# ON RUN-LENGTH-CONSTRAINED BINARY SEQUENCES

Zaharias M. Psillakis

Department of Physics, University of Patras, Patras, Greece

## ABSTRACT

*A class of binary sequences, constrained with respect to the length of zero runs, is considered. For such sequences, termed $(d, k)$-sequences, new combinatorial and computational results are established. Explicit expressions for enumerating $(d, k)$-sequences of finite length are obtained. Efficient computational procedures for calculating the capacity of a $(d, k)$-code are given. A simple method for constructing a near-optimal $(d, k)$-code is proposed. Illustrative numerical examples demonstrate further the theoretical results.*

## KEYWORDS

*Run-length-limited sequences, $(d, k)$-codes, enumerative combinatorics, continued fraction approximation, Shannon capacity*

## 1. INTRODUCTION

In many digital communication and recording systems the structure of a bit (binary or $0-1$) stream has to be restricted because of several technical reasons [3, 15] in terms of run lengths; for instance, with respect to the length of runs of 0s. Hereafter, as a 0-run we mean a sequence of consecutive 0s, the number of which is referred to as its length. One common encoding/decoding method in such systems is the run-length-limited (RLL) code. This code usually accompanies a non-return-to-zero-inverted (NRZI) code which encodes/decodes the physical channel signals of alternating polarity.

Let the channel signals be represented by a sequence $\{y_i\}_{i\geq 0}$, $y_i \in \{-1, 1\}$ or $y_i \in \{-, +\}$. By the sequence $\{y_i\}_{i\geq 0}$, a $0-1$ sequence $\{x_i\}_{i\geq 0}$, $x_i \in \{0, 1\}$, can be obtained, and vice versa. More specifically, assuming an initial positive polarity $y_{-1} = 1$, then for $i \geq 0$ it holds

$$x_i = 1, \text{ if } y_i y_{i-1} < 0;\ 0, \text{ otherwise} \tag{1}$$

and

$$y_i = y_{i-1}(-1)^{x_i}. \tag{2}$$

*Example 1.* For instance using (1) and (2), the sequence of channel signals 1 -1 -1 1 1 1 1 -1 -1 1 -1 1 1 -1 $\cdots$ (or equivalently the sequence + - - + + + + - - + - + + - $\cdots$) would be converted to the 0-1 sequence 0 1 0 1 0 0 0 1 0 1 1 1 0 1 $\cdots$, and vice versa. $\diamondsuit$

After this recording, the focus in a communication system is the sequence $\{x_i\}_{i\geq 0}$ which satisfies certain constraints with respect to the length of 0-runs. One of the most studied such constrained sequences is the $(d, k)$-constrained sequence where $d$ and $k$ are two non-negative integers. The set of $(d, k)$-constrained sequences defines a $(d, k)$-code or a $(d, k)$-system.

More specifically, we say (Tang and Bahl [14]) that a $0-1$ sequence $\{x_i\}_{i\geq 0}$, $x_i \in \{0,1\}$, is a $(d,k)$-constrained or $(d,k)$-limited sequence, in short a $dk$-sequence, if it satisfies simultaneously the following two conditions:
$C1$. $d$-constraint - two successive 1s are separated by a 0-run of length at least $d$.
$C2$. $k$-constraint - every 0-run has length at most $k$.

If only condition $C1$ is satisfied, the sequence is said to be $d$-constrained (with $k=\infty$). If only condition $C2$ is satisfied the sequence is $k$-constrained (with $d=0$). For $d=0$ and $k=\infty$ the sequence is an unconstrained $0-1$ sequence. From now on when we refer to a $(d,k)$-constrained sequence we assume that $0 \leq d < k \leq \infty$. See also [5].

The $(d,k)$-constrained sequences and $(d,k)$-systems are extensively used in digital communication, recording and storage information from Shannon's [12] era. The treatises [3, 10] cover the vast majority of the past and recent literature on $(d,k)$-sequences. The works [2, 4, 6, 7, 13-15] provide additional information and describe methods have been used for deriving results related to such sequences. These methods include finite state sequential machines, recursive schemes, generating function method and maxentropic sequences.

The article is organized as follows. In Section 2, we present preliminary results along with three motivating and interrelating issues (Sections 2.1, 2.2 and 2.3) connected to $(d,k)$-sequences which demonstrate the aim of the article. In Section 3, we establish the methods employed in deriving our main results. More specifically, in Section 3.1 we obtain, via enumerative combinatorics, closed expressions for determining the number of $(d,k)$-constrained sequences of finite length. In Section 3.2 we establish two fast and accurate numerical procedures for computing the Shannon capacity of a $(d,k)$-code. In Section 3.3 we present an efficient technique, relied on the continued fraction approximation method, for constructing a near-optimal $(d,k)$-code. Numerical examples, presented in Sections 2 - 3, exemplify further our results. The article concludes with Section 4 followed by Appendix.

The vast majority of the presented results are new whereas some known results are recaptured using different methods and provide alternative formulae.

Throughout the article, for integers $n$, $m$, $\binom{n}{m}$ denotes the extended binomial coefficient, $\lfloor x \rfloor$ ($\lceil x \rceil$) stands for the greatest (least) integer less (greater) than or equal to $x$, $\delta_{i,j}$ denotes the Kronecker delta function for integer arguments $i$ and $j$, such that $\delta_{i,j}=1$, if $i=j$; 0, otherwise and $\mid A \mid$ denotes the cardinality of a set $A$. Also, we apply the convention $\sum_{\alpha}^{\beta}=0$, for $\alpha > \beta$, that is an empty sum is to be interpreted as zero.

## 2. MOTIVATION AND PRELIMINARY RESULTS

In this section, we consider three motivating issues connected with $(d,k)$-sequences along with some preliminary results related to them.

The first issue refers to the number of $(d,k)$-sequences and the corresponding numbers of two versions of such sequences. These numbers are required for noiseless encoding/decoding a source binary sequence to a binary $(d,k)$-sequence and vice versa. We provide, for the first time, a simple closed form expression for their computation. Such an expression, given in Section 3.1, has the advantage of being faster and working as well on any commercial computer, compared to existing recursive formulae in the literature; see e.g. [2, 3, 10, 14].

The second issue is connected to the evaluation of the Shannon capacity of a $(d,k)$-code.

For this crucial in information theory problem we propose, in Section 3.2, two new fast and accurate numerical procedures relied on numerical analysis and information theory. The provided expressions by both methods are directly implementable on any commercial computer.

The third issue is associated to the determination of the proper size of a noiseless fixed-rate block-code converting source words to code words. For this important theoretical and practical problem in discrete and applied mathematics we propose, in Section 3.3, a new and very efficient rational approximation to the irrational capacity of a $(d, k)$ block-code. This approximation has a near-optimal efficiency which is better than the existing rational approximations; see e.g. [3, 10].

Before we explicitly state the aforementioned topics, we give some necessary definitions and notation. After that we discuss the raised issues. Their solutions proposed in this article are given in Section 3.

Let $S_n(d, k)$ be the set of $(d, k)$-sequences of finite length $n$, $n > 0$, for some fixed $d$ and $k$ and $N_n(d, k) =\mid S_n(d, k) \mid$. In finite length sequences the $d$-constraint is not imposed on the 0-runs having at least one end bounded by the sequence's boundary. Hereafter, for finite-length sequences we will use the (unified) name $(d, k)$-sequences, denoted as $(d, k; n)$, and we reserve the names $dk$-sequences and $d$-sequences, for $0 \leq d < k \leq n$ and $0 \leq d \leq n < k = n + 1$, respectively. That is, for $d$-sequences we use the convention $k = n + 1$ for $k = \infty$. Clearly, a $k$-sequence is a particular case of a $dk$-sequence with $0 = d < k \leq n$. Consequently, we denote as $(0, n; n)$ an unconstrained $0 - 1$ sequence of length $n$.

