Bit-flipping Decoder Failure Rate Estimation for (v,w)-regular Codes

– Extended Version –

Alessandro Annechini and Alessandro Barenghi and Gerardo Pelosi *Member, IEEE*
Department of Electronics, Information and Bioengineering - DEIB
Politecnico di Milano, Milano, Italy
Email: alessandro.annechini@mail.polimi.it, alessandro.barenghi@polimi.it, gerardo.pelosi@polimi.it

***Abstract*—Providing closed form estimates of the decoding failure rate of iterative decoder for low- and moderate-density parity check codes has attracted significant interest in the research community over the years. This interest has raised recently due to the use of iterative decoders in post-quantum cryptosystems, where the desired decoding failure rates are impossible to estimate via Monte Carlo simulations. In this work, we propose a new technique to provide accurate estimates of the DFR of a two-iterations (parallel) bit flipping decoder, which is also employable for cryptographic purposes. In doing so, we successfully tackle the estimation of the bit flipping probabilities at the second decoder iteration, and provide a fitting estimate for the syndrome weight distribution at the first iteration. We numerically validate our results, providing comparisons of the modeled and simulated weight of the syndrome, incorrectly-guessed error bit distribution at the end of the first iteration, and two-iteration Decoding Failure Rates (DFR), both in the floor and waterfall regime for simulatable codes. Finally, we apply our method to estimate the DFR of LEDAcrypt parameters, showing improvements by factors larger than 2^{70} (for NIST category 1) with respect to the previous estimation techniques. This allows for a $\approx 20\%$ shortening in public key and ciphertext sizes, at no security loss, making the smallest ciphertext for NIST category 1 only 6% larger than the one of BIKE. We note that the analyzed two-iterations decoder is applicable in BIKE, where swapping it with the current black-gray decoder (and adjusting the parameters) would provide strong IND-CCA2 guarantees.**

I. Introduction

Low Density Parity Check (LDPC) codes, proposed in [1] are random binary codes characterized by a sparse parity check matrix. They have proven a remarkable engineering tool thanks to their efficient decoding algorithm and good correction power. These desirable features come at the cost of a significant difficulty in providing closed-form bounds for their Decoding Failure Rate (DFR). While the increased availability of computing power allows to estimate the said DFR via Monte Carlo simulations (i.e., sampling random error vectors with a given weight, trying to decode them and counting the number of failures), reliably estimating very low failure rates (e.g., 10^{-9} and below) still requires considerable time. One context where closed form estimates for the DFR of binary codes, belonging to the category of either Low or Moderate Dentity Parity Check (LDPC or MDPC) codes, are remarkably important is the design of post-quantum cryptosystems. The difference between LDPC and MDPC codes is the Hamming weight of a parity check matrix row, which is $\mathcal{O}(\log(n))$ for LDPC codes, and $\mathcal{O}(\sqrt{n\log(n)})$ for MDPC codes, where n is the code length. Indeed, post-quantum cryptosystems such as BIKE [2], a current fourth-round candidate in NIST's standardization process, and LEDACrypt [3] employ, as the private key, a quasi-cyclic LDPC/MDPC code. In both cases, whenever a decoding failure takes place during the decryption of a ciphertext, information regarding the private key is leaked to an (active) attacker [4]. To attain security against active attackers (formally captured by the IND-CCA2 definition) both cryptosystems require the DFR of the employed codes to be below 2^{-128}, for a decoder of choice. To this end, BIKE relies on an extrapolation of the behaviour of the iterative decoder [5] at lower values of DFR, while LEDACrypt employs a two-iteration bit flipping decoder for which it combines a first-iteration model [6], [7] with a conservative, code-specific, upper bound for the error correction capability of the second iteration [8], [9].

Cryptographic-grade low DFR values cannot be estimated via numerical simulation, and have therefore led to a further significant amount of research in providing closed form bounds for the DFR of a decoder, explicitly encouraged also by NIST [10]. J. Chaulet [11] provided an estimate of the distribution of the syndrome weights (before the first iteration of the iterative decoder acts on it) with a good fit on the average value, and a model for the probability of parity check equations to be unsatisfied at the first decoder iteration. The observations on the syndrome weight involve the fact that error vectors with remarkably low syndrome weight tend to be decoded with lower probability. In an affine line of work, the authors of [12], [13] observe that errors vectors having regularities such as runs of ones, or ones, placed at regular intervals are less frequently decoded by QC-MDPC iterative decoders. The authors of [14] and [15] observed that, while the aforementioned error vectors are indeed harder to decode, they appear to represent a relatively low quantity of the overall non-decodable errors. Going in a different direction, J-P. Tillich [8] provided a code-specific technique to determine the maximum weight of an error which is guaranteed to be corrected by an iterative decoder operating on a QC-LDPC/QC-MDPC code, and proved that the DFR falls exponentially quickly when the code length is increased, while keeping the weight of the parity check rows $\mathcal{O}(\sqrt{n\log(n)})$.

Contribution. In this work, we describe how to accurately estimate the DFR of a two-iterations (parallel) bit flipping decoder for (v, w)-regular codes, such as the LDPC/MDPC one used in [3]. In doing so, we provide a closed form estimate of the syndrome weight distribution, improving on the one by [11]. Furthermore, we propose a technique to assess the bit values in the error vector estimate processed by bit-flipping decoders that do not match the actual error vector values after the first iteration, improving the accuracy with respect to [6]. Finally, we propose a technique, relying on the previous two results, to estimate the mismatches between the error estimate and the actual error elaborated by the decoder after the second iteration, deriving an estimate for the two-iteration DFR, which provides a remarkably good fit in the waterfall region. We validate our results with extensive numerical simulations. [1]

Our results can either be employed alone to obtain a reliable estimate of the DFR of a randomly picked (v, w)-regular LDPC/MDPC code, or, combined with the conservative DFR bound for a single iteration from [9] to obtain a conservative, cryptographically useful, DFR estimate to design cryptosystems such as LEDACrypt [3] and the fourth-round NIST competition candidate BIKE [2]. In particular, we note that our technique allows to improve the DFR bounds currently employed in LEDACrypt [3] by a factor greater than 2^{70}, allowing a significant reduction in the required keysize and ciphertext size at no security loss.

II. Preliminaries

A binary linear code C of block length n is a k-dimensional vector subspace of $\mathbb{F}_2^n$, where $\mathbb{F}_2 = \{0, 1\}$ is the field with two elements, described via a generator matrix $G \in \mathbb{F}^{k\times n}$ such that $C = \{mG \mid m \in \mathbb{F}_2^k\}$, and is also referred to as a code $C[n, k]$. The rate of C, denoted $R(C)$, equals k/n, where k is the dimension of C (as a vector space over $\mathbb{F}_2$). Being a linear subspace of dimension k, the code C can be described as the kernel of a matrix $H \in \mathbb{F}^{(n-k)\times n}$ so that $C = \{c \in \mathbb{F}_2^n \mid H\, c^T = 0\}$, and $GH^T = 0 \in \mathbb{F}^{k\times(n-k)}$. The matrix H is called the parity-check matrix of the code C and, in general, any choice of H whose rows form a basis of the dual space $C_\perp = \{x \in \mathbb{F}_2^n \mid x\, c^T = 0\ \forall\ c \in C\}$ describes the same code. The parity-check matrix has also an interpretation as an adjacency matrix of a bipartite graph, a.k.a. the *factor graph* of the code, defined as follows. On the left side there are n vertices, called *variable nodes*, one for each codeword position. On the right side there are $r{=}n{-}k$ vertices, called *factor nodes* or *check nodes*, one for each parity-check (i.e., row of the parity-check matrix). The value (1 or 0) of each column in a row indicates whether the corresponding variable node is connected to the check node (value 1) or not (value 0). If all nodes on the same side have the same degree, the graph is categorized as *left-regular*, *right-regular*, or *biregular* if every vertex has the same degree.

Algorithm 1: Bit Flipping Algorithm

```
   Input: $\widetilde{c}$: $1 \times n$ error-affected codeword,
          $s$: $r \times 1$ syndrome,
          $H$: $r \times n$ parity-check matrix,
          iterMax: max number of of permitted iterations.
   Output: $\bar{e} = [\bar{e}_0, \dots, \bar{e}_{n-1}]$: bits of the error vector estimate,
           decodeOk: value indicating success, 1, or fail, 0.
 1 iter $\leftarrow 0$
 2 $\bar{e} \leftarrow 0_{1\times n-1}$ // bit vector
 3 while ($s \neq 0_{r\times 1}$ and iter $<$ iterMax) do
 4     for $j$ from $0$ to $n-1$ do
 5         upc$_j \leftarrow \langle s, h_{:,j}\rangle$ // each $s_i{\cdot}h_{i,j}$ used as an integer
 6     th $\leftarrow$ ThresholdChoice(iter, $s$)
 7     for $j$ from $0$ to $n-1$ do
 8         if (upc[$j$] $\geq$ th) then
 9             $\bar{e}_j \leftarrow \bar{e}_j \oplus 1$
10             $s \leftarrow s \oplus h_{:,j}$
11     iter $\leftarrow$ iter $+ 1$
12 if ($s = 0_{r\times 1}$) then
13     decodeOk $\leftarrow 1$
14 else
15     decodeOk $\leftarrow 0$
16 return $\bar{e}$, decodeOk
```

This work focuses on codes that admit a parity-check matrix all of whose rows and columns have at most a fixed constant number of set bits, referred to as (v, w)-regular codes, where w and v denote the number of 1's lying in any row and any column of the parity-check matrix, respectively, or equivalently with a factor graph exhibiting all check nodes with degree w and all variable nodes with degree v. Low Density Parity Check (LDPC) codes, originally studied by Gallager [1], are (v, w)-regular codes that admit a sparse parity-check matrix and sparse factor graph, and hence amenable to linear time decoding algorithms, exhibiting column and row weights in the range of $O(\log(n))$. The major significance of these codes is due to their ability to provide high reliable communications at code rates that are extremely close to channel capacity. Increasing moderately the number of non null elements in each row of the parity-check matrix up to costants in the range of $O(\sqrt{n\log(n)})$, the codes are also known as Moderate Density Parity Check Codes (MDPC) [16], [17].

In the following section we are going to focus on modeling the statistical error correction properties of the iterative (parallel) bit flipping decoding algorithm proposed by Gallager's [1], when applied to a generic (v, w)-regular binary code. As shown in Algorithm 1, the said decoding process takes as input the parity-check matrix $H = [h_{i,j}]$, $i{\in}\{0, \dots, r{-}1\}$, $j{\in}\{0, \dots, n{-}1\}$ of a code, and the value of a syndrome s of an error affected codeword, $\widetilde{c} = c + e$, where e is an unknown error vector with length n and Hamming weight equal to weight t, while c is a legit codeword: $s{=}H(c + e)^T = He^T$. After each iteration, the algorithm updates the value of the syndrome to be used for the next iteration to match the equality $s = H(\bar{e} \oplus e)^T$, and terminates as soon as $s{=}0$, indicating that $\bar{e}{=}e$, or after a predetermined maximum number of iterations yielding a decoding failure.

The initial value of $\bar{e}$ is the null vector $0_{1\times n}$ (line 2). Each

[1] code available at https://crypto.deib.polimi.it/DFR_codebase.zip

algorithm iteration is split up in three phases. In the first phase (lines 4-5), it computes the inner product between the bit vector s and the bits in each column of H, considering them as integers, obtaining a quantity known as the "unsatisfied parity-check [equation count]" (upc) bound to the j-th bit position in the error vector and stores such a value in a variable $\texttt{upc}_j$, $j \in \{0,\dots,n\}$. In the second phase, a threshold $\texttt{th}{\in}\left\{\lceil\frac{v+1}{2}\rceil,\dots,v\right\}$ is either looked up from a set of predetermined values $\{\texttt{th}_1\texttt{th}_2,\dots\}$, each of which associated to a specific iteration, or computed as a function of the current value of the syndrome and of the current iteration count. In the third phase (lines 7-10), the algorithm evaluates for each $j{\in}\{0,\dots,n{-}1\}$, if $\texttt{upc}_j$ is greater than the threshold $\texttt{th}$, and in the affermative case it flips the value in $\bar{e}_j$ (i.e., $\bar{e}_j \leftarrow \bar{e}_j{\oplus}1$) and updates the syndrome by adding to it the j-th column of H (i.e., $s \leftarrow s \oplus H_{:,j}$).

III. Decoding failure rate model

In the following, we describe our technique to model the syndrome weight distribution in III-A, our method to derive the probability distribution of flips made by the first iteration in III-B, and by the second iteration in III-C, concluding with our method to compute the estimate of the DFR of the bit-flipping decoder (see Algorithm 1) after two-iterations in III-D. In developing our techniques, we make the following assumption: *the rows of H are independently and uniformly random drawn from the set of binary vectors having length n and w asserted bits*, acknowledging that this is an approximation, as the weight of the parity-check matrix columns will be kept constant to v. We share this assumption in the analysis with the original paper by Gallager [1].

A. Modeling Syndrome Weight Distribution

Let us denote as $(e,s)_l$, $l{\in}\{0,1,2\dots,t\}$, a pair of values representing an error vector e and its corresponding syndrome $s = He$ both indexed by the weight of the error vector l. We consider a given syndrome and its corresponding unknown error vector with weight t as the last pair in the sequence $(e,s)_0,(e,s)_1,(e,s)_2,\dots,(e,s)_t$, where $(e,s)_0$ contains a null error vector and its null syndrome, while $(e,s)_l$, $l \geq 1$, denotes a pair with an error vector that includes the same set bits of the error vector in $(e,s)_{l-1}$ plus an additional single set bit that is uniformly randomly placed in one out of the $n{-}(l{-}1)$ available positions. Analogously, the syndrome value in $(e,s)_l$, $l \geq 1$, is assumed to differ from the one in $(e,s)_{l-1}$ due to the bitwise addition of the column of the parity-check matrix H selected by the said additional set bit.

