IASET: Journal of Applied Mathematics &
Statistical Sciences (IASET: JAMSS)
ISSN(P): Applied; ISSN(E): Applied
Vol. 1, Issue 2, Jul - Dec 2016; 1- 14

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

NUMERICAL COMPUTATIONS FOR THE DESIGN OF ELECTRONIC MAIL BOXES ON CANTOR SET

AHMED BUSERI ASHINE

Madda Walabu University, Department of Mathematics, Bale Robe, Ethiopia

ABSTRACT

This paper extends the algorithm introduced by **Error! Reference source not found**. By using the Cantor sets and cubic spline interpolating function in the design of electronic mail boxes. The cantor sets was introduced as the domain of the function for the mail design while spline functions were used as the formula. The password of the mailbox was calculated in line with that of cantor set of intervals and spline interpolating functions in respective of the governing polynomial function of degree N – 1. The software package termed as MATLAB was in a position to design and calculate the intended numerical values. Finally, the Newton-Raphson Method was used for the computational of the password and mathematically the interpretations were given.

KEYWORDS: Cantor set, spline, Newton-Raphson Method, Electronic Mail Design

1. INTRODUCTION

1.1. Background of the Study

Electronic mail is very important at present in terms of its wide or widely accepted medium of interpersonal communication for many years. It have ample amount of applications, especially in transaction of business activities such as people's communication, military activities, academics and others. Many researchers have been working in designing, improving, securing, making connections so fast, to benefit the good usage of electronic mail. For instance, [1] addressed the impact of electronic direct mail on the design of the messages using Chi square distribution**.** [2] studied ways of combating the corporate paper war: Electronic Mail Abuse paper war and electronic mail abuse. As cited on [3], [4] proposed two families of protocols to certify electronic mail with enabling to exchange a receipt extracting the ideas from [4] and [1]. [5] examines the research and develop a prototype object-based multimedia electronic mail system based on the ideas taken from [4] and [2].

Moreover, [3] came with the paper that concerns for designing new proposed algorithm for the design of electronic mail. Different mail serves have different mechanisms to control the customers mailing activities by designing their own system controlling algorithms. Hence, three ideas were composed in the design, Cantor sets, spline, and Newton-Raphson's method. As indicated on [3], Cantor sets have good topological properties represented in bounded, closure, compactness, measurable, infiniteness, and countable. So, it was used as the area (or domain) of the design. For the smoothness of numerical spline method [6], it issued as a functioning or controlling the design. For fast timer and less error, Newton-Raphson's methods were used for the computation of the approximated roots of the governing interpolating polynomial function which was derived from the cubic spline interpolating functions. The user name was served as initial as initial point while the roots were used as the password.

The Cantor set has many definitions and many different constructions. Although Cantor originally provided a purely abstract definition, the most accessible is the Cantor middle-thirds or ternary set construction. Begin with the closed real interval [0, 1] and divide it into three equal open intervals. Remove the central open interval $I_1 = \left(\frac{1}{3}, \frac{2}{3}\right)$ such that

$[0,1] - I_1 = [0,1] - \left(\frac{1}{3}, \frac{2}{3}\right)$

Next, subdivide each of these two remaining intervals into three equal open intervals and from each remove the central third. Let I_2 be the removed set, then $I_2 = \left(\frac{1}{3^2}, \frac{2}{3^2}\right) \cup \left(\frac{7}{3^2}, \frac{8}{3^2}\right)$ and $[0,1] - (I_1 \cup I_2) = \left[0, \frac{1}{3^2}\right] \cup \left[\frac{2}{3^2}, \frac{2}{3^2}\right] \cup \left[\frac{6}{3^2}, \frac{7}{3^2}\right] \cup \left[\frac{8}{3^2}, 1\right]$.

We can then subdivide each of the intervals that comprise $[0,1] - (I_1 \cup I_2)$ in to three subintervals, removing their middle thirds, and continue in the previous manner. The sequence of open sets I_n is then disjoint, and we traditionally define Cantor set C as the closed interval with the union of these I_n subtracted out. That is, $C = [0,1] - \cup I_n$.

And moreover, the Cantor set C is perfect and totally disconnected, nonempty, closed and nowhere dense, and uncountable. Although our construction of the Cantor set in the first section used the typical middle-thirds or ternary rule, we can easily generalize this one-dimensional idea to any length other than $\frac{1}{3}$, excluding of course the degenerate cases of 0 and 1.

