Tribonacci Matrix Error Control

Supervisor : Bibhas Chandra Dhara Sir

Group No 16

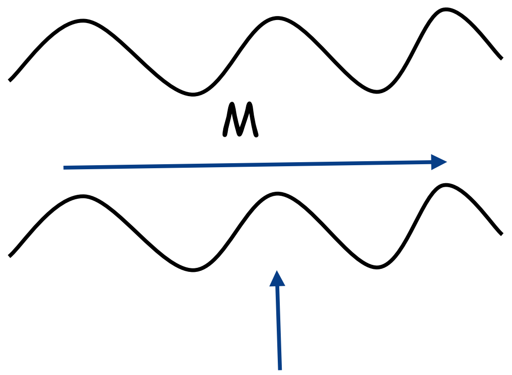
1. Karan Raj Bagri-001911001036
2. Swapnil Ghosh-001911001067
3. Chirag Jaiswal-001911001066

Referred from the research paper ("TRIBONACCI MATRICES AND A NEW CODING THEORY") by MANJUSRI BASU and MONOJIT DAS.

CONTENT

Why do we need error control?

SENDER



RECEIVER

(may receive M' != M)

Unreliable medium

(may cause corruption of bits in the transmitted message due to attenuation, noise, interference, etc.)

Why do we need error control? (contd.)

To ensure reliable communication over unreliable media, we need to

- Be able to detect the corruption when it occurs - ***Error Detection***
- Be able to correct the error caused by this corruption, either at the receiver's end itself (forward error correction) or by asking the sender to send the message again (backward error correction) - ***Error Correction***

The forward corrective ability of the Tribonacci Matrix Coding method is what stands out.

Applications:

A strong corrective ability is important when

* transmission error rate is high
* propagation delay is high

One such situation is **deep space communication**.

High propagation delay thanks to large distance between transmitter and receiver.
Between Earth and Mars: 4-21 minutes
Between Earth and Jupiter: 33-53 minutes

Error rates are relatively high too: 10^{-6} between Earth and Jupiter.

Tribonacci numbers

- The Tribonacci numbers t_k (k = 0, 1, 2, 3,…) are the generalization of the Fibonacci numbers defined by the recurrence relation

$$t_{k+1} = t_k + t_{k-1} + t_{k-2}, \quad \text{where } t_0 = t_1 = 0, \ \ t_2 = 1.$$

k	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
t_k	7	5	-8	4	1	-3	2	0	-1	1	0	0	1	1	2	4	7	13	24	44	81

Tribonacci Coding, Decoding Matrix

- *Tribonacci Matrix / Coding Matrix :*

$$M^k = \begin{pmatrix} t_{k+2} & t_{k+1}+t_k & t_{k+1} \\ t_{k+1} & t_k+t_{k-1} & t_k \\ t_k & t_{k-1}+t_{k-2} & t_{k-1} \end{pmatrix} \quad \textit{where } M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

- *Inverse Of Tribonacci Matrix / Decoding Matrix :*

$$M^{-k} = \begin{pmatrix} t_{k-1}^2 - t_{k-2}t_k & t_{k-2}t_{k+1} - t_{k-1}t_k & t_k^2 - t_{k-1}t_{k+1} \\ t_k^2 - t_{k-1}t_{k+1} & t_{k-1}t_{k+2} - t_kt_{k+1} & t_{k+1}^2 - t_kt_{k+2} \\ \begin{array}{c} t_{k-1}t_{k+1} + t_{k-2}t_{k+1} \\ -t_k^2 - t_{k-1}t_k \end{array} & \begin{array}{c} t_k^2 + t_kt_{k+1} - t_{k-1}t_{k+2} \\ -t_{k-2}t_{k+2} \end{array} & \begin{array}{c} t_kt_{k+2} + t_{k-1}t_{k+2} \\ -t_{k+1}^2 - t_kt_{k+1} \end{array} \end{pmatrix}$$

Some Properties of M^k Matrix

(1) $M^k = M^{k-1} + M^{k-2} + M^{k-3}$.

(2) $M^k M^l = M^l M^k = M^{k+l}$ $(k, l = 0, \pm 1, \pm 2, \pm 3 \ldots)$.

(3) $det\ M^k = 1$.

Tribonacci Coding Theory

❖Message Matrix (P) :

The desired message that we want to send to end-user in the channel.