*Example 2.* To have a picture of the numbers $N_n(d, k)$ and the sets $S_n(d, k)$ we consider for instance, $n = 4$, $d = 1$ and $k = 2$. Then, via the forthcoming formula (21), we find that among all $2^4 = 16$ $0-1$ sequences of length 4 there are $N_4(1, 2) = 5$ $(1, 2; 4)$ sequences. The latter sequences are the elements of $S_4(1, 2) = \{0010, 0100, 0101, 1001, 1010\}$. $\diamondsuit$

For $(d, k)$-sequences, let $S_n(i, j; d, k)$ and $S_n(1; d, k)$ be the subsets of $S_n(d, k)$ which contain those $(d, k; n)$ that start with an $i$ and end with a $j$, $i, j \in \{0, 1\}$, and those $(d, k; n)$ having their left most symbol 1, respectively. Such sequences are called $(i, j)$-restricted and 1-left-restricted $(d, k)$-constrained sequences, respectively. Moreover, let us denote $N_n^{(i,j)}(d, k) =\mid S_n(i, j; d, k) \mid$ and $N_n^{(1)}(d, k) =\mid S_n(1; d, k) \mid$. Clearly, for $0 \leq d < k \leq n$,

$$N_n(d, k) = \sum_{(i,j)\in\{0,1\}^2} N_n^{(i,j)}(d, k),\ N_n^{(1)}(d, k) = \sum_{j=0}^{1} N_n^{(1,j)}(d, k),\ N_n^{(0,1)}(d, k) = N_n^{(1,0)}(d, k). \quad (3)$$

## 2.1. Issue 1

The exact computation of the numbers $N_n(d, k)$ goes back to Shannon [12]. This is an interesting counting problem since, besides its own merit, the numbers $N_n(d, k)$, for $0 \leq d < k \leq n$ and $0 \leq d \leq n < k = n + 1$, are used in the enumeration coding system [14] for easily noiseless encoding/decoding $m$ unconstrained source bits to $n$ $(d, k)$-constrained channel bits. Readily, in such conversions it is true that

$$2^m \leq N_n(d, k). \quad (4)$$

In this coding scheme, noiseless encoding/decoding is achieved by changing the weighting system representing an integer number $x$, $0 \leq x \leq 2^m - 1$, i.e. an $x = (x)_{10}$, from the system $(2^{m-1}, 2^{m-2}, \ldots, 2^0)$ to the system $(N_{n-1}(d,k), N_{n-2}(d,k), \ldots, N_0(d,k))$ with $m$ and $n$ satisfying (4). More specifically, for $b_i, c_i \in \{0,1\}$, we have that

$$x = \sum_{i=0}^{m-1} b_i 2^i, \tag{5}$$

is encoded as

$$y = \sum_{i=0}^{n-1} c_i N_i(d, i+1), \text{ for } d-\text{sequences; } \sum_{i=0}^{n-1} c_i N_i(d,k), \text{ for } dk-\text{sequences,} \tag{6}$$

such that

$$y - x = 0, \text{ for } d-\text{sequences; } \sum_{i=0}^{n-k-1} N_i^{(1)}(d,k), \text{ for } dk-\text{sequences.} \tag{7}$$

Consequently, a $y$ written as in (6) is decoded as $x$ written as in (5) so that $y - x$ satisfies (7).

Notice that, using the number weighting system $(N_{n-1}(d,k), N_{n-2}(d,k), \ldots, N_0(d,k))$ there is a 1-1 mapping from the lexicographic ordered set $S_n(d,k)$ of $N_n(d,k)$ $(d,k;n)$ sequences onto the set of integer numbers $\{0, 1, \ldots, N_n(d,k) - 1\}$. Usually, the first $2^m$ such sequences are chosen for representing the numbers $0 \le x \le 2^m - 1$ and the rest $N_n(d,k) - 2^m$ sequences might be used as special patterns for checking and error detection purposes.

Readily, such encoding/decoding requires exact calculation of the numbers $N_n(d,k)$. In addition, using the enumerating coding system the exact computation of the numbers $N_n^{(1)}(d,k)$ is required as well.

### 2.2. Issue 2

When $m$ unconstrained source symbols (bits) are converted to $n$ $(d,k)$-constrained channel symbols (bits) and vice versa, the interest is focused on what should be the maximum rate

$$R = m/n,\ 0 < m \le n, \tag{8}$$

for certain values of $d$ and $k$, $0 \le d < k \le \infty$. It is well known [12] that the maximum value of $R$ that can be achieved is termed Shannon capacity (or asymptotic information rate). The capacity, denoted by $C(d,k)$, is governed by the specified constraints $d, k$ and is given by

$$C(d,k) = \lim_{n\to\infty} C_n(d,k), \tag{9}$$

where

$$C_n(d,k) = [\log_2 N_n(d,k)]/n,\ n \ge 1. \tag{10}$$

Clearly, $C(0,\infty) = 1$. Moreover, it holds [5]

$$C(d,\infty) = C(d-1, 2d-1),\ d \ge 1. \tag{11}$$

By (9)-(10), the evaluation of $C(d,k)$ involves the calculation of $N_n(d,k)$. Simple lower and upper bounds on $C(d,k)$ have been provided in [15].

## 2.3. Issue 3

One of the main problems in coding via an $S_n(d,k)$ code is that of the construction of an efficient and state independent noiseless block code with source words of length $m$ and code words of length $n$. For given values of $d$ and $k$, this problem calls for determining positive integers $m$ and $n$, such that the code rate $R = m/n$, approaches from below the asymptotic information rate $C = C(d,k)$. In other words, for

$$\eta = R/C \leq 1, \tag{12}$$

denoting the code's efficiency, with $C > 0$, the problem reduces to that of finding non-negative integers $m$ and $n$ for which $\eta$ becomes maximum or equivalently the relative efficiency

$$\eta^{'} = 1 - \eta, \tag{13}$$

becomes minimum. Such a $[m, n; d, k]$ code is called near optimal $(d,k)$-code. Clearly, the sizes $m$ and $n$ of the source and the code words, respectively, with rate $R = m/n \leq C$, must satisfy (4). For details on fixed-rate codes see e.g the surveys of [3, 10].

## 3. METHODOLOGY AND MAIN RESULTS

In this section, we establish the methods employed in deriving our results connected to the three motivating issues 2.1, 2.2 and 2.3. The corresponding results on the three relevant topics are presented in Sections 3.1, 3.2 and 3.3, respectively.

### 3.1. Enumeration of $(d,k)$-sequences of finite length

Next, we introduce for the first time a closed form solution regarding the enumeration of $(d,k)$-sequences. This is accomplished by determining explicitly the numbers $N_n(d,k)$, $N_n^{(1)}(d,k)$ and $N_n^{(i,j)}(d,k)$, $(i,j) \in \{0,1\}$, for $n > 0$, via a combinatorial technique.

To that end, for $n = 0$, we define $N_0^{(1)}(d,k) = N_0(d,k) = 1$ and we start with $d$-sequences and then continue with $dk$-sequences. For the latter sequences, we first recall two combinatorial numbers from [9].

