Sarthak Agarwal

Professor Mehrdad Khosravi

Math 2A

16 March 2019

Applications of Differential Equations Computer Science and Engineering

Richard Courant defines “mathematics as an expression of the human mind reflects the active will, the contemplative reason, and the desire for aesthetic perfection. Its basic elements are logic and intuition, analysis and construction, generality and individuality.” The basics elements stated are essential pillars in the field of mathematics but are also the fundamental building blocks of Computer Science and Engineering. Dynamical systems and differential equations are an integral part of Applied mathematics, and therefore their applications in the field of Computer Science and engineering are immense. From math used in building complex units required in the making of the processor to the math involved in coding complex algorithms, computer engineers use differential equations as a tool to find solutions. In a world where Artificial Intelligence and Machine learning have taken over many tasks that were done by humans, we need very accurate results and approximations. Many tools associated with Differential equations, enable us to get the best results to meet the expected results that lead us in the direction of achieving greater objectives in various fields. One of the areas that integrate the concepts from mathematics like Differential Equation and computer science is Numerical analysis.

Numerical analysis is the area of mathematics and computer science that creates, analyzes, and implements algorithms for solving the problems of continuous mathematics numerically. These real-world problems originate from real-world applications of algebra, calculus, etc., where the values of used variables and sets change continuously depending on factors like time. With the development of computer science and the availability of digital computers realistic we are capable of solving complex mathematical models in science, engineering, and business. With the demand for computers in such sophisticated fields, the use of numerical analysis has also increased exponentially. Because of the high risks and stakes involved in the area mentioned above, there is a need to solve these mathematical models from the real world to get the results to the highest level of precision and accuracy.

Mathematicians and engineers that work closely with numerical analysis have a variety of tools in their tool kit which they use to manipulate the mathematical problems to integrate the solution with the code and machinery to get things to work. There are three categories into which numerical analysts divide the equations into:

- ***Ordinary differential equations***
    - This systems of differential equations have the solution which are functions of only a single variable.
    - Examples:
        - Differential-algebraic equations are mixed systems of algebraic equations and ordinary differential equations.

        - Delay differential equations: Equations that help us find the rate of change of the solution depending on the state of the system at past times.

- ***Partial differential equations***
    - Includes differential equations in which the unknown solution is a function of more than one variable.
    - These equations occur in almost all areas of engineering, and many basic models of the physical sciences are given as partial differential equations.
    - Example:
        - Many equations that deal with fluid dynamics (EX: Navier-Stokes equations)
- ***Integral equations***
    - These equations involve the integration of an unknown function and linear equations.
    - Example:
        - The radiosity equation is a model for radiative heat transfer.

Sometimes solving a differential equation is practically impossible and approximations become the only way out. Approximations are an essential part of how the computer is built and how they compute solutions. From the straightforward tasks like calculation of arithmetic problems to estimates used in rocket science, we use approximate values. It is just about the precision to which a computer can approximate. At a fundamental level of finding solutions to differential equations, Euler's method comes handy. This first-order numerical procedure for solving ordinary differential equations with a given initial value is the most basic explicit method for numerical integration and helps computer science engineers to figure out the errors and approximate the benefits that they use to build software and hardware for specific research or product. However, Euler's method has its limitations and fails to approximate solutions in various cases. In complex situations, computer science engineers rely on Differential equation techniques and approximation theory and approach to fetch values, that eventually helps them to program the required software. The methods use computable functions p(x) to approximate the values of functions f(x) that are not easily computable or use approximations to simplify dealing with such complex functions and situations.

During the lectures, we have seen that Fourier and Laplace's transform make our lives easier by helping us manipulate a function of a real variable to a function of a complex variable and vice versa. Digital devices like oscilloscopes, computers, etc. receive information and commands in the form of signals which are in the form of wavelets and Fourier and Laplace play an essential role in manipulating and transforming these signals according to the requirement like filtering the frequency of the input signal/ waveform.

The Fourier Transform is defined as :

The Fourier transform (FT) of the function $f(x)$ is the function $F(\omega)$, where:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x}\, dx$$

and the inverse Fourier transform is

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x}\, d\omega$$

Recall that $i = \sqrt{-1}$ and $e^{i\theta} = \cos\theta + i \sin\theta$.

The Laplace Transform is defined as :

$$L\{f(t)\} = \bar{f}(s) = F(s) = \phi(s) = \int_0^{\infty} e^{-st} f(t) dt$$

The Fast Fourier Transform (FFT) is an application of the Fourier transform and is also one of the most essential algorithms in signal processing and data analysis. According to Wikipedia, FFT's importance derives from the fact that in signal processing and image processing it has made working in the frequency domain equally computationally feasible as working in the temporal or spatial domain.

Derivation of the *Fast Fourier Transform* algorithm

We all assume that the length of list is a power of 2.

