Approach Beyond the BEC

Jérémie Ezri*, Andrea Montanari† and Ruediger Urbanke*
*Ecole Polytechnique Fédérale de Lausanne
Communication Theory Laboratory, CH-1015 Lausanne, Switzerland
Email: jeremie.ezri@epfl.ch and Ruediger.urbanke@epfl.ch
†Electrical Engineering and Statistics Departments
Stanford University, Stanford, CA
Email: montanari@stanford.edu

***Abstract*— We want to extend the approximation of the error probability via a scaling approach from the BEC to general binary-input memoryless output-symmetric (BMS) channels. In particular, we consider such scaling laws for regular LDPC ensembles and message-passing (MP) decoders with a finite number of messages.**

We first show how to re-derive the scaling law for transmission over the BEC using an "EXIT-like" curve instead of the density evolution curve of the peeling decoder. The advantage of the new derivation is that the new expression of the scaling parameter α only contains quantities that can be meaningfully interpreted also for general message-passing algorithms. In particular, this expression only depends on the curvature of the EXIT-like curve as well as the variance of the messages, both taken at the critical channel parameter.

We discuss how to compute these quantities for general MP algorithms and we evaluate the expressions for the specific cases of the Gallager algorithm A as well as the Decoder with Erasures and compare the resulting predictions on the error probability with simulation results.

I. Introduction

Sparse graph codes under iterative decoding are of interest because of their ability to closely approach the channel capacity. The *asymptotic* behavior of such codes is relatively well understood. For practical applications, however, one would like to find the best code of a *fixed length*. Unfortunately, codes optimized for the asymptotic case are in general not optimal for the finite-length case. Indeed, the speed of convergence to the asymptotic limit can significantly vary from one ensemble to another.

We are thus interested in finding a good finite-length approximation which can be used as an efficient finite-length optimization tool in order to find the code best suited for any given application. To that end our approach needs to be as flexible as possible, applicable to the large variety of ensembles, channels, and message-passing decoders that are of interest for real applications.

Scaling laws seem to be a good tool in order to solve this problem. They have been widely used in statistical physics (a good source on this topic is the book by Fisher [4]). The basic idea of scaling laws applied to coding theory (as introduced by Montanari [6]) is that all finite-length error probability curves are up to higher order error terms scaled versions of a single mother curve $f(z)$. The exact scaling law for transmission over the BEC was characterized by Amraoui et. al. in [1]. Further, in [2] one can find explicit analytic expressions of the scaling parameters. We are interested in extending the known results for the BEC to general channels. We are faced with several hurdles. Both the proof of the scaling law as well as the derivation of the scaling parameters in the case of the BEC are strongly linked to the so called *peeling* decoder, originally introduced by Luby et. al. [5]. But unfortunately no generalization of the peeling decoder to general channels is known. As a first step we therefore show an alternative way of computing the scaling parameters for the BEC: this alternative derivation uses EXIT-like functions (see Section II), a concept that readily extends to the general case. The new formula can be meaningfully interpretation not only for he BEC but for any BMS channel and any message-passing decoder. This leads to a conjecture for the scaling parameter for the general case. We then show how to perform the computation of the scaling parameter for the general case and conclude with some applications.

II. EXIT-Like Curves

Consider a generic message-passing (MP) decoder with a finite message alphabet. Without loss of generality we assume that the messages take values in $\{-m, \cdots, m\}$, where $m \in \mathbb{N}$. We assume that the MP decoder fulfills the standard symmetry conditions both at the variable nodes as well as the check nodes. We do allow the MP decoding rules to be time dependent for a finite number of steps before they settle on a fixed rule. Finally, we assume that transmission takes place over a family of BMS channels and that the initial quantization of the received values at the decoder is done in a symmetric fashion as well so that we can make the all-one codeword assumption.