*Lemma 1.* The coefficient

$$\begin{aligned} H_{i,r-i}(\alpha, m, k_1, k_2) \quad = \quad & \sum_{j_1=0}^{\lfloor \frac{\alpha}{k_1+1} \rfloor} \binom{i}{j_1} \sum_{j_2=0}^{\lfloor \frac{\alpha-(k_1+1)j_1}{k_2+1} \rfloor} (-1)^{j_1+j_2} \binom{r-i}{j_2} \\ & \times \binom{\alpha + m - (k_1+1)j_1 - (k_2+1)j_2 - 1}{\alpha - (k_1+1)j_1 - (k_2+1)j_2}, \end{aligned} \tag{14}$$

is the number of allocations of $\alpha$ indistinguishable balls into $m$ distinguishable urns, $i$ specified of which have capacity $k_1$ and each of $r-i$ specified urns has capacity $k_2$, $0 \leq r \leq m$, $0 \leq i \leq r$, $k_1 \geq 0$, $k_2 \geq 0$. Notice that, $H_{i,r-i}(\alpha, m, k_1, k_2)$ equivalently gives the number of integer solutions of the equation $x_1 + x_2 + \ldots + x_m = \alpha$ such that $0 \leq x_j \leq k_1$, $1 \leq j \leq i$; $0 \leq x_{i+j} \leq k_2$, $1 \leq j \leq r-i$; $x_j \geq 0$, $r+1 \leq j \leq m$. $\diamondsuit$

As a particular case of (14) we get the following corollary.

*Corollary 1.* The coefficient

$$H(\alpha, r, k) = \sum_{j=0}^{\lfloor \frac{\alpha}{k+1} \rfloor} (-1)^j \binom{r}{j} \binom{\alpha + r - (k+1)j - 1}{\alpha - (k+1)j}, \tag{15}$$

is the number of allocations of $\alpha$ indistinguishable balls into $r$ distinguishable urns where each urn is occupied by at most $k$ balls. Equivalently, $H(\alpha, r, k)$ gives the numbers of integer solutions of the linear equation $x_1 + x_2 + \ldots + x_r = \alpha$, with the restrictions $0 \leq x_i \leq k$, $i = 1, 2, \ldots, r$. $\diamondsuit$

We next give results referring in computing the numbers $N_n(d,k)$, $N_n^{(1)}(d,k)$ and $N_n^{(i,j)}(d,k)$ by closed form expresions. The proofs of selected equations of Theorem 1 are given in the Appendix. The proofs of the rest equations can be obtained by similar reasoning and the definitions of the prementiond numbers.

*Theorem 1.* The numbers $N_n(d,k)$, $N_n^{(1)}(d,k)$ and $N_n^{(i,j)}(d,k)$ are given by:

Case I: $d$-sequences, $0 \leq d \leq n < k = n+1$.

$$N_n(d,k) = 2^n,\ d = 0, \tag{16}$$

$$N_n(d,k) = 1 + \sum_{s=1}^{\lfloor \frac{n+d}{d+1} \rfloor} \binom{n - (s-1)d}{s}, \quad 0 < d \leq n. \tag{17}$$

Case II: $dk$-sequences, $0 \leq d < k \leq n$.
For $d = 0$, i.e. $k$-sequences,

$$N_n(d,k) = \sum_{s=\lfloor \frac{n}{k+1} \rfloor}^{n} H(n-s, s+1, k), \tag{18}$$

$$N_n^{(1)}(d,k) = \sum_{s=1}^{n} H(n-s, s, k), \tag{19}$$

$$\begin{aligned} N_n^{(i,j)}(d,k) \ &= \sum_{s=\max\{1, \lceil \frac{n+(i+j-1)k}{k+1} \rceil\}}^{n-2+i+j} H_{2-i-j,s-1}(n-s-2+i+j, s+1-i-j, k-1, k) \\ &\quad + \delta_{k,n}\delta_{i+j,0}. \end{aligned} \tag{20}$$

For $d > 0$, i.e. "standard" $dk$-sequences,

$$N_n(d,k) = \sum_{s=1}^{\lfloor \frac{n+d}{d+1} \rfloor} H_{s-1,2}(n-s-(s-1)d, s+1, k-d, k) + \delta_{k,n}, \tag{21}$$

$$N_n^{(1)}(d,k) = \sum_{s=1}^{\lfloor \frac{n+d}{d+1} \rfloor} H_{s-1,1}(n-s-(s-1)d, s, k-d, k), \tag{22}$$

$$
\begin{aligned}
N_n^{(i,j)}(d,k) &= \sum_{s=1}^{\lfloor \frac{n+d}{d+1} \rfloor} H_{s-1,2-i-j}(n-s-(s-1)d-2+i+j, s+1-i-j, k-d, k-1) \\
&\quad +\delta_{k,n}\delta_{i+j,0}. \qquad (23)
\end{aligned}
$$

*Remark 1.* For $dk$-sequences, $0 \leq d < k \leq n$, we can compute $N_n(d,k)$ and $N_n^{(1)}(d,k)$ using the expressions provided for $N_n^{(i,j)}(d,k)$ and relationship (3) but then additional summations are needed. Nevertheless, (3) can be used for double checking the validity among the several formulae given for the calculation of the numbers $N_n(d,k)$, $N_n^{(1)}(d,k)$ and $N_n^{(i,j)}(d,k)$. $\diamondsuit$

*Example 3.* To get a sense of the magnitude of the numbers $N_n^{(i,j)} = N_n^{(i,j)}(d,k)$, $N_n^{(1)} = N_n^{(1)}(d,k)$, $N_n = N_n(d,k)$, we consider $(d,k) = (2,10)$ and $n = 16, 32$. $(2,10)$ code is employed by commercial CDs and DVDs. Then by (21)-(23) we have: $N_{16}^{(0,0)} = 263$, $N_{16}^{(0,1)} = N_{16}^{(1,0)} = 123$, $N_{16}^{(1,1)} = 57$, $N_{16}^{(1)} = 180$, $N_{16} = 566$, and $N_{32}^{(0,0)} = 107104$, $N_{32}^{(0,1)} = N_{32}^{(1,0)} = 49985$, $N_{32}^{(1,1)} = 23329$, $N_{32}^{(1)} = 73314$, $N_{32} = 230403$. Clearly, these numbers satisfy the first two equations of (3), too. $\diamondsuit$

## 3.2. Computation of the capacity of a $(d,k)$-code

As it is already mentioned, the evaluation of the capacity $C(d,k)$ involves by definition, Eqs. (9) - (10), the calculation of the numbers $N_n(d,k)$. Alternatively, it can approximately be calculated by Perron-Frobenius theory using a proper root of a characteristic polynomial. In the latter case, among the different approaches used, we employ a version of that employed in [4]. More specifically, $C(d,k)$ can approximately be computed as

$$C(d,k) = \log_2 \lambda,\ \lambda = 1/\rho_0, \qquad (24)$$

where $\rho_0$ is the smallest positive real root, with $\rho_0 < 1$, of the equation

$$h(\rho_0; d,k) = 0,\ h(z; d,k) = 1 - z - z^{d+1} + z^{k+2}. \qquad (25)$$

In fact, $\rho_0 \in [1/2, 1)$ so that $0 < C(d,k) \leq 1$ and $C(d,k) = 1$ iff $(d,k) = (0,\infty)$.

We next give two new numerical methods for the computation of $C(d,k)$, for $0 \leq d < k \leq \infty$. An example clarifying their basic features is given. Both methods are fast and accurate provided they are enriched with some additional computational techniques to ensure convergence in a prespecified level of accuracy; that is, in a predetermined number, say $\nu$, of significant decimal digits (SDD). The latter statement means that an (unknown) actual real value, say $\xi$, is well approximated by the term $\xi_n$ of a properly defined sequence $\{\xi_n\}_{n\geq 0}$, of real numbers, i.e.

$$\xi_n \simeq \xi \text{ if } \mid \xi_n - \xi_{n'} \mid \leq \varepsilon \mid \xi_n \mid,\ n > n',\ \varepsilon = 0.5 \times 10^{-\nu},\ \nu \geq 1. \qquad (26)$$

Usually in computer applications we take $n' = n/2$ or $n' = n-1$. Moreover, let

$$e_n = \xi - \xi_n, \qquad (27)$$

be the error between $\xi$ and $\xi_n$ and $\hat{e}_n$ be a computable estimate of $e_n$. In order to get rapid convergence, we include acceleration schemes in the body of our methods. These formulae are based on Aitken's and Richardson's extrapolation ideas which have been successfully used in numerous topics of numerical analysis. For details see e.g. [1, 8].