That is:

$$f = (f[1], f[2], \ldots, f[n]) \; ; \; n = 2^k \text{ for some } k.$$

We will also use complex exponentials

Specifically

$$\omega[p,q] = e^{2\pi i q/p}$$

here p = length of list

q = relation to Index

We write discrete Fourier transform of f by specifying m^{th} term:

$$\mathcal{F}f[m] = \sum_{k=0}^{n-1} f[k]\, \omega[n,-km]$$

Specifically we write sum above as the 2 sums below:-

$$\mathcal{F}f[m] = \sum_{k=0}^{n/2-1} f[2k]\, \omega[n,-2km] + \sum_{k=0}^{n/2-1} f[2k+1]\, \omega[n,-(2k+1)m]$$

Now we re write these pieces as Fourier transform.
We must replace the occurences of n in complex exponentials with $n/2$.

$$\{\omega[n,-2km] = e^{-2\pi i km/n} = \omega[n/2,-km]$$

For Second summation, we observe that

$$\omega[n,-(2k+1)m] = \omega[n,-2km] \cdot \omega[n,-m]$$

Now we have

$$\mathcal{F}_f[m] = \sum_{k=0}^{n/2-1} f[2k]\,\omega[n/2, -km] + \omega[n,-m] \sum_{k=0}^{n/2-1} f[2k+1]\,\omega[n/2, -km]$$

We denote list of even-index enteriesof f by f_{even} and vice versa for f_{odd}, we see that we have a combination of 2 Fourier Transforms of length $n/2$,

That is,

$$\mathcal{F}_f = \mathcal{F}_{f_{even}} + \omega[n,-m]\,\mathcal{F}_{f_{odd}}$$

Recalling our primer on discreet Fourier transform, we naturally extended the signal involved. So indeed, the length-$n/2$ Fourier transforms satisfy the following identity for each m

$$\mathcal{F}_{f_{even}}[m] = \mathcal{F}_{f_{even}}[m+n/2]$$
$$\mathcal{F}_{f_{odd}}[m] \quad \mathcal{F}_{f_{odd}}[m+n/2]$$

Plug in $m+n/2$ in above summations;
we get,

$$\mathcal{F}_f[m+n/2] = \sum_{k=0}^{\frac{n}{2}-1} f[2k]\,\omega[n/2 - (m+n/2)k] + \omega[n, (m+n/2)] \sum_{k=0}^{n/2-1} f[2k+1]\,\omega[n/2 - (m+n/2)k]$$

Now we can use the easy to prove identity

$$\omega[n/2 - (m+n/2)k] = \omega[n/2 - mk]\,\omega[n/2, -kn/2]$$

We can see right hand term is $e^{2\pi i k} = 1$

$$\omega[n, -(m+n/2)] = \omega[n,-m]\,\omega[n,-n/2] = -\omega[n,-m]$$

this simplyfying the massive formula above to more familiar form:-

$$\mathcal{F}_f[m+n/2] = \sum_k f[2k]\,\omega[n/2, -mk] - \omega[n,-m] \sum_k f[2k+1]\,\omega[n/2, -mk]$$

We Finally have: $\mathcal{F}_f[m] = \mathcal{F}_{f_{even}}[m] + \omega[n,-m]\,\mathcal{F}_{f_{odd}}[m]$

$\mathcal{F}_f[m+n/2] = \mathcal{F}_{f_{even}}[m] - \omega[n,-m]\,\mathcal{F}_{f_{odd}}[m]$

Base Case is $\mathcal{F}([a]) = [a]$

Fast Fourier Transform (FFT algorithm) has revolutionized the computer science, networking industry and most importantly transformed our lives by efficiently making signal processing more effective. “Digital audio and video, graphics, mobile phones, radar and sonar, satellite transmissions, weather forecasting, economics, and medicine all use the Fast Fourier Transform algorithm crucially.” (*The Fast Fourier Transform*)

Some fields where the FFT algorithm can be used are:

- Fast large integer and polynomial multiplication
- Filtering algorithms
- Fast algorithms for discrete cosine or sine transform

Apart from learnings from the research, one can also visualize the use of the differential equations in the field of ‘Data Structures’ which is an integral part of computer science related studies. A hash table is a data structure that implements an associative array abstract data type, a structure that can map keys to values. This data structure makes a programmer fetch and store data in a very organized manner using a technique called hashing. In today's date, hash tables are extensively used in the industry because of the security and organization features it carries with itself. We can use ordinary differential equations for work related to a hashing function, an active topic in computer algorithm research. A hashing function that is a result of a differential equation that may arise from equations like Euler Lagrange equations can be used to optimize the hashing algorithm and in turn, making the hash function more efficient and effective in storing, fetching and searching the data.

I believe that computer science is a branch of a tree called mathematics. Differential equations and their applications form the foundation of applied mathematics, and therefore it has a significant impact on both research and applications in the field of computer science. Over the years, the applications of the differential equations, like FFT algorithm, hashing methods, applications in the field of numerical analysis, etc., have revolutionized the computer science and engineering industry. I would use the concepts of Differential equations from the class and the research for the project to develop a better and stronger understanding of computer science and engineering that would help me prepare for a brighter prospect in my future endeavors.

Works Cited

"Richard Courant Quotes." *BrainyQuote*, Xplore, www.brainyquote.com/quotes/richard_courant_205929?src=t_mathematics.

Atkinson, Kendall E. "Numerical Analysis." *Scholarpedia*, 2007, www.scholarpedia.org/article/Numerical_analysis#Systems_of_linear_and_nonlinear_equations.

Heckbert, Paul. "Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm." www.cs.cmu.edu, 27 Jan. 1998, www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf.

"Fast Fourier Transform." Wikipedia, Wikimedia Foundation, 16 Mar. 2019, en.wikipedia.org/wiki/Fast_Fourier_transform#Applications.

j2kun. "The Fast Fourier Transform." Math ∩ Programming, 6 Feb. 2017, jeremykun.com/2012/07/18/the-fast-fourier-transform/.