Lecture 5 - Workshop 1

Scheduled: 2024/02/27
Latest Revision: 2024/05/09

Introduction

Many, if not most objects around us have at some point been represented as a CAD (computer-aided design) drawing. This is true for both physical objects, which nowadays may be directly manufactured from CAD drawings using modern automated fabrication machines (such as for example 3D printers), or digital objects, such as the scalable fonts in the PDF document you are looking at. CAD programs use polynomials to represent pieces of curves in 2D or surfaces in 3D. For example, a straight line segment may be exactly represented using a polynomial of degree one (affine function), whereas other segments or surfaces are approximated using higher degree (most often, cubic) polynomials. At points where different segments meet, we may impose further conditions on the approximation. In particular, we may require that the values, derivatives, and/or higher order derivatives of the approximating polynomial pieces agree at the meeting points, which creates curves and surfaces with different levels of smoothness. An example is provided in Figure 1, where the outline of the symbol “5” is represented with the help of a piecewise-polynomial curve. The advantage of representing objects/fonts in this fashion is that they can be easily manipulated: shifted/scaled/rotated/printed in an arbitrary resolution without sacrificing the quality of the object’s representation.

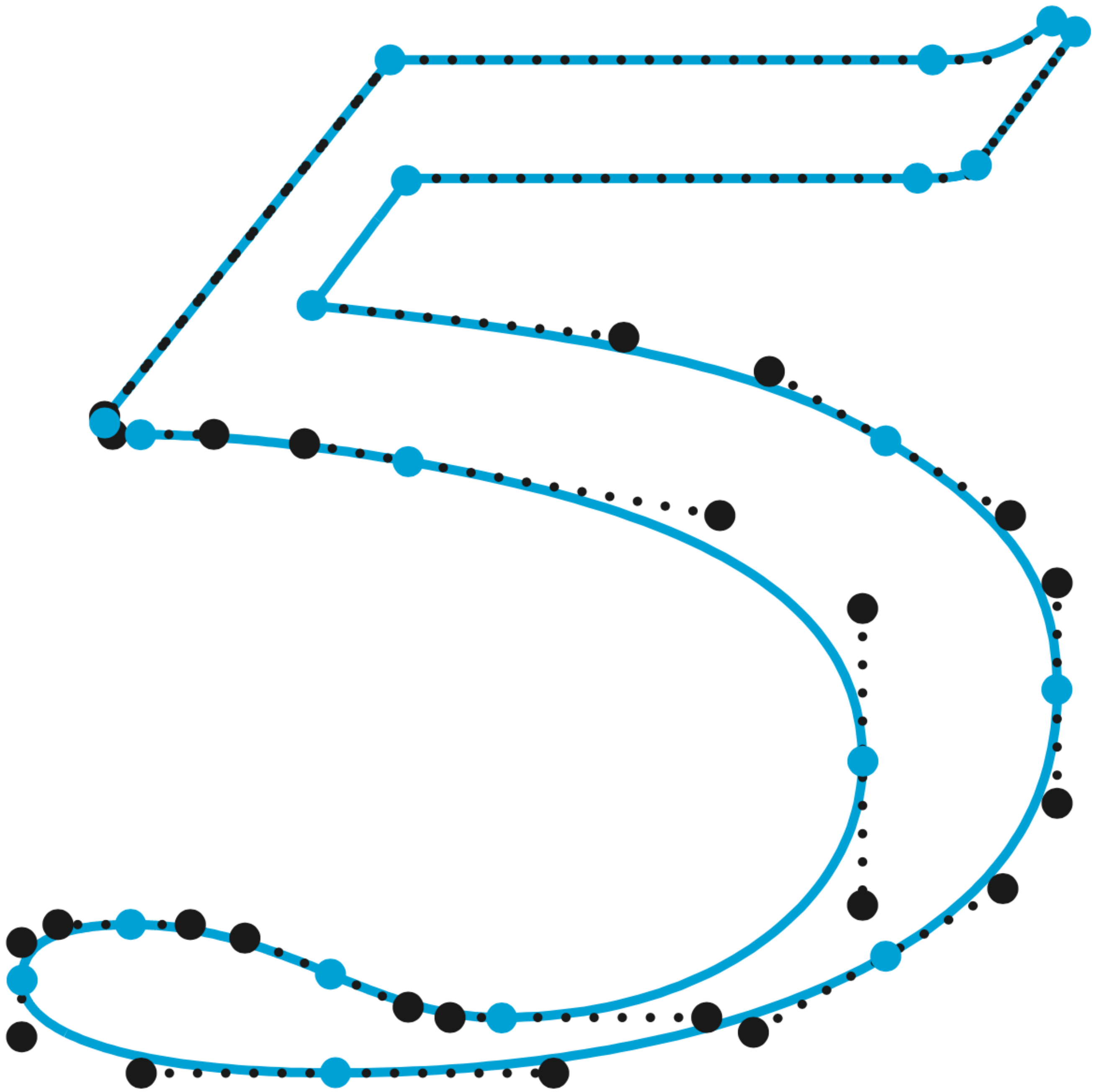


Figure 1: Representation of the outline of the symbol “5” with the help of a piecewise-polynomial curve. Each blue point corresponds to a meeting point of two different “pieces”/polynomials. The full curve consists of 21 pieces.

In the questions that follow we will explore applications of linear algebra in polynomial data interpolation and extrapolation, and then move on to the discussion of Bezier curves as utilized in PostScript and many other applications. From the linear algebraic perspective these questions are quite similar. That is, they boil down to formulating an appropriate system of linear algebraic equations and then solving it.

> **Polynomial interpolation & Extrapolation**
>
> **Polynomial interpolation** in linear algebra is a method used to find a polynomial function that passes through a given set of data points.
>
> **Extrapolation**, as opposed to interpolation, involves estimating values outside the range of known data points. In other words, it's the process of extending a curve or function beyond the range of observed data.