Let the channel be parameterized by $\mathtt{h}$, the entropy of the channel. For every $\mathtt{h} \in [0,1]$, initialize the density evolution process with (a suitably quantized version of) the density $\mathsf{a}_{\text{BMS}(\mathtt{h})}$. Let $\underline{x}$ denote the fixed-point density of the messages at the output of the variable nodes. Note that $\underline{x}$ encodes a probability mass function. To derive from this family of fixed point pairs an EXIT-like curve we can e.g. plot $(\mathtt{h}, 1 - \underline{x}_m)$.

In words, we measure the fraction of messages which are not "m"-messages at the fixed point as a function of the channel parameter. Since below the threshold $1-\underline{x}_m$ is for most MP decoders equal to zero, whereas above the threshold it will have a non-zero value this will give us a curve which is somewhat reminiscent of a standard EXIT curve (of the overall code). This also explains why we call it an "EXIT-like" curve. Our language below will reflect this particular choice but many other choices are possible and sometimes more convenient. Let x denote our choice. Consider the EXIT-like curve depicted in Figure 1. It is given by

$$x(\mathtt{h}) = \begin{cases} (\mathtt{h},0), & \mathtt{h}\in[0,\mathtt{h}^\star) \\ (\mathtt{h}(x),x), & \mathtt{h}\in(\mathtt{h}^\star,1]\leftrightarrow x\in(x^\star,1] \end{cases} \tag{1}$$

where $(\mathtt{h}^\star, x^\star)$ is the fixed-point pair at the threshold. In order for an EXIT-like curve to be suitable for our derivation it must have the following property: at the threshold, the curve must have an infinite slope and the second derivative must be strictly non-zero. Although this is not easy to glean from the picture, this is true for the specific curve shown. We will say that an EXIT-like curve is *regular* if it has the above property.

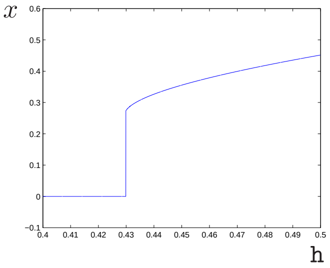

Fig. 1. EXIT-like curve for the (3,6)-code over the BEC: $\mathtt{h}^\star \approx 0.42944$.

III. New Derivation of Scaling Law

Consider transmission over the BEC. Imagine the following experiment. We start with a randomly chosen graph from the ensemble and all bits erased. Choose a random bit and reveal it to the decoder. Next run the iterative decoding process until it is stuck and the decoder hits a fixed point. This gives rise to one fixed point pair (H, X), where H equals the number of not yet revealed variables and where X is equal to the number of still erased messages. We continue now in this fashion, each time choosing at random a new bit which we then reveal and running the decoder until it is stuck. This gives us a sequence of fixed point pairs (H_t, X_t). If we normalize these random variables (H_t to n, the length of the ensemble, and X_t to $n\Lambda'(1)$, the number of edges) and connect the normalized points then we get a "curve." If we increase the length of the ensemble and plot such random curves, then we observe a concentration and it is therefore meaningful to define the average such curve as our EXIT-like curve. Its exact description is easy to write down in terms of density evolution quantities. Of course, for any finite length the individual instances exhibit some variation around this expected curve.