### 3.2.1. Numerical method 1

The first method is relied on the definition of the numbers $C_n = C_n(d,k)$, $0 \le d < k \le n$, through (10). $C_n$ is then extrapolated to the limit using a proper version of Aitken-Richardson extrapolation scheme. More specifically, improved estimate $\hat{C}_n$ of $C_n$ along with an estimate $\hat{e}_n$ of $e_n = C(d,k) - C_n$ are given by the expressions

$$\hat{C}_n = (\theta_n C_n - C_{n/2})/(\theta_n - 1), \tag{28}$$

$$\hat{e}_n = (C_n - C_{n/2})/(\theta_n - 1), \tag{29}$$

$$\theta_n = (C_{n/2} - C_{n/4})/(C_n - C_{n/2}), \tag{30}$$

provided $\theta_n > 1$, so that eventually, for $n' = n/2$ and fixed $\varepsilon$, (26) implies

$$\hat{C}_n \simeq C(d,k). \tag{31}$$

In practice, to get rapid convergence we start with an $n = n_0 = 2^{\lceil \log_2 k \rceil}$ and then we repeatedly doubling up $n$. Moreover, the case $(d,\infty)$, $d \ge 1$, is easily solved using transformation (11).

### 3.2.2. Numerical method 2

The second method is based on the solution of (25) via the introduced iterative scheme, a predictor-corrector fixed point iteration,

$$z_n = (z_{n-1} + \tilde{z}_n)/2,\ \tilde{z}_n = 1 - z_{n-1}^{d+1}(1 - z_{n-1}^{k-d+1}),\ n \ge 1, \tag{32}$$

with

$$\begin{gathered} z_0 = 2^{-C_0},\ C_0 = (C_\ell + C_u)/2,\ C_\ell = 2\beta/(k+d+2), \\ C_u = \min\{1, \beta/(d+1)\},\ \beta = \log_2(k-d+1). \end{gathered} \tag{33}$$

The scheme is properly combined with Aitken's extrapolation of $z_n$ via the error estimate $\hat{e}_n$ of $e_n = \rho_0 - z_n$ and the improved estimate $\hat{z}_n$ of $z_n$, given by

$$\hat{z}_n = z_n + \hat{e}_n \tag{34}$$

$$\hat{e}_n = (z_n - z_{n-1})\phi_n/(1 - \phi_n), \tag{35}$$

$$\phi_n = (z_n - z_{n-1})/(z_{n-1} - z_{n-2}),\ n \ge 2, \tag{36}$$

provided $0 <| \phi_n |< 1$, so that eventually, for $n' = n - 1$ and fixed $\varepsilon$, (26) implies

$$\hat{z}_n \simeq \rho_0,\ -\log_2 \rho_0 \simeq C(d,k). \tag{37}$$

The case $(d,\infty)$, $d \ge 0$, is easily handled by setting $k$ equal to a number larger than $d$, e.g. $k = 1000(d+1)$.

*Example 4.* For illustrating methods 1 and 2, we present two indicative cases. In both cases we have used the tolerance $\varepsilon = 0.5 \times 10^{-8}$, i.e. $\nu = 8$ SDD. More specifically, in Part (A) of the example we consider a case in which we know an analytic expression of $C = C(d, k)$. Consequently, we use this knowledge to exemplify how the numerical methods work. This will be useful to understand the results of our methods in the majority of the cases for which we have no analytic results. Such a case is presented in Part (B) of the example.

Table 1. Results computed via numerical method 1. $a_b \equiv a \times 10^b$.

| $n$ | $C_n$ | $e_n = C - C_n$ | $\theta_n$ | $\hat{e}_n$ | $\hat{C}_n$ | $C - \hat{C}_n$ |
|---|---|---|---|---|---|---|
| | $(d,k) = (1,\infty)$ | | | | | |
| 1 | 1.000000000 | | | | | |
| 2 | 0.792481250 | | | | | |
| 4 | 0.750000000 | $-0.557580864_{-1}$ | $0.488494919_{+1}$ | $-0.109348278_{-1}$ | 0.739065172 | $-0.448232586_{-1}$ |
| 8 | 0.722669964 | $-0.284280506_{-1}$ | $0.155437961_{+1}$ | $-0.492984146_{-1}$ | 0.673371550 | $0.208703641_{-1}$ |
| 16 | 0.708461897 | $-0.142199835_{-1}$ | $0.192355764_{+1}$ | $-0.153840610_{-1}$ | 0.693077836 | $0.116407748_{-2}$ |
| 32 | 0.701351907 | $-0.710999312_{-2}$ | $0.199832436_{+1}$ | $-0.712192423_{-2}$ | 0.694229983 | $0.119311107_{-4}$ |
| 64 | 0.697796910 | $-0.355499656_{-2}$ | $0.199999924_{+1}$ | $-0.355499926_{-2}$ | 0.694241911 | $0.270084510_{-8}$ |
| 128 | 0.696019412 | $-0.177749828_{-2}$ | $0.200000000_{+1}$ | $-0.177749828_{-2}$ | 0.694241914 | $0.333066907_{-15}$ |
| | $(d,k) = (2,10$ | | | | | |
| 16 | 0.571541140 | | | | | |
| 32 | 0.556681249 | | | | | |
| 64 | 0.549239227 | | $0.199675434_{+1}$ | $-0.746625552_{-2}$ | 0.541772971 | |
| 128 | 0.545518220 | | $0.200000243_{+1}$ | $-0.372099774_{-2}$ | 0.541797222 | |
| 256 | 0.543657716 | | $0.200000000_{+1}$ | $-0.186050339_{-2}$ | 0.541797213 | |
| 512 | 0.542727465 | | $0.200000000_{+1}$ | $-0.930251695_{-3}$ | 0.541797213 | |

Table 2. Results computed via numerical method 2. $a_b \equiv a \times 10^b$.

| $n$ | $z_n$ | $e_n = \rho_0 - z_n$ | $\phi_n$ | $\hat{e}_n$ | $\hat{z}_n$ | $\rho_0 - \hat{z}_n$ |
|---|---|---|---|---|---|---|
| | $(d,k) = (1,\infty)$ | | | | | |
| 0 | 0.702261830 | $-0.842278417_{-1}$ | | | | |
| 1 | 0.604545076 | $0.134889128_{-1}$ | | | | |
| 2 | 0.619535164 | $-0.150117479_{-2}$ | -0.153403453 | $-0.199369196_{-2}$ | 0.617541472 | $0.492517170_{-3}$ |
| 3 | 0.617855672 | $0.178316412_{-3}$ | -0.112040120 | $0.169211877_{-3}$ | 0.618024884 | $0.910453477_{-5}$ |
| 4 | 0.618055020 | $-0.210314990_{-4}$ | -0.118695418 | $-0.211511402_{-4}$ | 0.618033869 | $0.119641190_{-6}$ |
| 5 | 0.618031506 | $0.248265287_{-5}$ | -0.117955346 | $0.248097559_{-5}$ | 0.618033987 | $0.167728587_{-8}$ |
| 6 | 0.618034282 | $-0.293034340_{-6}$ | -0.118043263 | $-0.293057690_{-6}$ | 0.618033989 | $0.233505437_{-10}$ |
| | $(d,k) = (2,10$ | | | | | |
| 0 | 0.604400511 | | | | | |
| 1 | 0.692994650 | | | | | |
| 2 | 0.686228686 | | -0.076370330 | $0.480056776_{-3}$ | 0.686708743 | |
| 3 | 0.686990943 | | -0.112660544 | $-0.771809284_{-4}$ | 0.686913762 | |
| 4 | 0.686906166 | | -0.111217566 | $0.848494107_{-5}$ | 0.686914651 | |
| 5 | 0.686915610 | | -0.111394055 | $-0.946521531_{-6}$ | 0.686914663 | |
| 6 | 0.686914558 | | -0.111374598 | $0.105401887_{-6}$ | 0.686914663 | |

(A) Let $(d, k) = (1, \infty)$. Then, we know the analytic result $C(1, \infty) = \log_2 \lambda$, where $\lambda = (1 + \sqrt{5})/2$ is the golden mean. Thus, we have $C = C(1, \infty) \doteq 0.69424191363062$.