The studies done by [3] shows as that the Newton-Raphson Method (or simply Newton's Method) is another so-called local method to determine the root of an equation or function. This method uses a single starting point (as opposed to the bounds required by the bisection method), and repeatedly uses a derivative to project a line to the axis of the root in question.

The Electronic Mail Design

The Electric Mail Designer focuses on two terms, the (ID) and the password. It is clearly the (ID) is public while the password should be top secret. So, suitable mathematics must be used carefully for issue sub-domain for each mail which is not related with other mails and it is not possible to insert other domain in the sub-domains series.

Two producers were introduced, the one, named as (send message) used for putting the message in box mail while the second, named as (open mail) used for owner mail.

When a function f defined on the interval $[x_0, y_N]$ and a set of nodes $\{x_0, x_1, ..., x_N\}$ such that $a = x_0, x_1, ..., x_N = b$.

A cubic spline interpolating S for f is a function that satisfies the following conditions [3]:

1. $S(x)$ is a cubic polynomial, denoted $S_i(x)$ on subintervals $[x_i, x_{i+1}]$ for each $0,1,..., N-1$

2. $S(x_i) = f(x_{i+1})$ for each $i = 0,1,2,...., N-2.$

3. $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$ for each $i = 0,1,2,...., N-2.$

4. $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})$ for each $i = 0,1,2,...., N-2.$

5. $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})$ for each $i = 0,1,2,...., N-2.$

6. $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$ for each $i = 0,1,2,...., N-2.$

7. One of the following set of boundary conditions is satisfied

   $S''(x_0) = 0 = S''(x_N)$ for free or natural boundary and or each $i = 0,1,2,...., N-2.$

   $S'(x_0) = f''(x_0)$ and $S'(x_N) = f''(x_N)$ for coupled boundary.

   Remark: To construct the cubic spline interpolating S for the function f which on the values.

   Let $a = x_0 < x_1 < ... < x_N = b$ satisfying $S''(x_0) = S''(x_N)$ and

   $S(x) = S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ for $x_i \le x \le x_{i+1}$:

Picard's Theorem: If $f(x, y)$ and $\frac{\partial f}{\partial y}$ are both continuous functions on closed rectangle R, then through each point (x_0, y_0) in the interior of R, then there exists a unique curve of the equation $\frac{\partial f}{\partial y} = f(x, y)$ that passes through it.

2. DESIGN AND METHODOLOGY

2.1. Procedure of the Study

The procedure the researcher used was the producer used by [3] in his paper. In addition to these the following procedures were employed.

Step 1: The interval of Cantor set was selected.

Step 2: The domain name of one user was selected.

Step 3: By using MATLAB the domain name was converted into its data set points.

Step 4: The cubic interpolating Spline function with their respective conditions were defined.

Step 5: The diagonal coefficient matrix were obtained.

Step 6: The coefficient were evaluated on their proper interval.

Step 7: Interpolating polynomial of degree $N - 1$ was obtained.

Step 8: Using Newton-Raphson method the iteration was done.

Step 9: The system's password was obtained for the chosen domain name.

3. RESULTS AND DISCUSSIONS

3.1. Preliminaries

A polynomial spline of degree m is a function $S(x)\ for\ a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$ which satisfies the following conditions:

1. For $x \in [x_i, x_{i+1}]$, $S(x) = S_i(x)$: polynomial of degree $\leq m$.
2. $S(^{m-1})$ exists and continuous at the interval points

   i.e $\lim_{x \to x_i^-} S_{i-1}^{(m-1)}(x) = \lim_{x \to x_i^+} S_i^{(m-1)}(x)$

<u>Definition</u>: A cubic spline $S(x)$ is a piecewise defined function that satisfies the following conditions:

1. $S(x) = S_i(x)$ is a cubic polynomial on each sub interval $[x_i, x_{i+1}]$ for $i = 0,1,\dots, N-1$.
2. $S_i(x) = u_i$ for $i = 0,1,\dots, N-1$. (S have to interpolate all the points)
3. $S(x), S'(x)$ and $S''(x)$ are continuous on $[a, b]$ (S is smooth). So, we write the m cubic polynomial pieces as $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$, $i = 0,1,\dots, N-1$. where a_i, b_i, c_i & d_i represent 4'N unknown coefficients.