Let $\hat{X}_{\mathtt{h}} = X_{\mathtt{h}}/(n\Lambda'(1))$ denote the random variable which we get if we look at X_t at the time when $H_t = \mathtt{h}n$. In a similar way, let $\hat{H}_x = H_x/n$ denote the random variable which we get if we look at H_t at the *first* time when $X_t = xn\Lambda'(1)$. Imagine that we are sitting just above the threshold at $\mathtt{h} = \mathtt{h}^\star + \Delta\mathtt{h}$. This corresponds to the parameter $x = x^\star + \sqrt{\Delta\mathtt{h}}c$, due to the assumption that the EXIT-like curve is regular. Note that $\mathbb{E}[\hat{H}_x] = \mathtt{h}^\star + \Delta\mathtt{h}$. Let $\sigma_{\mathtt{h}}^2 = \mathbb{E}[(\hat{H}_x - \mathbb{E}[\hat{H}_x])^2]$ and note that $\sigma_{\mathtt{h}}$ is a continuous function of the channel parameter so that we can consider $\sigma_{\mathtt{h}^\star} = \lim_{\mathtt{h}\to\mathtt{h}^\star}\sigma_{\mathtt{h}}$. Consider again the decoding process as described above, which gives rise to the sequence of random variable (H_t, X_t). Assume that we measure the X-component of this process for a particular instance. We claim that if $\hat{X}_t < x^\star$ then with high probability the decoder for this instance will finish without revealing any further bits. On the other hand, if $\hat{X}_t > x^\star$ then with high probability further revelations of bits will be necessary. This is due to the shape of the EXIT-like curve: around the point $x^\star$ the curve has an infinite slope, indicating that the decoder is at the brink of successful decoding with the given information. Hence, if $\hat{H}_{x^\star} < \mathtt{h}$ then the given instance would not have decoded successfully with high probability if we had transmitted over a channel with parameter $\mathtt{h}$, and conversely it would have been successful with high probability if $\hat{H}_{x^\star} > \mathtt{h}$. Our probability of error estimate is therefore

$$P_B(\mathtt{h}) = \mathbb{P}\{\hat{H}_{x^\star} < \mathtt{h}\} = Q\left(\frac{\Delta\mathtt{h}}{\sigma_{\mathtt{h}^\star}}\right).$$

In the last step we have assumed that the distribution of $\hat{H}_{x^\star}$ is Gaussian with mean $\mathtt{h}^\star$ and standard deviation $\sigma_{\mathtt{h}^\star}$. This was shown to hold in the original derivation in [1].

In general it is not an easy task to compute $\sigma_{\mathtt{h}^\star}$ directly. But we can relate it to the variance of the messages which is amenable to an analysis. Define $\mathcal{V} = \frac{\mathbb{E}[(X_{\mathtt{h}} - \mathbb{E}[X_{\mathtt{h}}])^2]}{n\Lambda'(1)}$. Then some calculations show that

$$\begin{aligned} \sigma_{\mathtt{h}^\star}^2 &= \frac{1}{n\Lambda'(1)}\left(\frac{\partial^2\mathtt{h}(x)}{\partial x^2}|_\star\right)^2 \lim_{x\to x^\star}(x-x^\star)^2\mathcal{V} \\ &= \frac{1}{n\Lambda'(1)}\left(\frac{\partial^2\mathtt{h}(x)}{\partial x^2}|_\star\right)^2 \lim_{x\to x^\star}\left(\frac{(x-x^\star)}{1-\gamma\lambda_2(x)}\right)^2\xi. \end{aligned}$$

The last step warrants some remarks. When $x \to x^\star$, i.e., when the channel parameter approaches the critical parameter, then the variance $\mathcal{V}$ diverges. Indeed we will show in Section IV that $\mathcal{V}$ has the form

$$\frac{\xi}{(1-\gamma\lambda_2(x))^2} + O\left(\frac{1}{1-\gamma\lambda_2(x)}\right),$$

where $\gamma := (\mathtt{l}-1)(\mathtt{r}-1)$ only depends on the degree distribution and where $\lambda_2(x)$ is a function so that $\lim_{x\to x^\star}\gamma\lambda_2(x) = 1$. (As we will see later, λ_2 is the second largest eigenvalue of a matrix related to the density evolution process.)

Now we are able to conjecture the following scaling law.

Conjecture 1: Consider a BMS($\mathtt{h}$) and an EXIT-like curve which is *regular*. Then

$$P_B \simeq Q\left(\frac{\sqrt{n}(\mathtt{h}-\mathtt{h}^\star)}{\alpha}\right), \tag{2}$$

where $\alpha = \frac{\partial^2 \mathtt{h}(x)}{\partial x^2}|_\star \lim_{x\to x^\star} \frac{(x-x^\star)}{1-\gamma\lambda_2(x)} \sqrt{\frac{\xi}{\Lambda'(1)}}$.