(A1) In order to employ method 1 we use (11) and we get $C(1, \infty) = C(0, 1)$ and consequently we search for the capacity $C = C(0, 1)$. The results of the iterations of scheme (28)-(31), $C_n, \hat{C}_n$ with $n_0 = 2^{\lceil \log_2 1 \rceil} = 1$ are given in the upper part of Table 1 along with $\theta_n$, $e_n = C - C_n$, $C - \hat{C}_n$ and $\hat{e}_n$. As we can see the values of $\theta_n$ are converging to 2 such that $\theta_n > 1$ and the estimate $\hat{e}_n$ is an accurate indicator of the true error $e_n$. For instance, with $n = 128$, $C_{128} \doteq 0.696019412$, $e_{128} = C - C_{128} \doteq -0.177749828 \times 10^{-2}$, $\hat{e}_{128} \doteq -0.177749828 \times 10^{-2}$ and $\hat{C}_{128} \doteq 0.694241914$ with $C - \hat{C}_{128} \doteq 0.333066907 \times 10^{-15}$. Thus, the extrapolate $\hat{C}_{128}$ is more accurate than $C_{128}$ with $\hat{e}_{128}$ being a very accurate estimate of the error $e_{128}$. Finally, since $\mid \hat{C}_{128} - \hat{C}_{64} \mid / \hat{C}_{128} \doteq 0.39 \times 10^{-8} < 0.5 \times 10^{-8}$, it holds $\hat{C}_{128} \simeq C \doteq 0.694241914$, and $\hat{C}_{128}$ coincides in at least 8 SDD with the exact capacity $C(1, \infty)$.

(A2) The true root is $\rho_0 = 1/\lambda \doteq 0.61803398874989$. The results of the iterations of scheme (32)-(37), $z_n, \hat{z}_n$ with $k = 1002$, are given in the upper part of Table 2 along with the values of $\phi_n$, $e_n = \rho_0 - z_n$, $\rho_0 - \hat{z}_n$ and the estimate $\hat{e}_n$ of $e_n$. The values of $\phi_n$ are converging to -0.1180 such that $0 <| \phi_n |< 1$ and the estimate $\hat{e}_n$ is an accurate indicator of the true error $e_n$. For example, with $n = 6$, $z_6 \doteq 0.618034282$, $e_6 \doteq -0.293034340 \times 10^{-6}$, $\hat{e}_6 \doteq -0.293057690 \times 10^{-6}$ and $\hat{z}_6 \doteq 0.618033989$ with $\rho_0 - \hat{z}_6 \doteq 0.233505437 \times 10^{-10}$. Thus, the extrapolate $\hat{z}_6$ is more accurate than $z_6$ with $\hat{e}_6$ being an accurate estimate of the error $e_6$. Moreover, since $| \hat{z}_6 - \hat{z}_5 | /\hat{z}_6 \doteq 0.27 \times 10^{-8} < 0.5 \times 10^{-8}$, we have $\hat{z}_6 \simeq \rho_0 \doteq 0.618033989$, and $C(1, \infty) \simeq -\log_2 \hat{z}_6 = \hat{C}_6 \doteq 0.694241914$ which means that $\hat{C}_6$ coincides in at least 8 SDD with the exact capacity $C(1, \infty)$.

Therefore, (A1) and (A2) imply that both numerical methods 1 and 2 are rapid and accurate. That is, both methods give in a few iteration steps, the number of which is controlled by the desired accuracy, accurate and satisfactory results.

(B) An illustrative case for which we do not known analytically the capacity $C(d, k)$ is, for instance, with $(d, k) = (2, 10)$; i.e. for the values of $d$ and $k$ we used in Example 3. In the following we employ similar reasoning and interpretation of the results as in case (A).

(B1) Using scheme (28)-(31) with $n_0 = 2^{\lceil \log_2 10 \rceil} = 16$ and doubling up n then, i.e. for $n = 32, 64, ...$, we compute and depict the results of the iterations $C_n, \hat{C}_n$ along with the values of $\theta_n$ and the estimate $\hat{e}_n$ of the (unknown) true error $e_n$, in the lower part of Table 1. By the respective entries of the table we observe that the values of $\theta_n$ tend rapidly to 2, so that $\theta_n = 2 > 1$, for $n \geq 64$. Consequently, for $n = 512$, we have $C_{512} \doteq 0.542727465$, $| C_{512} - C_{256} | /C_{512} \doteq 0.1714 \times 10^{-2}$, $\hat{C}_{512} \doteq 0.541797213$, $| \hat{C}_{512} - \hat{C}_{256} | /\hat{C}_{512} \doteq 0.4922 \times 10^{-12} < 0.5 \times 10^{-8}$, $\hat{e}_{512} \doteq -0.930252 \times 10^{-3}$, and $\hat{C}_{512} - C_{512} \doteq -0.930252 \times 10^{-3}$. The latter numerical values imply that $C(2, 10) \simeq \hat{C}_{512}$ in at least 8 SDD, that is $C(2, 10) \doteq 0.54179721$ rounded in 8 SDD, as well as that $\hat{e}_{512}$ and $\hat{C}_{512} - C_{512}$ are accurate estimates of $e_{512} = C(2, 10) - C_{512}$.

(B2) The results of the iterations of scheme (32)-(37), $z_n, \hat{z}_n$ are given in the lower part of Table 2 along with the values of $\phi_n$, and the estimate $\hat{e}_n$ of $e_n$. The values of $\phi_n$ are converging to -0.1114 such that $0 <| \phi_n |< 1$, $n \geq 2$. Accordingly, with $n = 6$, we have $z_6 \doteq 0.686914558$, $| z_6 - z_5 | /z_6 \doteq 0.153 \times 10^{-5}$, $\hat{z}_6 \doteq 0.686914663$, $| \hat{z}_6 - \hat{z}_5 | /\hat{z}_6 \doteq 0.2166 \times 10^{-9} < 0.5 \times 10^{-8}$, $\hat{e}_6 \doteq -0.105 \times 10^{-6}$ and $\hat{z}_6 - z_6 = 0.105 \times 10^{-6}$. The last three numerical values imply that $\hat{z}_6 \simeq \rho_0$, with $\rho_0$ the (unknown) actual root of (25), in at least 8 SDD, that is $\rho_0 \doteq 0.689614663$, as well as that $\hat{e}_6$ and $\hat{z}_6 - z_6$ are accurate estimates of $\rho_0 - z_6$. Consequently, $C(2, 10) \simeq -\log_2 \hat{z}_6 \doteq 0.54179721$ rounded in 8 SDD.

Therefore, (B1) and (B2) imply that $C(2, 10) \doteq 0.54179721$ is accurate in at least 8 SDD. Both numerical methods 1 and 2 are converging to the latter value using almost the same number of iterations. Accordingly, we claim that similar results can be found, via the introduced numerical methods 1 and 2, for any unknown capacity $C(d, k)$. $\diamondsuit$

## 3.3. Construction of a near-optimal $(d, k)$-code

In this section, we employ, for the first time, the method of continued fraction approximation (CFA) in order to construct fixed-rate near-optimal $(d, k)$-codes for given values of $d$ and $k$. These codes have a fixed rate $R$ close to a degree of accuracy to the maximum information rate $C$. Hereafter we denote, as usually, by $\mathcal{R}$ the set of real numbers and by $Q = \{i/j; i, j \in Z, j \neq 0\}$ the set of rational numbers, where $Z = \{0, \pm 1, \pm 2, \ldots\}$ is the set of integer numbers. Consequently, we denote by $\bar{Q}$ the set of irrational numbers.