   From, cubic polynomial pieces between each data points we have:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, i = 0,1,\dots, N-1. \quad (1)$$

$$S'(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \quad (2)$$

$$S''(x) = 2c_i + 6d_i(x - x_i) \quad (3)$$

Let $S_i(x) = u_i$ for $i = 0,1,\dots, N-1$. Since $x_i \in [x_i, x_{i+1}]$ (4)

$$S(x_i) = S_i(x_i)$$

$$u_i = S_i(x)$$

$$u_i = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

$$u_i = a_i \text{ or each } i = 0,1,\dots, N-1. \quad (5)$$

From continuous properties of cubic spline method across each interval we have

$$S(x_i) = S_i(x_i)$$

$$S(x_i) = S_{i-1}$$

$$S_i(x_i) = S_{i-1}$$

$$S_i(x_i) = S_{i-1} \; i = 0,1,....,N-1. \tag{6}$$

From (5) we have $S_i = a_i$ and

$$S_{i-1}(x) = a_{i-1} + b_{i-1}(x - x_{i-1}) + c_{i-1}(x - x_{i-1})^2 + d_{i-1}(x - x_{i-1})^3$$

So, $a_i = a_{i-1} + b_{i-1}(x - x_{i-1}) + c_{i-1}(x - x_{i-1})^2 + d_{i-1}(x - x_{i-1})^3$

For $i = 0,1,....,N-1.$ and $h = x_i - x_{i-1}$

$$a_i = a_{i-1} + b_{i-1}h + c_{i-1}h^2 + d_{i-1}h^3 \tag{7}$$

To make a curve smooth across each interval, the derivative must be equal at the data points.

i.e $S'_i(x_i) = S'_{i-1}(x_i)$

⇨ $$S_i'(x_i) = b_i \tag{8}$$

$$S'_{i-1}(x) = b_{i-1} + 2c_{i-1}(x - x_{i-1}) + 3d_{i-1}(x - x_{i-1})^2$$

$$b_i = b_{i-1} + 2c_{i-1}h^2 + 3d_{i-1}h^2 \text{ for } \qquad i = 0,1,....,N-1. \tag{9}$$

From equation (3) $S_i''(x) = 6d_i(x - x_i) + 2c_i$

$$S_i''(x_i) = 2c_i \text{ for } i = 0,1,....,N-1. \tag{10}$$

Lastly, since $S_i''(x_i)$ has to be continuous across the interval,

$$S''_i(x_{i+1}) = 6d_i(x_{i+1} - x_i) + 2c_i$$

$$S''_{i+1}(x_{i+1}) = 6d_i(x_{i+1} - x_i) + 2c_i \tag{11}$$

And letting $h = x_{i+1} - x_i$, using the conclusion from (10) and (11):

$$S''_{i+1}(x_{i+1}) = 6d_i(x_{i+1} - x_i) + 2c_i$$

$$2c_{i+1} = 6d_i h + 2c_i \tag{12}$$

The equation can be much simplified by substituting M_i for $S''_i(x_i)$ and expressing the above equation in terms of M_i and u_i. This makes the determination the weights, $a_i, b_i, c_i \;\&\; d_i$ a much easier task. Each c_i can be represented by:

$$S''_i(x_i) = 2c_i \Rightarrow M_i = 2c_i \Rightarrow c_i = \frac{M_i}{2} \tag{13}$$

And a_i has already been determined to be $a_i = u_i$

Similarly using equation (12) d_i can be written as:

$$2c_{i+1} = 6d_i h + 2c_i \Rightarrow 6d_i h = 2c_{i+1} - 2c_i$$

$$\Rightarrow \quad d = \frac{2c_{i+1} - 2c_i}{6h} = \frac{2\left(\frac{M_{i+1}}{2}\right) - 2\left(\frac{M_i}{2}\right)}{6h}$$

$$\Rightarrow d = \frac{M_{i+1} - M_i}{6h} \tag{14}$$

From equation (7) b_i can be written as:

$$a_{i+1} = a_i + b_i h + c_i h^2 + d_i h^3 + a_{i+1}$$

$$\Rightarrow \quad b_i = \frac{-a_i - c_i h^2 - d_i h^3 + a_{i+1}}{h}$$

$$\Rightarrow \quad b_i = \frac{u_{i+1} - u_i}{h} - \frac{h}{6}(M_{i+1} + 2M_i) \tag{15}$$

We now have our equation for determining the weight of our $N-1$ equations:

$$u_i = a_i,\ b_i = \frac{u_{i+1} - u_i}{h} - \frac{h}{6}(M_{i+1} + 2M_i),\ c_i = \frac{M_i}{2} \text{ and } d = \frac{M_{i+1} - M_i}{6h} \tag{16}$$

