American University of Armenia

Numerical Analysis Project
November 2016

Project: Trigonometric Interpolation

Fall 2016

Sargsyan Grigor

Aim of the project

So far we have only discussed interpolations for functions, which are not periodic. But if the function is periodic or if we at least know it should have acted as a periodic function it will be a good choice to choose a trigonometric function instead of a algebraic polynomial to interpolate the given function at the given data points, or if the function is not periodic at the moment, but we are sure it has to repeat its values with some periodicity. The aim of the project is to discuss the method of trigonometric interpolation. First we will define, then consider the properties and examples of trigonometric interpolation, and we will introduce interpolation using complex numbers. We will also discuss a simple real life problem example and compare trigonometric interpolation with algebraic polynomial interpolation. After reading this project paper the reader should be able to:

- understand the concept of Trigonometric interpolation,
- distinguish cases in interpolation and bring examples of real life problems, where trigonometric interpolation is the most useful,
- construct an interpolating function including only sines and cosines such that the function will pass through the given data points of another function.

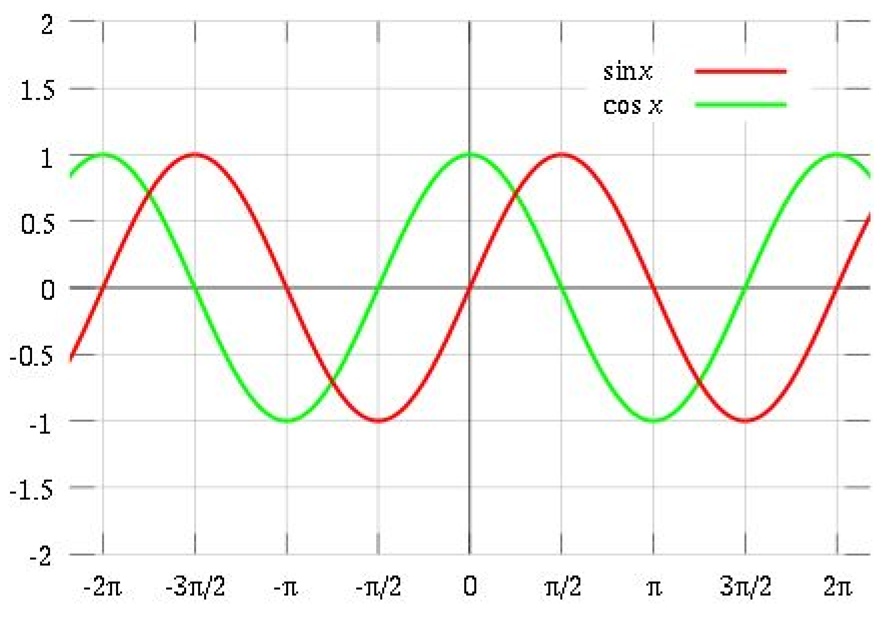
Contents

Chapter 1

Introduction

1.1 About trigonometric functions

Trigonometric functions although are known from the ancient times, but have various applications nowadays. The most famous of these functions are **sine**, **cosine** and **tangent**, which are defined on a right angled triangle. Besides showing the relation of an angle and other measures(for example the sides of the triangle), another main reason of being so popular and useful is their property of being periodic with a certain constant periodicity.



As we will see in chapter 4 there are many physical phenomena which are modeled using sin, cos and tan.

Trigonometric polynomial is defined as finite sum of the functions $sin(nx)$ and $cos(nx)$ (degrees of trigonometric functions should be 1 and coefficients of x can be any real numbers. The general formula of trigonometric polynomial is given by

$$t(x) = a_0 + \sum_{n=1}^{N} a_n cos(nx) + \sum_{n=1}^{N} b_n sin(nx) .$$

If we rewrite the above as

$$t(x) = a_0 + \sum_{n=1}^{N} a_n cos(nx) + i \sum_{n=1}^{N} b_n sin(nx)$$

it will be *complex trigonometric polynomial*.

Besides many usefulnesses and applications of trigonometric functions, in this project we will be concentrating on constructing a *trigonometric interpolator.* In other words we will discuss how to find a trigonometric polynomial that will approximate another function most accurately.

1.2 Formulation of the problem

Just like in other forms of interpolation we are given a set of data points x_n's and and a function $y = f(x)$, and our aim is to construct a trigonometric polynomial $t(x)$ that will pass through all the points

$$x_n,\ n \in \{0,\ 1,\ 2,\ \ldots,\ N-1\}$$

(total N points). That is

$$t(x_n) = f(x_n),\ \ n \in \{0,\ 1,\ 2,\ \ldots,\ N-1\} .$$

As we already saw a trigonometric polynomial has the form

$$t(x) = a_0 + \sum_{k=1}^{K} a_k cos(kx) + \sum_{k=1}^{K} b_k sin(kx)$$

with degree K .