---

1 - Polynomial interpolation

We begin with polynomial interpolation, where the goal is to find a polynomial passing through some datapoints. The resulting polynomial may be used for evaluating the model between datapoints, which is known as the interpolation of the data, or elsewhere, which is typically referred to as the extrapolation. Consider a dataset given by n pairs of real numbers

$$(t_1, \beta_1), (t_2, \beta_2), \ldots, (t_n, \beta_n) \tag{1}$$

which we interpret as n points in $\mathbb{R}^2$. Our underlying assumption is going to be that

$$t_i \neq t_j, \text{ for } i \neq j, \text{ and } i = 1, \ldots, n, \ j = 1, \ldots, n. \tag{2}$$

We would like to find a polynomial

$$p_x(t) = x_1 + x_2 t + \cdots + x_n t^{n-1} \tag{3}$$

of degree at most $n-1$, which passes through these points. That is, we would like to solve the system of n equations

$$\begin{cases} p_x(t_1) = \beta_1, \\ p_x(t_2) = \beta_2, \\ \quad\vdots \\ p_x(t_n) = \beta_n, \end{cases} \tag{4}$$

for the unknown *vector* of polynomial coefficients $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$. This is a system of linear algebraic equations and can as such be represented in the form $Ax = b$, for some suitably constructed matrix A and the vector b.

What does $Ax = b$ represents

- **A**: This represents the coefficient matrix. In the context of polynomial interpolation, A is a matrix constructed from the t_i values of the given dataset. Each row of A corresponds to one equation in the system, and each column corresponds to a coefficient of the polynomial.

  x: This represents the vector of unknowns, which are the coefficients of the polynomial. In the context of polynomial interpolation, x is a vector containing the coefficients $x_1, x_2, \ldots, x_n$.

  b: This represents the vector of **known values**, which are the dependent variable values β_i from the dataset. In the context of polynomial interpolation, b is a vector containing the values $\beta_1, \beta_2, \ldots, \beta_n$.

Question 1.

Question 1.1. Equation (3) allows us to identify polynomials $px(t)$ of degree no larger than $n-1$ with $n-$vectors of their coefficients $x \in \mathbb{R}^n$. Identify the polynomials, that correspond to the standard unit vectors $e_1, \ldots, e_n \in \mathbb{R}^n$. That is, write down the explicit formulae for $p_{e_1}(t), \ldots, pe_n(t)$.

The Answer

Here is equation (3)

$$p_x(t) = x_1 + x_2 t + \cdots + x_n t^{n-1} \tag{3}$$

In equation (3) the $x_1, x_2, \ldots x_n$ are scalars and make up the vector of unknowns

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

meaning we are multiplying the the entries of vector x with the $t^0, t^1, \ldots t^n$. Therefore if we replace the vectors x with the unit vectors e_1, e_2 or e_3 we get

$$\begin{gathered} p_{e_1} = 1 * t + 0 * t + 0 * t_2 + \cdots + 0 * t^{n-1} = 1 \\ p_{e_2} = 0 + 1 * t + 0 * t^2 + \cdots + 0 * t^{n-1} = t \\ p_{e_3} = 0 + 0 * t + 1 * t^2 + \cdots + 0 * t^{n-1} = t^2 \end{gathered}$$

Hence if we replace the vector x with the unit vector e_n we get

$$p_{e_n} = t^{n-1}$$

Question 1.2. Let us now consider a linear transformation (you do not have to prove, that it is in fact linear) $F : \mathbb{R}^n \to \mathbb{R}^n$, given by the formula

$$F(x) = \begin{bmatrix} p_x(t_1) \\ p_x(t_2) \\ \vdots \\ p_x(t_n) \end{bmatrix} \tag{5}$$

where the polynomial $p_x(t)$ and the vector $x \in \mathbb{R}^n$ are related via (3). Write down the matrix A corresponding to this linear transformation, that is, such that $F(x) = Ax$.

The Answer

Recall equation (3)

$$p_x(t) = x_1 + x_2 t + \cdots + x_n t^{n-1} \tag{3}$$

and we have t_i for $i = 1, \ldots, n$. Then, as an example, we construct the rows for $p_x(t_1)$, $p_x(t_2)$ and $p_x(t)3$

$$\begin{aligned} p_x(t_1) &= x_1 + x_2 t_1 + \cdots + x_n t_1^{n-1} \\ p_x(t_2) &= x_1 + x_2 t_2 + \cdots + x_n t_2^{n-1} \\ p_x(t_3) &= x_1 + x_2 t_3 + \cdots + x_n t_3^{n-1} \end{aligned}$$

Hence, the matrix A can be constructed as follows.

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \ldots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \ldots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \ldots & t_n^{n-1} \end{bmatrix}$$

Question 1.3. The *fundamental theorem of algebra* implies, that the only polynomial of degree no larger than $n - 1$, which satisfies the equations

$p_x(t_1) = p_x(t_2) = \cdots = p_x(t_n) = 0$ is a zero polynomial, that is, a polynomial, whose coefficients are all zeros. Explain why this means that the columns of A, constructed in the previous step, are linearly independent. Conclude that the columns of A form a basis in $\mathbb{R}^n$

(comment, still have no idea what the *fundamental theorem of algebra* is used for in here)

The Answer: Part one, explaining that the columns are linearly independent.

Recall from lecture 2 the definition of linear independence and dependence of columns lecture 3.

Definition of linear independence

Collection of n-vectors $\vec{a}_1, \ldots, \vec{a}_k$ (with $k \geq 1$) is called linearly independent if it is not linearly dependent, which means that Equation 5.1

$$\beta_1 \vec{a}_1 + \ldots + \beta_k \vec{a}_k = \vec{0}$$

only hold for $\beta_1 = \ldots = \beta_k = 0$. In other words, the only linear combination of the vectors that equals the zero vector is the linear combination with all coefficients zero.
An equivalent definition is, for n-vectors $a_1, \ldots, a_k$, no a_i can be expressed as a linear combination of the other vectors.

Linear dependence of columns.
We can express the concepts of linear dependence and independence in a compact form using matrix-vector multiplication. The columns of a matrix A are linearly dependent if $Ax = 0$ for some $x \neq 0$.

The columns of a matrix A are linearly independent if $Ax = 0$ implies $x = 0$.

we know from **Q.1.2** that the linear transformation F has a corresponding matrix A

$$F(x) = \begin{bmatrix} p_x(t_1) \\ p_x(t_2) \\ \vdots \\ p_x(t_n) \end{bmatrix} = Ax$$

And the matrix A can be represented in terms of its columns

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^{n-1} \end{bmatrix} = [a_1, a_2, \dots a_n]$$

And $A\vec{x} = \vec{0}$ is is true if and only if $\vec{x} = \vec{0}$

therefore, the only way $A\vec{x} = \vec{0}$ is that $n-$vector x is a zero vector, because there is no way the columns $a_1 + a_2 + \dots + a_n = \vec{0}$. Hence, the columns of A are linearly independent.

Part two, explaining that the columns form a basis.

