LOG(M) PROJECT ZERO

YOUNGWOO KWON

There are many complex equations with multiple solutions in the world. Today, we will learn how to find solutions of the specific polynomial called Orthogonal Polynomial.

1. Introduction

What does the orthogonal polynomial mean? Two things are called orthogonal if they disturb everything with each other, make their summation of the product equals zero. Let's write down this in a mathematical way. There is a polynomial called P(x), and there is a polynomial called Q(x). The product of to polynomials is P(x)*Q(x), and if we integrate, sum all of them in a certain range, those things, it must equal to zero if we want two say those are orthogonal. Mathematicians write these sentences as 'P and Q are orthogonal if and only if $\int P(x) * Q(X) = 0$'. They also add w(x) in this equation, called weight function. So the final definition of the orthogonal would be 'P and Q are orthogonal if and only if $\int P(x) * Q(X) * w(x) = 0$' [2]

As Q or w becomes more complicated, P will also become more complicated accordingly. Our brains will also be complicated to get the polynomial P. One good way to get this P is just let our computers do all the stressful works. Provide a Q and w(x) to our computer, and say "give me the orthogonal polynomial of Q!" This is a very, very good way, but unfortunately, computer usually don't know what the orthogonal polynomial is!

After all, minimal translation is essential. One good translation is to change the integration equation formula into matrix equation formula. Using some theories, we can find some μ_ks from w(x) and make a matrix to find P using μ_ks. This is also a good way - at least the computer knows what the matrix equation is. One problem here is, the computer may do the math roughly. If the computer does the work roughly and the precision is not high enough, we may not get the polynomial P we want.

Another way to get a polynomial P is to process a little bit of calculation using theories and then throw the equation in a form that is a little bit easier for the computer to calculate. Thanks to this little consideration, computers can show higher precision than before, and the result will be more like the polynomial P we want.

Of all the considerations we can do, what we're going to look at today is using a recursion relationship to give a better equation to find P. Assume there are many Qs with the relationship, such as $e^{ix}, e^{2ix}, e^{3ix}, ...$ There must be a corresponding orthogonal polynomial P for each Q. What if there is also a relationship between these P? If the relationship between P can provide less complicated equation to find Ps than $\int P(x) * Q(X) = 0$ for unknown P, we can use this relationship to find more precise P. [1, Chapter 2]

We find $x^2+7x = 8$ more difficult than $2x+1 = 3$, and $x^3-3x^2+3x+2 = 3$ more difficult than $x^2+7x = 8$. Similarly, If the degree of the P increases, the solution of P from the computer has less precision. We are working on how to make our solutions more precise in higher n. The two methods introduced above are good ways to make the answer precisely, and mathematicians keep working on theories that can give us better precision.

References

[1] Yang Chen Estelle Basor and Torsten Ehrhardt. Painlevé v and time dependent jacobi polynomials. *Journal of Physics A: Mathematical and Theoretical*, 43(1), 2009.

[2] Gabor Szegö. *Orthogonal polynomials by Gabor Szegö*. 1937.