Project 2_V3

October 28, 2021

0.0.1 TMA4215 Numerisk Matematikk

Høst 2021 – Tuesday, October 19, 2021

R. Bergmann, E. Çokaj, O. P. Hellan

1 Project 2: Bézier Curves and Interpolation

1.0.1 Notes

Groups. This project is a group project and can be solved in groups of *up to three* students. Feel free to use for example the Forum to find each other. Eventually it will be possible to register groups in Inspera. One persion (per group) can create a group and will get a PIN code which can be used by the other group members for registering.

Requirements for submission. The submission is in Inspera. Each group must submit their onw report. It is not allowed to copy from other groups.

All code – also the tests – should be in individual cells that can just be run (as soon as the necessary functions are defined). Functions should only be used in cells *after* their definition, such that an evaltuation in order of the notebook does not yield errors.

It is not possible to have an extension for this project.

Supervision. For questions the usual time, Thursday, 18.15–20.00 can be used. Questions can also be asked in the Mattelab forum.

1.0.2 Submission Deadline

Tuesday, November 9, 2021.

1.1 Introduction

In this project we consider another possibility to perform interpolation with piecewise polynomials, namely from the family of parametrized curves.

Let $\mathbf{p}_0, \ldots, \mathbf{p}_n \in \mathbb{R}^d$ (usually $d = 2$ or $d = 3$) denote $n + 1$ ordered points.

Then the nth degree Bézier curve is defined by

$$\mathbf{c}(t) = \mathbf{b}_n(t; \mathbf{p}_0, \ldots, \mathbf{p}_n) = \sum_{i=0}^{n} B_{i,n}(t)\mathbf{p}_i,$$

where $B_{n,i}(t)$ are the nth degree *Bernstein polynomials*. We use the first notation, $\mathbf{c}(t)$ when the points are clear from context, and the second, $\mathbf{b}(t;\mathbf{p}_0,\dots,\mathbf{p}_n)$ to emphasize the dependency of the nodes and/or the degree n.

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \qquad i = 0,\dots,n,$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ denotes the binomial coefficient.

To get familiar with the first few Bernstein polynomials it might be good to write down $B_{0,0}, B_{0,1}, B_{1,1}, B_{0,2}, B_{1,2}$, and $B_{2,2}$.

1.2 Problem 1: Properties of Bernstein polynomials

Let $n \in \mathbb{N}$ be given. We consider the Bernstein polynomials $B_{i,n}(t)$, $i = 0,\dots,n$.

1. Show that $B_{i,n}(t) \geq 0$ for all $t \in [0,1]$.

[ ]:

2. Show that the Bernstein polynomials for $0 < i < n$ can be recursively defined by

$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t), \qquad t \in [0,1].$$

   How does this look like for $B_{0,n}$ and $B_{n,n}$?

[ ]:

3. Show that the $B_{i,n}(t)$ form a partition of unity, i.e.

$$\sum_{i=0}^{n} B_{i,n}(t) = 1 \qquad \text{for } t \in [0,1].$$

   Hint: Use induction by n.

[ ]:

4. Show that the derivative is given by

$$B'_{i,n}(t) = n(B_{i-1,n-1}(t) - B_{i,n-1}(t))$$

   with $B_{-1,n-1}(u) \equiv B_{n,n-1}(u) \equiv 0$.

[ ]:

5. Implement a function `Bernstein(i,n,t)` that evaluates $B_{i,n}$ at t and plot all functions $B_{i,n}$, $i = 0,\dots,n$ for $n = 3$ and $n = 9$.

[3]:

1.3 Problem 2: Properties of (composite) Bézier curves

We consider the points $\mathbf{p}_0, \dots, \mathbf{p}_n \in \mathbb{R}^d$ and $\mathbf{q}_0, \dots \mathbf{q}_n \in \mathbb{R}^d$ and their corresponding Bézier curves $\mathbf{c}(t) = \mathbf{b}(t; \mathbf{p}_0, \dots, \mathbf{p}_n)$ and $\mathbf{d}(t) = \mathbf{b}(t; \mathbf{q}_0, \dots, \mathbf{q}_n)$, respectively.