Recall from lecture 2 the Independence-dimension inequality and the definition of basis

Independence-dimension inequality and basis

Independence-dimension inequality
if the n-vectors $a_1, \dots, a_k$ are linearly independent then $k \leq n$.

In other words a *linearly independent collection* of $n-$vectors can have at most n elements.
Put another way *Any* collection *of* $n+1$ *or more* $n-$*vectors is* linearly dependent*.

Basis
A collection of n linearly independent n-vectors (a collection of linearly independent vectors of the maximum size) is called *basis*.
If the n-vectors $a_1, \ldots, a_n$ are a basis, then any n-vector b can be written as a linear combination of them.

The columns of A are independent, for them to be a basis then the number of the columns of A must be the same as the number of rows/dimensions/equations i.e the matrix A is square.

we know that each row in A represented a the polynomials $p_x(t)$ for t_i for $i = 1, \ldots, n$. hence there are n rows. And from part one we saw that there are n columns. Hence the columns of A form a basis.

Question 1.4. Let us define the vector $b \in \mathbb{R}^n$ by

$$b = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

Note that the system (4) is equivalent to $F(x) = b$, which is in turn equivalent to $Ax = b$. Use the conclusion of the previous point to explain, why for an arbitrary choice of datapoints in (1), (2) there exists an interpolating polynomial $px(t)$ of degree at most $n-1$ passing through these datapoints, and why such polynomial is unique.

The Answer

The equation $F(x) = b$ is equivalent to $Ax = b$, where A is the matrix of coefficients constructed from the given datapoints and x is the vector of coefficients of the interpolating polynomial $p_x(t)$.

From the conclusion of the previous point (1.3), we know that the columns of A form a basis in $\mathbb{R}^n$.

If the n-vectors $a_1, \ldots, a_n$ are a basis (in this case the columns of A), then any n-vector b can be written as a linear combination of them.

Interpolation Polynomial: Therefore, for any arbitrary choice of datapoints $(t_1, \beta_1), (t_2, \beta_2), \ldots, (t_n, \beta_n)$ satisfying the conditions in (2), there exists a unique interpolating polynomial $px(t)$ of degree at most $n-1$ passing through these datapoints. This is because solving the system $Ax = b$ for x yields the coefficients of the unique polynomial that interpolates the given datapoints.

Uniqueness: The uniqueness of the interpolating polynomial stems from the fact that the columns of A form a basis in $\mathbb{R}^n$. Therefore, the system $Ax = b$ has a unique solution for any given $n-$vector b. Hence, there is only one set of coefficients x that satisfies $Ax = b$,

---

Question 1.5. Use Gaussian elimination (by hand) to find the quadratic polynomial passing through the points (1, 4),
(2, 0), and (3, 12). Plot the graph of the resulting polynomial and the points in the dataset.

The Answer

The quadratic polynomial denoted as $p_x(t) = x_1 + x_2 t + x_3 t^2$

Then we can set three equations based on the given points

$$\text{For } t=1: p_x(1) = x_1 + x_2 * 1 + x_3 * 1^2 = 4$$
$$\text{For } t=2: p_x(2) = x_1 + x_2 * 2 + x_3 * 2^2 = 0$$
$$\text{For } t=3: p_x(3) = x_1 + x_2 * 3 + x_3 * 3^2 = 12$$

Then we can rewrite the three equations as an augmented matrix

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 1 & 2 & 4 & 0 \\ 1 & 3 & 9 & 12 \end{array}\right]$$

Then we perform row operations to get the matrix into row-echelon form.

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 1 & 2 & 4 & 0 \\ 1 & 3 & 9 & 12 \end{array}\right]$$

Perform: $R_3 = R_3 - R_1$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 1 & 2 & 4 & 0 \\ 0 & 2 & 8 & 8 \end{array}\right]$$

Perform: $R_2 = R_2 - R_1$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 0 & 1 & 3 & -4 \\ 0 & 2 & 8 & 8 \end{array}\right]$$

Perform: $R_3 = R_3 - 2R_2$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 0 & 1 & 3 & -4 \\ 0 & 0 & 2 & 16 \end{array}\right]$$