It is known [5] that the capacity $C(d,k)$, $0 \leq d < k \leq \infty$, is an irrational number, i.e. $C = C(d,k) \in \bar{Q}$ for all parameters $d = 0$, $0 < k < \infty$ and all $1 \leq d < k \leq \infty$, except for the case $d = 0$ and $k = \infty$, for which $C(0,\infty) = 1$. Accordingly, the code's rate $R$ can never attain 100% of the capacity $C$, for $(d,k) \neq (0,\infty)$, but only a fraction of it. The problem of determining, at least theoretically and feasible also in practice, integers $m$ and $n$ so that the ratio $R = m/n$ approximates well from below the capacity $C$ is closely connected to a diophantine approximation of a real number $x$ by a sequence of rational numbers $c_r$, $r \geq 0$. For details on CFA see e.g. [11].

### 3.3.1. Convergents of CFA

One of the common methods used to get a diophantine approximation to an $x \in \mathcal{R}$ is that via a continued fraction expression of the form $\alpha_0 + \frac{1}{\alpha_1 + \frac{1}{\alpha_2 + \frac{1}{\ddots}}}$, usually abbreviated as $[\alpha_0; \alpha_1, \alpha_2, \ldots]$, which is finite if $x \in Q$ and infinite if $x \in \bar{Q}$, . The terms $\alpha_i$, $i \geq 0$, are called the partial quotients, and if $\alpha_i \in Z$, $i \geq 0$, and $\alpha_i \geq 1$, $i \geq 1$, the continued fraction is said to be simple.

Consider an $x \in \mathcal{R}$. To calculate a finite if $x \in Q$ or an infinite if $x \in \bar{Q}$ continued fraction of $x$ set $x_0 = x$ and for $i = 0, 1, 2, \ldots$, define recursively the sequence $\alpha_0, \alpha_1, \alpha_2, \ldots$, as follows:

$$\text{Set}\, \alpha_i = \lfloor x_i \rfloor,\, \text{and if}\, f_i = x_i - \alpha_i \neq 0\, \text{then set}\, x_{i+1} = 1/f_i\, \text{and repeat; else stop.} \tag{38}$$

In this way, $[\alpha_0; \alpha_1, \alpha_2, \ldots]$ and $[\alpha_0; \alpha_1, \alpha_2, \ldots, \alpha_r]$, with $r$ finite is a continued fraction of $x$, for $x \in \bar{Q}$ and $x \in Q$, respectively. Moreover, defining recursively the integers $m_0, m_1, \ldots,$ and $n_0, n_1, \ldots,$ by

$$m_0 = \alpha_0, n_0 = 1; m_1 = \alpha_0\alpha_1 + 1, n_1 = \alpha_1;$$

$$m_r = \alpha_r m_{r-1} + m_{r-2}, n_r = \alpha_r n_{r-1} + n_{r-2},\, \text{for}\, r \geq 2, \tag{39}$$

the fraction

$$c_r = m_r/n_r = [\alpha_0; \alpha_1, \alpha_2, \ldots, \alpha_r], \tag{40}$$

with $c_r \in Q$, is termed the $r$th convergent of the continued fraction. By its construction, $c_r$ is finite if $x \in Q$ and $\lim_{r\to\infty} c_r = x$ if $x \in \bar{Q}$. In addition, if $r$ is an even positive integer, it holds

$$c_1 > c_3 > \cdots > c_{r-1},\, c_0 < c_2 < \cdots < c_r. \tag{41}$$

The convergence $c_r = m_r/n_r$, $r \geq 1$, is the best rational approximation to an $x \in \bar{Q}$, in the sense that $c_r$ is closer to $x$ than any other rational number with denominator less than $n_r$ (see e.g. Theorem 10.15 in [11]). Moreover, any sufficiently close approximation to an $x \in \bar{Q}$ must be a convergent $c_r$ of an infinite simple continued fraction $[\alpha_0; \alpha_1, \alpha_2, \ldots]$ of $x$.

### 3.3.2. Error criterion of CFA

For $x \in \bar{Q}$, with $x = [\alpha_0; \alpha_1, \alpha_2, \ldots]$ and $c_r = m_r/n_r$, it holds

$$e_r =\mid x - c_r \mid < b_r = 1/n_r^2,\, r \geq 0. \tag{42}$$

Consequently, because of (42), we can compute a convergent $c_r$ within a desired accuracy, say $\varepsilon = 0.5 \times 10^{-\nu}$, $\nu \geq 1$. To that end, for $x \neq 0$ a good enough approximation $c_r$ to $x$, being true in at least $\nu$ significant decimal digits, can be obtained by calculating a convergent $c_r$, $r \geq 0$, until an $r = \tilde{r}$ is determined, via (38)-(40), such that the relation

$$b'_{\tilde{r}} = b_{\tilde{r}} / \mid x \mid \leq \varepsilon, \tag{43}$$

is satisfied. Then the relative error $e'_{\tilde{r}}$ is

$$e'_{\tilde{r}} = e_{\tilde{r}} / \mid x \mid < \varepsilon, \tag{44}$$

since by (42), we have $e'_{\tilde{r}} < b'_{\tilde{r}}$. Hence,

$$c_{\tilde{r}} = m_{\tilde{r}} / n_{\tilde{r}} = [\alpha_0; \alpha_1, \alpha_2, \ldots, \alpha_{\tilde{r}}], \tag{45}$$

is the desired CFA to $x$, true within $\varepsilon$.

### 3.3.3. CFA of a $(d, k)$-code capacity

In the sequel we will apply the CFA technique for determining rational approximation to the capacity of a given $(d, k)$-code. This approximation, as a convergent, provides best rational code rate $R$ for a $(d, k)$-code with given $d$ and $k$ parameters. To that end, let

$$\tilde{C}_r = \tilde{C}_r(d, k) = m_r / n_r, \; r \geq 0, \tag{46}$$

be the $r$th convergent of $C = C(d, k)$, $0 \leq d < k$. As we have already mentioned, $C \in \bar{Q}$ for $(d, k) \neq (0, \infty)$. From (41), we get that legitimate rational approximations to $C$ are only the even numbered convergents $\tilde{C}_r$, $r = 0, 2, \ldots$, since $\tilde{C}_r < C$ or $\eta_r = \tilde{C}_r / C < 1$, for $r \geq 0$. These $\tilde{C}_r$s are suitable candidates for code rates of the form $R = m_r / n_r$, $r \geq 0$.

In order to give an illustrating example clarifying further CFA in obtaining rational code rates, we next present Example 5. Similar results can also be obtained for any $(d, k)$-code.

*Example 5.* Let $(d, k) = (2, 10)$. Then, by part (B) of Example 4, $C = C(d, k) = 0.54179721$. Assume that $\varepsilon = 0.5 \times 10^{-6}$. Then by (38)-(40), (42)-(45) we compute $\tilde{r} = 8$ and $\tilde{C}_{\tilde{r}} = [0; 1, 1, 5, 2, 12, 1, 5, 3]$. Table 3 depicts integers $m_r$ and $n_r$ for $r = 2, 4, 6, 8$. Their ratios $\tilde{C}_r = m_r / n_r$ provide best rational approximate code rates $R_r = m_r / n_r$ to code's capacity $C$. The entries of the table also show the respective ratios $R_r = \tilde{C}_r$ along with the relative code's efficiency $\eta'_r = C'_r$, where $C'_r = (C - \tilde{C}_r)/C$ is the relative error between $C$ and the convergent $\tilde{C}_r$ with relative error bound $b'_r$.