In this problem, we will also consider *composite Bézier curves*, or piecewise Bézier curves, e.g. a curve $\mathbf{s}\colon [0,2] \to \mathbb{R}^d$ defined by

$$\mathbf{s}(t) = \begin{cases} \mathbf{c}(t) & \text{for } 0 \leq t < 1 \\ \mathbf{d}(t-1) & \text{for } 1 \leq t \leq 2. \end{cases}$$

1. Compute the first two derivatives $\mathbf{c}'(t)$ and $\mathbf{c}''(t)$ of $\mathbf{c}(t)$.

[ ]:

2. What values does $\mathbf{c}(t)$ attend at its end points? State $\mathbf{c}(0)$ and $\mathbf{c}(1)$.

[ ]:

3. Prove that the following properties hold:
   1. $\mathbf{c}'(0) = n(\mathbf{p}_1 - \mathbf{p}_0)$,
   2. $\mathbf{c}''(0) = n(n-1)(\mathbf{p}_0 - 2\mathbf{p}_1 + \mathbf{p}_2)$,
   3. $\mathbf{c}'(1) = n(\mathbf{p}_n - \mathbf{p}_{n-1})$,
   4. $\mathbf{c}''(1) = n(n-1)(\mathbf{p}_n - 2\mathbf{p}_{n-1} + \mathbf{p}_{n-2})$.

[ ]:

4. Use the recursion property of Bernstein polynomials to prove the recursive definition

$$\mathbf{b}_n(t; \mathbf{p}_0, \dots, \mathbf{p}_n) = (1-t)b_{n-1}(t; \mathbf{p}_0, \dots, \mathbf{p}_{n-1}) + tb_{n-1}(t; \mathbf{p}_1, \dots, \mathbf{p}_n).$$

[ ]:

5. The recursion from point 4 of this problem can be used to define the so-called “de Casteljau” algorithm to evaluate $\mathbf{b}_n(t_0; \mathbf{p}_0, \dots, \mathbf{p}_n)$ at $t_0 \in [0,1]$ algorithm:

   Starting with $\mathbf{p}_{0,i}(t_0) = \mathbf{p}_i$ compute for $k = 1, \dots, n$ and $i = 0, \dots, n-k$ the intermediate points

$$\mathbf{p}_{k,i}(t_0) = (1-t_0)\mathbf{p}_{k-1,i}(t_0) + t_0\mathbf{p}_{k-1,i+1}(t_0),$$

   then $\mathbf{b}_n(t_0; \mathbf{p}_0, \dots, \mathbf{p}_n) = \mathbf{p}_{n,0}$.

   Implement a function `deCasteljau(P,t)` where `P` is a vector – of $n+1$ points – to evaluate the corresponding Bézier curve at t.

   This function should also return (as a second return value) a vector `Pvecs` that contains a vector of points for every “level” k considered.

   Plot the corresponding curve for the points

$$\mathbf{p}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{p}_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} 6 \\ -3 \end{pmatrix}, \quad \mathbf{p}_4 = \begin{pmatrix} 8 \\ 0 \end{pmatrix},$$

including one line per "level" k connecting the points when evaluating the curve at $t_0 = \frac{1}{3}$

[ ]:

6. Consider a *composite Bézier curve* $\mathbf{s}(t)$ as described in the beginning of this problem. Assume we want $\mathbf{s}(t)$ to be a $C^{(k)}$, $k = 0, 1, 2$ function. Then surely, increasing the class k increases the dependent properties we have to impose.

   1. What are the critical points of $\mathbf{s}(t)$ to investigate for the property to be a $C^{(k)}$ function?
   2. Which properties have to hold for continuity ($k = 0$)?
   3. Which properties have to hold for $\mathbf{s}(t)$ to be continuously differentiable ($k = 1$) ?
   4. Which properties have to hold for $\mathbf{s}(t)$ to be twice continuously differentiable ($k = 2$) ?

   Try to simplify the conditions for the third and fourth point based on the properties you derived before.

   5. What changes if we want $\mathbf{s}(t)$ to be periodic, i.e. $s(t) = s(2+t)$ for all t for the continous differentiability case $k = 1$?

[ ]:

1.4 Problem 3: Interpolation with (composite, cubic) Bézier curves

The most prominent variant are – similar to B splines – again those Bézier curves that yield cubic polynomials, i.e. $\mathbf{b}_3(t; \mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$. To obtain a spline, we consider *composite Bézier curves*, i.e. we "stitch together" several Bézier curves (as considered for the case of 2 curves in the last problem):

Given a number m of segments and $\mathbf{p}_{0,i}, \mathbf{p}_{1,i}, \mathbf{p}_{2,i}, \mathbf{p}_{3,i}$ for $i = 1, \ldots, m$, then we define

$$\mathbf{B}(t) = \begin{cases} b_3(t - i + 1; \mathbf{p}_{0,i}, \mathbf{p}_{1,i}, \mathbf{p}_{2,i}, \mathbf{p}_{3,i}) & \text{for } i - 1 \leq t < i \text{ and each } i = 1, \ldots, m \end{cases}$$

1. Implement a function `compositeBézier(P, t)` that evaluates $\mathbf{B}(t), t \in [0, m]$, where $P = (\mathbf{p}_{j,i})_{j=0,i=1}^{3,m}$ denotes a matrix of control points. Note that you can obtain the degree and the number of segments from the size of `P`.

