Int. J. Open Problems Compt. Math., Vol. 11, No. 4, December 2018
ISSN 1998-6262;
www.i-csrs.org

Coding theory on balancing numbers

Bandhu Prasad

Department of Mathematics, Kandi Raj College
Kandi - 742137, India.
e-mail:bandhu_iit@rediffmail.com

Received 15 August 2018; Accepted 29 October 2018

(Communicated by Nedal Tahat)

Abstract

In this paper, we introduced $Q^n_{1,2}$ matrix whose elements are balancing numbers and developed a new coding and decoding method followed from $Q^n_{1,2}$ matrix. We established the relations between the code matrix elements, error detection and correction for this coding theory. Correction ability of this method is 93.33%.

Keywords: *Balancing numbers, k-Balancing numbers, Balancing constant, Code matrix.*
2010 Mathematics Subject Classification: 11B39, 11B83.

1 Introduction

A positive integer n is called a balancing number if $1+2+3+\cdots+(n-1) = (n+1)+(n+2)+(n+3)+\cdots+(n+r)$ for some nonnegative integer r and calling r as the balancer corresponding to n. First few balancing numbers are 1,6,35,204,1189 with balancers 0,2,14,84,492 respectively. Balancing numbers are studied and generalized in many ways [1,2,3]. In [4], Szakaacs defined multiplying balancing numbers. A positive integer n is called a multiplying balancing number if $1\cdot2\cdot3\cdot\cdots\cdot(n-1) = (n+1)\cdot(n+2)\cdot(n+3)\cdot\cdots\cdot(n+r)$ for some positive integer r which is called as multiplying balancer corresponding to the multiplying balancing number n.

2 Preliminaries

2.1 k-balancing numbers

For any positive number k, k-balancing numbers, $\{B_{k,n}\}_{n=0}^{\infty}$, defined by the recurrence relation

$$B_{k,n+1} = 6kB_{k,n} - B_{k,n-1}, \quad \text{for} \;\; n \geq 1 \tag{1}$$

with initial terms

$$B_{k,0} = 0, B_{k,1} = 1.$$

For $k = 1$, (1) gives the sequence of balancing numbers and the characteristic equation for (1) is given by

$$\alpha^2 - 6k\alpha + 1 = 0 \tag{2}$$

where the roots are $\alpha_1 = 3k + \sqrt{9k^2 - 1}$ and $\alpha_2 = 3k - \sqrt{9k^2 - 1}$.
For $k = 1$, $\alpha_1 = 3 + 2\sqrt{2}$, $\alpha_2 = 3 - 2\sqrt{2}$, α_1, α_2 are conjugate of each other and $\lambda = 3 + 2\sqrt{2}$ is known as balancing constant.

Lemma 1

For any integer $n \geq 1$, $\alpha_1^{n+2} = 6k\alpha_1^{n+1} - \alpha_1^n$ and $\alpha_2^{n+2} = 6k\alpha_2^{n+1} - \alpha_2^n$.

Proof:

Since α_1 and α_2 are roots of $\alpha^2 - 6k\alpha + 1 = 0$.
Therefore, $\alpha_1^2 - 6k\alpha_1 + 1 = 0$ and $\alpha_2^2 - 6k\alpha_2 + 1 = 0$.
Hence, $\alpha_1^{n+2} - 6k\alpha_1^{n+1} + \alpha_1^n = 0 \Longrightarrow \alpha_1^{n+2} = 6k\alpha_1^{n+1} - \alpha_1^n$ and $\alpha_2^{n+2} - 6k\alpha_2^{n+1} + \alpha_2^n = 0 \Longrightarrow \alpha_2^{n+2} = 6k\alpha_2^{n+1} - \alpha_2^n$.

Lemma 2 Binet's formula

The closed form of nth k-balancing number is $B_{k,n} = \frac{\alpha_1^n - \alpha_2^n}{\alpha_1 - \alpha_2}$, where $\alpha_1 = 3k + \sqrt{9k^2 - 1}$ and $\alpha_2 = 3k - \sqrt{9k^2 - 1}$.