Then we have

$$x_1 + x_2 + x_3 = 4$$
$$x_2 + 3x_3 = -4$$
$$2x_3 = 16$$

Then we have

$x_3 = 16/2 = 8$

back substitute x_3 into equation two and solve for x_2

$x_2 = -4 - (3 * 8) = -28$

back substitute x_2 and x_3 intro equation 1

$x_1 = 4 - (-28) - 8 = 24$

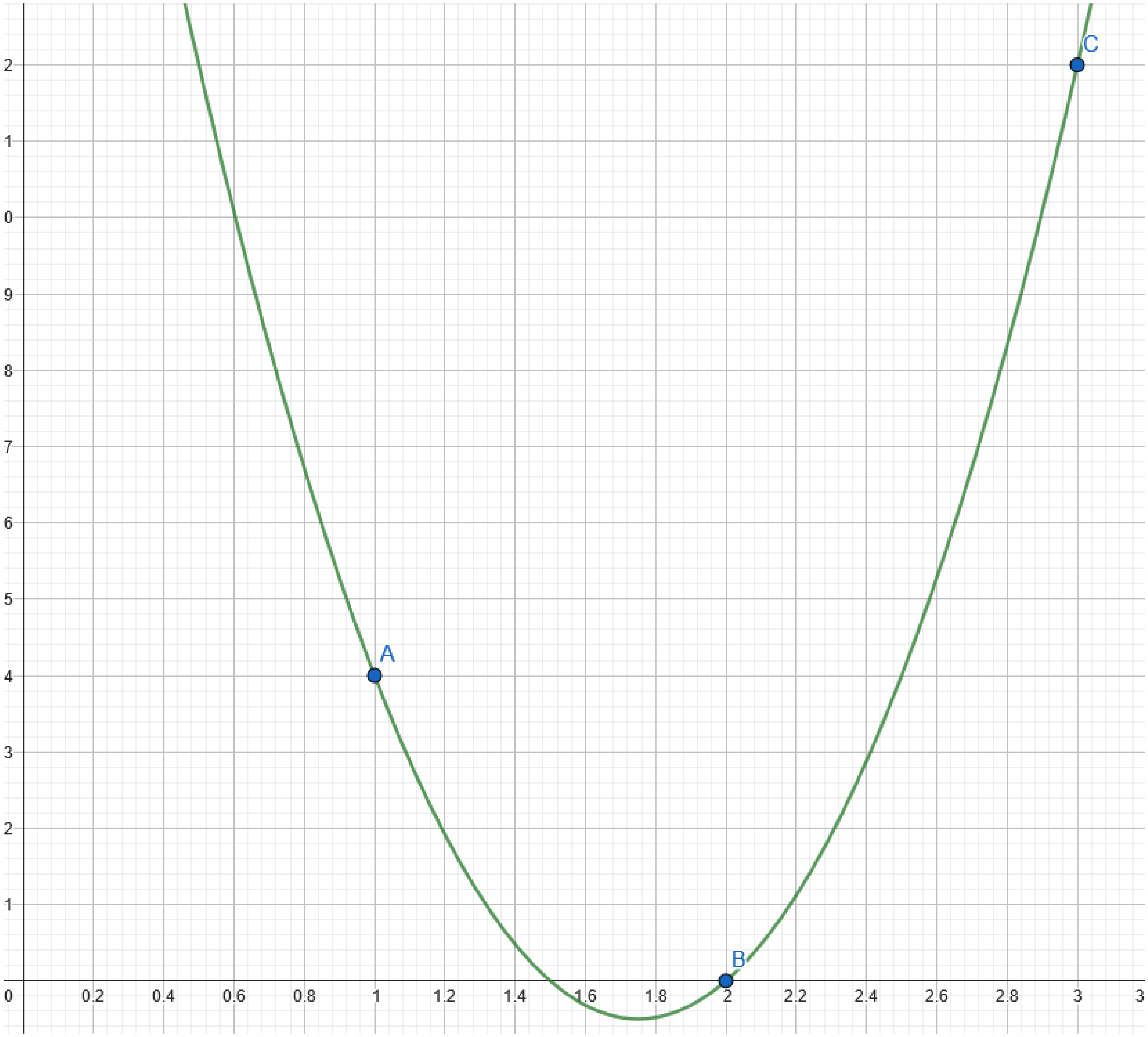
Hence,

$$\begin{cases} x_1 = 24 \\ x_2 = -28 \\ x_3 = 8 \end{cases}$$

Here is the quadratic polynomial is

$$p_x(t) = 24 + 28t + 8t^2$$

And here is the graph of the resulted polynomial plotted with datapoints and passing through them.



Question 1.6. Determine the fourth degree polynomial passing through the datapoints

$(0, \cos(0)) = (0, 1)$,

$(\pi/6, \cos(\pi/6)) = (\pi/6, \sqrt{3}/2)$,

$(\pi/4, \cos(\pi/4)) = (\pi/4, \sqrt{2}/2)$,

$(\pi/3, \cos(\pi/3)) = (\pi/3, 1/2)$,

$(\pi/2, \cos(\pi/2)) = (\pi/2, 0)$.

Do not perform Gaussian elimination by hand! Instead, write down the system of linear algebraic equations for this case and solve it using

Matlab or Python's numpy. See the examples of how to do this available on moodle.

Plot the interpolating polynomial and the function $\cos(x)$ on the interval $[-1,2]$. Using the computed interpolating polynomial, find an approximation to $\cos(1)$ and compare it with the "exact" value.

The Answer

To determine the fourth-degree polynomial passing through the given data points, let's denote the polynomial as

$$p_x(t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3 + x_5 t^4$$

Then we can set five equations based on the given points

$$\begin{gathered}
\text{For } t = 0: \quad p_x(0) = 1 \\
\text{For } t = \tfrac{\pi}{6}: \quad p_x(\tfrac{\pi}{6}) = x_1 + x_2 \tfrac{\pi}{6} + x_3 (\tfrac{\pi}{6})^2 + x_4 (\tfrac{\pi}{6})^3 + x_5 (\tfrac{\pi}{6})^4 = \tfrac{\sqrt{3}}{2} \\
\text{For } t = \tfrac{\pi}{4}: \quad p_x(\tfrac{\pi}{4}) = x_1 + x_2 \tfrac{\pi}{4} + x_3 (\tfrac{\pi}{4})^2 + x_4 (\tfrac{\pi}{4})^3 + x_5 (\tfrac{\pi}{4})^4 = \tfrac{\sqrt{2}}{2} \\
\text{For } t = \tfrac{\pi}{3}: \quad p_x(\tfrac{\pi}{3}) = x_1 + x_2 \tfrac{\pi}{3} + x_3 (\tfrac{\pi}{3})^2 + x_4 (\tfrac{\pi}{3})^3 + x_5 (\tfrac{\pi}{3})^4 = \tfrac{1}{2} \\
\text{For } t = \tfrac{\pi}{2}: \quad p_x(\tfrac{\pi}{2}) = x_1 + x_2 \tfrac{\pi}{2} + x_3 (\tfrac{\pi}{2})^2 + x_4 (\tfrac{\pi}{2})^3 + x_5 (\tfrac{\pi}{2})^4 = 0
\end{gathered}$$

The augmented matrix is

$$\left[\begin{array}{ccccc|c}
1 & 0 & 0 & 0 & 0 & 1 \\
1 & \frac{\pi}{6} & \left(\frac{\pi}{6}\right)^2 & \left(\frac{\pi}{6}\right)^3 & \left(\frac{\pi}{6}\right)^4 & \frac{\sqrt{3}}{2} \\
1 & \frac{\pi}{4} & \left(\frac{\pi}{4}\right)^2 & \left(\frac{\pi}{4}\right)^3 & \left(\frac{\pi}{4}\right)^4 & \frac{\sqrt{2}}{2} \\
1 & \frac{\pi}{3} & \left(\frac{\pi}{3}\right)^2 & \left(\frac{\pi}{3}\right)^3 & \left(\frac{\pi}{3}\right)^4 & \frac{1}{2} \\
1 & \frac{\pi}{2} & \left(\frac{\pi}{2}\right)^2 & \left(\frac{\pi}{2}\right)^3 & \left(\frac{\pi}{2}\right)^4 & 0
\end{array}\right]$$