We model the Hamming weight of each syndrome in the previous sequence as an instance of a discrete random variable $\mathcal{W}_l$ bound to a probability mass function $\Pr(\mathcal{W}_l = y)$, with $l{\in}\{0,\dots,t\}$, $y{\in}\{0,\dots,r\}$, which is in turn represented as an array $\texttt{wp}_{(l)} = [\texttt{wp}_{(l),0},\dots,\texttt{wp}_{(l),x},\dots,\texttt{wp}_{(l),r}]$.
Starting from the distribution of the weight of the syndrome of a null error vector, $\texttt{wp}_{(0)} = [1,0,\dots,0]$, the random variable $\mathcal{W}_t$ associated to the weight of the syndrome at hand coincides with final state of discrete-time non-omogeneous Markov chain with $r{+}1$ states. Such a Markov chain is uniquely defined by $\texttt{wp}_{(0)}$ and the transition matrices $\texttt{P}_{(l)} = [p_{x,y,l}]_{x,y\in\{0,\dots,r\}}$. Specifically, the distribution of each random variable $\mathcal{W}_l$ is derived through the following vector-matrix multiplication $\texttt{wp}_{(l)} = \texttt{wp}_{(l-1)} \cdot \texttt{P}_{(l)}$, with $l{\in}\{1,\dots,t\}$, where each transition probability $p_{x,y,l} = \Pr(\mathcal{W}_l = y|\mathcal{W}_{l-1} = x)$ is a function of the starting and ending weight of the syndrome as well as of the step l considered along the chain.

In the following we model the number of flips induced on any syndrome bit, subsequently we derive the probabilities of flipping up a clear bit and flipping down a set bit of a syndrome along the aforementioned chain, and finally compute the statistical distribution of the weight of a syndrome and the transition probabilities $p_{x,y,l}$. We denote as $\mathcal{F}_l{\in}\{0,\dots,\min(w,l)\}$ the discrete random variable modeling the count of flips of any single bit of the syndrome of an error vector with weight l, during the computation of the syndrome itself. The probability mass function of $\mathcal{F}_l$ follows an hypergeometric distribution $\phi_l(f,l){=}\Pr(\mathcal{F}_l = f) = \frac{\binom{w}{f}\binom{n-w}{l-f}}{\binom{n}{l}}$.
Indeed, the l set positions in the error vector select l positions in any single row of H, which in turn corresponds to a syndrome bit. Whenever one of such selected row positions contains one of the w set bits out of n, the syndrome bit corresponding to the row at hand is flipped. Following any syndrome bit along the sequence $(e,s)_0,\dots,(e,s)_{l-1},(e,s)_l,\dots$, we note that $\Pr(\mathcal{F}_l{=}f{+}1\,|\,\mathcal{F}_{l-1}{=}f) = \frac{w-f}{n-l}$, while the event $\mathcal{F}_{l-1} = f$ implies that the syndrome bit is either clear or set, depending on f being either even or odd, respectively.

As a consequence the probability of flipping at step l any syndrome bit that was cleared at step $l{-}1$, depends on the value of the step, $\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to1}}(l)$, and can be derived as $\frac{\sum_f \Pr(\mathcal{F}_l=f+1|\mathcal{F}_{l-1}=f)\Pr(\mathcal{F}_{l-1}=f)}{\sum_f \Pr(\mathcal{F}_{l-1}=f)}$, where the variable f ranges over the even values in $\{0,\dots,\min(w,l)\}$; analogously, the probability of flipping at step l any syndrome bit that was set at step $l{-}1$, also depends from the value of the step, $\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to0}}(l)$, and can be derived by applying the same formula with $f{\in}\{0,\dots,\min(w,l)\}$ ranging over odd values:

$$\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to1}}(l){=}\left(\overset{\min(l,w)}{\underset{0,\texttt{even}}{\textstyle\sum_f}}\left(\frac{w-f}{n-l}\cdot\phi_l(f,l)\right)\right)\Big/\left(\overset{\min(l,w)}{\underset{0,\texttt{even}}{\textstyle\sum_f}}\phi_l(f,l)\right)$$

$$\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to0}}(l){=}\left(\overset{\min(l,w)}{\underset{0,\texttt{odd}}{\textstyle\sum_f}}\left(\frac{w-f}{n-l}\cdot\phi_l(f,l)\right)\right)\Big/\left(\overset{\min(l,w)}{\underset{1,\texttt{odd}}{\textstyle\sum_f}}\phi_l(f,l)\right)$$

We now analyze the change of the syndrome weight from step $l{-}1$ to step l, to derive the probability mass function $p_{x,y,l}{=}\Pr(\mathcal{W}_l = y|\mathcal{W}_{l-1} = x)$, with $x,y \in \{0,\dots,r\}$. Since the column weight of the parity-check matrix is v, the overall amount of flips among all r bits of the syndrome is v. As a consequence, the weight y of the syndrome at step l is obtained from flipping up a clear bits out of $r{-}x$, and flipping down $v{-}a$ set bits out of x, for all admissible values of a, i.e., $a{\in}\{\max\{0,v-x\},\dots,\min\{r-x,v\}\}$. The weight of the syndrome after the l-th step is complete is thus $y = x + a - (v-a)$, or equivalently it holds that

$r-y=r-x-a+(v-a)$, from which we derive $a=\frac{y-x+v}{2}$. Given $\mathcal{W}_{l-1}=x$ and a specific value of a, define two events: $\mathtt{E}_{1,a}$: a bits are flipped up in $r-x$ flip trials; $\mathtt{E}_{2,a}$: $v-a$ bits are flipped down in x flip trials. The probability mass function $\Pr(\mathtt{E}_{1,a})$=$\varphi(x,a,l)$ describes the probability of flipping up a bits of the syndrome at step $l-1$, while $\Pr(\mathtt{E}_{2,a})=\psi(x,a,l)$ describes the probability of flipping down $v-a$ bits of the at the step $l-1$. Note that, depending on the weight of the syndrome at step $l-1$, there are $\binom{r-x}{a}$ possibile patterns for the said flip-ups and $\binom{x}{v-a}$ possibile patterns for the said flips down. We thus obtain $\varphi(x,a,l)$ and $\psi(x,a,l)$ as follows:

$$\varphi(x,a,l)=\binom{r-x}{a}\left(\pi_{\substack{l-1\rightarrow l\\ \mathtt{flip}\,0\rightarrow 1}}(l)\right)^{a}\left(1-\pi_{\substack{l-1\rightarrow l\\ \mathtt{flip}\,0\rightarrow 1}}(l)\right)^{(r-x)-a}$$

$$\psi(x,a,l)=\binom{x}{v-a}\left(\pi_{\substack{l-1\rightarrow l\\ \mathtt{flip}\,1\rightarrow 0}}(l)\right)^{v-a}\left(1-\pi_{\substack{l-1\rightarrow l\\ \mathtt{flip}\,1\rightarrow 0}}(l)\right)^{x-(v-a)}$$

Given a specific value for a and $\mathcal{W}_{l-1}=x$, the event modelling the v flips in the transition of the syndrome weight from x to y=x+$2a$−v, from step $l-1$ to step l, is $\mathtt{E}_{1,a}\cap\mathtt{E}_{2,a}$. The event is bound to the probability mass function $\Pr(\mathtt{E}_{1,a})\cdot\Pr(\mathtt{E}_{2,a})=\varphi(x,a,l)\cdot\psi(x,a,l)$ since they are independent (indeed, they take place on two disjoint set of syndrome bit positions).

Considering all the admissible values for a, the probability mass function $\Pr(\mathcal{W}_{l-1}=x)=\omega(x,l)$, models the probability of moving to any admissible syndrome weight y at step l, is computed as:

$$\omega(x,l)=\sum\nolimits_{i}^{\min\{r-x,v\}}_{\max\{0,v-x\}}\left(\varphi(x,i,l)\cdot\psi(x,i,l)\right).$$

Thus, the transition probability $p_{x,y,l}$=$\Pr(\mathcal{W}_l=y|\mathcal{W}_{l-1}=x)$ can be written as a function of the step count l, and of both the starting and ending weights of the syndrome.

$$p_{x,y,l}=\begin{cases}1, & l=1,x=0,y=v\\ \rho(x,y,l), & \substack{l\geq 2\\ \max(0,x-v)\leq y\leq\min(x+v,r)\\ y\equiv_2(x+v)}\\ 0, & \text{otherwise}\end{cases}$$

with

$$\rho(x,y,l)=\frac{\varphi(x,\frac{y-x+v}{2},l)\cdot\psi(x,\frac{y-x+v}{2},l)}{\omega(x,l)},$$

where, in the special case $l=1,x=0,y=v$, $\pi_{\substack{l-1\rightarrow l\\ \mathtt{flip}\,1\rightarrow 0}}(l)$ is not defined. The value $p_{0,v,1}=1$ is quantitatively justified since, during the first step, a null syndrome (therefore with $x=0$) will deterministically turn into a weight v syndrome when a column of H is added.

The pseudo-code of the procedure to derive the statistical distribution of the weight of the syndrome of an error vector with weight t, $\Pr(\mathcal{W}_t=y)$, yielding the corresponding array of probabilities $\mathtt{wp}_{(t)}=[\mathtt{wp}_{(t),0},\ldots,\mathtt{wp}_{(t),r}]$ is shown by Algorithm 2.

```
Algorithm 2: SYNDROME WEIGHT DISTRIBUTION
   Input: $(v,w)$-regular code parameters.
          $v$: column weight; $w$: row weight;
          $t$: error vector weight, $t\geq 1$;
          $r$: num. of rows of the parity-check matrix;
          $n$: num. of columns of the parity-check matrix
   Output: $\mathtt{wp}_{(t)}=[\mathtt{wp}_{(t),0},\ldots,\mathtt{wp}_{(t),r}]$
 1 $\mathtt{wp}\leftarrow[0,\ldots,0]$
 2 $\mathtt{wp}[v]\leftarrow 1$ // initialized as $\mathtt{wp}_{(1)}$
 3 for $l$ from $2$ to $t$ do
 4     $\mathtt{wp\_prev}\leftarrow\mathtt{wp}$
 5     for $y$ from $0$ to $r$ do
 6         $\mathtt{wp}[y]\leftarrow 0$
 7         for $i$ from $\max(0,y-v)$ to $\min(y+v,r)$ do
 8             $\mathtt{p}\leftarrow$ COMPUTEPMATRIXELEMENT$(i,y,l)$
 9             $\mathtt{wp}[y]\leftarrow\mathtt{wp}[y]+\mathtt{wp\_prev}[i]\cdot\mathtt{p}$
10 return $\mathtt{wp}$
```

The procedure start by initializing the vector containing the discrete distribution $\mathtt{wp}$ to the one of a weight t=1 error vector (lines 1–2). Subsequently, the computation iterates $t-1$ times the procedure computing the non-homogeneous Markov chain (lines 3–9). The chain is computed with two nested loops, iterating on the possible weights (loop at 5–9) and the non-null values of the transition matrix columns $\mathtt{P}_{(l)}$ (loop at lines 7–9). The values of the matrix elements are computed by the COMPUTEPMATRIXELEMENT, which can be computed with time complexity $\mathcal{O}(n\cdot\log^3(n))$. While a straightforward implementation would have a higher computational cost, applying the *memoization* technique improves significantly the efficiency. Memoization is applicable whenever the computation of a function depends only on its input parameters (i.e., computing it twice on the same inputs yields the same result). Memoization keeps a lookup table of the function results, indexed by their corresponding input values. When implemented in this fashion, Algorithm 2 has an overall time complexity of $\mathcal{O}\left(t\cdot r\cdot v\cdot n\cdot\log^3(n)\right)=\mathcal{O}\left(t\cdot v\cdot n^2\cdot\log^3(n)\right)$.

B. First Iteration of a Bit Flipping Decoder

Given a (v,w)-regular binary code with a $r\times n$ parity-check matrix $H=[h_{i,j}]$ where $i\in\{0,\ldots,r-1\}$, $j\in\{0,\ldots,n-1\}$, and the value of a syndrome s=He^T derived from an unknown error vector e with weight t, in the following a parity-check equation is defined as $\sum_{j=0}^{n-1}h_{i,j}e_j=s_i$, where e_j are the unknowns, $h_{i,j}$ are the known coefficients and s_i the constant known term. The equation is said to be *satisfied* if s_i=0, *unsatisfied* if s_i=1. We are going to consider the (parallel) bit flipping decoding algorithm introduced in the previous section which iteratively estimates the most likely value $\bar{e}$ of the error vector e, given s and H, startimng from the initial value $\bar{e}=0_{1\times n}$.

Our goal is to estimate the statistical distribution of the random variable $\mathcal{E}_{(\mathtt{iter})}$ modeling the Hamming weight of $\bar{e}\oplus e$ after the $\mathtt{iter}$-th iteration of the decoding algorithm, i.e., the count of differences between the actual error vector e and the current estimated one $\bar{e}$. The probability mass function $\Pr(\mathcal{E}_{(\mathtt{iter})}=\mathrm{d})$, $\mathrm{d}\in\{0,\ldots,n\}$, will be considered only with $\mathtt{iter}>0$, because $\Pr(\mathcal{E}_{(0)}=t)$=1, before the beginning of the decoding algorithm, when $\bar{e}$=0. The probability mass function $\Pr(\mathcal{E}_{(\mathtt{iter})} = \mathrm{d})$ will be obtained, using the result in the previous section about the distribution of the weitgh of the syndrome of an error vector of weight t, (i.e., $\mathcal{W}_t = y$), as follows:

$$\Pr(\mathcal{E}_{(\mathtt{iter})} = \mathrm{d}) = \sum_{y=0}^{r} \left(\Pr\left(\mathcal{E}_{(\mathtt{iter})} = \mathrm{d} \mid \mathcal{W}_t = y\right) \Pr(\mathcal{W}_t = y)\right).$$

From now on, the goal of the analysis is going to be the estimate of the probability $\Pr\left(\mathcal{E}_{(\mathtt{iter})} = \mathrm{d} \mid \mathcal{W}_t = y\right)$, $y{=}\mathtt{wt}(s)$: $\mathtt{iter}{=}1$ in this subsection, and $\mathtt{iter}{=}2$ in the next one. Furthermore, for the sake of brevity, in the definition of any event and in the formulas of probability mass functions we are going to omit any reference to the weight of the syndrome.