Proof:

We will proof this result by mathematical induction. It is clearly that this result is true for $n = 0$ and $n = 1$. Assume that this result is true for all j such that $0 \leq j \leq m$ for some positive integer m.
Therefore, $B_{k,m+1} = 6kB_{k,m} - B_{k,m-1} = 6k\frac{\alpha_1^m - \alpha_2^m}{\alpha_1 - \alpha_2} - \frac{\alpha_1^{m-1} - \alpha_2^{m-1}}{\alpha_1 - \alpha_2} = \frac{\alpha_1^{m-1}(6k\alpha_1 - 1) - \alpha_2^{m-1}(6k\alpha_2 - 1)}{\alpha_1 - \alpha_2} = \frac{\alpha_1^{m+1} - \alpha_2^{m+1}}{\alpha_1 - \alpha_2}$.
Hence the result.

Lemma 3

For any integer $n \geq 1$, $\alpha_1^n = \alpha_1 B_{k,n} - B_{k,n-1}$ and $\alpha_2^n = \alpha_2 B_{k,n} - B_{k,n-1}$.

Proof:

We know that $\alpha_1^2 = 6k\alpha_1 - 1 = \alpha_1 B_{k,2} - B_{k,1}$
$\alpha_1^3 = 6k\alpha_1^2 - \alpha_1 = (36k^2 - 1)\alpha_1 - 6k = \alpha_1 B_{k,3} - B_{k,2}$

Consequently, $\alpha_1^n = \alpha_1 B_{k,n} - B_{k,n-1}$ and similarly $\alpha_2^n = \alpha_2 B_{k,n} - B_{k,n-1}$. Hence the result.

Lemma 4

Any two consecutive k-balancing numbers are relatively prime that is $(B_{k,n}, B_{k,n-1}) = 1$.

Proof:

Let us suppose that $B_{k,n}$ has original dividend and $B_{k,n-1}$ has original divisor. Then from Euclidean Algorithm, We have

$$B_{k,n} = 6kB_{k,n-1} - B_{k,n-2}$$

$$B_{k,n-1} = 6kB_{k,n-2} - B_{k,n-3}$$

$$\vdots$$

$$B_{k,3} = 6kB_{k,2} - B_{k,1}$$

$$B_{k,2} = 6kB_{k,1} - B_{k,0}$$

Therefore, $(B_{k,n}, B_{k,n-1}) = B_{k,1} = 1$.

Lemma 5

For any positive integers n and m, $B_{k,m}$ divides $B_{k,nm}$.

Proof:

It is clearly that $B_{k,m}$ divides $B_{k,nm}$ for $n = 1$. Let us suppose that it true for all $t \geq 1$. Then we have to show that it is also true for $t + 1$.

$B_{k,m(t+1)} = B_{k,mt+m} = B_{k,mt}B_{k,m+1} - B_{k,mt-1}B_{k,m} = k_1 B_{k,m}B_{k,m+1} - B_{k,mt-1}B_{k,m}$.

Hence $B_{k,m}$ divides $B_{k,nm}$.

Lemma 6

For any positive integers n and l, $B_{k,ln-1}$ and $B_{k,n}$ are relatively prime that is $(B_{k,ln-1}, B_{k,n}) = 1$.

Proof:

Let $d = (B_{k,ln-1}, B_{k,n})$. Then d divides $B_{k,ln-1}$ and $B_{k,n}$. But $B_{k,n}$ divides $B_{k,ln}$ from the properties of balancing numbers. Therefore d divides $B_{k,ln-1}$ and $B_{k,ln}$. But we know that $(B_{k,ln-1}, B_{k,ln}) = 1$. Hence $d = 1$. Therefore, $(B_{k,ln-1}, B_{k,n}) = 1$.

Lemma 7

For positive integers m, n, l, r with $m = ln + r$, $(B_{k,m}, B_{k,n}) = (B_{k,n}, B_{k,r})$.

Proof:

We have $(B_{k,m}, B_{k,n}) = (B_{k,ln+r}, B_{k,n}) = (B_{k,ln}B_{k,r+1} - B_{k,ln-1}B_{k,r}, B_{k,n}) = (B_{k,ln-1}B_{k,r}, B_{k,n}) = (B_{k,r}, B_{k,n})$.