   Test your function with the 3-segment cubic composite B spline given by

$$\begin{aligned}
\mathbf{p}_{0,1} &= \begin{pmatrix} -1 \\ 0 \end{pmatrix}, & \mathbf{p}_{1,1} &= \begin{pmatrix} -1 \\ \frac{1}{3} \end{pmatrix}, & \mathbf{p}_{2,1} &= \begin{pmatrix} -\frac{1}{3} \\ 1 \end{pmatrix}, & \mathbf{p}_{3,1} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
\mathbf{p}_{0,2} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}, & \mathbf{p}_{1,2} &= \begin{pmatrix} \frac{1}{3} \\ 1 \end{pmatrix}, & \mathbf{p}_{2,2} &= \begin{pmatrix} 1 \\ \frac{1}{3} \end{pmatrix}, & \mathbf{p}_{3,2} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
\mathbf{p}_{0,3} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}, & \mathbf{p}_{1,3} &= \begin{pmatrix} -\frac{1}{3} \\ 1 \end{pmatrix}, & \mathbf{p}_{2,3} &= \begin{pmatrix} \frac{1}{3} \\ -1 \end{pmatrix}, & \mathbf{p}_{3,3} &= \begin{pmatrix} 0 \\ -1 \end{pmatrix}
\end{aligned}$$

   and plot the resulting (complete) curve $\mathbf{B}$.

[ ]:

2. Use the properties derived so far to derive an algorithm for the following problem:

   Given data points $\mathbf{a}_0, \dots, \mathbf{a}_{m-1} \in \mathbb{R}^d$ and velocities $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{m-1}, \in \mathbb{R}^d$.

   Find the *periodic composite cubic Bézier curve* $\mathbf{B}(t)$ that maps from $[0, m]$ to $\mathbb{R}^d$ with the following properties

   - $\mathbf{B}(0) = \mathbf{B}(m)$,
   - $\mathbf{B}'(0) = \mathbf{B}'(m)$,
   - $\mathbf{B}(i) = \mathbf{a}_i$ for $i = 0, \dots, m-1$,
   - $\mathbf{B}'(i) = \mathbf{v}_i$ for $i = 0, \dots, m-1$.

   You can for example first sketch the algorithm or a few ideas in LaTeX.

   Then implement a function `interpolate_periodic(A,V)` where `A` is the vector of the interpolation points $[\mathbf{a}_0, \dots, \mathbf{a}_{m-1}]$ and `V` is the vector of the velocities $[\mathbf{v}_0, \dots, \mathbf{v}_{m-1}]$.
   The function should return a matrix `P` like in the first part to be able to plot the result.

[ ]:

3. Take your favourite letter from the alphabet. Draw its outline – i.e. a closed curve surrounding the letter– on a graph paper (those with a regular 2D grid) and take a few measurements of points and velocities. Use this data to illustrate how your function from 2 works.

   Hint: a good idea is to take a letter without holes that only consist of one component like `t` or `T` (maybe not the little boring `l`).

   Bonus Task: Ignore the hint and do something fancy with `å`, `æ`, `ø`, or even `ß` (though that can be done with one outline in most this fonts).

[ ]:

4. Take the resulting control points from part 3 of this problem and create the following new curves with new sets of points given by
   1. `Q` where each $\mathbf{q}_{i,j} = 2\mathbf{p}_{i,j}$
   2. `R` where each $\mathbf{r}_{i,j} = \begin{pmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{pmatrix} \mathbf{p}_{i,j}$
   3. `S` where each $\mathbf{s}_{i,j} = \begin{pmatrix} 1 & \frac{1}{2} \\ 1 & 0 \end{pmatrix} \mathbf{p}_{i,j} + \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
   4. `T` where each $\mathbf{t}_{i,j} = \begin{pmatrix} -1 & -\frac{1}{2} \\ 1 & 0 \end{pmatrix} \mathbf{p}_{i,j} + \frac{1}{2}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ where each $i = 0, 1, 2, 3$ and $j = 0, \dots, m-1$

[ ]:

1.5 Problem 4: Optimisation with Bézier curves

Similar to Problem 3, assume we have a composite cubic Bézier curve $\mathbf{B}(t)$ (here just not necessarily periodic) with m segments, i.e. control points $\mathbf{p}_{0,1}, \mathbf{p}_{1,1}, \mathbf{p}_{2,1}, \mathbf{p}_{3,1}, \mathbf{p}_{0,2}, \mathbf{p}_{1,2}, \dots, \mathbf{p}_{3,m}$. We denote its segments by $\mathbf{c}_i \colon [i-1, i] \to \mathbb{R}^2$, for $i = 1, \dots, m$. Then $\mathbf{B}(t)$ is defined on $[0, m]$. Assume further that $\mathbf{B}(t)$ is $C^{(1)}$.