Therefore, in order to compute α, we need to compute $\frac{\partial^2 h(x)}{\partial x^2}|_\star$, $\lim_{x\to x^\star} \frac{(x-x^\star)}{1-\gamma\lambda_2(x)}$, as well as ξ. The most difficult quantity is the last one. We will therefore focus our attention on its derivation.

IV. Computation of α

As we have seen in the previous section, in order to compute the scaling parameter we have to compute the message-variance on the graph. For the sequel it is notationally slightly easier to take $x = \underline{x}_m$ (instead of $x = 1-\underline{x}_m$). As we pointed out before, the final result is the same. More precisely, let

$$\mathcal{V}_n^\ell = \frac{\mathbb{E}[(X_{\mathtt{h}}^\ell - \mathbb{E}[X_{\mathtt{h}}^\ell)])^2]}{n\Lambda'(1)},$$

where n indicates the length of the code and ℓ the number of iterations we perform. We want to determine the constant ξ where

$$\xi = \lim_{x\to x^*} \lim_{\ell\to\infty} \lim_{n\to\infty} \mathcal{V}_n^\ell (1-\gamma\lambda_2(x))^2.$$

Let us start by discussing how to compute $\mathcal{V}_n^\ell$. First note that we can express $X_{\mathtt{h}}^\ell$ as $X_{\mathtt{h}}^\ell = \sum_{i=1}^{n\Lambda\prime(1)} \mathbb{1}_{\{\mu_i^\ell=m\}}$, where μ_i^ℓ is the variable-to-check message sent at iteration ℓ along edge i. We have

$$\mathcal{V}_n^\ell = \frac{\mathbb{E}[(X_{\mathtt{h}}^\ell - \mathbb{E}[X_{\mathtt{h}}^\ell])^2]}{n\Lambda'(1)} = \sum_i \left(\mathbb{P}\{\mu_1^\ell = m, \mu_i^\ell = m\} - x^2\right).$$

Order all edges according to their distance from edge 1. Here we say that two variable nodes have distance l if there is a path connecting them which contains exactly $l-1$ intermediate variable nodes. The correlation is an exponentially decreasing function in the distance. On the other hand, the number of pairs of edge (emanating from a variable node) that have distance l and that face in the opposite direction increases like γ^l, where $\gamma = (\mathtt{l}-1)(\mathtt{r}-1)$. If the exponential decrease in the correlation is faster than γ^{-l} then the total contribution to the correlation is dominated by pairs that are close. Indeed this is what happens above the threshold. As we approach the threshold from above, however, the correlation extends further and further and exactly at the threshold the two exponents are equal.

More precisely, consider the chain of length l depicted in Figure 2 consisting of an alternating sequence of variable and check nodes. We label, from left to right, the variable nodes from 0 to l and the check nodes from $\hat{1}$ to $\hat{l}$. For the messages in the chain, we use the notation $\mu_{i\to\hat{i}}^t$, where the subscript corresponds to the edge and the direction of the message and the superscript corresponds to its time instance. Further, we denote by ν_i^t ($\nu_{\hat{i}}^t$) the message stemming from the $\mathtt{l}-2$ ($\mathtt{r}-2$) *remaining* incoming messages at the variable (check) node i ($\hat{i}$) at time t. Let us say that edge 1 corresponds to what in the figure is denoted as $\hat{0} \leftarrow 0$. We are interested in the correlation of the message on this edge (which is denoted by $\mu_{\hat{0}\leftarrow 0}$ with the message which is sent on edge $l \to \widehat{l+1}$ (which is denoted by $\mu_{l\to\widehat{l+1}}$). These are pairs of edges which face in the "opposite" direction. In the same way we also have pairs of edges which face in the "same" direction. A detailed calculation shows that the dominant contribution stems from pairs facing in the opposite direction. We will therefore only explain the computation for this case.