Now our goal is to compute these coefficients $a_0,\ a_1,\ a_2, \ldots, a_K$ and $b_1,\ b_2, \ldots, b_K$. That is a total of $2K+1$ numbers to complete the trigonometric polynomial. Say we want to interpolate the function $f(x) = x^2$ with three nodes - $x_0 = 1,\ x_1 = 2,\ x_2 = 3$, *i.e.* we want to find a trigonometric interpolant $t(x)$ such that $t(1) = f(1) = 1$, $t(2) = f(2) = 4$ and $t(3) = f(3) = 9$. That is

$$t(x) = a_0 + a_1 cos(x) + a_2 cos(2x) + a_3 cos(3x) + b_1 sin(x) + b_2 sin(2x) + b_3 sin(3x)$$

Thus we will have to calculate $2 \cdot 3 + 1 = 7$ coefficients in order for our new function to pass through all the 3 data points. But if we write this as a system of equations, then we will have 3 equations with 7 unknowns, which is impossible to solve in most cases. But as we will see in the next chapter this problem can be reduced to a much simpler one when involving complex numbers.

Chapter 2

Constructing Trigonometric Interpolator

2.1 About the solution of the Problem

We defined the form of the trigonometric polynomial as

$$t(x) = a_0 + \sum_{k=1}^{K} a_k cos(kx) + \sum_{k=1}^{K} b_k sin(kx)$$

Since both sin and cos are 2π periodic, its easy to see that the given nodes(data points) can correspond to a number in the semi open interval $[0; 2\pi)$ or $(0; 2\pi]$. The existence of the $t(x)$ depends on how large is N. There is always a solution when $N \leq 2K + 1$, that is the number of data points is less or equal to the number of coefficients. There is a unique interpolator $t(x)$ if and only if $N = 2K + 1$. Otherwise the existence of the solution will depend on the chosen x_n's. These theorems follow immediately after introducing complex numbers and formulating the problem in the complex plane. In general we don't require that the given nodes are equally spaced in $[0; 2\pi]$ interval. But in case of equidistant nodes will have special case and a simpler problem which will be discussed later.

2.2 The Problem in the Complex plane

Recall Euler's formula that for any real θ we have

$e^{i\theta} = cos(\theta) + isin(\theta)$

Where i is the imaginary unit. It follows from this formula that

$sin(x) = \frac{e^{ix} - e^{-ix}}{2i}$ and $cos(x) = \frac{e^{ix} + e^{-ix}}{2}$.

Therefore we can rewrite our formula for trigonometric polynomial as

$t(x) = \sum_{k=-K}^{K} c_k e^{ikx}$

Where $c_k = \frac{1}{2}(a_k + ib_k)$ and $c_{-k} = \frac{1}{2}(a_k - ib_k)$.

If we let $e^{ix} = z$, the form of $t(x)$ will be $\sum_{k=-K}^{K} (c_k \cdot z^k)$. This clearly reduces our problem to finding a polynomial interpolant. Also the properties of polynomial interpolation will apply to trigonometric interpolation, for example the existence or the uniqueness. Moreover, de Moivre's formula proves that our function $t(x)$ will be a trigonometric polynomial, as $e^{ni\theta} = cos(n\theta) + isin(n\theta)$.

2.3 Solution of the Problem

2.3.1 Odd number of data points

Now that we defined the form of our interpolant in the complex plane let's study its behaviour. Recall that for algebraic polynomials we use Lagrange polynomial to find the interpolating function. Lagrange polynomial was defined by

$$L(x) = \sum_{i=0}^{n}(f(x_i) \cdot l_i(x)), \quad \text{where } l_i(x) = \prod_{0\le j\le n} \frac{x-x_j}{x_i-x_j} .$$

If the number of given data points is odd, that is $N = 2K + 1$, using the Lagrange polynomial formula our function will be in the form

$$t(x) = \sum_{k=0}^{2K}(f(x_k) \cdot l_k(x)), \text{ where } l_k(x) = e^{-iKx+iKx_k} \prod_{m=0,m\neq k}^{2K} \frac{e^{ix}-e^{ix_m}}{e^{ix_k}-e^{ix_m}} .$$

This yields that $l_k(x) = \prod_{m=0,m\neq k}^{2K} \frac{sin\frac{1}{2}(x-x_m)}{sin\frac{1}{2}(x_k-x_m)}$.