These systems can be handled more conveniently by putting them in Matrix form as follows

From (9), $b_{i-1} = b_i + 2c_i h + 3d_i h^2$ for $i = 0,1,\ldots, N-1$.

$$\Rightarrow \quad 3d_i h^2 + 2c_i h = b_{i-1} - b_i \tag{17}$$

When we substitute the values of equation (16) into (17) and rearrange the values; we get:

$$M_i + 4M_{i+1} + M_{i+2} = \frac{6}{h^2}\left[u_i - 2u_{i+1} + u_{i+2}\right],\ \text{for } i = 0,1,\ldots, N-1. \tag{18}$$

By substitution the values of for i i in to for -1, we get:

$$M_{i-1} + 4M_i + M_{i+1} = \frac{6}{h^2}\left[u_{i-1} - 2u_i + u_{i+1}\right] \text{for } i = 0,1,\ldots, N-1. \tag{19}$$

Definition 1: The Cantor set C is defined as $C = \bigcap_{n=1}^{\infty} I_n$ where I_{n+1} is constructed by trisecting I_n and removing the middle third, I_0 being the closed real interval $[0,1]$.

Several intersecting properties of the Cantor set are immediately apparent. Since it is defined as the set of points not excluded, the "size" of the set can be thought of as the proportion of the interval $[0,1]$ removed. If we add up the contribution from $\frac{2}{3}$ removed n times find that $\sum_{n=0}^{\infty} \frac{2^n}{3^{n+1}} = \frac{1}{3} + \frac{2}{9} + \frac{4}{27} + ... = \frac{1}{3}\left(\frac{1}{1-\frac{2}{3}}\right) = 1$ where the geometric sum has its well-known solutions. As a result, the proportion remaining "in" the Cantor set is $1-1=0$, and it can contain no intervals of non-zero length. For assume by contradiction it does contain some interval (a,b). Choose $n \in N$ such that $\frac{1}{3^n} < b-a$. Since the Cantor set is contained in the finite intersection of closed intervals, all of length less than $(b-a)$, we have that this intersection and so C cannot contain (a,b).

Theorem 3.1.1: The Cantor set is nonempty.

Proof: Let consider the interval I_n as defined on the above Cantor set definition. Each trisection of I_n to from I_{n+1} leaves exactly two end points. For example removing $\left(\frac{1}{3}, \frac{2}{3}\right)$ from $[0,1]$ leaves the points $P_0 = \frac{1}{3} and P_1 = \frac{2}{3}$. In fact, since the Cantor set is the infinite intersection of each I_n, contains the end points of each subinterval, and is clearly non empty. In fact it is infinite.

Definition 2: A subset A of a metric space M is nowhere dense if its closure has an empty interior. That is if $\text{int}(\overline{A}) = \varnothing$.

Theorem 3.1.2: A Cantor set is closed and nowhere dense.

We have already seen that C is the intersection of closed sets, which implied that C is itself closed. Furthermore, as previously discussed the Cantor set contains no intervals of non-zero length, and so, $\text{int}(C) = \varnothing$

Definition 3: A metric space M is totally disconnected if, for any $\varepsilon > 0$ and $p \in M$ there exists a closed subset U of M such that $p \in U \subset M_{\varepsilon}(p)$. That is, there is an arbitrarily small clopen neighborhood centered on every point of M. With this definition we can proven two more important facts about the Cantor set.

Theorem 3.1.3: The cantor set C is perfect and totally disconnected.

Proof: Fix any $\varepsilon > 0$ and point $p \in C$. Let $n \in N$ be sufficiently large such that $\frac{1}{3^n} < \varepsilon$. Then , P is guaranteed to be in one of the intervals (I_n for some $n \in N$) that make up C, each of length $\frac{1}{3^n}$, the endpoints of the Cantor set in this interval are infinite number and contained in the open interval $(p-\varepsilon, p+\varepsilon)$, so P is a cluster point of C, $M_{\varepsilon}(p)$ containing an infinite number of points. And since we are considering any $p \in C$, C is perfect. Furthermore, this interval I_n is closed in R and in the Cantor set C as well. Since $I_n^{\ c} = C \setminus I_n$ consists of a countable number of closed intervals, itself closed. We can then represent C as the disjoint union of two clopen sets, $(C \cap I_n)$ and $(C \cap I_n^{\ c})$, the result being that the Cantor set C is totally disconnected.