As mentioned above. The dominant term contributing to the variance is due to edge pairs which are facing in the opposite direction. This means that as a function of the parameter x, $x > x^*$, $\mathcal{V}$ behaves like

$$(\mathtt{l}-1)\left(\sum_{l=0}^{\infty} \gamma^l (\mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{l\to\widehat{l+1}}^l = m\} - x^2)\right).$$

Consider a chain of even length l. Such a chain has a variable node at position $l/2$ which is exactly in the middle of the chain.

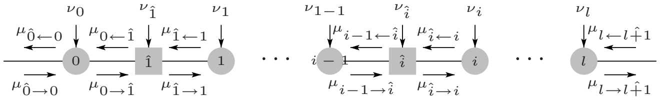

Fig. 2. Chain of alternating variable and check nodes.

Note that $\mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{l\to\widehat{l+1}}^l = m\}$ can be written as

$$\sum_{r,s} \mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{\widehat{l/2}\to l/2}^{l/2} = s \mid \mu_{\widehat{l/2}\leftarrow l/2}^{l/2} = r\}$$
$$\mathbb{P}\{\mu_{l\to\widehat{l+1}}^l = m, \mu_{\widehat{l/2}\leftarrow l/2}^{l/2} = r \mid \mu_{\widehat{l/2}\to l/2}^{l/2} = s\} \qquad (3)$$

where we have used the fact that $\mu_{\widehat{l/2}\to l/2}^{l/2}$ and $\mu_{\widehat{l/1}\leftarrow l/2}^{l/2}$ are independent. As we will see now, each of the two terms which appear in the sum can be computed recursively. We can write $\mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{\hat{i}\to i}^i = s \mid \mu_{\hat{i}\leftarrow i}^{l-i} = r\}$ as

$$\sum_{j,k,u,v} \mathbb{P}\{\mu_{\hat{i}\to i}^i = s \mid \mu_{i-1\to\hat{i}}^{i-1} = j, \nu_{\hat{i}}^{i-1} = u\}$$
$$\mathbb{P}\{\mu_{i-1\leftarrow\hat{i}}^{l-i+1} = k \mid \mu_{\hat{i}\leftarrow i}^{l-i} = r, \nu_{\hat{i}}^{l-i} = v\}\mathbb{P}\{\nu_{\hat{i}}^{i-1} = u, \nu_{\hat{i}}^{l-i} = v\}$$
$$\mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{i-1\to\hat{i}}^{i-1} = j \mid \mu_{i-1\leftarrow\hat{i}}^{l-i+1} = k\}, \qquad (4)$$

and in a similar manner we can express $\mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{i\to\widehat{i+1}}^i = j \mid \mu_{i\leftarrow\widehat{i+1}}^{l-i} = k\}$. Let us define define two vectors of length $(2m+1)^2$,

$$\underline{c}_{j,k}^{l,i} = \mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{\hat{i}\to i}^i = j \mid \mu_{\hat{i}\leftarrow i}^{l-i} = k\},$$
$$\hat{\underline{c}}_{j,k}^{l,i} = \mathbb{P}\{\mu_{\hat{0}\leftarrow 0}^l = m, \mu_{i\to\widehat{i+1}}^i = j \mid \mu_{i\leftarrow\widehat{i+1}}^{l-i} = k\}.$$

Note that by symmetry we also have

$$\underline{c}_{j,k}^{l,i} = \mathbb{P}\{\mu_{l\to\widehat{l+1}}^l = m, \mu_{l-i\leftarrow l-\hat{i}+1}^i \mid \mu_{l-i\to l-\hat{i}+1}^{l-i}\},$$
$$\hat{\underline{c}}_{j,k}^{l,i} = \mathbb{P}\{\mu_{l\to\widehat{l+1}}^l = m, \mu_{\widehat{l-i}\leftarrow l-i}^i \mid \mu_{\widehat{l-i}\to l-i}^{l-i}\}.$$