Lemma 8
For positive integers m, n $(B_{k,m}, B_{k,n}) = B_{k,(m,n)}$.
Proof:
Using Euclidean Algorithm, for $m \geq n$, We have
$m = q_0 n + r_1$, $0 \leq r_1 < n$
$n = q_1 r_1 + r_2$, $0 \leq r_2 < r_1$
$\vdots$
$r_{n-2} = q_{n-1} r_{n-1} + r_n$, $0 \leq r_n < r_{n-1}$
$r_{n-1} = q_n r_n + 0$
From the properties of balancing numbers, $(B_{k,m}, B_{k,n}) = (B_{k,n}, B_{k,r_1}) = (B_{k,r_1}, B_{k,r_2}) = \cdots = (B_{k,r_{n-1}}, B_{k,r_n}) = B_{k,r_n}$. We know that B_{k,r_n} divides $B_{k,r_{n-1}}$ as r_n divides r_{n-1}. Therefore, $(B_{k,r_{n-1}}, B_{k,r_n}) = B_{k,r_n} = B_{k,(m,n)}$.
Lemma 9
For positive integers m, n that are relatively prime, $B_{k,m} B_{k,n}$ divides $B_{k,nm}$.
Proof:
We know that $B_{k,m}$ divides $B_{k,nm}$ and $B_{k,n}$ divides $B_{k,nm}$. From the properties of balancing numbers, $(B_{k,m}, B_{k,n})$ divides $B_{k,nm}$. But $(B_{k,m}, B_{k,n}) = B_{k,(m,n)} = B_{k,1} = 1$.
Therefore, $B_{k,m} B_{k,n}$ divides $B_{k,nm}$.

3 Main results

3.1 Balancing matrix, $Q_{1,2}$ and its properties

In this section, we define a new balancing matrix, $Q_{1,2}$ of order 2

$$Q_{1,2} = \begin{pmatrix} 6 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} B_{1,2} & -B_{1,1} \\ B_{1,1} & B_{1,0} \end{pmatrix} \tag{3}$$

The nth power of the $Q_{1,2}$ is given by $Q_{1,2}^n$

$$Q_{1,2}^n = \begin{pmatrix} B_{1,n+1} & -B_{1,n} \\ B_{1,n} & -B_{1,n-1} \end{pmatrix} \tag{4}$$

The determinant of $Q_{1,2}$, det $Q_{1,2} = 1$ and determinant of $Q_{1,2}^n$, det $Q_{1,2}^n = 1$ i.e. $B_{1,n}^2 - B_{1,n+1} B_{1,n-1} = 1$, which is known as Cassini formula for balancing numbers.

3.2 Balancing coding and decoding method

In this paper, we introduce a new coding theory, which is known as balancing coding and decoding method. In this method, we represent the message in the form of nonsingular square matrix, M of order 2 and we represent the balancing matrix, $Q^n_{1,2}$ of order 2 as coding matrix and its inverse matrix $(Q^n_{1,2})^{-1}$ as a decoding matrix. We represent a transformation $M \times Q^n_{1,2} = E$ as balancing coding and a transformation $E \times (Q^n_{1,2})^{-1} = M$ as balancing decoding. We represent the matrix, E as code matrix.

3.2.1 Example of the balancing coding and decoding method

Let us represent the initial message in form of the nonsingular square matrix, M of order 2

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}. \tag{5}$$

Assume that all elements of the matrix are positive integer i.e., m_1, m_2, m_3, $m_4 > 0$. Let us now select for any value of n for a $Q^n_{1,2}$ matrix. We simply write for $n = 2$

$$Q^2_{1,2} = \begin{pmatrix} B_{1,3} & -B_{1,2} \\ B_{1,2} & -B_{1,1} \end{pmatrix} = \begin{pmatrix} 35 & -6 \\ 6 & -1 \end{pmatrix}. \tag{6}$$