Table 3. $d = 2$, $k = 10$, $C = C(d, k) = 0.54179721$. The integers $m_r$ and $n_r$ are such that $R = R_r = m_r / n_r$ are the best rational approximate code rates for the code's capacity $C$.

| $r$ | $m_r$ | $n_r$ | $R_r$ | $\eta'_r$ | $b'_r$ |
|---|---|---|---|---|---|
| 2 | 1 | 2 | 0.500000 | $0.7715 \times 10^{-1}$ | $0.4614 \times 10^{0}$ |
| 4 | 13 | 24 | 0.541667 | $0.2409 \times 10^{-3}$ | $0.3204 \times 10^{-2}$ |
| 6 | 175 | 323 | 0.541796 | $0.2850 \times 10^{-5}$ | $0.1769 \times 10^{-4}$ |
| 8 | 3286 | 6065 | 0.541797 | $0.2393 \times 10^{-7}$ | $0.5018 \times 10^{-7}$ |

By the entries of Table 3 we infer that: (a) There are no block codes with code words of length less than the depicted $n_r$ for which their relative efficiency $\eta_r^{'} = C_r^{'}$ is less than the showed $C_r^{'}$ in the table. (b) For any two successive $r_1$ and $r_2$, $r_1, r_2 \in \{2, 4, 6, 8\}$, with $r_2 > r_1$ it holds $C_{r_1}^{'} > C_{r_2}^{'}$. The latter implies that all legitimate ($R < C$) code rates $R = m/n$, with $m_{r_1} \leq m \leq m_{r_2}$, $n_{r_1} \leq n \leq n_{r_2}$, satisfy the relation $C_{r_2}^{'} \leq \eta^{'} \leq C_{r_1}^{'}$, where $\eta^{'} = 1 - R/C$. Accordingly, (a) and (b) suggest bounds for the efficiency which may be obtained in approximating $C(d, k)$ by rational rates of the form $R = m/n$. In other words, (a) and (b) set limits on every possible pair $(m, n)$ that could be used as a potential candidate for obtaining a feasible block code with rate $R$.

As an illustration, let $r_1 = 4$ and $r_2 = 6$. Then $\eta_{r_1}^{'} = 0.2409 \times 10^{-3}$ and $\eta_{r_2}^{'} = 0.2850 \times 10^{-5}$, respectively. According to (a) and (b), $\eta_{r_1}^{'}$ and $\eta_{r_2}^{'}$ imply that:

First, there are no code words of length less than or equal to 24 (for $r = 4$) and 323 (for $r = 6$), so that the code's relative efficiency $\eta^{'}$ is less than or equal to $0.2409 \times 10^{-3}$ and $0.2850 \times 10^{-5}$, respectively.

Second, for integer pairs $(m, n)$, with $13 \leq m \leq 175$, $24 \leq n \leq 323$ and $(m/n) < C$, the inequality $0.2850 \times 10^{-5} \leq \eta^{'} \leq 0.2409 \times 10^{-3}$, is satisfied. These pairs $(m, n) \in \Gamma_1 \cup \Gamma_2$, with $\Gamma_1 = \{(i, j) : i = 13\ell, j = 24\ell, \ell = 1, 2, \ldots, 13\}$ and $\Gamma_2 = \{175, 323\}$. Clearly, for all $(m, n) \in \Gamma_1$ it holds $\eta^{'} = 0.2409 \times 10^{-3}$ since these pairs are multiples of $(13, 24)$.

Third, in case we want we use block codes with smaller code words, but having in mind that they would be less efficient, we can search for codes with relative efficiency between and including $\eta_2^{'}$ and $\eta_4^{'}$. These codes have $(m, n) \in \cup_{i=1}^{8} \Delta_i$, with $\Delta_1 = \{(i, j) : i = \ell, j = 2\ell, \ell = 1, 2, \ldots, 12\}$, $\Delta_2 = \{(12, 23)\}$, $\Delta_3 = \{(11, 21)\}$, $\Delta_4 = \{(10, 19)\}$, $\Delta_5 = \{(9, 17)\}$, $\Delta_6 = \{(8, 15)\}$, $\Delta_7 = \{(7, 13)\}$ and $\Delta_8 = \{(13, 24)\}$, with increasing $\eta = (m/n)/C$ or decreasing $\eta^{'} = 1 - \eta$. More specifically, these codes have $\eta_{\Delta_1}^{'} = 0.7715 \times 10^{-1}$, $\eta_{\Delta_2}^{'} = 0.3702 \times 10^{-1}$, $\eta_{\Delta_3}^{'} = 0.3320 \times 10^{-1}$, $\eta_{\Delta_4}^{'} = 0.2857 \times 10^{-1}$, $\eta_{\Delta_5}^{'} = 0.2286 \times 10^{-1}$, $\eta_{\Delta_6}^{'} = 0.1562 \times 10^{-1}$, $\eta_{\Delta_7}^{'} = 0.6157 \times 10^{-2}$, and $\eta_{\Delta_8}^{'} = 0.2409 \times 10^{-3}$, respectively. The same relations, hold also for the multiples of $(m, n)$, i.e. $(\ell m, \ell n)$, $\ell = 1, 2, \ldots$. Among the pairs $(m, n) \in \cup_{i=1}^{8} \Delta_i$, we observe that the next more efficient pairs after $(13, 24)$ are, in order of decreasing efficiency, the pairs $(7, 13)$, $(8, 15)$, $(9, 17)$, $(10, 19)$, $(11, 21)$, $(12, 23)$, $(1, 2)$ and of course their multiples. Their efficiency $\eta = (m/n)/C$, including $(13, 24)$, in decreasing order are: 99.98%, 99.38%, 98.44%, 97.71%, 97.14%, 96.68%, 96.30% and 92.29%, respectively. Consequently, all the mentioned block codes, with $R = m/n$, have efficiency $\eta = R/C > 90\%$, suggesting that they are noticeably good ones.

For instance, we select the near-optimal $(2, 10)$-codes $[7, 13; 2, 10]$, $[13, 24; 2, 10]$, having $R = 0.538462$, $R = 0.541667$ and $\eta = 99.38\%$, $\eta = 99.98\%$, respectively. Therefore, both $7{-}to{-}13$ and $13 - to - 24$ codes are very efficient with $\eta > 99\%$. Furthermore, in coding these $m$-bit, with $m \in \{7, 13\}$, source words to their $n$-bit, with $n \in \{13, 24\}$, channel representation, we can consider, for example, the first $2^m = 128, 8192$ for $m = 7, 13$, numbered in decimal representation by the integers $0, 2^0, \ldots, 2^m - 1$, from the $N_n(2, 10) = 183, 11421$ for $n = 13, 24$, $(2, 10)$-constrained $n$-bit channel words, and vice versa. Some of the rest $N_n(2, 10) - 2^m = 55, 3229$ for $n = 13, 24$, $(2, 10)$-sequences, might be used as special patterns for checking or error detection as they obey the specified $(2, 10)$ constraints.