Further, define the matrices

$$\hat{\mathrm{C}}_{(r,s),(j,k)}^{l,i} = \sum_{u,v} \mathbb{P}\{\mu_{\hat{i}\to i}^i = s \mid \mu_{i-1\to\hat{i}}^{i-1} = j, \nu_{\hat{i}}^{i-1} = u\}$$
$$\mathbb{P}\{\mu_{i-1\leftarrow\hat{i}}^{l-i+1} = k \mid \mu_{\hat{i}\leftarrow i}^{l-i} = r, \nu_{\hat{i}}^{l-i} = v\}$$
$$\mathbb{P}\{\nu_{\hat{i}}^{i-1} = u, \nu_{\hat{i}}^{l-i} = v\}$$

$$\begin{aligned}\mathrm{C}^{l,i}_{(j,k),(r,s)} &= \sum_{u,v} \mathbb{P}\{\mu^{i}_{i\to i\hat{+}1} = j \mid \mu^{i}_{\hat{i}\to i} = s, \nu^{i}_{i} = u\} \\ &\qquad \mathbb{P}\{\mu^{l-i}_{\hat{i}\leftarrow i} = r \mid \mu^{l-i}_{i\leftarrow \hat{i}} = k, \nu^{l-i}_{i} = v\} \\ &\qquad \mathbb{P}\{\nu^{i}_{i} = u, \nu^{l-i}_{i} = v\}\end{aligned}$$

This gives $\underline{c}^{l,i} = \hat{\mathrm{C}}^{l,i}\underline{\hat{c}}^{l,i-1} = \hat{\mathrm{C}}^{l,i}\mathrm{C}^{l,i-1}\underline{c}^{l,i-1}$. The initial condition is $\underline{c}^{l,0}_{j,k} = \underline{y}_{j}$ if $k = m$ and 0, otherwise. We can now rewrite (3) as

$$\mathbb{P}\{\mu^{l}_{\hat{0}\leftarrow 0} = m, \mu^{l}_{l\to l\hat{+}1} = m\} = \sum_{r,s} \underline{c}^{l,l/2}_{s,r}\ \underline{\hat{c}}^{l,l/2}_{r,s}$$

$$= \left(\prod_{i=1}^{2/l} \hat{\mathrm{C}}^{l,i}\mathrm{C}^{l,i-1}(\underline{c}^{l,0})^{T}\right)^{T} \mathrm{F}\mathrm{C}^{l,l/2} \left(\prod_{i=1}^{2/l} \hat{\mathrm{C}}^{l,i}\mathrm{C}^{l,i-1}(\underline{c}^{l,0})^{T}\right)$$

where F is a $(2m+1)^2 \times (2m+1)^2$ permutation matrix which switches the order of the two indices. More precisely, $\mathrm{F}_{(i,j),(k,l)} = 1$ if $i = l$ and $j = k$ and 0 otherwise.