Then the inverse of $Q^2_{1,2}$ is given by

$$(Q^2_{1,2})^{-1} = \begin{pmatrix} -B_{1,1} & B_{1,2} \\ -B_{1,2} & B_{1,3} \end{pmatrix} = \begin{pmatrix} -1 & 6 \\ -6 & 35 \end{pmatrix}. \tag{7}$$

Then the coding of the message (5) consists of the multiplication of the initial matrix (6) that is

$$M \times Q^2_{1,2} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} 35 & -6 \\ 6 & -1 \end{pmatrix} = \begin{pmatrix} 35m_1 + 6m_2 & -6m_1 - m_2 \\ 35m_3 + 6m_4 & -6m_3 - m_4 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} = E \tag{8}$$

where $e_1 = 35m_1 + 6m_2,\ e_2 = -6m_1 - m_2,\ e_3 = 35m_3 + 6m_4,\ e_4 = -6m_3 - m_4$. Then the code message, $E = e_1,\ e_2,\ e_3,\ e_4$ is sent to a channel. The decoding of the code message, E is given by following way,

$$\begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \begin{pmatrix} -1 & 6 \\ -6 & 35 \end{pmatrix} = \begin{pmatrix} -e_1 - 6e_2 & 6e_1 + 35e_2 \\ -e_3 - 6e_4 & 6e_3 + 35e_4 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} = M.$$

3.3 Determinant of the code matrix, E

The code matrix, E is defined by the following formula $E = M \times Q^n_{1,2}$. According to the matrix theory [5,6] we have

$$DetE = Det(M \times Q^n_{1,2}) = DetM \times DetQ^n_{1,2} = DetM \times (1)^n = DetM. \tag{9}$$

3.4 Relations between the code matrix elements

We can write the code matrix, E and the initial message, M as the following

$$E = M \times Q_{1,2}^{n} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} B_{1,n+1} & -B_{1,n} \\ B_{1,n} & -B_{1,n-1} \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix}$$

and

$$M = E \times (Q_{1,2}^{n})^{-1} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \begin{pmatrix} -B_{1,n-1} & B_{1,n} \\ -B_{1,n} & B_{1,n+1} \end{pmatrix} = \begin{pmatrix} -e_1 B_{1,n-1} - e_2 B_{1,n} & e_1 B_{1,n} + e_2 B_{1,n+1} \\ -e_3 B_{1,n-1} - e_4 B_{1,n} & e_3 B_{1,n} + e_4 B_{1,n+1} \end{pmatrix} \tag{10}$$

.

Since $m_1,\ m_2,\ m_3,\ m_4$ are positive integers, we have

$$m_1 = -e_1 B_{1,n-1} - e_2 B_{1,n} > 0, \tag{11}$$

$$m_2 = e_1 B_{1,n} + e_2 B_{1,n+1} > 0, \tag{12}$$

$$m_3 = -e_3 B_{1,n-1} - e_4 B_{1,n} > 0, \tag{13}$$

$$m_4 = e_3 B_{1,n} + e_4 B_{1,n+1} > 0. \tag{14}$$

From (11) and (12) we get

$$\frac{B_{1,n}}{B_{1,n-1}} < -\frac{e_1}{e_2} < \frac{B_{1,n+1}}{B_{1,n}}. \tag{15}$$

From (13) and (14) we get

$$\frac{B_{1,n}}{B_{1,n-1}} < -\frac{e_3}{e_4} < \frac{B_{1,n+1}}{B_{1,n}}. \tag{16}$$

Therefore, for large value of n we get

$$-\frac{e_1}{e_2} \approx \lambda, \quad -\frac{e_3}{e_4} \approx \lambda \text{ where } \lambda = 3 + 2\sqrt{2}. \tag{17}$$

3.5 Error detection and correction

3.5.1 Error detection

The main aim of the coding theory is the detection and correction of errors arising in the code message, E under influence of noise in the communication channel. The most important idea is using the property of determinant of the matrix as the check criterion of the transmitted message, E. Let the initial message, M is given by