Theorem 3.1.4: Cantor set C is Compact.

Proof: Each C_n is a finite union of closed sets, so C_n is closed for $\forall_n$. Then, $C = \cap C_n$ is also closed. Also, C is bounded since $C \subseteq [0,1]$. So, by Heine-Borel theorem, C is compact.

Generalization, although our construction of Cantor set used the typical “middle third” or ternary rule, we can easily generalize this one dimensional idea to any length other than $\frac{1}{3}$, excluding of course the degenerate cases of 0 and 1.

3.1. Main Results

In this section we are going to treat and compute the actual system based password of g-mail domain name. With the wide spreading of the internet and the World Wide Web, our society is becoming more and more dependent on communication data which are transmitted over computer networks. A large number of transactions involving a growing number of people have been actually replaced by their digital analogues, in which electronic “objects” are exchanged among two more parties. An example comes from the diffusion of the electronic mail services which allows users to exchange messages containing text or multimedia files.

Because of its features, such as low cost, rapidity and accessibility the email service is increasingly used in place of ordinary mail.

In many cases, email messages are recognized as recipient’s evidences of online transactions, such as buying airlines tickets, or submission of papers in conferences or journals, and so on. However, the user of email poses some problems, since in its simplest form the email service does not have many features that are usually required in such cases. The standard email service is based on Simple Mail Transfer Protocol and Post Office Protocol, which do not offer guarantees on the delivery and the integrity of the messages. Messages are usually stored and transmitted in plain text allowing a malicious adversary to tap the connection during the transfer and making him able to access sensible data.

Now, a day’s most of the mail users are in a position to use mail address from the prominent serves such as Gmail, Yahoo, Hotmail, and the like. Gmail server has 6518MB of storage, which allows the users the ability to save their own email without worrying that any new emails will not get through because they will reach the allowed limit.

Now for just comparison, the researcher used mail address or domain names of users of Gmail and Yahoo server. The reason why these two servers were selected was that most of the users in the world (about 98% **Error! Reference**

source not found were using these two were sketched through MATLAB by using cubic interpolating spline function which was defined on Cantor set.

For the analysis, the domain name 'saamhookoo@gmail.com' was taken and by using MATLAB built in package which corresponds each character (twenty of them) of the domain into numbers (twenty corresponding numbers), (i.e 's' is represented by '115','a' by '97', etc), which can be written as a vector

$$[115\;97\;97\;109\;104\;111\;111\;107\;111\;111\;64\;103\;109\;97\;105\;108\;46\;99\;111\;109]\ .$$

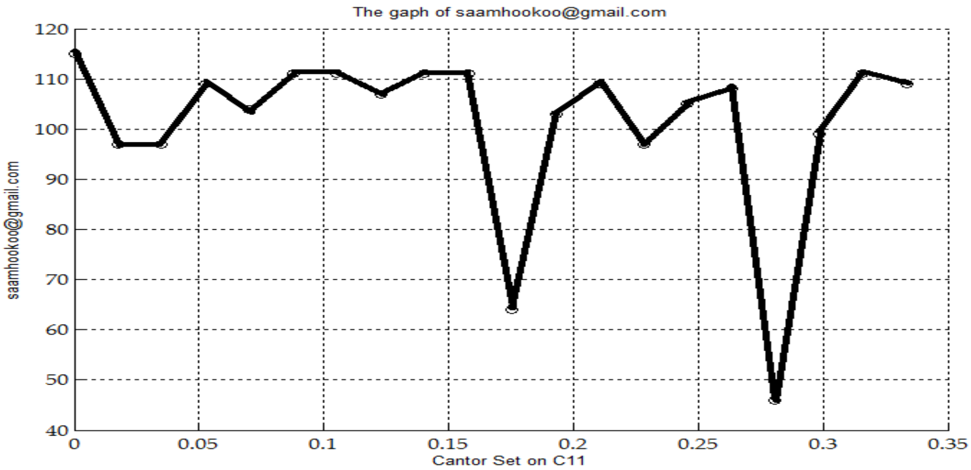

Figure 1

Example: Design the cubic spline of domain name saamhookoo@gmail.com

From (19) and equation (1 – 18) the coefficients of the spline interpolating function was calculated by reducing the expressions into tri-diagonal. That is,

$$M_{i-1} + 4M_i + M_{i+1} = \frac{6}{h^2}[u_{i-1} - 2u_i + u_{i+1}], \text{ for } i = 1,2,\ldots, N-1 .$$