After finding our desired interpolant in this form we can eventually get to trigonometric polynomial using trigonometric identities.

2.3.2 Even number of data points

Now suppose we have even number of nodes, *i.e.* $N = 2K$ nodes. In this case we will have

$$l_k(x) = e^{-iKx+iKx_k} \frac{e^{ix}-e^{i\alpha_k}}{e^{ix_k}-e^{i\alpha_k}} \prod_{m=0,m\neq k}^{2K-1} \frac{e^{ix}-e^{ix_m}}{e^{ix_k}-e^{ix_m}} \quad .$$

Which is equivalent to saying that

$$l_k(x) = \frac{sin\frac{1}{2}(x-\alpha_k)}{sin\frac{1}{2}(x_k-\alpha_k)} \prod_{m=0,m\neq k}^{2K} \frac{sin\frac{1}{2}(x-x_m)}{sin\frac{1}{2}(x_k-x_m)}$$

, where α_k's can be any number.

Chapter 3

Applications in life

3.1 Usefulness and realizations of Trigonometric Interpolations

As already mentioned above interpolating by a trigonometric polynomial has its full usefulness when the function we are interpolating is periodic function with a certain periodicity T ($f(x) = f(x+T) = f(x-T)$). Such conditions are met in navigation, physics, engineering and in other spheres as well. There are many physical phenomena that are easy to model using periodic functions. For example sound or light waves, length of the day, sunlight intensity or average temperature throughout a year are all described and studied using trigonometric functions.

3.2 Real life Problem Example

Suppose we want a function that will approximate the temperature of the year, if we are given some data points. But as we know the temperature repeats every year with about same periodicity we know for sure that our interpolating function must be a periodic one, *i.e.* we will have to do trigonometric interpolation to achieve our desired function. Say we have the following nodes:

$x_0 = 1$ $x_1 = 100$ $x_2 = 200$

$y_0 = -10$ $y_1 = 5$ $y_2 = 35$

This means that we measured the temperature at day 1 and got a result of -10. In other words at January 1 the temperature was $-10^o C$ etc..

Now let's construct a trigonometric polynomial using the techniques we learned in previous chapters. In this case we have odd number of data points $N = 2K + 1$. Applying our data to our formula we get

$$t(x) = 1 \cdot \frac{sin\frac{1}{2}(x-100)}{sin(\frac{99}{2})} \cdot \frac{sin\frac{1}{2}(x-200)}{sin(\frac{199}{2})} - 100 \cdot \frac{sin\frac{1}{2}(x-1)}{sin(\frac{99}{2})} \cdot \frac{sin\frac{1}{2}(x-200)}{sin(50)} + 200 \cdot \frac{sin\frac{1}{2}(x-1)}{sin(\frac{199}{2})} \cdot \frac{sin\frac{1}{2}(x-100)}{sin(50)}$$

This is not the best approximation of the temperature of the year because we will have some points where the temperature is more then $300^o C$. Also we will have not real rapid changes from day to day. But that problem is a matter of choosing good number of data points and with good distances, but that topic is beyond the scope of our project. This proved that our choices were bad, but the idea here is that if we did a polynomial interpolation instead we would have worse approximation. Now we have maximum temperature degree is $< 400^0 C$, not very useful approximation but at least it's not infinity as it would be in polynomial interpolation.

Chapter 4

Further Studies, not covered in the Project

Tightly connected topics to trigonometric interpolations are Discrete Fourier Transform(DFT), Fast Fourier Transform(FFT) etc.. For example DFT transforms a real valued function to a complex valued function of frequency and can be used to solve trigonometric interpolation when nodes are equidistant, with a faster algorithm. FFT can be used in special cases too for rapidly calculating the coefficients. Using FFT it has been derived that the coefficients c_k in the equation

$t(x) = \sum_{k=-K}^{K} c_k e^{ikx}$ can be calculated

by the formula $c_k = \frac{1}{2\pi} \int_0^{2\pi} f(x)\, e^{-ikx} dx$.

In the form of $t(x) = a_0 + \sum_{k=1}^{K} a_k cos(kx) + \sum_{k=1}^{K} b_k sin(kx)$, the coefficients $a_k,\ b_k$ can also be calculated by the formulas $a_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)\, cos(kx)\, dx$, and similarly

$b_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)\, sin(kx)\, dx$, where T is the periodicity of our initial function $f(x)$.

References

https://en.wikipedia.org/wiki/Trigonometric_interpolation