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \tag{18}$$

where all elements m_1, m_2, m_3, m_4 of the matrix, M are positive integers. Now determinant of M is

$$Det\ M = m_1 m_4 - m_2 m_3 \tag{19}$$

and the code message, E

$$E = M \times Q_{1,2}^n. \tag{20}$$

So,

$$Det\ E = Det\ (M \times Q_{1,2}^n) = Det\ M \times Det\ Q_{1,2}^n = Det\ M \times (1)^n = Det\ M. \tag{21}$$

This shows that the determinant of the initial message, M is connected with the determinant of the code message, E by a very simple relation. The identity (21) gives the new method of the error detection based on the application of the $Q_{1,2}^n$ matrix. The essence of the method consists that the sender calculates the determinant of the initial message, M represents in the matrix form (18) and sends it to the channel after the code message, E (20). The receiver calculates the determinant of the code message, E (20) and compares the determinant of the initial message of M (18) received from the channel. If this comparison corresponds to (21) it means that the code message, E (20) is correct and the receiver can decode the code message, E (20) otherwise the code message, E (20) is not correct. Error detection is the first step in communication of messages.

3.5.2 Error correction

The possibility of restoration of the code message, E can be done by using the property of the $Q_{1,2}^n$ matrix. For selecting $n = 2$, $Q_{1,2}^n$ matrix will be

$$Q_{1,2}^2 = \begin{pmatrix} B_{1,3} & -B_{1,2} \\ B_{1,2} & -B_{1,1} \end{pmatrix} = \begin{pmatrix} 35 & -6 \\ 6 & -1 \end{pmatrix}. \tag{22}$$

Then the coding of the message (18) consists of the multiplication of the initial matrix (22) that is

$$M \times Q_{1,2}^2 = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} 35 & -6 \\ 6 & -1 \end{pmatrix} =$$

$$\begin{pmatrix} 35m_1 + 6m_2 & -6m_1 - m_2 \\ 35m_3 + 6m_4 & -6m_3 - m_4 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} = E \tag{23}$$

where $e_1 = 35m_1 + 6m_2$, $e_2 = -6m_1 - m_2$, $e_3 = 35m_3 + 6m_4$, $e_4 = -6m_3 - m_4$. After constructing the code matrix, E, we calculate the determinant of the initial matrix, M (18). The determinant is sent to the communication channel after the code message, $E = e_1, e_2, e_3, e_4$. Assume that the communication channel has the special means for the error detection in each of elements e_1, e_2, e_3, e_4 of the code message, E. Assume that the first element e_1 of E is received with the error. Then, we can represent the code message in the matrix form

$$E' = \begin{pmatrix} u & e_2 \\ e_3 & e_4 \end{pmatrix} \tag{24}$$

where u is the destroyed element of the code message, E but the rest matrix entries must be correct and equal to the following:

$$e_2 = -6m_1 - m_2; e_3 = 35m_3 + 6m_4; e_4 = -6m_3 - m_4. \tag{25}$$

Then, according to the properties of the coding method, we can write the following equation for calculation of u

$$ue_4 - e_2e_3 = u(-6m_3 - m_4) - (-6m_1 - m_2)(35m_3 + 6m_4) = (m_1m_4 - m_2m_3). \tag{26}$$

From (26), we get

$$u = 35m_1 + 6m_2. \tag{27}$$

Comparing the calculated value (27) with the entry e_1 of the code matrix, E given with (23) we conclude that $u = e_1$. Thus, we have restored the code message, E using the property of determinant of the $Q_{1,2}^n$ matrix. But in the real situation usually we do not know what element of the code message is destroyed. In this case, we suppose different hypotheses about the possible destroyed elements and then we test these hypotheses. However, we have one more condition for the elements of the code matrix, E that all its elements are integers. Our first hypothesis is that we have the case of single error in the code matrix, E received from the communication channel. It is clear that there are four variants of the single errors in the code matrix, E:

$$(a) \begin{pmatrix} u & e_2 \\ e_3 & e_4 \end{pmatrix} (b) \begin{pmatrix} e_1 & v \\ e_3 & e_4 \end{pmatrix} (c) \begin{pmatrix} e_1 & e_2 \\ w & e_4 \end{pmatrix} (d) \begin{pmatrix} e_1 & e_2 \\ e_3 & z \end{pmatrix} \tag{28}$$