In the analysis of the first iteration of the decoding algorithm, we denote as $\mathcal{S}_i$, $i{\in}\{0,\ldots,r\}$, the random variables modeling the value taken by the i-th bit of the syndrome, s_i, at the beginning of each iteration of the decoding algorithm. Therefore, for each unsatisfied and satisfied parity-check equation, the probability to observe a clear or set constant term is: $\Pr(\mathcal{S}_i = 0) = \frac{r-y}{r}$, $\Pr(\mathcal{S}_i = 1) = \frac{y}{r}$, respectively.

We denote as $\mathtt{E}_{(i,j),0}$ or $\mathtt{E}_{(i,j),1}$ the event of an error bit being either clear or set, respectively, in a position j captured by one of the w set coefficients of the row $h_{i,:}$ in the i-th parity-check equation, i.e.: $\mathtt{E}_{(i,j),0}{=}\{h_{i,j} = 1 \,\text{and}\, e_j = 0\}$, and $\mathtt{E}_{(i,j),1}{=}\{h_{i,j} = 1 \,\text{and}\, e_j = 1\}$, respectively.

Observe that, using $\mathcal{F}_t$ from the previous section as the random variable modeling the count of bitflips determining s_i, for any i, in the computation $s = He^T$, the probabilities $\Pr(\mathcal{S}_i = 0)$ and $\Pr(\mathcal{S}_i = 1)$ can be written also as:

$$\Pr(\mathcal{S}_i = 0) = \Pr\left(\bigcup_{f=0,\mathtt{even}}^{\min(t,w)} (\mathcal{F}_t = f)\right) = \sum_{f=0,\mathtt{even}}^{\min(t,w)} \Pr(\mathcal{F}_t = f),$$

$$\Pr(\mathcal{S}_i = 1) = \Pr\left(\bigcup_{f=1,\mathtt{odd}}^{\min(t,w)} (\mathcal{F}_t = f)\right) = \sum_{f=1,\mathtt{odd}}^{\min(t,w)} \Pr(\mathcal{F}_t = f),$$

where the last step is mutuated by the fact that the events $(\mathcal{F}_t = f)$ are disjoint. Note that $\mathcal{F}_t = f$ implies that there are f variables of the i-th parity-check equation which are both set to 1 and have a corresponding coefficient set to 1 (out of the w set ones present in the equation). Here, $\Pr(\mathcal{F}_t = f)$ is a shorthand for $\Pr(\mathcal{F}_t = f \,|\, \mathcal{W}_t = y)$, and the formula for the probability mass function of $\Pr(\mathcal{F}_t = f)$ reported in the previous section cannot be applied anymore. For the complete derivation of $\Pr(\mathcal{F}_t = f \,|\, \mathcal{W}_t = y)$, the reader is referred to Appendix A.

Consider now, $p_{\mathtt{unsat}|0} = \Pr\left(\mathcal{S}_i = 1 \mid \mathtt{E}_{(i,j),0}\right)$, that is the probability that, given an error variable $e_j{=}0$, the constant term s_i of the i-th parity-check equation, where the variable is involved with a coefficient $h_{i,j}{=}1$, is $s_i{=}1$ (the equation is unsatisfied). Since the rows of the parity-check matrix are assumed to be independent and with the same weight, we have that $p_{\mathtt{unsat}|0}$ is independent from the index of the parity-check equation i:

$$p_{\mathtt{unsat}|0} = \frac{\Pr\left(\mathtt{E}_{(i,j),0} \,|\, \mathcal{S}_i{=}1\right) \Pr(\mathcal{S}_i{=}1)}{\Pr\left(\mathtt{E}_{(i,j),0} \,|\, \mathcal{S}_i{=}1\right) \Pr(\mathcal{S}_i{=}1) + \Pr\left(\mathtt{E}_{(i,j),0} \,|\, \mathcal{S}_i{=}0\right) \Pr(\mathcal{S}_i{=}0)}$$

The factors $\Pr\left(\mathtt{E}_{(i,j),0} \mid \mathcal{S}_i = 0\right)$, $\Pr\left(\mathtt{E}_{(i,j),0} \mid \mathcal{S}_i = 1\right)$ are:

$$\Pr\left(\mathtt{E}_{(i,j),0} \mid \mathcal{S}_i = 0\right) = \\ = \frac{\Pr\left(\mathtt{E}_{(i,j),0} \cap \mathcal{S}_i{=}0\right)}{\Pr(\mathcal{S}_i{=}0)} = \frac{\sum_{f=0,\mathtt{even}}^{\min(t,w)} \Pr\left(\mathtt{E}_{(i,j),0} \,|\, \mathcal{F}_t{=}f\right) \Pr(\mathcal{F}_t{=}f)}{\sum_{f=0,\mathtt{even}}^{\min(t,w)} \Pr(\mathcal{F}_t{=}f)}$$

$$\Pr\left(\mathtt{E}_{(i,j),0} \mid \mathcal{S}_i = 1\right) = \\ = \frac{\Pr\left(\mathtt{E}_{(i,j),0} \cap \mathcal{S}_i{=}1\right)}{\Pr(\mathcal{S}_i{=}1)} = \frac{\sum_{f=1,\mathtt{odd}}^{\min(t,w)} \Pr\left(\mathtt{E}_{(i,j),0} \,|\, \mathcal{F}_t{=}f\right) \Pr(\mathcal{F}_t{=}f)}{\sum_{f=1,\mathtt{odd}}^{\min(t,w)} \Pr(\mathcal{F}_t{=}f)}.$$

Note that $\Pr\left(\mathtt{E}_{(i,j),0} \mid \mathcal{F}_t = f\right){=}\frac{w-f}{w}$, since knowing $\mathcal{F}_t = f$ for the i-th parity-check equation implies that there are $w{-}f$ variables e_j, among the w ones with their coefficient $h_{i,j} = 1$, which are equal to zero.

Consider now $p_{\mathtt{unsat}|1} = \Pr\left(\mathcal{S}_i = 1 \mid \mathtt{E}_{(i,j),1}\right)$, i.e., the probability that, given an error variable $e_j = 1$, the constant term s_i of the i-th parity-check equation, where the variable is involved with a coefficient $h_{i,j}{=}1$, is $s_i{=}1$ (the equation is unsatisfied). The expression for $p_{\mathtt{unsat}|1}$ can be obtained with a derivation analogous to the one for $p_{\mathtt{unsat}|0}$, with the only difference that $\Pr\left(\mathtt{E}_{(i,j),1} \mid \mathcal{F}_t = f\right) = \frac{f}{w}$, since there are f variables e_j equal to 1 among the w ones involved in the i-th parity-check equation.

We are now equipped to describe the distribution of the random variables $\mathcal{U}_j$ modelling the values of the $\mathtt{upc}_j$ variables computed at each decoder iteration. We are interested in modelling in particular $\Pr\left(\mathcal{U}_j = u \mid e_j = 0\right)$ and $\Pr\left(\mathcal{U}_j = u \mid e_j = 1\right)$. We recall that $\mathtt{upc}_j = \langle s, h_{:,j}\rangle$, where $h_{:,j}$ has only v elements set to 1; as a consequence, the range of $\mathcal{U}_j$ is $u \in \{0,\ldots,v\}$. Therefore the value $\mathtt{upc}_j$ equals the sum of v constant terms of the parity-check equations indexed by the support (i.e., the set of positions of non null elements) of the j-th column of H, $\mathrm{Supp}(h_{:,j})$. Without loss of generality, partition the support into two subsets, $\mathrm{Supp}(h_{:,j})_0$ and $\mathrm{Supp}(h_{:,j})_1$, where $\mathrm{Supp}(h_{:,j})_0$ contains all the indices of the satisfied parity-check equations, i.e., for all $i \in \mathrm{Supp}(h_{:,j})_0$, we have $s_i{=}0$, while $\mathrm{Supp}(h_{:,j})_1$ contains all the indices of the unsatisfied parity-check equations, i.e., for all $i \in \mathrm{Supp}(h_{:,j})_1$, we have $s_i{=}1$. From this we have that $|\mathrm{Supp}(h_{:,j})_1| = u = \mathtt{upc}_j$, while $|\mathrm{Supp}(h_{:,j})_0| = v{-}u$. Given u, there are $\binom{v}{u}$ possible ways of partitioning $\mathrm{Supp}(h_{:,j})$. Consider the following event:

$$\left(\bigcap_{i\in\mathrm{Supp}(h_{:,j})_1} (\mathcal{S}_i = 1 | \mathtt{E}_{(i,j),0})\right) \cap \left(\bigcap_{i\in\mathrm{Supp}(h_{:,j})_0} (\mathcal{S}_i = 0 | \mathtt{E}_{(i,j),0})\right),$$

modeling one of the possible combinations of syndrome values leading to $\mathtt{upc}_j = u$, under the assumption that $e_j = 0$, Noting that the components of the intersection are all independent, under the hypothesis of the independence of the rows of H, we obtain that the probability of the event is $p_{\mathtt{unsat}|0}^{u}(1 - p_{\mathtt{unsat}|0})^{v-u}$. As a consequence,

$$\Pr\left(\mathcal{U}_j = u \mid e_j = 0\right) = \binom{v}{u}\, p_{\mathtt{unsat}|0}^{u}\, (1 - p_{\mathtt{unsat}|0})^{v-u}.$$

Following the same reasoning, we obtain

$$\Pr\left(\mathcal{U}_j = u \mid e_j = 1\right) = \binom{v}{u}\, p_{\mathtt{unsat}|1}^{u}\, (1 - p_{\mathtt{unsat}|1})^{v-u}.$$

Note that, in an out-of-order decoder, the decision to flip a given $\bar{e}_j$ is taken independently from the value of the values of $\texttt{upc}_{j'}$ for $j' \neq j$, as the $\texttt{upc}$ values do not change until all the flips are completed. We now compute $p_{\texttt{flip}|0} = \Pr(\texttt{upc}_j \geq \texttt{th}|e_j = 0)$, that is the probability that the decoder flips the j-th error estimate value $\bar{e}_j$, when $e_j = 0$, and the corresponding alternatives. $p_{\texttt{flip}|0}$ corresponds to the union of all events where the $\texttt{upc}_j$ value is equal or greater than the threshold $\texttt{th}$, we therefore have

$$p_{\texttt{flip}|0} = \sum_{a=\texttt{th}}^{v} \Pr\left(\mathcal{U}_j = a \mid e_j = 0\right).$$

We denote with

$$p_{\neg\texttt{flip}|0} = 1 - p_{\texttt{flip}|0} = \sum_{a=0}^{\texttt{th}-1} \Pr\left(\mathcal{U}_j = a \mid e_j = 0\right)$$

decides not to flip the said error estimate bit.

Through an analogous reasoning, we obtain $p_{\texttt{flip}|1}$ (the probability that the decoder flips an error estimate bit, assuming that the corresponding error vector bit is set) as

$$p_{\texttt{flip}|1} = \sum_{a=\texttt{th}}^{v} \Pr\left(\mathcal{U}_j = a \mid e_j = 1\right),$$

while we have

$$p_{\neg\texttt{flip}|1} = 1 - p_{\texttt{flip}|1} = \sum_{a=0}^{\texttt{th}-1} \Pr\left(\mathcal{U}_j = a \mid e_j = 1\right).$$

Denote with $\bar{e}_{(i)}$ the value of $\bar{e}$ after the bit flips of the i-th iteration have been applied. Given the previous probabilities, we consider the event $\texttt{E}_{(\mathrm{d}_+)} = |(\{0,\dots,n-1\} \setminus \mathrm{Supp}(e)) \cap \mathrm{Supp}(\bar{e}_{(1)})| = \mathrm{d}_+$, that is d_+ flips of $\bar{e}$ happen on the $n-t$ positions where $e_j = 0$ (effectively increasing the amount of discrepancies), and the event $\texttt{E}_{(\mathrm{d}_-)} = |\mathrm{Supp}(e) \cap \mathrm{Supp}(\bar{e}_{(1)})| = \mathrm{d}_-$, that is d_- flips happen on the t positions where $e_j = 1$ has a set bit (effectively decreasing the number of discrepancies). We have that, through a counting argument

$$\delta_+(\mathrm{d}_+) = \Pr\left(\texttt{E}_{(\mathrm{d}_+)}\right) = \binom{n-t}{\mathrm{d}_+}(p_{\texttt{flip}|0})^{\mathrm{d}_+}(1-p_{\texttt{flip}|0})^{n-t-\mathrm{d}_+}$$

and analogously

$$\delta_-(\mathrm{d}_-) = \Pr\left(\texttt{E}_{(\mathrm{d}_-)}\right) = \binom{t}{\mathrm{d}_-}(p_{\texttt{flip}|1})^{\mathrm{d}_-}(1-p_{\texttt{flip}|1})^{t-\mathrm{d}_-}.$$

Having obtained closed formulas for $\delta_+(\mathrm{d}_+)$ and $\delta_-(\mathrm{d}_-)$, we observe that the events $\texttt{E}_{(\mathrm{d}_+)}$ and $\texttt{E}_{(\mathrm{d}_-)}$ act on disjoint subsets of the bits of e and are thus independent. We are thus able to obtain $\Pr\left(\mathcal{E}_{(1)} = \mathrm{d} \mid \mathcal{W}_t = y\right)$ considering the set $\mathbf{D}$ of pairs $(\mathrm{d}_+, \mathrm{d}_-)$ such that $\mathrm{d} = t - \mathrm{d}_- + \mathrm{d}_+$:

$$\Pr\left(\mathcal{E}_{(1)} = \mathrm{d} \mid \mathcal{W}_t = y\right) =$$

$$\Pr\left(\bigcup_{(\mathrm{d}_+,\mathrm{d}_-)\in\mathbf{D}} (\texttt{E}_{(\mathrm{d}_+)} \cap \texttt{E}_{(\mathrm{d}_-)})\right) = \sum_{(\mathrm{d}_+,\mathrm{d}_-)\in\mathbf{D}} \delta_+(\mathrm{d}_+)\cdot\delta_-(\mathrm{d}_-)$$

Note that the conditioning on the value of $\mathcal{W}_t$ is embedded in the fact that $(\mathcal{W}_t = y)$ is employed to derive $\Pr(\mathcal{S}_i = 0)$ and $\Pr(\mathcal{S}_i = 1)$, which are needed to obtain $p_{\texttt{flip}|0}$ and $p_{\texttt{flip}|1}$.