1. Due to the property of $\mathbf{B}(t)$ being continuous, we have $\mathbf{p}_{3,i} = \mathbf{p}_{0,i+1}$ for $i = 1, \dots, m-1$, so we can omit "storing" the redundant data of $\mathbf{p}_{3,i}$.

Similarly due to the differentiability we can express $\mathbf{p}_{2,i}$ using $\mathbf{p}_{0,i+1}$ and $\mathbf{p}_{1,i+1}$ for each $i = 1, \ldots, m-1$. What does this expression look like?

Bonus question: The "data" we store for $\mathbf{p}_{2,i}$ and/or $\mathbf{p}_{1,i+1}$ is just one vector in $\mathbb{R}^d$. Can we phrase this information in terms of the velocity $\mathbf{v}_{i+1}$ at $\mathbf{p}_{0,i+1}$, i.e. such that from this velocity we can recover both "neighboring" points?

[ ]:

2. We want to minimize the (squared) acceleration of the curve

$$F(\mathbf{P}) = \int_0^m \|\mathbf{B}''(t)\|^2 \mathrm{d}t,$$

with respect to the remaining control points

$$\mathbf{P} = [\mathbf{p}_{0,1}, \mathbf{p}_{1,1}, \mathbf{p}_{0,2}, \mathbf{p}_{1,2}, \ldots, \mathbf{p}_{0,m-1}, \mathbf{p}_{1,m-1}\mathbf{p}_{0,m}, \mathbf{p}_{1,m}, \mathbf{p}_{2,m}, \mathbf{p}_{3,m},].$$

For simplicity we only consider one segment, i.e. for $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \in \mathbb{R}^2$, we consider the cubic Bézier curve $\mathbf{b}_3(t; \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$.

First derive a closed form for the integral

$$\int_0^1 \|\mathbf{b}_3''(t; \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\|_2^2 \, \mathrm{d}t$$

in order to derive the gradient of

$$\tilde{F}(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = \int_0^1 \|\mathbf{b}_3''(t; \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\|_2^2 \, \mathrm{d}t$$

and with respect to the control points $\mathbf{q}_i, i = 0, \ldots, 3$.

While we do not want to write down the whole gradient of F, please sketch how you can use the result of $\tilde{F}$ to compute the gradient of F. Remember that $\mathbf{B}(t)$ is continuously differentiable.

[ ]:

3. Look at the first order optimality conditions of the optimisation task to minimize $\tilde{F}$ from the last part. How can we find such a minimiser? Is the solution unique?

For the overall problem F we even have to take into account the properties from the fist part. What about the solution now? Is it unique? You may argue intuitively here or provide a concrete example of two minimisers for a 2-segment curve, i.e. $m = 2$.

[ ]:

4. Assume we extend the problem to have some (data) points $\mathbf{d}_i \in \mathbb{R}^2$, $i = 0, \ldots, m$ given and we extend the problem to

$$G_\lambda(\mathbf{P}) = \frac{\lambda}{2} \sum_{i=0}^{m} \|\mathbf{d}_i - \mathbf{B}(i)\|_2^2 + \int_0^m \|\mathbf{B}''(t)\|^2 \mathrm{d}t, \qquad \text{for some} \quad \lambda > 0$$

We again can first look at the simplified problem: Given two points $\mathbf{s}, \mathbf{e} \in \mathbb{R}^2$ consider for some $\lambda > 0$ the function for one segment, namely

$$\begin{aligned}\tilde{G}_\lambda(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) &= \frac{\lambda}{2}\Big(\|\mathbf{s} - \mathbf{b}_3(0; \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\|_2^2 + \|\mathbf{e} - \mathbf{b}_3(1; \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\|_2^2\Big) + \tilde{F}(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\\ &= \frac{\lambda}{2}\Big(\|\mathbf{s} - \mathbf{q}_0\|_2^2 + \|\mathbf{e} - \mathbf{q}_3\|_2^2\Big) + \tilde{F}(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\end{aligned}$$

Does this change the question about uniqueness? Without programming / testing, just intuitively: What does this model do, if you let λ tend to zero? What does it do, if you let λ tend to ∞?

Similarly to part 3 of this problem, what does change for G_λ in comparison to F concerning uniqueness? what does the λ change here (if very large or very small)?

[ ]:

5. Use the previous parts to derive a gradient descent algorithm to minimize $G_\lambda(\mathbf{P})$ with respect to the control points $\mathbf{P}$. You may use a constant step size.

   Take as an example your letter from Problem 3 as input $\mathbf{P}$ for your algorithm and two different values of λ. How does the letter change?

[ ]: