min(yvals) - padding, max(yvals) + padding])
    pylab.title('Points and their convex hull')
    pylab.xlabel('x')
    pylab.ylabel('y')
    pylab.show()

# The points to test
points = [(1, 2), (31/6, 3), (-8/3, 4), (0, 5)]
# Calling the function
show_hull(points)
```

Appendix II

Code from task 4.

```
from scipy.special import comb
import numpy
import pylab

def bernsteinPol(n, t0):
    """
    Constructs and evaluates the Bernstein Polynomials of a given degree
        at point t0.
    :param n: degree
    :param t0: value of t
    :return: list of values of the polynomials at t=t0
    """
    return [comb(n,j)*(1-t0)**(n-j)*t0**j for j in range(n+1)]

def plotBernstein(n,t0=0,t1=1,steps=30):
    """
    Plots the Bernstein Polynomials of any given degree.
    :param n: degree
    :param t0: start point of interval
    :param t1: end point of interval
    :param steps: number of grid points to plot
    """
    # Construct the grid of t values
    tgrid = numpy.linspace(t0,t1,steps)

    # Evaluate the Bernstein polynomials at each grid point
    ugrid = zip(*[bernsteinPol(n,t) for t in tgrid])

    # Plot each polynomial
    for ind, i in enumerate(ugrid):
        pylab.plot(tgrid,i,label='$j={}$'.format(ind))
    pylab.title('Bernstein Polynomials of Degree {}'.format(n))
    pylab.xlabel('$t$')
    pylab.ylabel('$B^{}_j(t)$'.format(n))
    pylab.legend(loc = 'upper right')
    pylab.show()

if __name__ == '__main__':
    plotBernstein(4)
```