where u, v, w, z are destroyed elements. In this case we can check different hypotheses (28). For checking the hypothesis (a), (b), (c), (d) we can write the following algebraic equations based on the checking relation (21):

$$ue_4 - e_2e_3 = Det\ M(\text{a possible single error is in the element } e_1), \tag{29}$$

$$e_1e_4 - ve_3 = Det\ M(\text{a possible single error is in the element } e_2), \tag{30}$$

$$e_1e_4 - e_2w = Det\ M(\text{a possible single error is in the element } e_3), \tag{31}$$

$$e_1z - e_2e_3 = Det\ M(\text{a possible single error is in the element } e_4). \tag{32}$$

It follows from (29)-(32) four variants for calculation of the possible single errors.

$$u = \frac{Det\ M + e_2e_3}{e_4}, \tag{33}$$

$$v = \frac{-Det\ M + e_1e_4}{e_3}, \tag{34}$$

$$w = \frac{-Det\ M + e_1e_4}{e_2}, \tag{35}$$

$$z = \frac{Det\ M + e_2e_3}{e_1}. \tag{36}$$

The formula (33)-(36) give four possible variants of single error but we have to choice the correct variant only among the cases of the integer solutions u, v, w, z; besides, we have to choice such solutions, which satisfies the additional checking relations (17). If calculations by formulas (33)-(36) do not give an integer result we have to conclude that our hypothesis about single error is incorrect or we have error in the checking element Det M. For the latter case we can use the approximate equalities (17) for checking a correctness of the code matrix, E. By analogy we can check all hypotheses of double error in the code matrix. As example let us consider the following case of double errors in the code matrix, E

$$\begin{pmatrix} u & v \\ e_3 & e_4 \end{pmatrix} \tag{37}$$

where u, v are the destroyed elements of the code message. Using the first checking relation (21) we can write the following algebraic equation for the matrix (37):

$$ue_4 - ve_3 = Det\ M. \tag{38}$$

However, according to the second checking relation (17) there is the following relation between u and v :

$$u \approx -\lambda v. \tag{39}$$

It is important to emphasize that (38) is Diophantine one. As the Diophantine equation (38) has many solutions we have to choice such solutions u, v which satisfy to the checking relation (39). By analogy one may prove that using checking relations (17), (21) by means of solution of the Diophantine equation similar to (38) we can correct all possible double errors in the code matrix. However, we can show by using such approach there is a possibility to correct all possible triple errors in the code matrix, E, for example $\begin{pmatrix} u & v \\ w & e_4 \end{pmatrix}$ etc. where u, v, w are destroyed elements. Thus, our method of error correction is based on the verification of different hypotheses about errors in the code matrix by using the checking relations (17), (21) and by using the fact that the elements of the code matrix are integers. If all our solutions do not bring to integer solutions it means that the checking element Det M is erroneous or we have the case of four fold error in the code matrix, E and we have to reject the code matrix, E as defective and not correctable. Our method allows to correct 14 cases among $({}^4C_1 +^4 C_2 +^4 C_3 +^4 C_4) = 2^4 - 1 = 15$ cases. It means that correction ability of the method is equal to $\frac{14}{15} = 0.9333 = 93.33\%$.

3.6 Redundancy of the balancing coding method

For code redundancy, we use the concept of absolute and relative redundancy. Absolute redundancy, r is given by $r = s - l$ where s is the number of bits in the code message and l is the number of bits in initial message. The relative redundancy, R_d is given by $R_d = \frac{s-l}{s}$. Let $l = 4t$. For encoding, the initial message is represented in the form of data matrix, M

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$$

which means that each elements of the data matrix consists t bits. Let us consider the encoding matrix, $Q_{1,2}^n$

$$Q_{1,2}^n = \begin{pmatrix} B_{1,n+1} & -B_{1,n} \\ B_{1,n} & -B_{1,n-1} \end{pmatrix}.$$

The code matrix, E

$$E = M \times Q_{1,2}^n = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix}.$$