For $i = 1$ we have the following expressions: $M_0 + 4M_i + M_2 = \frac{6}{h^2}[u_0 - 2u_i + u_2]$

In a similar fashion one can obtain a 20X20 tri-diagonal matrix for the left hand expression and a column matrix say $g_i's$ for the right hand side expression as follows. Let A be a coefficient matrix for the left hand side expression.

$$g_i = \frac{6}{h^2}(u_{i-1} - 2u_i + u_{i+1}) \text{ for } i = 1,2,\ldots,18 \tag{20}$$

And $u_i's$ were calculated directly from the domain name saamhookoo@gmail.com which was transformed into vector representation format by using MATLAB

$$U = u_i = [u_0, u_1, \ldots, u_{17}]^T \tag{21}$$

$$= [115\quad 97\quad 97\quad 109\quad 104\quad 111\quad 111\quad 107\quad 111\quad 111\quad 64\quad 103\quad 109\quad 97\quad 105\quad 108\quad 46\quad 99\quad 111\quad 109]^T$$

For $i = 0,1,\ldots,17$

$$\begin{bmatrix}
1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} * \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \\ M_{10} \\ M_{11} \\ M_{12} \\ M_{13} \\ M_{14} \\ M_{15} \\ M_{16} \\ M_{17} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \\ g_9 \\ g_{10} \\ g_{11} \\ g_{12} \\ g_{13} \\ g_{14} \\ g_{15} \\ g_{16} \\ g_{17} \\ g_{18} \end{bmatrix} \tag{22}$$

From equation (20) $g_i{}'s$ for i = 1, 2,…., 18 , were calculated by using MATLAB and the result were displayed as follows in matrix form

g_i = [3.8911e+005 2.5941e+005 -3.6749e+005 2.5941e+005 -1.5132e+005 -8.6469e+004

1.7294e+005 -8.6469e+004 -1.0160e+006 1.8591e+006 -7.1337e+005 -3.8911e+005

4.3235e+005 -1.0809e+005 -1.4051e+006 2.4860e+006 -8.8631e+005 -3.0264e+005]^T , where i=1,2,⋯18.

From equation (22) M i ' s were computed by taking the following expression.

$M_i = A^{-1} * g_i$ for i=1,2,….,18. So, A^{-1} is computed and the result was as follows:

$$A^{-1}=1.0e+009*\begin{matrix}
0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0006 & -0.0021 & 0.0079 & -0.0294 & 0.1096 & -0.4089 & 1.5259 & -5.6946 \\
0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0006 & -0.0021 & 0.0079 & -0.0294 & 0.1096 & -0.4089 & 1.5259 \\
0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0006 & -0.0021 & 0.0079 & -0.0294 & 0.1096 & -0.4089 \\
0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0006 & -0.0021 & 0.0079 & -0.0294 & 0.1096 \\
0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0006 & -0.0021 & 0.0079 & -0.0294 \\
0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0006 & -0.0021 & 0.0079 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0006 & -0.0021 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0002 & 0.0006 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0002 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & -0.0000 & -0.0000 & 0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0.0000 & -0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0000 & 0.0000 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000
\end{matrix}$$

From equation (16) we have

$$u_i = a_i,\ b_i = \frac{u_{i+1}-u_i}{h} - \frac{h}{6}(M_{i+1} - 2M_i),\ c_i = \frac{M_i}{2} \text{ and } d_i = \frac{M_{i+1}-M_i}{6h}$$

Before we are going to use Newton-Raphson Method first we have to determine the governing function f that agrees with S on the mesh points. Since we have 20 data sets, need a 20x20 matrix.

Let $f(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + ... + \alpha_{20} x^{19}$ (23)

From the property of spline function S and its governing function f we have:

$f(x_i) = S(x_i) = S_i(x_i)$, For all i= 1,2,3, … , 20. Consider the first interval of the Cantor set. (Say)

$C_{11} = \left[0, \frac{1}{3}\right]$. The step size or the discretization length $h = \frac{x_N - x_1}{N} = \frac{\frac{1}{3}-0}{20} = \frac{1}{60}$.