C. Second Iteration of the Bit Flipping Decoder

We now tackle the modeling of the second iteration of the out-of-order bit flipping decoder, with the end of describing $\Pr(\mathcal{E}_{(2)} = \mathrm{d})$. To this end, we will derive $\Pr(\mathcal{E}_{(2)} = \mathrm{d}|\mathcal{W}_t = y)$, and then obtain $\Pr(\mathcal{E}_{(2)} = \mathrm{d})$ combining the results thanks to the knowledge of the distribution of $\mathcal{W}_t$.

To this end, we partition the bits of $\bar{e}_{(1)}$ into four classes. Each class is labeled with a pair (a, b), where, for each bit $\bar{e}_j$ belonging to the class, we have $a = e_j, b = e_j \oplus \bar{e}_{(1),j}$. From now on, we denote as $\mathbf{J}_{a,b}$ with $a, b \in \{0, 1\}$ the sets of positions of the bits in the class (a, b). We are going to estimate the probabilities of flips being applied to the bits in each of the four (a, b) classes considering their values obtained after the first iteration $\bar{e}_{(1)}$, which we will denote as $p_{\texttt{flip}|00}$, $p_{\texttt{flip}|01}$, $p_{\texttt{flip}|10}$, and $p_{\texttt{flip}|11}$, and we will derive them as a function of the cardinalities of the sets, $|\mathbf{J}_{0,1}| = \epsilon_{01}$ and $|\mathbf{J}_{1,1}| = \epsilon_{11}$.

Note that the analysis of the previous section allows us to compute the probability distributions $\Pr(|\mathbf{J}_{a,b}| = m)$ for all $a, b \in \{0, 1\}, m \in \{0, \dots, n\}$; these, combined with the probabilites of flipping each element in a position indexed by $\mathbf{J}_{a,b}$ allow us to derive the number of discrepancies left after the second iteration $\texttt{wt}(e \oplus e_{(2)})$.

Indeed, we compute the probability that ϵ_{01} bits of the error estimate have been flipped by the first iteration, despite them being clear in the actual error vector, $\Pr(\epsilon_{01} = \mathrm{d}_+)$ as $\delta_+(\mathrm{d}_+)$; similarly we compute the probability that ϵ_{11} bits have not been flipped by the first iteration, despite them being set in the actual error vector $\Pr(\epsilon_{11} = t - \mathrm{d}_-) = \delta_-(\mathrm{d}_-)$. Note that $\Pr(\epsilon_{00} = n - t - \mathrm{d}_+) = \delta_+(\mathrm{d}_+) = \Pr(\epsilon_{01} = \mathrm{d}_+)$ as the number of non-flipped error estimate bits sharing the position with the $n-t$ error vector elements containing a 0 is $n - t - \mathrm{d}_+$. Analogously, we have that $\Pr(\epsilon_{10} = \mathrm{d}_-) = \delta_-(\mathrm{d}_-) = \Pr(\epsilon_{11} = t - \mathrm{d}_-)$.

For the sake of clarity, $\textsc{Bin}(\texttt{trials}, \texttt{succpr}, \texttt{nsucc})$ will indicate the probability mass function of obtaining $\texttt{nsucc}$ success events out of $\texttt{trials}$ independent events with a success probability of $\texttt{succpr}$ each.

We now describe the probability of flipping up an error estimate bit, given that it appears in a satisfied parity-check equation and the corresponding error bit value is 0, that is: $\Pr\left(\texttt{upc}_j \geq \texttt{th} \mid e_j = 0, h_{i,j} = 1, s_i = 0\right)$. To compute this probability, we employ $p_{\texttt{unsat}|0}$ from the previous analysis, i.e., the probability that a parity-check equation is unsatisfied, given that one of the involved error vector bits is 0. The probability

$$p_{\texttt{flip}|0,\texttt{OneEqSat}} = \Pr\left(\texttt{upc}_j \geq \texttt{th} \mid e_j = 0, h_{i,j} = 1, s_i = 0\right)$$

$$p_{\texttt{flip}|0,\texttt{OneEqSat}} = \sum_{a=th}^{v-1} \textsc{Bin}(v-1, p_{\texttt{unsat}|0}, a),$$

i.e., the probability of the union of the events where $a \geq \texttt{th}$ parity-check equations are unsatisfied, knowing a priori that one out of the v parity-check equations is satisfied, hence reducing the number of trials to $v - 1$.

We compute the probability

$$p_{\mathtt{flip}|0,\mathtt{OneEqUnsat}} = \Pr\left(\mathtt{upc}_j \geq \mathtt{th} \mid e_j = 0, h_{i,j} = 1, s_i = 1\right)$$

$$p_{\mathtt{flip}|0,\mathtt{OneEqUnsat}} = \sum_{a=th-1}^{v-1} \textsc{Bin}(v-1, p_{\mathrm{unsat}|0}, a),$$

following the same line of reasoning of $p_{\mathtt{flip}|0,\mathtt{OneEqSat}}$, while taking care of the fact that the parity-check equation with the a-priori known value is now unsatisfied, thus decreasing the minimum number of required successes to $\mathtt{th}-1$, besides decreasing the number of trials to $v-1$ as before.
We move onto computing

$$p_{\neg\mathtt{flip}|1,\mathtt{OneEqSat}} = \Pr\left(\mathtt{upc}_j < \mathtt{th} \mid e_j = 1, h_{i,j} = 1, s_i = 0\right),$$

that is, the probability of not flipping a value in the error estimate $\bar{e}_j$, given that the corresponding value of the error vector is $e_j = 1$, and the parity-check equation in which it appears is satisfied ($s_i = 0$). We employ $p_{\mathrm{unsat}|1}$ to obtain $p_{\neg\mathtt{flip}|1,\mathtt{OneEqSat}} = \sum_{a=0}^{\mathtt{th}-1} \textsc{Bin}(v-1, p_{\mathrm{unsat}|1}, a)$ as the union of the events where $a < \mathtt{th}$ parity-check equations are unsatisfied, out of the $v-1$ where the error bit e_j is involved, as the outcome of one of the parity-check equations is known a-priori to be satisfied. Finally, through an analogous line of reasoning, we obtain

$$p_{\neg\mathtt{flip}|1,\mathtt{OneEqUnsat}} = \Pr\left(\mathtt{upc}_j < \mathtt{th} \mid e_j = 1, h_{i,j} = 1, s_i = 1\right)$$

$$p_{\neg\mathtt{flip}|1,\mathtt{OneEqUnsat}} = \sum_{a=0}^{\mathtt{th}-2} \textsc{Bin}(v-1, p_{\mathrm{unsat}|1}, a).$$

1) Modeling the $\bar{e}$ bits indexed by $\mathbf{J}_{0,0}$: To derive the expression of $p_{\mathtt{flip}|00}$, we now model the effect of the bit-flips made by the first iteration on the satisfaction value of the parity-check equations where a bit $\bar{e}_a, a \in \mathbf{J}_{0,0}$ is involved. To this end, consider a parity-check equation where an error bit $e_a, a \in \mathbf{J}_{0,0}$ is involved together with $\mathtt{tc}$ asserted bits of the error e, and define the following probability:

$$\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01}) =$$

$$\Pr\left(|\mathrm{Supp}(h_{i,:}) \cap \mathbf{J}_{0,1}| \text{ is odd } \mid \exists a \in \mathbf{J}_{0,0}, h_{i,a} = 1, \langle e, h_{i,:}\rangle = \mathtt{tc}\right),$$

that is, the probability that such a parity-check equation involves an odd number of terms (i.e., bits in $e \oplus \bar{e}_{(1)}$), which are in positions where the first iteration performed a flip-up of $\bar{e}$, and the corresponding error vector bit is clear. The choice to consider an odd number of terms is mutuated by the fact that, if an additional odd number of set terms are involved in the parity-check equation, its satisfaction value will toggle. By contrast, having an additional even number of set terms would not toggle the satisfaction value. The expression of $\chi_{\uparrow\mathtt{odd}}$ depends on the parity of $\mathtt{tc}$. Considering the case where $\mathtt{tc}$ is even, we compose the expression of $\chi_{\uparrow\mathtt{odd}}$ starting from two binomial distributions, $\eta(\mathtt{tc}, l)$ and $\zeta(\mathtt{tc}, l, \epsilon_{01})$. The distribution $\eta(\mathtt{tc}, l) = \textsc{Bin}(w - \mathtt{tc} - 1, p_{\mathtt{flip}|0,\mathtt{OneEqSat}}, l)$ describes the probability that, out of the $w - \mathtt{tc} - 1$ variables involved in the parity-check equation, l have positions in $\mathbf{J}_{0,1}$, i.e. are discrepancies resulting from the flip-ups made by the first iteration, considering that the probability that each such flip-up takes place in the first iteration is $p_{\mathtt{flip}|0,\mathtt{OneEqSat}}$. The possible positions for the flip-ups to take place are $w - \mathtt{tc} - 1$, since, out of the w positions involved in the equation, $\mathtt{tc}$ have a corresponding error position containing a 1, and one belongs to $\mathbf{J}_{0,0}$, which implies that the bit was not flipped. The distribution $\zeta(\mathtt{tc}, l, \epsilon_{01}) = \textsc{Bin}(n - w - (t - \mathtt{tc}), p_{\mathtt{flip}|0}, \epsilon_{01} - l)$ models the probability that the remaining $\epsilon_{01} - l$ flip-ups causing a discrepancy, which took place in the first iteration, took place on $\epsilon_{01} - l$ positions out of the $n - w - (t - \mathtt{tc})$ ones which are not involved in the parity-check equation (hence $n - w$), and not containing a set bit (hence, we subtract the set bits not involved in the equation $t - \mathtt{tc}$. Combining the $\eta(\mathtt{tc}, l)$ and $\zeta(\mathtt{tc}, l, \epsilon_{01})$ we obtain $\xi(\mathtt{tc}, \epsilon_{01}) =$

$$\sum_{l'=\max(0,\epsilon_{01}-(n-w-(t-\mathtt{tc})))}^{\min(\epsilon_{01},w-\mathtt{tc}-1)} \eta(\mathtt{tc}, l') \cdot \zeta(\mathtt{tc}, l', \epsilon_{01})$$

which represents the probability distribution of the union of the events where the one modeled by $\eta(\mathtt{tc}, l')$ and the one modeled by $\zeta(\mathtt{tc}, l', \epsilon_{01})$ take place simultaneously, for a given l'. The value l' is upper bounded by the smallest value between the amount of flip-up affected positions ϵ_{01} and the places where they should happen $w - \mathtt{tc} - 1$, while being lower bounded by $\epsilon_{01} - (n - w - (t - \mathtt{tc}))$, that is, the amount of flip-up affected positions which cannot be fit among the $n - w$ not involved in the parity check equation, and further excluding the $t - \mathtt{tc}$ ones where the error vector bits not involved in the parity check equation are set. Thus, informally, $\xi(\mathtt{tc}, \epsilon_{01})$ represent the probability that any admissible number of flip-ups l' of the first iterations are on positions involved in the parity check equation containing $\mathtt{tc}$ positions where the corresponding error bit is set, plus a position where the corresponding error bit is clear, and the corresponding error estimate bit was not flipped.

We obtain $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01})$ for the case where $\mathtt{tc}$ is even as:

$$\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01}) = \sum_{l=1, l \text{ odd}}^{\min(\epsilon_{01},w-\mathtt{tc}-1)} \frac{\eta(\mathtt{tc}, l) \cdot \zeta(\mathtt{tc}, l, \epsilon_{01})}{\xi(\mathtt{tc}, \epsilon_{01})},$$

that is, we consider the union of all disjoint events where an odd number l of bit flips from the first iteration affect the positions involved in the parity check equation at hand. The case for $\mathtt{tc}$ is analogous, save for replacing $p_{\mathtt{flip}|0,\mathtt{OneEqSat}}$ with $p_{\mathtt{flip}|0,\mathtt{OneEqUnsat}}$ in the definition of $\eta(\mathtt{tc}, l)$ since, if the number of set error bits $\mathtt{tc}$ involved in the parity check equation is odd, the equation will be unsatisfied.

We employ the same approach used to derive $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01})$ to obtain $\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11}) =$

$$\Pr\left(|\mathrm{Supp}(h_{i,:}) \cap \mathbf{J}_{1,1}| \text{ is odd } \mid \exists a \in \mathbf{J}_{0,0}, h_{i,a} = 1, \langle e, h_{i,:}\rangle = \mathtt{tc}\right),$$

where we still consider a parity check equation involving a position in $\mathbf{J}_{0,0}$, and $\mathtt{tc}$ positions where the error vector e contains set bits, but, differently from $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc})$ we model the event of an odd amount of the positions in $\mathbf{J}_{1,1}$ being involved in the parity check equation itself. We recall that the positions in $\mathbf{J}_{1,1}$ are characterized by being the ones where the first iteration did not flip $\bar{e}$, while the error vector has a bit equal to 1 in it. We start with the case of even $\mathtt{tc}$ values. Defining $\nu(\mathtt{tc}, l) = \textsc{Bin}(\mathtt{tc}, p_{\neg\mathtt{flip}|1,\mathtt{OneEqSat}}, l)$, we have the probability distribution that l positions in $\mathbf{J}_{1,1}$ are among the $\mathtt{tc}$ ones involved in the parity check equation. This is equivalent to saying that, among the $\mathtt{tc}$ positions involved in the parity check equation where the error e contains a set bit, l are not flipped by the first iteration in $\bar{e}$, thus resulting in l discrepancies being maintained after the first iteration itself. Each one of such maintaining actions takes place with probability $p_{\neg\mathtt{flip}|1,\mathtt{OneEqSat}}$, i.e., the probability that a position is not flipped, given that the corresponding error bit is set, and the parity check equation in which it is involved is satisfied (indeed, $\mathtt{tc}$ is even).

