**Design and Calculating Bounded Area**

**Introduction**

1. **Part A Working with a Given Function and Designing with Functions**
   1. **Overview**

Part A of the task requires a graphic design that is clearly described mathematically. It also asks to integrate two given functions and into the design. A good mathematical expression given in this section will be important in Part B when evaluating the area. To achieve this, a series of methods will be introduced, and based on them, a rather complex not but overly elaborate pattern is expected.

* 1. **Calculate the area of the given region**

Given a function . The second function It is created by vertically translating down by 2 units. The region bounded by the two curves is

(see Figure 1)

To calculate its area, there are 2 methods.

|  |  |  |
| --- | --- | --- |
| This area can be evaluated by the formula:  This is a general formula to calculate the region bounded by a pair of functions in a given domain. To use this formula, the equation of is required. By the definition of translation, . |  | Figure 1. Region *D0* |
| An easier way: the region between 2 curves is formed by translation. Hence, when evaluating the integral in Eq. , every infinitesimal rectangle of the Riemann summation has the same length , so the total area is equal to the area of a 4.5 by 2 rectangle (see the orange rectangles in Figure 2). |  | Figure 2. Put all infinitesimal rectangles together |

* 1. **A rough sketch of the final design**
  2. **Design of region *D1***
     1. **Affine transformation and parametric equation**

The task asks to integrate the region between the given functions and into the final design, and it is encouraged to use transformations including reflection, translation, or dilations, which are **affine transformations**.

A simple definition of an affine transformation is to perform a linear transformation on a vector space, and additionally, a translation. Generally, an affine transformation on vector in can be written as

where is a 2 by 2 matrix representing a linear transformation, and represents a translation. The transformation of a function is more complicated. For this specific purpose, since what is transformed is the whole shape of a function , which means the actual subject of the transformation is every point on the plane, namely

This equation naturally satisfies the restriction on the **domain.** But note that after the transformation, the relation between and will probably no longer obey the restriction of a function (one-to-one correspondence). To better describe this kind of curve, **parametric equations** are introduced. Hence, a curve can be written as

where vector P records the position of a point on the curve at time t. Consequently, the whole curve is the trajectory of the point. According to Eq. 1.2, a general affine transformation on a parametric equation is

Parameterise is simply letting , so

And , the parametric form of , is given by transformation:

Now all kinds of transformation can be applied to. The region looks too long in the vertical direction. To compress vertically to 1/5 of the original size and stretch horizontally to four times the original size requires the scale transformation as in following table.

|  |  |  |
| --- | --- | --- |
| To compress vertically to 1/5 of the original size and stretch horizontally to four times the original size requires the matrix  By the way, if the graph is sclaed by a negative factor, the graph will be flipped.  Since there is no translation, b is the zero vector. |  |  |

There are two very good things about affine transformations. First, they are composable. For example, the transformation to first scale the given curves and , copy a pair of curves and , flip this pair of curves around the y-axis, translate them to align with the original curves, rotate them all by 180 degrees, and finally translate them to 10 units beneath the origin, can be written as

|  |  |
| --- | --- |
|  |  |

Another advantage of affine transformations is that the amount of the area changed in the transform can be easily calculated by the **determinant** of the transformation matrix. This will be further discussed in Part B.

* + 1. **Non-linear transformation and Jacobian**

Here, I want the shape to be more “enclosed”, by curlingit into a circular shape. This requires a **non-linear transformation**. A non-linear transformation does not guarantee that the parallel lines in the original graph remain parallel, and hence do not have a constant matrix to describe the transformation for every point on the Cartesian plane. A general way is to establish a **map**. For this purpose, the map is:

since the new graph’s x-coordinates have a range from -18 to 18. This **curl** transform gives a shape shown in Fig. 1.As a matter of convenience, after the transformation, the curves shown in Fig. 1 are named .

|  |  |
| --- | --- |
| Fig.1. |  |

Though this transform do not have a matrix to describe it, it do have a local linear behaviour when one looks it closely enough. This sort of linear behaviour can be measured by Jacobian. For transformation , the Jacobian is

The method to evaluate the transformed area based on the Jacobian will be further discussed in Part B.