From equation (23) above we have

$$f(x_1) = \alpha_1 + \alpha_2(x_1) + \alpha_3(x_1)^2 + ... + \alpha_{20}(x_1)^{19} = 115 \text{ Since } x_1 = 0$$

$$\Rightarrow f(0) = \alpha_1 + \alpha_2(0) + \alpha_3(0)^2 + ... + \alpha_{20}(0)^{19} = 115,$$

$$\Rightarrow f(0) = \alpha_1 = 115$$

For x_2, $f(x_2) = \alpha_1 + \alpha_2(x_2) + \alpha_3(x_2)^2 + ... + \alpha_{20}(x_2)^{19}$, $x_2 = h = \frac{1}{60}$.

$$f(x_2) = f(h) = \alpha_1 + \alpha_2(h) + \alpha_3(h)^2 + \ldots + \alpha_{20}(h)^{19} = 97$$

$$f(x_3) = f(2h) = \alpha_1 + \alpha_2(2h) + \alpha_3(2h)^2 + \ldots + \alpha_{20}(2h)^{19} = 97$$

....

$$f(x_{20}) = f(20h) = \alpha_1 + \alpha_2(20h) + \alpha_3(20h)^2 + \ldots + \alpha_{20}(20h)^{19} = 109$$

So, the coefficient matrix E (say), can be expressed as:

$$E = \begin{bmatrix} (0h) & (0h)^2 & (0h)^3 & \ldots & (0h)^{19} \\ (1h) & (1h)^2 & (1h)^3 & \ldots & (1h)^{19} \\ (2h) & (2h)^2 & (2h)^3 & \ldots & (2h)^{19} \\ \ldots & \ldots & \ldots. & \ldots & \ldots. \\ (20h) & (20h)^2 & (20h)^3 & \ldots & (20h)^{19} \end{bmatrix} \text{ and } \alpha_i = \begin{bmatrix} \alpha_1 \\ \alpha 2 \\ \alpha 3 \\ \ldots. \\ \alpha_{20} \end{bmatrix} \text{ for } i = 1,2,\ldots,20$$

Thus, $$\begin{bmatrix} (0h) & (0h)^2 & (0h)^3 & \ldots & (0h)^{19} \\ (1h) & (1h)^2 & (1h)^3 & \ldots & (1h)^{19} \\ (2h) & (2h)^2 & (2h)^3 & \ldots & (2h)^{19} \\ \ldots & \ldots & \ldots. & \ldots & \ldots. \\ (20h) & (20h)^2 & (20h)^3 & \ldots & (20h)^{19} \end{bmatrix} * \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \ldots \\ \alpha_{20} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \ldots \\ a_{20} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \ldots \\ \alpha_{20} \end{bmatrix} = \begin{bmatrix} (0h) & (0h)^2 & (0h)^3 & \ldots & (0h)^{19} \\ (1h) & (1h)^2 & (1h)^3 & \ldots & (1h)^{19} \\ (2h) & (2h)^2 & (2h)^3 & \ldots & (2h)^{19} \\ \ldots & \ldots & \ldots. & \ldots & \ldots. \\ (20h) & (20h)^2 & (20h)^3 & \ldots & (20h)^{19} \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \ldots \\ a_{20} \end{bmatrix} \qquad (24)$$

From these expressions, we have values of α_i's and one can also find the inverse of coefficient matrix E^{-1}. So, by multiplying E^{-1} by α_i's we will have the following values.

$$\Rightarrow \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \ldots \\ \alpha_{20} \end{bmatrix} = \begin{bmatrix} 115 \;\; 2.3224e+007 - 4.7923e+009 \; 4.2859e+011 - 2.2408e+013 \; 7.7563e+014 - 1.9023e+016 \\ 3.4465e+017 - 4.7388e+018 \; 5.0323e+019 \; ? \\ -4.1717e+020 \; 2.7136e+021 - 1.3850e+022 \; 5.5171e+022 - 1.6951e+023 \; 3.9352e+023 \\ -6.6706e+023 \; 7.7850e+023 - 5.5889e+023 \; 1.8596e+023 \end{bmatrix}^T$$

This computation leads us to get the governing function f with their respective conditions as displayed in the following lines

$f(x)$ = 115 + (6234133056968353 / 268435456)* x^1-(2512548999448485 / 524288)* x^2+ (1755502906181383 / 4096)* x^3

- (5736392578750985 / 256)* x^4+ (6205046247124737 / 8)* x^5- (19023112608783088)* x^6 + (344651962205307456)* x^7

- (4738841091626776576)* x^8 + (50323312017772527616)* x^9 - (417172156483298852864)* x^{10}

+ (2713558697764479041536)* x^{11} - (13849897859614441996288)* x^{12} + (55170702892070133039104)* x^{13}