Defining $\lambda(\mathtt{tc}, l, \epsilon_{11}) = \textsc{Bin}(t - \mathtt{tc}, p_{\neg\mathtt{flip}|1}, \epsilon_{11} - l)$ we model the probability of the remaining $\epsilon_{11} - l$ positions where the first iteration did not flip the error estimate in correspondence with a set error bit to be among the $t - \mathtt{tc}$ positions where the error vector is set, and which are not among the ones involved in the parity check equation. The probability of each such position not being flipped is indeed $p_{\neg\mathtt{flip}|1}$. With an approach similar to what was done for $\xi(\mathtt{tc}, \epsilon_{01})$, we combine $\nu(\mathtt{tc}, l)$ and $\lambda(\mathtt{tc}, l, \epsilon_{11})$ to obtain $\theta(\mathtt{tc}, \epsilon_{11}) =$

$$\sum_{l'=\max(0,\epsilon_{11}-(t-\mathtt{tc}))}^{\min(\epsilon_{11},tc)} \nu(\mathtt{tc}, l') \cdot \lambda(\mathtt{tc}, l', \epsilon_{11})$$

We thus compose $\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11})$ as:

$$\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11}) = \sum_{l=1, l\,\mathrm{odd}}^{\min(\epsilon_{11},\mathrm{tc})} \frac{\nu(\mathtt{tc}, l) \cdot \lambda(\mathtt{tc}, l, \epsilon_{11})}{\theta(\mathtt{tc}, \epsilon_{11})}.$$

Employing $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{0,1}|)$ and $\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{1,1}|)$ we are now able to define the probability that a satisfied parity check involving a position in $\mathbf{J}_{0,0}$ becomes unsatisfied after the flips made by the first iteration, $p_{00|\mathtt{BecomeUnsat}}$. The change in the satisfaction value of the parity check equation takes place in two cases after the flips of the first iteration are applied. The first case is when an odd number of discrepancies is added (i.e. an odd number of positions of $\mathbf{J}_{0,1}$ are involved in the parity check equation), and an even number of discrepancies is maintained (i.e. an even number of positions of $\mathbf{J}_{1,1}$ are involved in the parity check equation). For the sake of description, consider the number of set error bits $\mathtt{tc}$ involved in the parity check equation to be known: the probability of the previous event is $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{0,1}|)(1 - \chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{1,1}|))$. The second case is when an even number of discrepancies is added, and an odd number of discrepancies is maintained. The probability of such an event, fixing the number of set error bits $\mathtt{tc}$ involved in the parity check, is modeled by $(1 - \chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{0,1}|))\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{1,1}|)$. Therefore, the probability that, fixing the value of $\mathtt{tc}$, a specific flip pattern from the first iteration makes a parity check equation containing a position in $\mathbf{J}_{0,0}$ switch from containing an even number of set terms to containing an odd one is $\gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, |\mathbf{J}_{0,1}|, |\mathbf{J}_{1,1}|) = \chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{0,1}|)(1 - \chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{1,1}|)) + (1 - \chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{0,1}|))\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, |\mathbf{J}_{1,1}|)$. Observe that any pair of events, with fixed values of $\mathtt{tc}$, say, $\mathtt{tc}', \mathtt{tc}''$, are disjoint if $\mathtt{tc}' \neq \mathtt{tc}''$.

Furthermore, note that, for the events to take place, a certain number of positions, $\mathtt{tc}$ should be present among the ones in the parity check equations. The probability that a certain amount, $\mathtt{tc}$, among the t ones where the error vector is set, falls within the w involved in the parity check equation, knowing that one of the w is taken by the currently analyzed position in $\mathbf{J}_{0,0}$ is denoted as $\Pr(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{0,0})$, and its explicit derivation can be found in the Appendix B.

It is thus possible to compute the overall probability $p_{00|\mathtt{BecomeUnsat}}(\epsilon_{01}, \epsilon_{11})$ as

$$\frac{\sum_{\mathtt{tc}=0,\mathtt{tc}\text{ even}}^{\min(t,w-1)} \Pr(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{0,0}) \cdot \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11})}{\sum_{\mathtt{tc}=0,\mathtt{tc}\text{ even}}^{\min(t,w-1)} \Pr(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{0,0})}$$

where the denominator is justified by the fact that, in a satisfied parity check, only even values of $\mathtt{tc}$ are possible. We now obtain $p_{00|\mathtt{StayUnsat}}(\epsilon_{01}, \epsilon_{11})$ through an analogous line of reasoning:

$$\frac{\sum_{\mathtt{tc}=1,\mathtt{tc}\text{ odd}}^{\min(t,w-1)} \Pr(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{0,0}) \cdot \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11})}{\sum_{\mathtt{tc}=1,\mathtt{tc}\text{ odd}}^{\min(t,w-1)} \Pr(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{0,0})}.$$

Note that $\gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11})$ is employed in the same fashion in building $p_{00|\mathtt{StayUnsat}}$ as it defines the events which result in the parity check equation being unsatisfied after the flips of the first iteration.

We model with $\mu(\mathtt{nsat}, \mathtt{nunsat}, a, \epsilon_{01}, \epsilon_{11})$ the probability that, as a result of the flips from the first iteration, among the v parity check equations which involve a given position in $\mathbf{J}_{0,0}$, out of the $v-a$ satisfied ones $\mathtt{nsat}$ become unsatisfied, while, out of the a unsatisfied ones $\mathtt{nunsat}$ stay unsatisfied. We obtain $\mu(\mathtt{nsat}, \mathtt{nunsat}, a) =$
$\textsc{Bin}(v - a, p_{00|\mathtt{BecomeUnsat}}(\epsilon_{01}, \epsilon_{11}), \mathtt{nsat})\cdot$
$\textsc{Bin}(a, p_{00|\mathtt{StayUnsat}}(\epsilon_{01}, \epsilon_{11}), \mathtt{nunsat})$

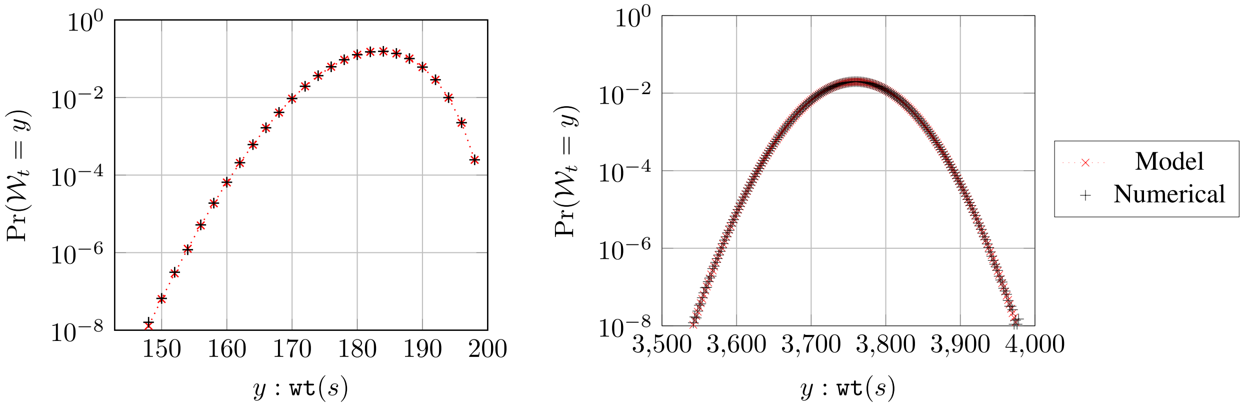
Consider now that we can obtain the probability distribution of the j-th upc being valued a, given that $j \in \mathbf{J}_{0,0}$ as:

$$\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,0}) = \begin{cases} \frac{\textsc{Bin}(v, p_{\mathtt{unsat}|0}, a)}{p_{\neg\mathtt{flip}|0}} & \text{if } x < \mathtt{th}_{(1)} \\ 0 & \text{otherwise} \end{cases}$$

Finally, we obtain $p_{\mathtt{flip}|00}$ as a function of ϵ_{01} and ϵ_{11}, $p_{\mathtt{flip}|00}(\epsilon_{01}, \epsilon_{11})$:

$$p_{\mathtt{flip}|00}(\epsilon_{01}, \epsilon_{11}) = \sum_{a=0}^{\mathtt{th}_{(1)}-1} \Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,0})\cdot$$
$$\left(\sum_{\mathtt{nsat}=0}^{v-a} \sum_{\substack{\mathtt{nunsat}\,= \\ \max(0,\,\mathtt{th}_{(2)}\,-\,\mathtt{nsat})}}^{a} \mu(\mathtt{nsat}, \mathtt{nunsat}, a, \epsilon_{01}, \epsilon_{11}) \right)$$

The probabilities $p_{\mathtt{flip}|01}(\epsilon_{01}, \epsilon_{11}), p_{\mathtt{flip}|10}(\epsilon_{01}, \epsilon_{11})$ and, $p_{\mathtt{flip}|11}(\epsilon_{01}, \epsilon_{11})$ are obtained through an analogous line of reasoning (full derivation in Appendix C).



(a) $n = 4400, k = 2200, v = 11, t = 18$

(b) $n_0 = 4, p = 13397, v = 83, t = 66$

Fig. 1. Numerical validation of the model of syndrome weight distribution, simulation on a (v, w)-regular code parity check matrix, picking a communications-grade code parameter set (left) and a cryptography grade code parameter set (right). Numerical results obtained with 10^9 random syndrome samples

D. Two-iterations DFR estimation

Having obtained the probability of flipping an error estimate bit at the second iteration, depending on which set among $\mathbf{J}_{0,0}, \mathbf{J}_{0,1}, \mathbf{J}_{1,0}$ and $\mathbf{J}_{1,1}$ contain its position, and on the cardinality of $\mathbf{J}_{0,1}$ and $\mathbf{J}_{1,1}$, we are able to coalesce $\Pr(\mathcal{E}_{(2)} = 0|\mathcal{W}_t = y, |\mathbf{J}_{0,1}| = \epsilon_{01}, |\mathbf{J}_{1,1}| = \epsilon_{11})$, that is the probability that no discrepancies are left after the second iteration, i.e., the decoding algorithm terminates after it.

We recall that, from the analysis of the first iteration, we are able to compute $\Pr(\epsilon_{01} = \mathrm{d}_+)$ as $\delta_+(\mathrm{d}_+)$ and $\Pr(\epsilon_{11} = t - \mathrm{d}_-) = \delta_-(\mathrm{d}_-)$. Thanks to this knowledge, we are able to derive the probability of performing a correct decoding $\Pr(\mathcal{E}_{(2)} = 0|\mathcal{W}_t = y)$, combining together the aforementioned probabilities, as follows:

$$\sum_{\epsilon_{01},\epsilon_{11}\in\{0,\dots,n-t\}\times\{0,\dots,t\}} \Big(\delta_+(\epsilon_{01})\cdot\delta_-(t-\epsilon_{11})\cdot$$

$$\cdot(1-p_{\mathtt{flip}|00}(\epsilon_{01},\epsilon_{11}))^{n-t-\epsilon01}\cdot p_{\mathtt{flip}|01}(\epsilon_{01},\epsilon_{11})^{\epsilon_{01}}\cdot$$

$$\cdot(1-p_{\mathtt{flip}|10}(\epsilon_{01}))^{t-\epsilon_{11}}\cdot p_{\mathtt{flip}|11}(\epsilon_{01},\epsilon_{11}))^{\epsilon_{11}}\Big)$$

Finally, we derive the Decoding Failure Rate (DFR) after the second decoder iteration as $\mathrm{DFR} = 1 - \Pr(\mathcal{E}_{(2)} = 0)$

$$\mathrm{DFR} = 1 - \sum_{y=0}^{r} \Pr(\mathcal{E}_{(2)} = 0|\mathcal{W}_t = y)\Pr(\mathcal{W}_t = y)$$

We note that the obtained value provides an average of the DFR value over all the (v, w) regular codes. While this is desirable when designing code parameters for a target DFR, it does not employ the information provided by the knowledge of a specific matrix H, when available. We note that our result can be fruitfully combine with the ones from [6], [8], which exploit the knowledge of the (quasi cyclic) parity check matrix values H to derive an amount of discrepancies between the error vector and its estimate that a bit flipping decoder corrects in a single iteration, denoted as τ in [6] and in the following. To combine our results with the aforementioned ones, given the matrix H, compute τ, and subsequently obtain the compute $\Pr(\mathcal{E}_{(2)} = 0|\mathcal{W}_t = y)$ excluding from the weighted sum employed to derive it all the terms resulting in an amount of discrepancies after the first iteration $\leq \tau$.

IV. Numerical validation

In this section, we provide numerical validations of *i)* our syndrome weight estimation technique, *ii)* the probability distributions of the discrepancies (i.e., differences between the error vector estimate and the actual error vector) after the first iteration, $\delta_+(\mathrm{d}_+)$ and $\delta_-(\mathrm{d}_-)$, *iii)* the two-iterations decoding failure rate of a parallel bit-flipping decoder, while varying the code density and the number of errors; *iv)* we report the result of applying our two-iterations DFR estimation technique, in combination with the result of [9] to re-evaluate the DFR of the code parameters proposed in the LEDAcrypt specification [3].