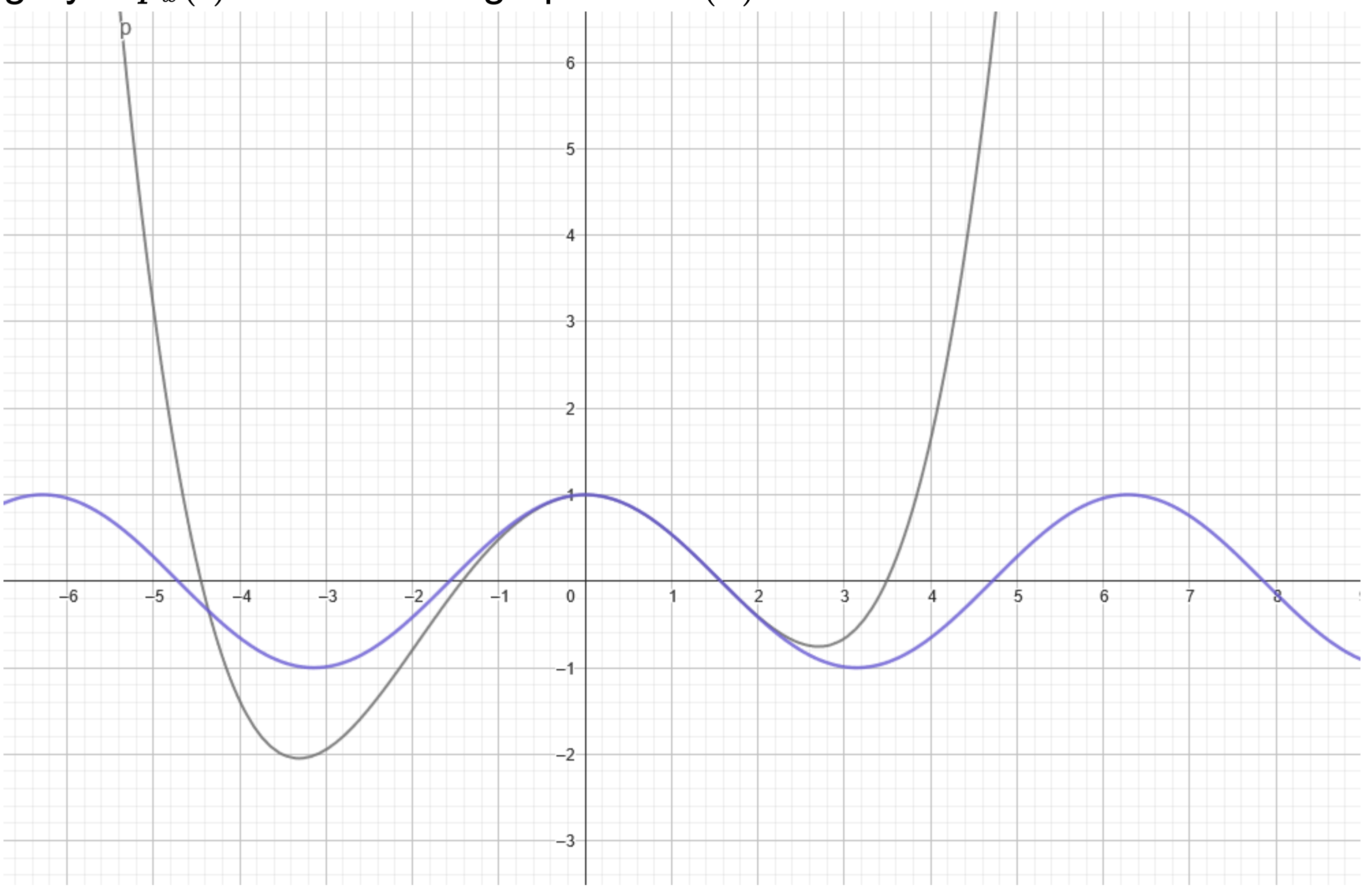
Hence

$$x = \begin{bmatrix} 1 \\ 1/298 \\ -186/361 \\ 11/470 \\ 11/382 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.0034 \\ -0.5152 \\ 0.0234 \\ 0.0288 \end{bmatrix}$$

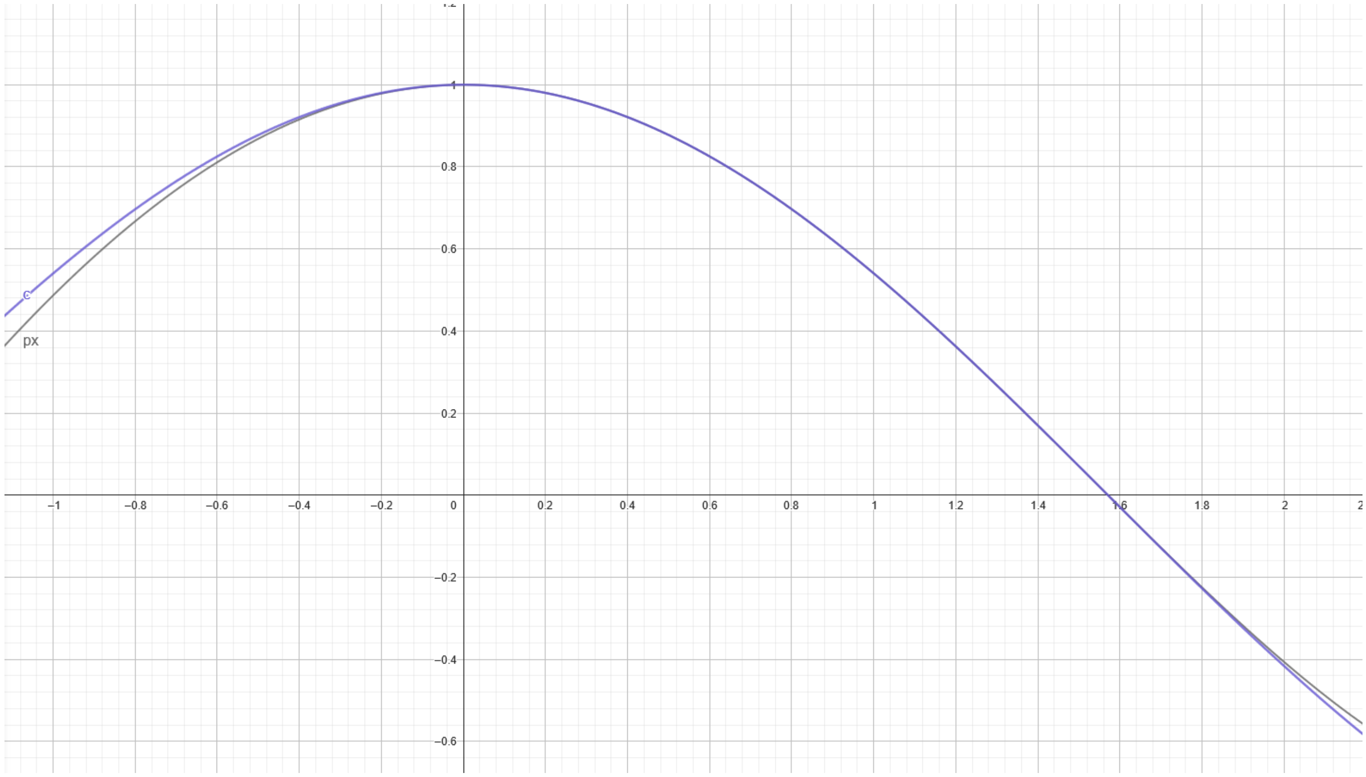
Here is fourth degree polynomial passing through the datapoints

$$p_x(t) = 1 + 0.0034t - 0.5152t^2 + 0.0234t^3 + 0.0288t^4$$

Here is an overview of the $p_x(t)$ and $\cos(x)$ plotted in GeoGebra. The grey is $p_x(t)$ and the blue graph is $\cos(x)$.