- (169511024182191928115200)* x^{14} + (393524688526126484553728)* x^{15} - (667058782115569686544384)* x^{16}

+ (778499302591270089654272)* x^{17} - (558889075366920378646528)* x^{18} + (185955788415513546194944)* x^{19}

Now, the next step is to use Newton-Raphson Method to approximate the password of the user name saamhookoo@gmail.com

Newton-Raphson method is defined as

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{24}$$

By putting

$$x_0 = (a_i's)^T = [115\ 97\ 97\ 109\ 104\ 111\ 111\ 107\ 111\ 111\ 64\ 103\ 109\ 97\ 105\ 108\ 46\ 99\ 111\ 109]^T.$$

Since $(a_i's)^T$ is a row matrix, governing function has been written in $f((a_i's)^T)$.

Then take the first derivatives of f and substitute in equation (23), and iterate the function till the absolute error $\xi = |P_a - P_{app}| \leq \delta$ and

$$\delta = 1.0\times10^{-6}, \xi = absoluteerror,$$

$P_a = actualpassword$, and $P_{app} = theapproximatepassword$

Now the derivatives of $f(x)$ is computed and the result was recorded as follows:

$f'(x) =$ (6234133056968353/ 268435456) - (2512548999448485/ 262144)*x +(5266508718544149/ 4096)*x^2

- (5736392578750985/64)*x^3 - (31025231235623685/8)*x^4 - (114138675652698528)*x^5

+ (2412563735437152192)*x^6 - (37910728733014212608)*x7 + (452909808159952748544)*x^8

- (4171721564832988528640)*x^9 -(29849145675409269456896)*x^{10} -(166198774315373303955456)*x^{11}

+(717219137596911729508352)*x^{12}-(2373154338550686993612800)*x^{13}+ (5902870327891897268305920)*x^{14}

-(10672940513849114984710144)*x^{15}+(13234488144051591524122624)*x^{16}-(10060003356604566815637504)*x^{17}

+ (3533159979894757377703936)*x^{18}

Let $P_i's$ be the successive iterated value of the function up to the range of tolerable error. Then the following are the results obtained.

$$P_0 = (a_i's)^T = [115\ 97\ 97\ 109\ 104\ 111\ 111\ 107\ 111\ 111\ 64\ 103\ 109\ 97\ 105\ 108\ 46\ 99\ 111\ 109]^T.$$

$P_1 = P_0 - \left(\frac{f(p_0)}{f'(P_0)} \right)$; where P_1 is the first iterated value. Since matrix division is not allowed, the iteration was done in an element wise manner. Then by continuing this procedure up to the 38^{th} (to have a uniform and consistent method for all) iterations we will obtain the best approximated value which is termed as the system's password. Hence, we can design any mail server by this fashion, thus numerical analysis is key for the design.

4. CONCLUSIONS/DISCUSSIONS

On this paper, different mail servers such as yahoo.com, and gmail.com were treated to obtain the pattern or the similarities and differences among them. After the two servers design were seen the mathematical meaning of electronic mail design was analyzed thoroughly.

For any domain name or email address, we obtain a unique curve that uniquely identifies the user name. No two or more mail addresses have the same curve on any mail server. By Picard's theorem, this curves has a unique solution. This unique solution(s) was/were calculated by using Newton-Raphson method for its being fast and easily converges to the required solution.

The sent message have been checked where it is correct or corrupted, and validates the properties of Cantor set in line with its topological properties that makes the mail boxes stable.

REFERENCES

1. M. Cheung, "Factors Affecting the Design of Electronic Direct Mail Messages: Implications for Professional Communicators", IEEE, Transactions on Professional Communication, 54, 3, 2011, pp. 279-298.
2. M. M. Bothma, E.C. Hoffmann, "Combating the Corporate Paper War: Electronic Mail Abuse", Journal of Information Management, 10, 1, 2008, pp. 1-12.
3. Adil. Al-Rammahi Cantor Interpolating Spline to Design Electronic mail Boxes. 8, 2014, pp 31 – 33.
4. A. Bahreman**,** "Certified Electronic Mail", In Proceedings of the Network and Distributed Systems Security Conference, 1994, pp. 3-19.
5. S.C. Hui , "A multimedia Electronic Mail System on a Heterogeneous Environment", IEEE, Tenscon Proceeding, 1, 1993, pp. 61-64.
6. S. Lipschutz, "Theory and Problems in General Topology", Schaum's Outline Series; McGraw-Hill Book Company, 1965.