The numeric simulations were run on two Dell PowerEdge R630 nodes, each one endowed with two Intel Xeon CPU E5-2698 v4 (20 cores/40 threads each), and a Dell PowerEdge R7425 equipped with two AMD Epyc 7551 (32 cores/64 threads each), taking around 50k core-hours. The memory footprint of the simulations was small (<200MiB). The model results were obtained on a desktop equipped with an Intel Core i7-12700, and took approximately half a core-hour overall.

Figure 1 reports the comparison between our modeled (red) and the numerically estimated (black) distribution of syndrome weights $\Pr(\mathcal{W}_t = y)$ for a small $(11-22)$ regular LDPC code with code length $n = 4400$ and error weight $t = 18$ (Figure 1a), and for the code with rate $\frac{3}{4}$ employed in the LEDAcrypt specification [3] (Figure 1b), for NIST security category 1. In both cases, our estimation technique provides a very good match for the numerically simulated probabilities. We note that the asymmetric shape of Figure 1a is justified by the code being rather sparse, in turn causing little interaction among the bits of the columns of the parity check matrix being added to

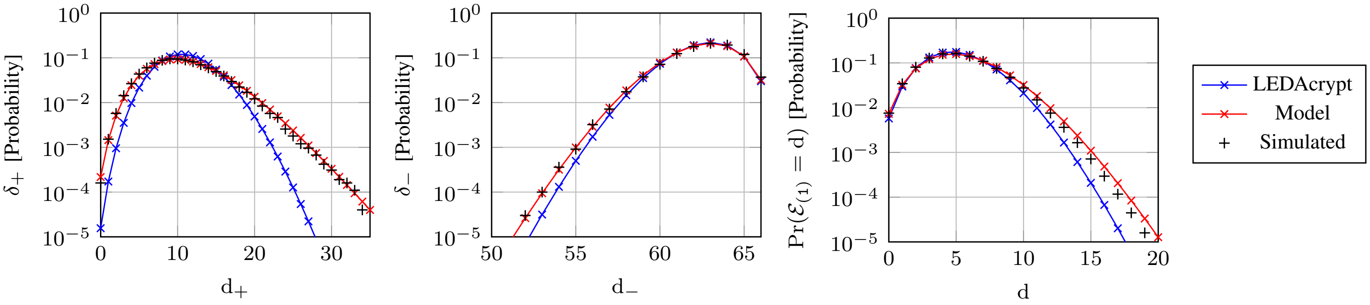
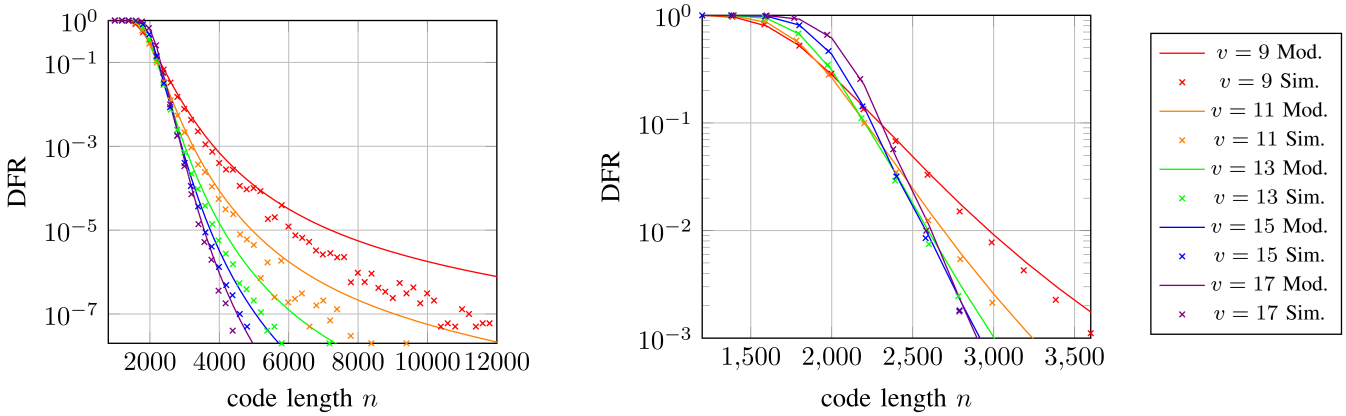


Fig. 2. Number flips on $\bar{e}_j$ which took place when $e_j = 0$ (d_+) and number of flips not made on $\bar{e}_j$ when $e_j = 1$ ($t - \mathrm{d}_-$) after the first iteration for the LEDAcrypt code with parameters $n_0 = 4, p = 13397, n = n_0 p, k = (n_0 - 1)p, v = 83$, results obtained with 10^5 randomly generated error vectors of weight $t = 66$ for each point.



(a) Waterfall and floor regions.

(b) Focus on the beginning of the waterfall region.

Fig. 3. Two iterations DFR values for $(v, 2v)$-regular LDPC codes, $v \in \{9, 11, 13, 15, 17\}$, with rate $\frac{k}{n} = \frac{1}{2}$, $t = 18$, parallel decoder employing majority thresholds, i.e., $\texttt{th1} = \texttt{th2} = \lceil \frac{v+1}{2} \rceil$. Each data point was obtained performing 10^8 decoding actions, or a sufficient number to obtain 100 decoding failures.

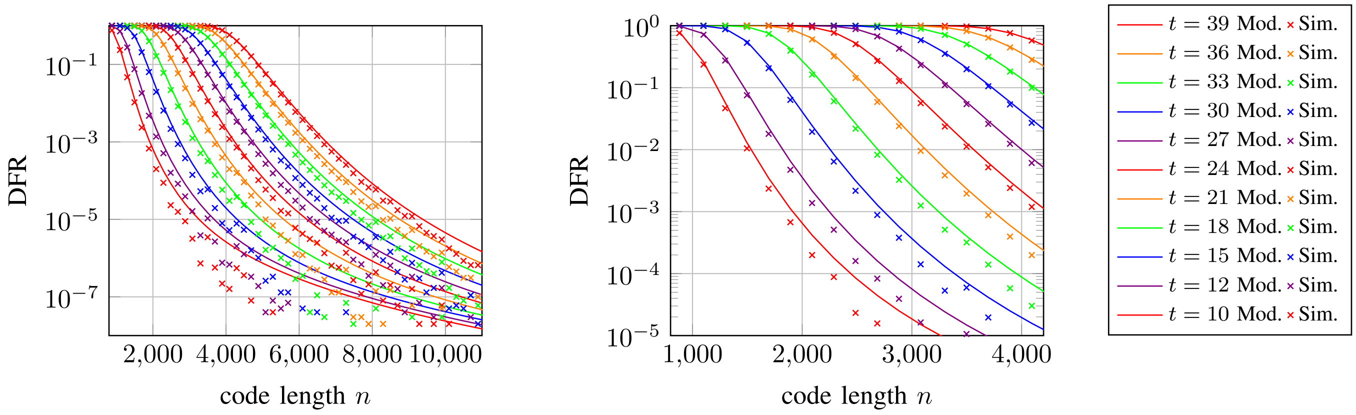
the syndrome. Indeed, a non-negligible amount of syndromes have the maximum weight $v \cdot t = 198$.

Figure 2 reports the distribution of d, d_+ and d_-. We recall that the aforementioned values are the number of discrepancies between the error vector and its estimate in the decoder after the first iteration (d), the number of discrepancies introduced by the first iteration flipping up positions where the error vector does not have a set bit (d_+), and the number of discrepancies removed by the first iteration not flipping a position of the estimate where the error vector has a set bit (d_-). In all three cases, our model yields a distribution that depends on the syndrome weight; the depicted values are thus obtained as the weighted average over all syndrome weights. As it can be seen, our model provides a closer fit to the sample distribution of d, d_+ and d_-, w.r.t. the one employed in [3].

We now validate the goodness of fit of our two-iterations DFR predictions against numerical simulations. To this end, we chose to sweep over the code length n, column weight v and error weight t parameters. All numerical simulations are done performing 10^8 decoding actions, or a sufficient number to obtain 100 decoding failures, whichever takes place first. Due to computation time constraints, we restricted our systematic numerical validation for the two-iterations DFR to $(v, 2v)$ regular codes with rate $\frac{1}{2}$.

We present our results examining first the effects of changing code length n and column weight v, as depicted in Figure 3. We chose the parameter regime in such a fashion that the code with the smallest column weight $v = 7$ achieves a floor regime well within the explored range of code lengths ($n \in \{1200, \ldots, 12000\}$), and visible failure rates. Figure 3a shows how our two-iteration DFR prediction provides a conservative estimate for the decoder behaviour when after the floor regime has been reached, while providing a remarkably good fit for the waterfall region. This is visible in a clearer fashion in Figure 3b, which provides a zoom on the waterfall region of all codes, showing the closeness between our model and the numerical simulations.

We move onto the analysis of robustness of our estimates, when changing the number of errors $t \in \{10, 13, \ldots, 37\}$ while retaining the same column weight v and exploring the code length range $\{800, \ldots, 11000\}$. The code length range to be explored, and the error weight were chosen with the



(a) Waterfall and floor regions.

(b) Focus on the beginning of the waterfall region.

Fig. 4. Two iterations DFR values for $(v, 2v)$-regular LDPC codes, $t \in \{10, \ldots, 39\}$, with rate $\frac{k}{n} = \frac{1}{2}$, $v = 11$, parallel decoder employing majority thresholds, i.e., $\texttt{th1} = \texttt{th2} = \lceil \frac{v+1}{2} \rceil$. Each data point was obtained performing 10^8 decoding actions, or a sufficient number to obtain 100 decoding failures.

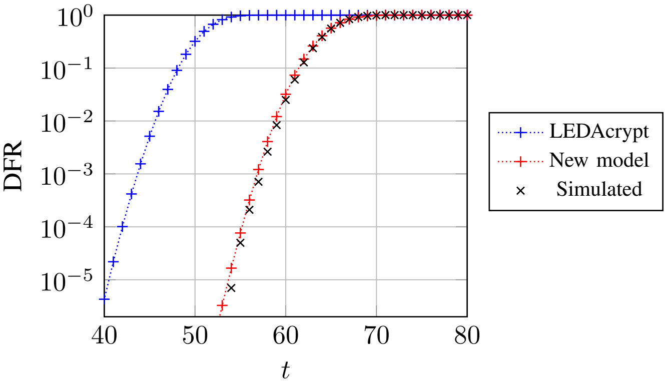


Fig. 5. Comparison of our two-iterations DFR estimation technique with the one currently employed in the LEDAcrypt parameter design. Code parameters matching the ones in the LEDAcrypt specifications [3], Section 4.1, Figure 4.1: $n_0 = 2, p = 4801, n = n_0 p, k = p, v = 45, \texttt{th1} = 25, \texttt{th2} = 25$. Simulation data obtained from 10^6 randomly generated error vectors of weight t, decoded with a 2-iterations parallel (*Out-of-place* in LEDAcrypt specification) decoder.

same intent as the previous exploration, i.e., an attempt at covering as much as possible the variety of behaviours when transitioning from a waterfall to a floor regime.

Figure 4 depicts the results of the validation campaign, providing the overview of the entire parameter sweep in Figure 4a, and a zoom on the smaller code lengths in Figure 4b. As it can be seen, our DFR estimation technique still provides consistent and accurate estimates for the waterfall region of the examined codes, while providing a conservative (i.e., higher) estimated DFR value in the floor regime of the codes.

Finally, we move onto the last step of our numerical validation, i.e., the use of the proposed two-iterations DFR estimate to analyze the current parameter sets from LEDAcrypt. First of all, we quantify the improvement in the estimate of two-iterations DFR, replicating the numeric experiment reported in the LEDAcrypt specification, Section 4.1. Figure 5 reports the result of numerically obtained DFR values on a relatively small code $n_0 = 2, p = 4801, n = n_0 p, k = p, v = 45$, obtained performing 10^6 decoding actions, while sweeping the range of error weight $\{40, \ldots, 80\}$. As it can be seen, we effectively improve the tightness of the DFR estimate by $\approx 10^5$, in the visible waterfall region of the code.

TABLE I
ESTIMATES FOR THE DFR OF A TWO-ITERATIONS OUT-OF-ORDER DECODING FROM THIS WORK, COMPARED TO THE ONES OF THE LEDACRYPT [3] SPECIFICATIONS. CODES ARE $[n_0 p, (n_0 - 1)p]$ QC-LDPC CODES, HENCE $(v, n_0 v)$ REGULAR

NIST Cat.	$\mathbf{n_0}$	**p**	**v**	**t**	τ	**LEDAcrypt DFR**	**Our DFR**
1	2	23371	71	130	10	2^{-64}	2^{-147}
	3	16067	79	83	9	2^{-64}	2^{-139}
	4	13397	83	66	8	2^{-64}	2^{-134}
	2	28277	69	129	11	2^{-128}	2^{-203}
	3	19709	79	82	10	2^{-128}	2^{-198}
	4	16229	83	65	9	2^{-128}	2^{-189}
3	2	40787	103	195	13	2^{-64}	2^{-190}
	3	28411	117	124	11	2^{-64}	2^{-181}
	4	22901	123	98	11	2^{-64}	2^{-167}
	2	52667	103	195	15	2^{-192}	2^{-309}
	3	36629	115	123	13	2^{-192}	2^{-300}
	4	30803	123	98	12	2^{-192}	2^{-295}
5	2	61717	137	261	17	2^{-64}	2^{-223}
	3	42677	153	165	14	2^{-64}	2^{-220}
	4	35507	163	131	13	2^{-64}	2^{-208}
	2	83579	135	260	18	2^{-256}	2^{-422}
	3	58171	153	165	16	2^{-256}	2^{-403}
	4	48371	161	131	15	2^{-256}	2^{-396}

This significant improvement in the tightness of the estimate allows us to re-analyze the parameters proposed in the LEDAcrypt specification, as reported in Table I.