and here is the $p_x(t)$ and $\cos(x)$ plotted in GeoGebra with the interval $[-1, 2]$.



Here is

$$\text{The exact value of } \cos(1) = 0.5403023058681$$
$$\text{The computed interpolating polynomial } \; p_x(1) = 0.5403203144717$$

Consider the Lagrange polynomials $L_i(t), i = 1, \dots, n$

$$\begin{aligned}
L_1(t) &= \frac{(t-t_2)(t-t_3)\cdots(t-t_n)}{(t_1-t_2)(t_1-t_3)\cdots(t_1-t_n)},\\
L_2(t) &= \frac{(t-t_1)(t-t_3)\cdots(t-t_n)}{(t_2-t_1)(t_2-t_3)\cdots(t_2-t_n)},\\
&\vdots\\
L_i(t) &= \frac{(t-t_1)(t-t_2)\cdots(t-t_{i-1})(t-t_{i+1})\cdots(t-t_n)}{(t_i-t_1)(t_i-t_2)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_n)},\\
&\vdots\\
L_n(t) &= \frac{(t-t_1)(t-t_2)\cdots(t-t_{n-1})}{(t_n-t_1)(t_n-t_2)\cdots(t_n-t_{n-1})}.
\end{aligned} \tag{6}$$

Thus in the numerator and the denominator of $L_i(t)$ we multiply the terms $(t - t_j)/(t_i - t_j)$ for all $j = 1, \ldots, n$
except for $j = i$

> ✏ **NOTE: Extra info I added from the internet, mainly ChatGPT and https://en.wikipedia.org/wiki/Lagrange_polynomial**
>
> Lagrange interpolation is a method used to approximate a function $f(t)$ given a set of data points (t_i, β_i), where t_i are the input values and β_i are the corresponding function values or outcomes. The goal is to construct an interpolating polynomial $p(t)$ that passes through these data points and can be used to estimate $f(t)$ at any point t within the range of the data.
>
> In this sense, you can draw a parallel to regression models. In both cases, you're trying to fit a mathematical model to a set of observed data points. However, there are some key differences:
>
> 1. In Lagrange interpolation, you're fitting a polynomial function to the data points, whereas in regression models, you might use various types of functions
> 2. Lagrange interpolation is a deterministic method, meaning that the interpolating polynomial passes exactly through the given data points.
> 3. Lagrange interpolation can be used for a small number of data points, but it becomes numerically unstable for a large number of points.

Question 1.7. Determine the Lagrange polynomials L_1, L_2, L_3 corresponding to the dataset in point 1.5. Check that they satisfy the equations

$$L_i(t_i) = 1, \qquad L_i(t_i) = 0, \qquad \text{when } i \neq j$$

The Answer

Recall the points in 1.5. which are
t_1, β_1 are(1, 4),
t_2, β_2 are (2, 0)
and t_3, β_3 (3, 12)

The Lagrange polynomial L_1

$$L_1(t) = \frac{(t - t_2)(t - t_3)}{(t_1 - t_2)(t_1 - t_3)} = \frac{(t - 2)(t - t_3)}{(1 - 2)(1 - 3)} = \frac{(t - 2)(t - 3)}{2}$$

The Lagrange polynomial L_2

$$L_2(t) = \frac{(t - 1)(t - 3)}{(2 - 1)(3 - 1)} = -(t - 1)(t - 3)$$

The Lagrange polynomial L_3

$$L_3(t) = \frac{(t - 1)(t - 2)}{(3 - 1)(3 - 2)} = \frac{(t - 1)(t - 2)}{2}$$

Checking that L_1, L_2 and L_3 satisfy the equations

$$L_i(t_i) = 1, \qquad L_i(t_j) = 0, \qquad \text{when } i \neq j \tag{7}$$

Additional info from ChatGPT

$L_i(t_i) = 1$ **when** $t = t_i$: This means that when you plug in the value of t_i into the Lagrange polynomial $L_i(t)$, you should get 1. In other words, each Lagrange polynomial should "pass through" its corresponding data point vertically at $t = t_i$. This property ensures that the Lagrange polynomial perfectly interpolates the data points.

$L_i(t_j) = 0$ **when** $t \neq t_i$ **for** $i \neq i$: This means that when you plug in any other t_j (where j is not equal to i) into the Lagrange polynomial $Li(t)$, you should get 0. In other words, each Lagrange polynomial should be 0 at all other data points except its corresponding one. This property ensures that the Lagrange polynomial doesn't "interfere" with the other data points; it only affects its corresponding data point.

Testing L_1

$$\begin{aligned} L_1(1) &= \tfrac{(1-2)(1-3)}{2} = 1 \\ L_1(2) &= \tfrac{(2-2)(2-3)}{2} = 0 \\ L_1(3) &= \tfrac{(3-2)(3-3)}{2} = 0 \end{aligned}$$

Testing L_2

$$\begin{aligned} L_2(1) &= -(1-1)(1-3) = 0 \\ L_2(2) &= -(2-1)(2-3) = 1 \\ L_2(3) &= -(3-1)(3-3) = 0 \end{aligned}$$

Testing L_3

$$\begin{aligned} L_3(1) &= \tfrac{(1-1)(1-2)}{2} = 0 \\ L_3(2) &= \tfrac{(2-1)(2-2)}{2} = 0 \\ L_3(3) &= \tfrac{(3-1)(3-2)}{2} = 1 \end{aligned}$$

As one can see, L_1, L_2 and L_3 satisfy the equations in (7).