* + 1. **Smoothness Consideration**

Generally, the smoothness can be measured by continuity , namely the nth derivatives at the two ends are equal. However, when dealing with a parametric curve, this approach will probably fail since the rate of change in position of a point with respect to t is not only related to the smoothness of the shape but also the smoothness of the motion (change in velocity). Thereby, the geometric continuity is introduced.

A curve or surface can be described as having continuity, with being the increasing measure of smoothness. Consider there are two curves, and their continuity is defined as:

: The curves touch at the join point.

: The curves also share a common tangent direction at the join point, which means the derivatives are collinear.

: The curves also share a common centre of curvature at the join point (Wikipedia, 2023).

Besides the conclusion , are smoother than , if n>m, it is also obvious that is stronger than ( ). For graph designs, is sufficiently smooth. Another property of continuity is that an affine transformation will not affect the original continuity, and actually, every continuous mapping can keep the original continuity.

To construct continuous joins, copy one set of curves shown in Fig. 1 and rotate them 180 degrees to get curves .

|  |  |
| --- | --- |
| Fig. 2. Graph A |  |

Curves give us Graph A as shown in Fig. 2, where there are four joins satisfying and continuity: , , , and . Take as an example.

**,**

,

Since the direction of the increment in the parameter is reversible in a parametric equation, Eq. 1.12 actually implies there is a continuity. By applying similar approaches to the other three joins, it can be found that all joins are continuous. Now, Graph A is good enough to be a part of the final design, and to form other graphs, some general methods are required.

**1.5 General modals to translate designs to mathematical expressions**

**1.5.1 Generate an interpolation curve from the given points based on the Discrete Fourier Transform**

Consider an arbitrary closed (end-to-end) parametric curve

Denote it as a complex function

Note that this complex function can be expressed as a sum of infinite simple orbiting complex numbers with different frequencies and complex coefficients to describe the phases and amplitude. Hence, there is

This series, known as the **Fourier series**, was proposed by Joseph Fourier in 1807. The reason why the frequency is is that for any closed curve, is true if and only if ; you can see this as a sort of period ( ).The **Fourier Transform** is used to transform a function from the original domain to the frequency domain, namely to get corresponding s for different frequencies s. For Eq. 1.13, the transform is

There are two perspectives to understand this equation.

1. Understand the Fourier Transform from the nature of Euler‘s formula

According to Eqs. 1.13 and 1.14.

According to **Euler’s formula** , the complex number really describes a “revolution”. Therefore, if integrate it through t, it will give zero. Proof:

Hence, all terms in Eq. 15 are killed except .

1. Understand the Fourier Transform from Hilbert space

**Hilbert space** is a vector space which have infinite dimensions. In this space, the vectors are functions, and there is an operation on them called the inner product, which is analogous to the dot product in the regular vector spaces. For example, in Hilbert space defined on , the inner product of two functions and is

where the bar stands for the complex conjugate. Thus, the Fourier transform is an inner product

Here is the interesting part. Taking any two different integers , there is

which means are orthogonal. Therefore, Eq. 1.18 is actually projecting onto infinite **orthogonal bases** , which gives the component of function in the direction of frequency , i.e. .

However, in the real-world problem, it is unlikely to get a continuous curve, and most of the time, we have to deal with discrete sample points. Suppose there are discrete points on a curve in complex form:

Apply the **Discrete Fourier Transform** on it,

And finally, the curve is formed by the sum of the first terms of the Fourier series:

A higher can give a more accurate fitting of the original curve, and a relatively small can filter the high-frequency details and give a gentler (no steep wiggle) shape. For example, 1000 sample points on the contour of the shape shown in Fig. 3 are extracted as shown in Fig. 4.

|  |  |
| --- | --- |
| C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png  Fig. 3 | C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_contour.png  Fig. 4 |

Using python to do some repeated numerical calculation, the graph keeping different number of terms are obtained

|  |  |  |
| --- | --- | --- |
|  |  |  |
| C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_2.png | C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_10.png | C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_50.png |
|  |  |  |
| C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_100.png | C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_200.png | **C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\badapple2.png_1000.png** |

**Bibliography**

Wikipedia. (2023). *Smoothness*. [online] Available at: https://en.wikipedia.org/wiki/Smoothness.

3Blue1Brown (2019). *But what is a Fourier series? From heat flow to drawing with circles*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=r6sGWTCMz2k&list=PL4VT47y1w7A1-T_VIcufa7mCM3XrSA5DD&index=4>.