More specifically, Table 4 depicts the source $(2^{m-1}, 2^{m-2}, \ldots, 2^0)$ and $(d, k) = (2, 10)$-code

$(N_{n-1}(d,k), N_{n-2}(d,k), \ldots, N_0(d,k))$ weighting systems for a fixed-rate block-code having source words of length $m$ bits and $(d,k)$-code words of length $n$ bits, with $m = 7, 13$ and $n = 13, 24$, respectively. Table 5 shows source words $(x)_{10}$, $(x)_2$, $(2,10)$-code words $(y)_2$, $(y)_{10}$ and converting factors $(y-x)_{10}$ for a code with source words of length $m = 7, 13$ and $(2,10)$-code words of length $n = 13, 24$ bits, respectively. For converting the presented in Table 5 source/code words the encoder/decoder employs eqs. (5)-(6). For instance, using a $[7,13;2,10]$ code the number $(x)_{10} = 55$ written as source word $(x)_2 = 0110111$, in the number system $(64,32,16,8,4,2,1)$, is converted to $(2,10)$-code word $(y)_2 = 0001001001000$, in the number system $(126,87,60,41,28,19,13,9,6,4,3,2,1)$, and vice versa, such that $(y)_{10} = 58$ and $(y-x)_{10} = 3$. $\diamondsuit$

Table 4. Weighting systems for block-codes with $m$ source bits and $n$ $(2,10)$-code bits.

| $m$ | $n$ | | weighting system |
|---|---|---|---|
| 7 | 13 | source | (64, 32, 16, 8, 4, 2, 1) |
| | | code | (126, 87, 60, 41, 28, 19, 13, 9, 6, 4, 3, 2, 1) |
| 13 | 24 | source | (4096, 2048, 1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1) |
| | | code | (7845, 5389, 3702, 2543, 1747, 1200, 824, 566, 389, 267, 183, 126, 87, 60, 41, 28, 19, 13, 9, 6, 4, 3, 2, 1) |

Table 5. Source $(x)_{10}$, $(2,10)$-code words $(y)_{10}$ and converting factors $(y-x)_{10}$, along with $(x)_2$ and $(y)_2$ words of length $m$ and $n$ bits, respectively.

| $m$ | $n$ | $(x)_{10}$ | $(x)_2$ | $(y)_2$ | $(y)_{10}$ | $(y-x)_{10}$ |
|---|---|---|---|---|---|---|
| 7 | 13 | 0 | 0000000 | 0000000000100 | 3 | 3 |
| | | 55 | 0110111 | 0001001001000 | 58 | |
| | | 64 | 1000000 | 0010000010001 | 67 | |
| | | 127 | 1111111 | 1000000001000 | 130 | |
| 13 | 24 | 0 | 0000000000000 | 000000000010000000000100 | 186 | 186 |
| | | 4091 | 0111111111011 | 001000010000000000100000 | 4277 | |
| | | 4096 | 1000000000000 | 001000010000000001000001 | 4282 | |
| | | 8191 | 1111111111111 | 100000001001000001001000 | 8377 | |

## 4. CONCLUSIONS

For an important class, in communication and data storage applications, of run-length-limited sequences termed $(d,k)$-constrained sequences: (a) We have obtained simple combinatorial expressions (Eqs. (16)-(23)) enumerating sequences of finite length. (b) We have provided efficient numerical procedures (Eqs. (28)-(37)) for determining the capacity of $(d,k)$-constrained codes. (c) We have proposed a simple method (Eqs. (43)-(46)) for constructing near-optimal $(d,k)$-codes, i.e. $(d,k)$-codes with rational rates almost equal to their irrational capacity.

## APPENDIX

*Proof of Equations* (21)-(22). We will determine the numbers $N_n(d,k)$ and $N_n^{(1)}(d,k)$ by approaching the problem via a model of distribution (allocation) of indistinguishable balls into distinguishable urns under certain conditions.

Let $s$ be the number of 1s and $n-s$ the number of 0s in the 0-1 sequence of length $n$. For a fixed $s \geq 1$, we consider that the $s$ 1s form $s+1$ for a sequence in $S_n(d,k)$ and $s$ for a sequence in $S_n(1;d,k)$ distinguishable urns, which are numbered as 1st, 2nd, 3rd, etc. Of the urns formed, $s-1$ are internal, i.e. each internal urn is formed between two successive 1s. First, $d$ 0s (considered as indistinguishable balls) are placed in each of the $s-1$ internal urns. Next, the remaining $n-s-(s-1)d$ 0s are distributed in (all) the urns, $s-1$ of which have limited capacity $k-d$ and the remaining urns (two and one, for a sequence in $S_n(d,k)$ and $S_n(1;d,k)$, respectively) have limited capacity $k$. According to (14) this number of distributions equals to $H_{s-1,2}(n-s-(s-1)d, s+1, k-d, k)$ for a sequence in $S_n(d,k)$ and $H_{s-1,1}(n-s-(s-1)d, s, k-d, k)$ for a sequence in $S_n(1;d,k)$.

Summing with respect to $s$ and taking into account that one more sequence is included in $S_n(d,k)$ when $s=0$ (there are no 1s in the sequence) and $n=k$, we get the results. $\diamondsuit$

*Proof of Equation* (23). Using similar reasoning as in the proof of Eqs (21)-(22) we consider again that $s \geq 1$ 1s in the sequence form $s+1-i-j$ urns, $s-1$ of which are internal, formed among the 1s. Then after placing $d$ 0s in each of the $s-1$ internal urns and one 0 in each of the $2-i-j$ external urns, the remaining $n-s-(s-1)d-2+i+j$ 0s are distributed in the internal urns each of limited capacity $k-d$ and in the external urns each of limited capacity $k-1$. Then we continue as in the proof of Eqs (21)-(22). Summing with respect to $s$ and taking into account that one more sequence is included in $S_n(i,j;d,k)$ when $s=0$ (regarding the case $i=j=0$) and $n=k$, we get the result. $\diamondsuit$

## ACKNOWLEDGEMENTS

The author wishes to thank the anonymous reviewers for the thorough reading and useful comments which helped to improve the article.

## REFERENCES

[1] K. E. Atkinson, An Introduction to Numerical Analysis, Second Edition, Wiley, Singapore, 1989.

[2] P.A. Franaszek, Sequence-state methods for run-length-limited coding, IBM J. Res. Develop. 14 (1970) 376-383.

[3] K.A.S. Immink, Codes for Mass Data Storage Systems, Second Edition Shannon Foundation Publishers, Eindhoven, 2004.

[4] P. Jacquet, W. Szpankowski, On $(d,k)$ sequences not containing a given word, IEEE International Symposium on Information Theory (2006) 1486-1489.

[5] N. Kashyap, P.H. Siegel, Equalities among capacities of $(d,k)$-constrained systems, SIAM J. Discrete Math. 17 (2003) 276-297.

[6] W.H. Kautz, Fibonacci codes for synchronization control, IEEE Trans. Infor. Theory, 11 (1965) 284-292.

[7] O.F. Kurmaev, Coding of run-length-limited sequences, Problems of Information Transmission, 37 (2001) 216-225.

[8] F.S Makri, Z.M. Psillakis, On $\ell$-overlapping runs of ones of length $k$ in sequences of independent binary random variables, Commun. Statist. Theor. Meth. 44 (2015) 3865-3884.

[9] F.S. Makri, Z.M. Psillakis, On the exact distributions of pattern statistics for a sequence of binary trials: A combinatorial approach, in: J. Glaz, M.V. Koutras (Eds.) Handbook of Scan Statistics, Springer Science+Business Media, LLC, 2019, pp. 1-20, https://doi.org/10.1007/978-1-4614-8414-1_48-1.

[10] B.H. Marcus, R.M. Roth, P.H. Siegel, An Introduction to Coding for Constrained Systems (draft version downloadable, Fith Edition, 2001). Revised and expanded version of "Constrained Systems and Coding for Recording Channels" by B.H. Marcus, R.M. Roth, P.H. Siegel, Chapter 20 in Handbook of Codind Theory, V. Pless and W.C. Huffman (Eds), Elsevier Press, Amsterdam, 1998, pp. 1635-1764.

[11] K.H. Rosen, Elementary Number Theory and Its Applications, Third Edition, Addison-Wesley Publ. Comp. Reading, 1993.

[12] C.E. Shannon, A mathematical theory of communication, Bell. Syst. Tech. J. 27 (1948) 379-423, 623-656.

[13] M.N. Simic, New RLL code for high density optical recording, TELSIKS (2011) 160-163.

[14] D.T. Tang, L.R. Bahl, Block codes for a class of constrained noiseless channels, Inf. Control 17 (1970) 436-461.

[15] E. Zehavi, J.K. Wolf, On runlength codes, IEEE Trans. Infor. Theory 34 (1988) 45-54.