LEDAcrypt employs quasi-cyclic codes, with rates among $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}$, providing security-equivalent parameter sets for all the rates. For each rate and security level, the LEDAcrypt specification proposed two set of parameters, one with DFR low enough to formally guarantee the resistance against an active attacker (IND-CCA2 security), and one with a practically low enough DFR 2^{-64} so that the event of an attacker succeeding in causing a failure in decryption is extremely unlikely. Applying our estimate techniques, we observe that the parameter sets which were proposed with DFR $\leq 2^{-64}$ for NIST security category 1 actually guarantee a DFR $< 2^{-128}$, in turn fully meeting the requirement for IND-CCA2 security ($\leq 2^{-128}$, for category 1), and indeed leaving further margin for the reduction of the code size. The parameters proposed with DFR $\leq 2^{-64}$ for NIST security category 3 come relatively close to the requirement for IND-CCA2 security ($\leq 2^{-192}$, for category 3), albeit not meeting it already, while their analogues for NIST category 5 have DFRs decidely higher than the required $\leq 2^{-256}$. These results point to the possibility a further reduction of the code sizes for LEDAcrypt at NIST security category 1 by more than 20% with respect to the currently proposed ones, while a significant reduction is expected also for category 3 and 5.

V. Conclusion

In this work, we presented a new technique to estimate the decoding failure rate of two-iteration parallel bit flipping decoders. Our technique relies on a sound model for the syndrome weight distribution, and on a novel approach to determining if a bit of the error vector estimation is flipped at the second iteration of the parallel bit flipping decoder. We validated numerically our approach showing a good fit of our prediction to the actual DFRs in the waterfall region, and a conservative estimate of the DFR in the floor region. In light of our results, we analyzed the DFR estimates made by LEDAcrypt in their parameter design, and have shown that it is possible to reduce the chosen code lengths by $\approx 20\%$ for NIST category 1, while retaining a suitable DFR.

Acknowledgment

This work was carried out with partial financial support of the Italian MUR (PRIN 2022 project POst quantum Identification and eNcryption primiTives: dEsign and Realization (POINTER) ID-2022M2JLF2)

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962. [Online]. Available: https://doi.org/10.1109/TIT.1962.1057683

[2] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Güneysu, C. A. Melchor, R. Misoczki, E. Persichetti, J. Richter-Brockmann, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor, "BIKE: Bit Flipping Key Encapsulation. Round 4 Submission," [Online] Available: https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf, 2023.

[3] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "LEDAcrypt - version 3.0 Specification," [Online] Available: https://www.ledacrypt.org/documents/LEDAcrypt_v3.pdf, 2023.

[4] Q. Guo, T. Johansson, and P. S. Wagner, "A Key Recovery Reaction Attack on QC-MDPC," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1845–1861, 2019. [Online]. Available: https://doi.org/10.1109/TIT.2018.2877458

[5] N. Sendrier and V. Vasseur, "On the decoding failure rate of QC-MDPC bit-flipping decoders," in *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, ser. Lecture Notes in Computer Science, J. Ding and R. Steinwandt, Eds., vol. 11505. Springer, 2019, pp. 404–416. [Online]. Available: https://doi.org/10.1007/978-3-030-25510-7_22

[6] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "A Failure Rate Model of Bit-flipping Decoders for QC-LDPC and QC-MDPC Code-based Cryptosystems," in *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020 - Volume 2: SECRYPT, Lieusaint, Paris, France, July 8-10, 2020*, P. Samarati, S. D. C. di Vimercati, M. S. Obaidat, and J. Ben-Othman, Eds. ScitePress, 2020, pp. 238–249. [Online]. Available: https://doi.org/10.5220/0009891702380249

[7] ——, "Analysis of In-Place Randomized Bit-Flipping Decoders for the Design of LDPC and MDPC Code-Based Cryptosystems," in *E-Business and Telecommunications - 17th International Conference on E-Business and Telecommunications, ICETE 2020, Online Event, July 8-10, 2020, Revised Selected Papers*, ser. Communications in Computer and Information Science, M. S. Obaidat and J. Ben-Othman, Eds., vol. 1484. Springer, 2020, pp. 151–174. [Online]. Available: https://doi.org/10.1007/978-3-030-90428-9_7

[8] J. Tillich, "The Decoding Failure Probability of MDPC Codes," in *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018.* IEEE, 2018, pp. 941–945. [Online]. Available: https://doi.org/10.1109/ISIT.2018.8437843

[9] P. Santini, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Analysis of the Error Correction Capability of LDPC and MDPC Codes Under Parallel Bit-Flipping Decoding and Application to Cryptography," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4648–4660, 2020. [Online]. Available: https://doi.org/10.1109/TCOMM.2020.2987898

[10] D. Moody, "Nist pqc: Looking into the future," https://csrc.nist.gov/Presentations/2022/nist-pqc-looking-into-the-future, November 2022.

[11] J. Chaulet, "Etude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques. (study of public key cryptosystems based on quasi-cyclic MDPC codes)," Ph.D. dissertation, Pierre and Marie Curie University, Paris, France, 2017. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01599347

[12] V. Vasseur, "Post-quantum cryptography: a study of the decoding of QC-MDPC codes. (Cryptographie post-quantique : étude du décodage des codes QC-MDPC)," Ph.D. dissertation, University of Paris, France, 2021. [Online]. Available: https://tel.archives-ouvertes.fr/tel-03254461

[13] ——, "QC-MDPC codes DFR and the IND-CCA security of BIKE," *IACR Cryptol. ePrint Arch.*, p. 1458, 2021. [Online]. Available: https://eprint.iacr.org/2021/1458

[14] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "Performance Bounds for QC-MDPC Codes Decoders," in *Code-Based Cryptography - 9th International Workshop, CBCrypto 2021, Munich, Germany, June 21-22, 2021 Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Wachter-Zeh, H. Bartz, and G. Liva, Eds., vol. 13150. Springer, 2021, pp. 95–122. [Online]. Available: https://doi.org/10.1007/978-3-030-98365-9_6

[15] S. Arpin, T. R. Billingsley, D. R. Hast, J. B. Lau, R. A. Perlner, and A. Robinson, "A Study of Error Floor Behavior in QC-MDPC Codes," in *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022, Virtual Event, September 28-30, 2022, Proceedings*, ser. Lecture Notes in Computer Science, J. H. Cheon and T. Johansson, Eds., vol. 13512. Springer, 2022, pp. 89–103. [Online]. Available: https://doi.org/10.1007/978-3-031-17234-2_5

[16] S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," *CoRR*, vol. abs/0911.3262, 2009. [Online]. Available: http://arxiv.org/abs/0911.3262

[17] R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," in *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12,*

2013. IEEE, 2013, pp. 2069–2073. [Online]. Available: https://doi.org/10.1109/ISIT.2013.6620590

Appendix A
Additional material

In this section, we derive an explicit formulation for the distribution of the number of asserted bits in each parity check before the first iteration, knowing the syndrome weight distribution, referred as $\Pr\left(\mathcal{F}_t = f \mid \mathcal{W}_t = y\right)$. To calculate $\Pr\left(\mathcal{F}_t = f \mid \mathcal{W}_t = y\right)$, we first formulate the problem in the following way:

$$\Pr\left(\mathcal{F}_t = f \mid \mathcal{W}_t = y\right) = \frac{\Pr(\mathcal{F}_t = f \cap \mathcal{W}_t = y)}{\Pr(\mathcal{W}_T = y)} =$$

$$= \Pr\left(\mathcal{W}_t = y \mid \mathcal{F}_t = f\right) \frac{\Pr(\mathcal{F}_t = f)}{\Pr(\mathcal{W}_t = y)}$$

While the two quantities on the right of the expression have been already calculated, the probability $\Pr\left(\mathcal{W}_t = y \mid \mathcal{F}_t = f\right)$ expresses the probability that the syndrome weight is equal to y, given the amount of flips applied to one of the syndrome bits. To derive this probability, we undergo a process similar to the one employed for the calculation of the syndrome weight distribution. We begin by defining a state vector $\texttt{wp}_{(l)} = [\texttt{wp}_{(l),0}, ..., \texttt{wp}_{(l),r-1}]$ modeling the number of set bit in the $r-1$ syndrome bits different from the selected one. As before, the starting state is the one corresponding to a null syndrome, thus $\texttt{wp}_{(0)} = [1, 0, ..., 0]$. Given the t total bit flips applied to the error vector, f of these will cause a flip in the selected bit and $v-1$ flips in the other $r-1$ syndrome bits, while the remaining $t-f$ error bits will not cause any flip in the selected parity check, while causing instead v flips in the $r-1$ other positions. To complete the definition of this modified discrete-time non-omogeneous Markov chain, we need to define the transition matrices $\texttt{P}_{(l)} = [p_{x,y,l}]_{x,y\in\{0,...,r\}}$. We can assume, without loss of generality, that the f flips affecting the selected syndrome bit happen after the other $t-f$ flips. Since we are considering the rows of the matrix H as independent from each other, the number of flips applied to a certain syndrome bit provides no information about the flips applied to the other $r-1$ positions, therefore the distribution of the number of flips applied during the process to a generic syndrome bit:

$$\phi_l(f, l) = \Pr(\mathcal{F}_l = f) = \frac{\binom{w}{f}\binom{n-w}{l-f}}{\binom{n}{l}}$$

remains unchanged. Subsequently, the definitions of $\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to 1}}(l)$ and $\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to 0}}(l)$, corresponding to the probability of flipping up or down a syndrome bit during step l, do not change either. The probabilities we need to modify are $\Pr(\texttt{E}_{1,a})$=$\varphi(x, a, l)$, corresponding to the probability of flipping up a bits, and $\Pr(\texttt{E}_{2,a}) = \psi(x, a, l)$, corresponding to the probability of flipping down $v-a$ (or $v-1-a$, depending on l) bits, where x is the number of asserted bits and l is the current step. The following modification is justified by the fact that only $r-1$ positions can be modified instead of r, and the number of bit flips happening at each step depends on the value of l. For $1 \le l \le t-f$, we have that v flips take place at each step, so the formulation of $\varphi(x, a, l)$ and $\psi(x, a, l)$ becomes:

$$\varphi(x, a, l) = \binom{r-1-x}{a}\left(\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to 1}}(l)\right)^{a}\left(1-\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to 1}}(l)\right)^{(r-1-x)-a}$$

$$\psi(x, a, l) = \binom{x}{v-a}\left(\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to 0}}(l)\right)^{v-a}\left(1-\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to 0}}(l)\right)^{x-(v-a)}$$

The probability distribution $\Pr(\mathcal{W}_{l-1} = x) = \omega(x, l)$ has to be modified in a similar way:

$$\omega(x, l) = \sum\nolimits_{i}_{\max\{0,v-x\}}^{\min\{r-1-x,v\}} \left(\varphi(x, i, l) \cdot \psi(x, i, l)\right).$$

Finally, the elements of the transition matrix $p_{x,y,l}$ indicating the probability of changing the number of asserted bits in the syndrome from x to y can be defined as:

$$p_{x,y,l} = \begin{cases} 1, & l = 1, x = 0, y = v \\ \rho(x, y, l), & \begin{array}{l} l \ge 2 \\ \max(0, x-v) \le y \le \min(x+v, r-1) \\ y \equiv_2 (x+v) \end{array} \\ 0, & \text{otherwise} \end{cases}$$

with

$$\rho(x, y, l) = \frac{\varphi(x, \frac{y-x+v}{2}, l) \cdot \psi(x, \frac{y-x+v}{2}, l)}{\omega(x, l)},$$

For $t-f+1 \le l \le t$, we apply a similar modification to the previous formulas, noting that in the last f steps only $v-1$ flips take place:

$$\varphi(x, a, l) = \binom{r-1-x}{a}\left(\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to 1}}(l)\right)^{a}\left(1-\pi_{\substack{l-1\to l\\ \texttt{flip}\,0\to 1}}(l)\right)^{(r-1-x)-a}$$

$$\psi(x, a, l) = \binom{x}{v-1-a}\left(\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to 0}}(l)\right)^{v-1-a}\left(1-\pi_{\substack{l-1\to l\\ \texttt{flip}\,1\to 0}}(l)\right)^{x-(v-1-a)}$$

$$\omega(x, l) = \sum\nolimits_{i}_{\max\{0,v-1-x\}}^{\min\{r-1-x,v-1\}} \left(\varphi(x, i, l) \cdot \psi(x, i, l)\right).$$

The definition of the transition probability for the last f rounds is the following:

$$p_{x,y,l} = \begin{cases} 1, & l = 1, x = 0, y = v-1 \\ \rho(x, y, l), & \begin{array}{l} l \ge 2 \\ \max(0, x-v+1) \le y \le \min(x+v-1, r-1) \\ y \equiv_2 (x+v-1) \end{array} \\ 0, & \text{otherwise} \end{cases}$$

with

$$\rho(x, y, l) = \frac{\varphi(x, \frac{y-x+v-1}{2}, l) \cdot \psi(x, \frac{y-x+v-1}{2}, l)}{\omega(x, l)},$$

Once the transition matrices have been derived, the Markov chain is well defined and the final state vector $\texttt{wp}_{(t)}$ can be obtained. The resulting vector contains the probability distribution of the number of asserted bits in the $r-1$ bits different from the selected one. The conditioned probability

$\Pr\left(\mathcal{W}_t = y \mid \mathcal{F}_t = f\right)$ can then be obtained by adding the selected bit to the count:

$$\Pr\left(\mathcal{W}_t = y \mid \mathcal{F}_t = f\right) = \begin{cases} \mathtt{wp}_{(t),y} & \text{if } f \text{ is even} \\ \mathtt{wp}_{(t),y-1} & \text{if } f \text{ is odd} \end{cases}$$