The message is represented using a matrix of order 3.

All elements of the matrix are non-negative integer.

$$p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9 \geq 0$$

$$P = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix}.$$

Encoding

- This Encoding process is done at sender side.
- We are encoding the Message Matrix (P) by multiplying it with the Tribonacci Coding Matrix (M^K).
- The encoded matrix is called as Code Matrix (E).

$$E = \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \qquad P = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix}.$$

$$E = P \times M^k.$$

Decoding

- ❖This decoding process takes place at receiver side.
- ❖We take the Tribonacci Decoding Matrix (M^{-k}) and multiply it with the received Code Matrix (E) .
- ❖The decoded matrix is our Message Matrix (P) if and only if there is no error/corruption done in Code Matrix (E), in the transmission channel.

$$E = \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \quad P = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix}.$$

$$E \times M^{-k} = P$$

Example

Message

10101100001011111011101011100000001010110001000011111101111000011011001110000 1001

10101100	00101111	01110101	11000000	10101100	01000011	1111011	11000011	01100111	00001001

Sender Side

Message Matrix

P =

172	**47**	**117**
192	67	251
195	103	9

Tribonacci Coding Matrix

M^8 =

81	**68**	**44**
44	37	24
24	20	13

Code Matrix

$E(P \times M^8) =$

18957	**15900**	**10298**
24673	20680	13400
20692	17376	11250

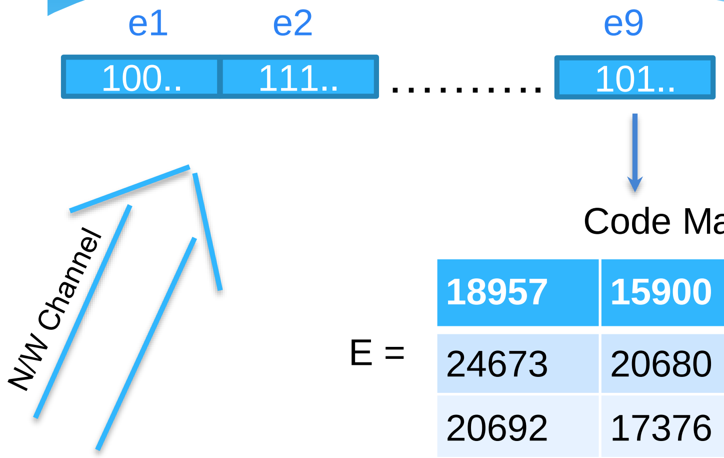
✚ |P| = -1338614

N/W Channel

01010101,….

Encoding

Example (cont)



Receiver Side

Code Matrix

E =

18957	15900	10298
24673	20680	13400
20692	17376	11250

Message Matrix

$P (E \times M^{-8}) =$

172	47	117
192	67	251
195	103	9

Tribonacci Decoding Matrix

$M^{-8} =$

1	-4	4
4	-3	-8
-8	12	5

Decoding

Detection of error in Code Matrix (E)

- Sender is sending the Code Matrix (E) and also the determinant of the Message Matrix (P) .

$$E = P \times M^k \xrightarrow{E = \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix},\quad det\, P}$$

- The sole reason for sending determinant of Message Matrix (P) is so that the receiver can verify that the received Code Matrix (E) is error free or not.

- The error is checked using this relation,

$$\det E = \det\left(P \times M^k\right) = \det P \times \det M^k = \det P \times 1 = \det P$$

- This relation means when that the determinant of the Code Matrix (E) received is equals to the determinant of the Message Matrix (P).

- When this relation is holds that means our received Code Matrix (E) is error free otherwise error is present in our Code Matrix (E).

Error Detection and Correction

- Detection of error is important as it helps to recover the Code Matrix (E) and eventually the Message Matrix (P).