The code message, E entering the channel consists of the five elements, the checking element Det M and the four elements of the code matrix e_1, e_2, e_3, e_4. The checking element Det M is the main source of redundancy of the code message entering the channel. We need to calculate the number of bits necessary for the representation of Det M. For this purpose, we calculate the maximal value of the Det M. It is obvious that the determinant of M can reach its maximal value when the product of m_1 and m_4 is maximal, the product of m_2 and m_3 is minimal. If we neglect the minimal value m_2m_3 in comparison to maximal value m_1m_4, we can estimate the maximum value of the Det M as follow $(\text{Det}M)_{max.} \simeq (2^t)(2^t) = 2^{2t}$. It follows that we need $2t$ bits for the binary representation of the Det M. The code matrix, E is redundant with respect to the initial(data) matrix, M. By comparing the code matrix elements e_1, e_2, e_3, e_4 with data matrix elements m_1, m_2, m_3, m_4, we conclude that for the case $n \geq 1$ the numerical values of the code matrix elements e_1, e_2, e_3, e_4 are more than the numerical values of the data matrix elements m_1, m_2, m_3, m_4. This means that for binary representation of the code matrix elements e_1, e_2, e_3, e_4 we need no less than $l = 4t$ bits. In order to obtain the lowest estimation of the redundancy of this coding method, we conclude that we need $2t$ bits for the representation of the checking element Det M and we need no less than $4t$ bits for the representation of the code matrix, E. Therefore, the representation of the code message entering the channel, we need no less than $6t$ bits. The ratio of the number of the checking bits $(2t)$ to the general number of bits $(6t)$ is the lowest estimation of the relative redundancy of this coding method. Relative redundancy, $R_d = \frac{1}{3} = 0.333 = (33.3\%)$.

3.7 Comparison of the balancing coding method to the classical coding method

The balancing coding method is based on matrix approach which possesses many peculiarities and advantages in comparison to classical (algebraic) coding method. The use of matrix theory for designing new error-correction codes is the first peculiarity of the this coding method. The large information units, in particular matrix elements, are objects of detection and correction of errors in this coding method. There is no theoretical restrictions for the value of the numbers that can be matrix elements whereas in algebraic coding theory there are very small information elements, bits and their combinations are the objects of detection and correction. This coding method has very high correction ability in comparison to classical coding method.

4 Conclusion

The balancing coding method is the main application of the $Q_{1,2}^n$ matrix. This coding method reduces to matrix multiplication, a well-known algebraic operation, which is realized very well in modern computers. The main practical peculiarity of this method is that large information units, in particular, matrix elements, are objects of detection and correction of errors. The elements of the initial matrix, M and therefore the elements of the code matrix, E can be the numbers of unlimited value. It means that theoretically this coding method allows to correct the numbers of unlimited value. The correction ability of this method is equal 93.33% that exceeds essentially all well-known correcting codes. The correction ability, detection ability of this coding method are very high in comparison with algebraic coding method.

5 Open Problem

There is a open problem to define a $Q_{1,n}$ matrix of order n whose elements are balancing numbers. This matrix will be useful to establish the error detection and correction and relations among the code matrix elements.

6 Acknowledgement

The author would like to thank the editor and referees for their comments, which have helped in improving the presentation and clarity of the paper.

References

[1] H. Belbachair and L. Szalay, *Balancing in direction (1,-1) in Pascals triangle*, Armenian J. Math., 6(1)(2014), 32-40.

[2] A. Berczes and K. Liptai and I. Pink., *On generalized balancing sequences*, Fibonacci Quart., 48(2)(2010), 121-128.

[3] G.K. Panda and S.S. Rout, *Gap balancing numbers*, Fibonacci Quart., 51(3)(2013), 239-248.

[4] T. Szakaacs, *Multiplying balancing numbers*, Acta Univ. Sapientiae Math., 3(2011), 90-96.

[5] A.P. Stakhov, *Introduction into algorithm measurement theory*, Moscow: Soviet Radio; 1977 (In Russian).

[6] A.P. Stakhov, *Fibonacci matrices, a generalization of the cassini formula and a new coding theory*, Chaos, Solitons and Fractals, 30(2006), 56-66.