Appendix B
Additional material-2

The aim of the following analysis is to calculate the probability that a certain amount, $\mathtt{tc}$, among the t ones where the error vector is set, falls within the w involved in the parity check equation, knowing that one of the w is taken by the currently analyzed position in $\mathbf{J}_{0,0}$. The analysis in the case of bits in $\mathbf{J}_{0,1}$, $\mathbf{J}_{1,0}$ and $\mathbf{J}_{1,1}$ is equivalent up to a matter of indexes. We denote as $\mathtt{E}_{(\mathbf{J}_{0,0})}$ the event of a selected bit to be in $\mathbf{J}_{0,0}$, and we underline the fact that the aleatory variable corresponding to the number of asserted bits included in a certain parity check corresponds to the previously defined variable $\mathcal{F}_t$. We begin by formulating the problem in the following way:

$$\Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathtt{E}_{(\mathbf{J}_{0,0})}\right) = \frac{\Pr(\mathcal{F}_t = \mathtt{tc} \cap \mathtt{E}_{(\mathbf{J}_{0,0})})}{\Pr(\mathtt{E}_{(\mathbf{J}_{0,0})})} =$$

$$= \frac{\Pr\left(\mathtt{E}_{(\mathbf{J}_{0,0})} \mid \mathcal{F}_t = \mathtt{tc}\right)\Pr(\mathcal{F}_t = \mathtt{tc})}{\sum_{f=0}^{\min(t,w)} \Pr\left(\mathtt{E}_{(\mathbf{J}_{0,0})} \mid \mathcal{F}_t = f\right)\Pr(\mathcal{F}_t = f)}$$

Where we remind the fact that all the probabilities are implicitly conditioned to the syndrome weight. The probability $\Pr(\mathcal{F}_t = f)$ has been already calculated, while $\Pr\left(\mathtt{E}_{(\mathbf{J}_{0,0})} \mid \mathcal{F}_t = f\right)$ indicates the probability of choosing a bit in $\mathbf{J}_{0,0}$ inside a parity check where f asserted bits are included. The event of choosing a bit in $\mathbf{J}_{0,0}$ implies selecting a clear bit ($e_i = 0$) and not flipping it during the first iteration:

$$\Pr\left(\mathtt{E}_{(\mathbf{J}_{0,0})} \mid \mathcal{F}_t = f\right) =$$

$$= \Pr\left(e_i = 0 \mid \mathcal{F}_t = f\right) \cdot \Pr\left(\mathtt{E}_{(\mathbf{J}_{0,0})} \mid \mathcal{F}_t = f \cap e_i = 0\right)$$

The first term, the probability of choosing a clear bit, is the following:

$$\Pr\left(e_i = 0 \mid \mathcal{F}_t = f\right) = \frac{w-f}{w}$$

The second term, the probability of not flipping the clear bit, depends on the outcome of the parity check, which in turn depends on the parity of f: $\Pr\left(\mathtt{E}_{(\mathbf{J}_{0,0})} \mid \mathcal{F}_t = f \cap e_i = 0\right) =$

$$\begin{cases} p_{\mathtt{flip}|0,\mathtt{OneEqSat}} & \text{if } f \text{ is even} \\ p_{\mathtt{flip}|0,\mathtt{OneEqUnsat}} & \text{if } f \text{ is odd} \end{cases}$$

With this final result, we can conclude the analysis.

Appendix C
Additional material-3

We now consider the case of the bit under examination being in $\mathbf{J}_{0,1}$. The expression of $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01})$ is (changes highlighted in blue):

$$\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01}) = \sum_{l=1, l \text{ odd}}^{\min(\epsilon_{01}-1, w-\mathtt{tc}-1)} \frac{\eta(\mathtt{tc}, l) \cdot \zeta(\mathtt{tc}, l, \epsilon_{01}-1)}{\xi(\mathtt{tc}, \epsilon_{01}-1)},$$

changing from the one when the examined position is in $\mathbf{J}_{0,0}$ only in replacing the occurrences of ϵ_{01} with $\epsilon_{01} - 1$: indeed, one of the positions $j \in \mathbf{J}_{0,1}$ involved in the equation being considered is actually the variable being considered.

The expression of the probability $\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}) = \Pr\left(|\mathrm{Supp}(h_{i,:}) \cap \mathbf{J}_{1,1}| \text{ is odd} \mid \exists a \in \mathbf{J}_{0,0}, h_{i,a} = 1, \langle e, h_{i,:}\rangle = \mathtt{tc}\right)$, does not change, as the fact that a is in $\mathbf{J}_{0,1}$ instead of $\mathbf{J}_{0,0}$ does not have any effect on its formulation.

$p_{01|\mathtt{BecomeUnsat}}$ is obtained as (changes highlighted in blue)

$$\frac{\sum_{\mathtt{tc}=0, \mathtt{tc}\text{ even}}^{\min(t,w-1)} \Pr(\mathcal{F}_t{=}\mathtt{tc} \mid \mathbf{J}_{0,1}) \cdot (1 - \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11}))}{\sum_{\mathtt{tc}=0, \mathtt{tc}\text{ even}}^{\min(t,w-1)} \Pr(\mathcal{F}_t{=}\mathtt{tc} \mid \mathbf{J}_{0,1})}$$

Indeed the two changes from $p_{00|\mathtt{BecomeUnsat}}$ involve the fact that the position being considered is in $\mathbf{J}_{0,1}$ instead of $\mathbf{J}_{0,0}$ and the fact that the contribution to the parity taking place on the $w-1$ positions of the equation excluding the one being considered should be even in order to have an unsatisfied check. Symmetrically, $p_{01|\mathtt{StayUnsat}}$ is obtained as (changes highlighted in blue)

$$\frac{\sum_{\mathtt{tc}=1, \mathtt{tc}\text{ odd}}^{\min(t,w-1)} \Pr(\mathcal{F}_t{=}\mathtt{tc} \mid \mathbf{J}_{0,1}) \cdot (1 - \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11}))}{\sum_{\mathtt{tc}=1, \mathtt{tc}\text{ odd}}^{\min(t,w-1)} \Pr(\mathcal{F}_t{=}\mathtt{tc} \mid \mathbf{J}_{0,1})}.$$

The probability distribution of the j-th upc being valued a, given that $j \in \mathbf{J}_{0,1}$ is (changes w.r.t $\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,0})$ highlighted in blue):

$$\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,1}) = \begin{cases} \frac{\textsc{Bin}(v, p_{\mathtt{unsat}|0}, a)}{p_{\mathtt{flip}|0}} & \text{if } x \geq \mathtt{th}_{(1)} \\ 0 & \text{otherwise} \end{cases}$$

Since the expression of μ does not change, we obtain $p_{\mathtt{flip}|01}(\epsilon_{01}, \epsilon_{11})$:

$$p_{\mathtt{flip}|01}(\epsilon_{01}, \epsilon_{11}) = \sum_{a=\mathtt{th}_{(1)}}^{v} \Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,1}) \cdot$$

$$\cdot \left(\sum_{\mathtt{nsat}=0}^{v-a} \sum_{\substack{\mathtt{nunsat} = \\ \max(0, \mathtt{th}_{(2)} - \mathtt{nsat})}}^{a} \mu(\mathtt{nsat}, \mathtt{nunsat}, a, \epsilon_{01}, \epsilon_{11}) \right)$$

since we are interested in the probability of performing a flip at the second iteration (which does not change from the computation of $p_{\mathtt{flip}|00}$), but the first iteration performed a flip, therefore the first-iteration upc value a should be above the first iteration threshold $\mathtt{th}_{(1)}$.

We now move to the case of a bit being in $\mathbf{J}_{1,0}$. The expression of $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01})$ is:

$$\eta(\mathtt{tc}, l) = \textsc{Bin}(w - \mathtt{tc}, p_{\mathtt{flip}|0,\mathtt{OneEqSat}}, l)$$

$$\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01}) = \sum_{l=1, l \text{ odd}}^{\min(\epsilon_{01}, w-\mathtt{tc})} \frac{\eta(\mathtt{tc}, l) \cdot \zeta(\mathtt{tc}, l, \epsilon_{01})}{\xi(\mathtt{tc}, \epsilon_{01})},$$

where the modified indexed are justified by the fact that the bit we are analyzing is not in the set of initially clear bits, so we do not need to subtract it in the calculation of $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01})$. For the same reason, $\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11})$ becomes:

$$\nu(\mathtt{tc}, l) = \mathrm{B}\textsc{in}(\mathtt{tc} - 1, p_{\neg\mathtt{flip}|1,\mathtt{OneEqSat}}, l)$$

$$\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11}) = \sum_{l=1, l \text{ odd}}^{\min(\epsilon_{11}, \mathtt{tc}-1)} \frac{\nu(\mathtt{tc}, l) \cdot \lambda(\mathtt{tc}, l, \epsilon_{11})}{\theta(\mathtt{tc}, \epsilon_{11})}$$

Where $\mathtt{tc} - 1$ is the number of asserted positions in the error vector where a flip may happen. $p_{10|\mathtt{BecomeUnsat}}$ is obtained as

$$\frac{\sum_{\mathtt{tc}=2, \mathtt{tc} \text{ even}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,0}\right) \cdot \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11})}{\sum_{\mathtt{tc}=2, \mathtt{tc} \text{ even}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,0}\right)}$$

substituting $w - 1$ with w (since a clear bit in the check is not guaranteed) and $\mathtt{tc} = 0$ with $\mathtt{tc} = 2$ (since an asserted bit is now guaranteed). $p_{10|\mathtt{StayUnsat}}$ is obtained as

$$\frac{\sum_{\mathtt{tc}=1, \mathtt{tc} \text{ odd}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,0}\right) \cdot \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11})}{\sum_{\mathtt{tc}=1, \mathtt{tc} \text{ odd}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,0}\right)}.$$

The probability distribution of the j-th upc being valued a, given that $j \in \mathbf{J}_{1,0}$ is (changes w.r.t $\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,0})$ highlighted in blue):

$$\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{1,0}) = \begin{cases} \frac{\mathrm{B}\textsc{in}(v, p_{\mathtt{unsat}|1}, a)}{p_{\mathtt{flip}|1}} & \text{if } x \geq \mathtt{th}_{(1)} \\ 0 & \text{otherwise} \end{cases}$$

Since the expression of μ does not change, we obtain $p_{\mathtt{flip}|10}(\epsilon_{01}, \epsilon_{11})$:

$$p_{\mathtt{flip}|10}(\epsilon_{01}, \epsilon_{11}) = \sum_{a=\mathtt{th}_{(1)}}^{v} \Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{1,0}) \cdot$$
$$\cdot \left(\sum_{\mathtt{nsat}=0}^{v-a} \sum_{\substack{\mathtt{nunsat} = \\ \max(0, \mathtt{th}_{(2)} - \mathtt{nsat})}}^{a} \mu(\mathtt{nsat}, \mathtt{nunsat}, a, \epsilon_{01}, \epsilon_{11}) \right)$$

Finally, we derive the flipping probability of a bit in $\mathbf{J}_{1,1}$. Switching from $\mathbf{J}_{1,0}$ to $\mathbf{J}_{1,1}$ does not change the definition of $\chi_{\uparrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{01})$, while $\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11})$ becomes:

$$\chi_{\leftrightarrow\mathtt{odd}}(\mathtt{tc}, \epsilon_{11}) = \sum_{l=1, l \text{ odd}}^{\min(\epsilon_{11}-1, \mathtt{tc}-1)} \frac{\nu(\mathtt{tc}, l) \cdot \lambda(\mathtt{tc}, l, \epsilon_{11} - 1)}{\theta(\mathtt{tc}, \epsilon_{11} - 1)}$$

where, with the same line of reasoning as the $\mathbf{J}_{0,1}$ case, we exclude one of the bits in $\mathbf{J}_{1,1}$ being it the one under consideration. $p_{11|\mathtt{BecomeUnsat}}$ is obtained as

$$\frac{\sum_{\mathtt{tc}=2, \mathtt{tc} \text{ even}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,1}\right) \cdot (1 - \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11}))}{\sum_{\mathtt{tc}=2, \mathtt{tc} \text{ even}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,1}\right)}$$

noting that, for the parity check to be unsatisfied, an even number of asserted bits must result from the other $w - 1$ bits. $p_{11|\mathtt{StayUnsat}}$ is obtained as

$$\frac{\sum_{\mathtt{tc}=1, \mathtt{tc} \text{ odd}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,1}\right) \cdot (1 - \gamma_{\mathtt{UnsatPostFlips}}(\mathtt{tc}, \epsilon_{01}, \epsilon_{11}))}{\sum_{\mathtt{tc}=1, \mathtt{tc} \text{ odd}}^{\min(t,w)} \Pr\left(\mathcal{F}_t = \mathtt{tc} \mid \mathbf{J}_{1,1}\right)}.$$

The probability distribution of the j-th upc being valued a, given that $j \in \mathbf{J}_{1,1}$ is (changes w.r.t $\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{0,0})$ highlighted in blue):

$$\Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{1,1}) = \begin{cases} \frac{\mathrm{B}\textsc{in}(v, p_{\mathtt{unsat}|1}, a)}{p_{\mathtt{flip}|1}} & \text{if } x < \mathtt{th}_{(1)} \\ 0 & \text{otherwise} \end{cases}$$

Since the expression of μ does not change, we obtain $p_{\mathtt{flip}|11}(\epsilon_{01}, \epsilon_{11})$:

$$p_{\mathtt{flip}|11}(\epsilon_{01}, \epsilon_{11}) = \sum_{a=0}^{\mathtt{th}_{(1)}-1} \Pr(\mathcal{U}_j = a, j \in \mathbf{J}_{1,1}) \cdot$$
$$\cdot \left(\sum_{\mathtt{nsat}=0}^{v-a} \sum_{\substack{\mathtt{nunsat} = \\ \max(0, \mathtt{th}_{(2)} - \mathtt{nsat})}}^{a} \mu(\mathtt{nsat}, \mathtt{nunsat}, a, \epsilon_{01}, \epsilon_{11}) \right)$$

With this analysis, we obtained the probability of performing a flip on each of the four category of bits, in turn allowing us to derive the final DFR estimation as a function of the probability of performing the correct flips during the second iteration of the decoder.