- The inter-relation among the Code Matrix (E) elements are as follows : -

$$\min\left\{\frac{t_{k+2}}{t_{k+1}+t_k}, \frac{t_{k+1}}{t_k+t_{k-1}}, \frac{t_k}{t_{k-1}+t_{k-2}}\right\} \le \frac{e_1}{e_2}, \frac{e_4}{e_5}, \frac{e_7}{e_8} \le \max\left\{\frac{t_{k+2}}{t_{k+1}+t_k}, \frac{t_{k+1}}{t_k+t_{k-1}}, \frac{t_k}{t_{k-1}+t_{k-2}}\right\}$$

$$\min\left\{\frac{t_{k+1}+t_k}{t_{k+1}}, \frac{t_k+t_{k-1}}{t_k}, \frac{t_{k-1}+t_{k-2}}{t_{k-1}}\right\} \le \frac{e_2}{e_3}, \frac{e_5}{e_6}, \frac{e_8}{e_9} \le \max\left\{\frac{t_{k+1}+t_k}{t_{k+1}}, \frac{t_k+t_{k-1}}{t_k}, \frac{t_{k-1}+t_{k-2}}{t_{k-1}}\right\}$$

$$\min\left\{\frac{t_{k+2}}{t_{k+1}}, \frac{t_{k+1}}{t_k}, \frac{t_k}{t_{k-1}}\right\} \le \frac{e_1}{e_3}, \frac{e_4}{e_6}, \frac{e_7}{e_9} \le \max\left\{\frac{t_{k+2}}{t_{k+1}}, \frac{t_{k+1}}{t_k}, \frac{t_k}{t_{k-1}}\right\}$$

The limit, $\alpha = \lim_{k\to\infty} \frac{t_k}{t_{k-1}}$ exists, called Tribonacci constant and is the one and only real root of the equation $x^3 - x^2 - x - 1 = 0$. Actually $\alpha = 1.83928675$.

❖ Using the relations mentioned previously we try to correct different folds of error.

❖ We can correct up to 8 folds of error.

$$\frac{{}^9C_1 + {}^9C_2 + {}^9C_3 + {}^9C_4 + {}^9C_5 + {}^9C_6 + {}^9C_7 + {}^9C_8}{{}^9C_1 + {}^9C_2 + {}^9C_3 + {}^9C_4 + {}^9C_5 + {}^9C_6 + {}^9C_7 + {}^9C_8 + {}^9C_9} = 0.9980$$

❖ We can achieve an accuracy of 99.80 %.

Other Methods

- ***Cyclic Redundancy Check***
  - In CRC we can only detect error (all single errors, double errors) but we cannot correct it, our algorithm can detect error but it's a bit computationally expensive.
  - In CRC along with message we also send some extra bits of length maximum degree of message polynomial, but in our algorithm the message size is fixed.
- ***Reed Solomon Code***
  - In RSC we can detect error as well as correct error, just like our algorithm .
  - Its computationally very expensive at sender side as we use to generate polynomial (p(x)) using Lagrange Interpolation.
- ***Hamming Code***
  - In Hamming Code we can detect and correct only a single bit of error of a binary number, which our algorithm can easily do and it's computationally less expensive than our algorithm.

IMPLEMENTATION AND ANALYSIS

Error Correction Procedure

1. Assume a t-fold error (initially t=1).
    a. Choose t entries from the 9 entries in the encoded matrix.
    a. For each such combination:
        i. Let the t elements be variables (effectively assuming that these are the locations of the errors).
        ii. Form a constraint satisfaction problem, considering |E| = |P| as well as the ratio ranges mentioned previously.
        iii. Use an appropriate solver to solve this problem, only allow for integer solutions.
        iv. If a solution is obtained, use it and terminate the procedure.