Question 1.8. Check that the interpolating polynomial you computed in point 1.5 may be written as a *linear combination* of Lagrange polynomials, that is

$$p_x(t) = \sum_{i=1}^{n} \beta_i L_i(i) \tag{8}$$

Explain, why this formula also holds in the general case. Hint: interpolating polynomial (of degree at most $(n-1)$ is unique! That is, if two polynomials (of degree at most $(n-1)$ agree at $t_1, t_2, \ldots, t_n$), they are identical.

The Answer

Recall this is the quadratic polynomial computed in 1.5.

$$p_x(t) = 24 + 28t + 8t^2$$

and the datapoints were

t_1, β_1 are(1, 4),

t_2, β_2 are (2, 0)

t_3, β_3 (3, 12)

To construct the interpolating polynomial as a linear combination of Lagrange polynomials we use the formula given in (8):

$$p_x(t) = \beta_1 L_1 + \beta_2 L_2 + \beta_3 L_3$$

Then we substitute L_1, L_2, and L_3 and β_1, β_2 and β_3

$$\begin{aligned} p_x(t) &= 4\left(\frac{(t-2)(t-3)}{2}\right) + 0\left(\frac{(t-1)(t-3)}{-1}\right) + 12\left(\frac{(t-1)(t-2)}{2}\right) \\ &= 2(t-2)(t-3) + 6(t-1)(t-2) \\ &= 8t^2 - 28t + 24 \end{aligned}$$

As one can see , the polynomial constructed using Lagrange polynomials is the same as the one constructed in 1.5.

Idk how to explain this, therefore, according to ChatGPT

The formula $\sum_{i=1}^{n} \beta_i L_i(t)$ holds in the general case because of the uniqueness of the interpolating polynomial.

The uniqueness of the interpolating polynomial states that if two polynomials of degree at most $n-1$ agree at n distinct points $t_1, t_2, \ldots, t_n$ then they are identical.

Given that our interpolating polynomial $px(t)$ passes through the points $(t_1, \beta_1), (t_2, \beta_2), \ldots, (t_n, \beta_n)$, we can express it as a linear combination of the Lagrange polynomials. This is because the Lagrange polynomials provide a set of basis functions that uniquely determine the interpolating polynomial.

Therefore, in the general case, the interpolating polynomial $px(t)$ can always be expressed as a linear combination of the Lagrange polynomials $Li(t)$, and this combination holds true due to the uniqueness property of the interpolating polynomial.

Question 1.9. Let us now denote the coefficients of Lagrange polynomials by c_{ij}. That is,

$$L_i(t) = c_{1i} + c_{2i}t + \cdots + c_{ni}t^{n-1} \tag{9}$$

Show that we have the equation

$$A \begin{bmatrix} c_{1i} \\ c_{2i} \\ \vdots \\ c_{ni} \end{bmatrix} = e_i$$

where e_i is the standard unit vector in $\mathbb{R}^n$. Hint: which dataset does the polynomial with coefficients $c_{1i}, \ldots, c_{ni}$ interpolate? See (9), (7).

The Answer

✎ **Resources Used**

This Wikipediae article: https://en.wikipedia.org/wiki/Vandermonde_matrix. It basically saying that matrix A is a Vandermonde matrix and is connect and used in combination with to Lagrange polynomials explained in this Wikipediae article: https://en.wikipedia.org/wiki/Lagrange_polynomial ChatGPT was also used

Using the hint: The Lagrange polynomial $L_i(t)$ interpolates the dataset $(t_1, 0), (t_2, 0), \ldots, (t_i, 1), \ldots, (t_n, 0)$, where $L_i(t_i) = 1$ and $L_i(t_j) = 0$ for all $j \neq i$ and . This follows from equation (7).

Recall the matrix A

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \ldots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \ldots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \ldots & t_n^{n-1} \end{bmatrix}$$

Now, consider the coefficients $c_{1i}, \ldots, c_{ni}$ in the Lagrange polynomial $L_i(t)$, as represented in equation (9). These coefficients determine the polynomial $L_i(t)$.

Then the polynomial $L_i(t)$.

$$L_i(t) = c_{1i} + c_{2i}t + \cdots + c_{ni}t^{n-1} \tag{9}$$

can express the polynomial $L_i(t)$ as a vector equation:

$$L_i(t) = \begin{bmatrix} 1 & t & t^2 & \ldots & t^{n-1} \end{bmatrix} \begin{bmatrix} c_{1i} \\ c_{2i} \\ \vdots \\ c_{ni} \end{bmatrix}$$

Where the column vector is a row of the matrix A

Here is the detailed calculation for getting the unit vector e_i from the coefficients of $L_i(t)$ and the matrix A

First we have the following equation given

$$A\begin{bmatrix} c_{1i} \\ c_{2i} \\ \vdots \\ c_{ni} \end{bmatrix} = e_i$$

Expanding on the LHS

$$\begin{aligned}
&= \begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^{n-1} \end{bmatrix} \begin{bmatrix} c_{1i} \\ c_{2i} \\ c_{3i} \\ \vdots \\ c_{ni} \end{bmatrix} \\
&= \begin{bmatrix} c_{11} & + & c_{12}t_1 & + & c_{13}t_1^2 & + & \dots & + & c_{1n}t_1^{n-1} \\ c_{21} & + & c_{22}t_2 & + & c_{23}t_2^2 & + & \dots & + & c_{2n}t_2^{n-1} \\ \vdots & & \vdots & & \vdots & & \ddots & & \vdots \\ c_{i1} & + & c_{i2}t_n & + & c_{i3}t_n^2 & + & \dots & + & c_{in}t_n^{n-1} \end{bmatrix} \\
&= \begin{bmatrix} L_1(t) \\ L_2(t) \\ \vdots \\ L_i(t) \end{bmatrix}
\end{aligned}$$

Recall equation (7)

$$L_i(t_i) = 1, \qquad L_i(t_j) = 0, \qquad \text{when } i \neq j \tag{7}$$

Hence the vector we get from the matrix-vector multiplication will have all its entries as 0 and expect one entry as 1 which is the i_{th} entry.

Therefore

$$\begin{bmatrix} L_1(t) \\ L_2(t) \\ \vdots \\ L_i(t) \end{bmatrix} = e_i$$

Question 1.10. Using the findings in 1.9, conclude that

$$AC = I$$

where the matrix C is given by

$$C = \begin{bmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \dots & c_{nn} \end{bmatrix}$$

Explain, in your own words, the action of the linear transformation $G : \mathbb{R}^n \to \mathbb{R}^n$ given by $G(y) = Cy$. That is, if $x = G(y)$, what is the relationship between x and y? Hint: if $x = G(y)$, what is $F(x)$?