2. t = t+1

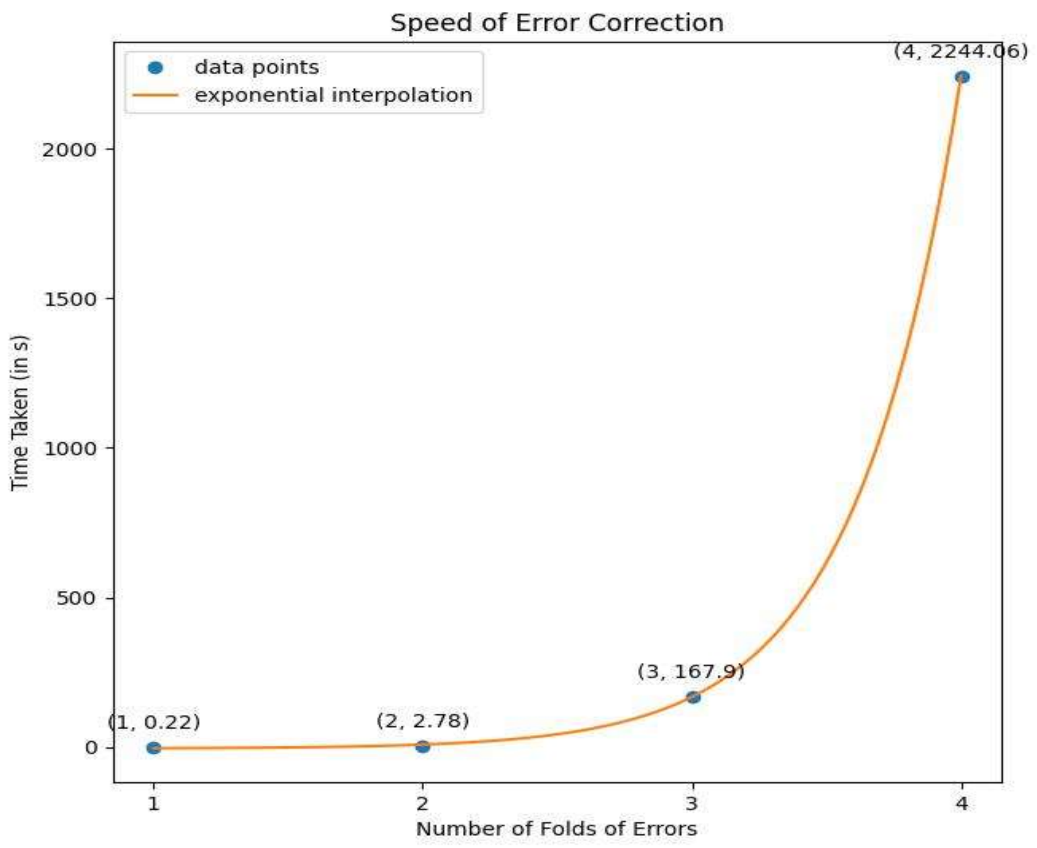
3. if t=9, error can’t be corrected. Terminate the procedure.

TOOLS USED

	Code editor
python™	Programming language
NumPy matplotlib PYOMO	For matrix operations, plotting and modelling the error correction problem
Coin-or/Couenne	Solving the error correction problem (minlp)
GitHub	Version Control

ISSUES:

1. **Time required for error correction increases sharply with increase in the number of folds of error.**



ISSUES :

2. **There is a chance (albeit small) that an error will not be detected.**

What if $E_{received}$!= E but $|E| == |E_{received}|$?

For example,

$$E = \begin{bmatrix} 376 & 345 & 214 \\ 426 & 284 & 215 \\ 528 & 352 & 256 \end{bmatrix}$$

Here, |E| = |Received| = 421104

$$\text{Received} = \begin{bmatrix} 376 & 345 & \mathbf{778} \\ 426 & 284 & 215 \\ 528 & 352 & 256 \end{bmatrix}$$

ISSUES :

To inhibit this occurrence:

Also check if the ratios are within bounds, during error detection.

$$\min\left\{\frac{t_{k+2}}{t_{k+1}+t_k}, \frac{t_{k+1}}{t_k+t_{k-1}}, \frac{t_k}{t_{k-1}+t_{k-2}}\right\} \leq \frac{e_1}{e_2}., \frac{e_4}{e_5}., \frac{e_7}{e_8} \leq \max\left\{\frac{t_{k+2}}{t_{k+1}+t_k}, \frac{t_{k+1}}{t_k+t_{k-1}}, \frac{t_k}{t_{k-1}+t_{k-2}}\right\}$$

$$\min\left\{\frac{t_{k+1}+t_k}{t_{k+1}}, \frac{t_k+t_{k-1}}{t_k}, \frac{t_{k-1}+t_{k-2}}{t_{k-1}}\right\} \leq \frac{e_2}{e_3}., \frac{e_5}{e_6}., \frac{e_8}{e_9} \leq \max\left\{\frac{t_{k+1}+t_k}{t_{k+1}}, \frac{t_k+t_{k-1}}{t_k}, \frac{t_{k-1}+t_{k-2}}{t_{k-1}}\right\}$$

$$\min\left\{\frac{t_{k+2}}{t_{k+1}}, \frac{t_{k+1}}{t_k}, \frac{t_k}{t_{k-1}}\right\} \leq \frac{e_1}{e_3}., \frac{e_4}{e_6}., \frac{e_7}{e_9} \leq \max\left\{\frac{t_{k+2}}{t_{k+1}}, \frac{t_{k+1}}{t_k}, \frac{t_k}{t_{k-1}}\right\}$$

Does not eliminate the problem but reduces the chances significantly.

ISSUES:

3. **There is a chance that the solution given by the solver isn't the desired one.**

   Worse when the number of folds of error is high.

   Example:

$$E = \begin{bmatrix} 34806 & 29206 & 18920 \\ 84605 & 71006 & 45992 \\ 71633 & 60130 & 38945 \end{bmatrix}$$

$$\text{Received} = \begin{bmatrix} 34806 & \mathbf{29207} & \mathbf{18901} \\ \mathbf{84600} & \mathbf{52006} & \mathbf{41942} \\ \mathbf{71603} & \mathbf{20122} & \mathbf{32132} \end{bmatrix} \xrightarrow[\text{Correction}]{\text{Error}} \begin{bmatrix} 34806 & 29210 & 18911 \\ 105903 & 88876 & 57541 \\ 108219 & 90788 & 58787 \end{bmatrix}$$

!= E

ISSUES :

To inhibit this occurrence:

Increase the value of k (in M^k).

$\downarrow$

When k increases, t_{k+1}/t_k stabilizes.

$\downarrow$

Range of the ratios e1/e3, e2/e3 (and so on) decreases (constraints get tighter).

$\downarrow$

Solution space gets smaller.

$\downarrow$

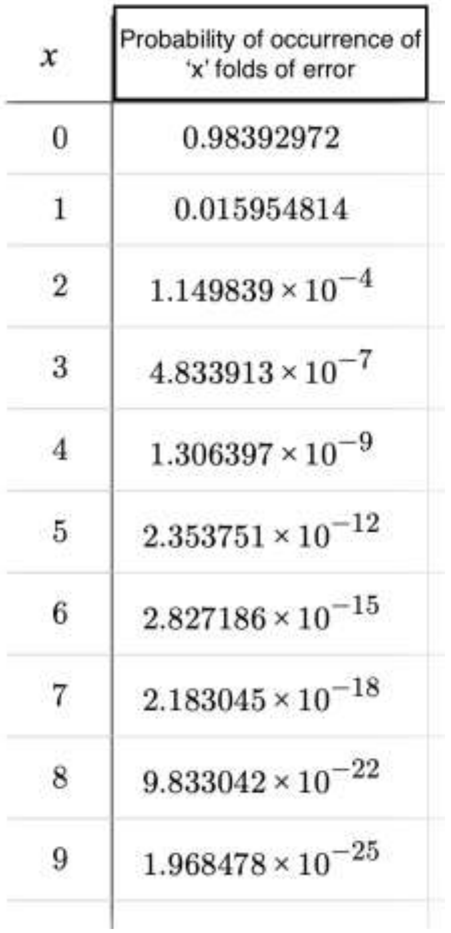
Less chance of obtaining an undesired solution.

ISSUES:

4. **This method assumes that |P| does not suffer any transmission errors.**

   Not a good assumption.

Probability that |P| suffers from a transmission error = **2.6×10^{-3}**



x	Probability of occurrence of 'x' folds of error
0	0.98392972
1	0.015954814
2	1.149839×10^{-4}
3	4.833913×10^{-7}
4	1.306397×10^{-9}
5	2.353751×10^{-12}
6	2.827186×10^{-15}
7	2.183045×10^{-18}
8	9.833042×10^{-22}
9	1.968478×10^{-25}

$N = 8$,
$k = 10$,
$\varepsilon = 10^{-4}$

SUMMARY

- **Computationally expensive:** Matrix operations, MINLP problem.
- **Too much overhead:**

  Size of message block = N bits &

  k (in M^k), then

  $$\text{Number of overhead bits} \approx \underbrace{(0.8k + 2)*9}_{\text{Extra bits in encoded matrix}} + \underbrace{(3N+2)}_{\text{Number of bits to represent } |P|}$$

- **4 inherent issues.** Can’t eradicate them, only reach a compromise.
- **Impractical.**

THANK YOU