The Answer

I remember reading that for product of two matrices to give the identity matrix, they have to be inverse of each other. Just like for the product of two numbers a and b to give 1, b has to be $b = 1/a$.

In 1.9. We found that for each Lagrange polynomial $L_i(t)$, the vector of its coefficients $[c_{1i}, c_{2i}, \dots, c_{ni}]^T$ (the transpose means that the vectors is a column not a row) multiplied by matrix A yields the standard unit vector e_i where e_i is the i_{th} column of the identity matrix I.

$$C = \begin{bmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \dots & c_{nn} \end{bmatrix} = [cL_1(t) \quad cL_2(t) \quad \dots \quad cL_n(t)]$$

where $cL_i(t)$ is represents the i_{th} column if C which is represents the coefficients of the polynomial $L_i(t)$. (note the the notation $cL_i(t)$ is made up by me).

Recall from lecture 4.1: Column interpretation of matrix-matrix product.

We can derive some additional insight into matrix multiplication by interpreting the operation in terms of the columns of the second matrix. Consider the matrix product of an $m \times p$ matrix A and a $p \times n$ matrix B, and denote the columns of B by b_k. Using block-matrix notation, we can write the product AB as

$$AB = A[b_1 \;\; b_2 \;\; \dots \;\; b_n] = [Ab_1 \;\; Ab_2 \;\; \dots \;\; Ab_n]$$

Thus, the columns of AB are the matrix-vector products of A and the columns of B. The product AB can be interpreted as the matrix obtained by "applying" A to each of the columns of B.

So in our case we have matrix A and and matrix C whose columns are the ***coefficients*** of the i_{th} Lagrange polynomial $L_i(t)$. Therefore multiplying A by C is essentially applying matrix A to each column vector of C.

$$AC = [A * cL_1 \quad A * cL_2 \quad \dots \quad A * cL_n] = [e_1 \quad e_2 \quad \dots \quad e_n] = I$$

Since each e_i is all zeros except for a 1 in the i_{th} position, AC essentially picks the i_{th} column from C, which is just $[c_{1i}, c_{2i}, \dots, c_{ni}]^T$. This is equivalent to the coefficients of the i_{th} Lagrange polynomial $L_i(t)$, which we showed in question 1.9.

---

2 - Bezier curves and PostScript fonts

Bezier curve is a piecewise-polynomial curve, where each piece is represented by cubic polynomials. In 2D we have a curve $(p(t), q(t))$, where $t \in \mathbb{R}$ is a parameter, such that $p(t)$ and $q(t)$ are piecewise cubic polynomials. We will denote by $[t_i, t_{i+1}]$ the parameter intervals, on which the curve is polynomial; thus we have the change from one polynomial to the next at $t_1, t_2, \ldots$. The values of $(p(t_i), q(t_i)), (p(t_{i+1}), q(t_{i+1}))$, and the slopes $(p'(t_i), q'(t_i)), (p'(t_{i+1}), q'(t_{i+1}))$ are controlled by the user, making it possible to approximate a variety of shapes. An example of a Bezier curve consisting of only one piece (so both coordinates are just cubic polynomials, and not piecewise-polynomials) is shown in Figure 2. Another example is shown in Figure 1, where the curve is clearly a piecewise-polynomial; one can see that the curve contains sharp "corners" corresponding to the places where even the first derivatives of the coordinates are discontinuous.

We shall focus on only one piece of the spline, as shown in Figure 2. The parameter t will vary in the interval $[t_1, t_2] = [0, 1]$. We put

$$\begin{aligned} p(t) &= p_1 + p_2 t + p_3 t^2 + p_4 t^3, \quad \text{and} \\ q(t) &= q_1 + q_2 t + q_3 t^2 + q_4 t^3, \end{aligned} \tag{10}$$

Thus we have 8 unknown coefficients $(p_1, \ldots, p_4, q_1, \ldots, q_4)$ to determine.

The *p-coefficients* can be determined independently from *q-coefficients*, and the procedure is exactly the same for both sets of coefficients. We will therefore focus only on *p-coefficients*.

The equations that $p(t)$ has to satisfy are

$$\begin{cases} p(0) = x_1, \\ p'(0) = 3(x_2 - x_1), \\ p'(1) = 3(x_4 - x_3), \\ p(1) = x_4, \end{cases} \tag{11}$$

where $(x1, \ldots, x4)$ are given by the user, for example see Figure 2. Note, that now the unknowns are called $(p1, \ldots, p4)$, while $(x1, \ldots, x4)$ are known constants.

Question 2.

Question 2.1. Write down the total matrix corresponding to the system of equations (11) for determining the unknown coefficients $(p_1, p_2, p_3, p_4) \in \mathbb{R}^4$. Solve the system (11) using Gaussian elimination to find the explicit formulas for $(p_1, p_2, p_3, p_4) \in \mathbb{R}^4$ in terms of the data $(x1, x2, x3, x4) \in \mathbb{R}^4$. Clearly the formulas for $(q_1, q_2, q_3, q_4) \in \mathbb{R}^4$ in terms of the data $(y_1, y_2, y_3, y_4) \in \mathbb{R}^4$ are analogous.

The Answer

First we the matrix from equation (11) we need the first derivative of $p(t)$

$$p'(t) = p_2 + 2p_3 t + 3p_4 t^2$$

Then the equation that construct the matrix are

$$\begin{aligned} p(0) &= p_1 = x_1, \\ p'(0) &= p_2 = 3(x_2 - x_1), \\ p'(1) &= p_2 + 2p_3 + 3p_4 = 3(x_4 - x_3), \\ p(1) &= p_1 + p_2 + p_3 + p_4 = x_4 \end{aligned}$$

We need to construct the augmented matrix but first here is the matrix of coefficients (lets call it M), the vectors of unknowns p and the RHS

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \qquad p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \qquad RHS = \begin{bmatrix} x_1 \\ 3(x_2 - x_1) \\ 3(x_4 - x_3) \\ x_4 \end{bmatrix},$$

What we are trying to solve is $M * p = RHS$. Therefore, the augmented matrix to perform Gaussian elimination on becomes

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & x_1 \\ 0 & 1 & 0 & 0 & 3(x_2 - x_1) \\ 0 & 1 & 2 & 3 & 3(x_4 - x_3) \\ 1 & 2 & 3 & 4 & x_4 \end{array}\right]$$

To be continued......