The above formula still looks quite complicated. Fortunately, it can be substantially simplified as follows. Let us look at the quantities that appear in the entries of the matrix $\mathrm{C}^{\hat{l},i}$. $\mathbb{P}\{\mu^{i}_{\hat{i}\to i} = s \mid \mu^{i-1}_{i-1\to \hat{i}} = j, \nu^{i-1}_{\hat{i}} = u\}$ is either 1 or 0 if the messages s, j and u satisfy the check node rule or not. The same is true for $\mathbb{P}\{\mu^{l-i+1}_{i-1\leftarrow \hat{i}} = k \mid \mu^{l-i}_{\hat{i}\leftarrow i} = r, \nu^{l-i}_{\hat{i}} = v\}$. There remains $\mathbb{P}\{\nu^{i-1}_{\hat{i}} = u, \nu^{l-i}_{\hat{i}} = v\}$. If $l = 2i$, $\mathbb{P}\{\nu^{i-1}_{\hat{i}} = u, \nu^{l-i}_{\hat{i}} = v\} = \mathbb{P}\{\nu^{i-1}_{\hat{i}} = u\}$ if $u = v$ and 0 otherwise. What happen if $l \neq 2i$? Contrary to the BEC, for a general discrete MP algorithm, the messages are not stable at the fixed point. More precisely, the expected number of messages that take on a particular value converges but the individual messages continue to flip. This is *not* a consequence of loops in the system but this phenomena also appears on the infinite tree. Consider a infinite support tree. We want to determine the joint probability that a message on a given edge is equal to i at a first time instance and equal to j at a second time instance. Happily it turns out that this joint probability does not depend on the two time instances (as long as they are distinct). So let us define $p(i,j)$ and $q(i,j)$ to be this joint probability for variable-to-check and check-to-variable messages respectively at the critical point. Assume we knew $p(i,j)$. Then it is simple to compute $q(i,j)$ using the standard message-passing rules. The equivalent statement is true if we knew $q(i,j)$ and wanted to compute $p(i,j)$. It follows that p and q can be determined as the solution to a fixed point recursion. Assume therefore that we have determined $p(i,j)$ and $q(i,j)$. By knowing $p(i,j)$, we can compute $g(u,v) = \mathbb{P}\{\nu^{t_1}_{\hat{i}} = u, \nu^{t_2}_{\hat{i}} = v\}$. Similarly we can compute $f(u,v) = \mathbb{P}\{\nu^{t_1}_{i} = u, \nu^{t_2}_{i} = v\}$ if we know $q(i,j)$. Note that f and g depend neither on the time instances t_1 and t_2 nor on the position i. So we can simplify to

$$\mathrm{C}^{l,i} = \left\{\begin{array}{cc}\mathrm{D} & \text{if } l = 2i \\ \mathrm{C} & \text{otherwise}\end{array}\right. \qquad \hat{\mathrm{C}}^{l,i} = \left\{\begin{array}{cc}\hat{\mathrm{D}} & \text{if } l = 2i-1 \\ \hat{\mathrm{C}} & \text{otherwise}\end{array}\right.$$

Define $\mathrm{M} = \hat{\mathrm{C}}\mathrm{C}$. Then we have

$$\mathbb{P}\{\mu^{l}_{\hat{0}\leftarrow 0} = m, \mu^{l}_{l\to l\hat{+}1} = m\} = \underline{c}^{l,0}(\mathrm{M}^{T})^{l/2}\mathrm{F}\mathrm{D}\mathrm{M}^{l/2}(\underline{c}^{l,0})^{T}.$$

So far we have only considered the case of even l. But a similar derivation for odd l shows that

$$\begin{aligned}&\mathbb{P}\{\mu^{l}_{\hat{0}\leftarrow 0} = m, \mu^{l}_{l\to l\hat{+}1} = m\} \\ &= \underline{c}^{l,0}(\mathrm{M}^{T})^{(l-1)/2}\mathrm{C}^{T}\mathrm{F}\hat{\mathrm{D}}\mathrm{C}\mathrm{M}^{(l-1)/2}(\underline{c}^{l,0})^{T}.\end{aligned}$$

It is convenient to treat both cases together. Define the matrix $\mathrm{K} = \mathrm{F}\mathrm{D} + \gamma\mathrm{C}^{T}\mathrm{F}\hat{\mathrm{D}}\mathrm{C}$. We note that K is always a symmetric matrix. Then the correlation for length $2l$ and $2l+1$ together are given by

$$(\mathtt{1}-1)\gamma^{2l}\left(\underline{c}^{l,0}(\mathrm{M}^{T})^{l}\mathrm{K}\mathrm{M}^{l}(\underline{c}^{l,0})^{T} - x^{2}(1+\gamma)\right), \qquad (5)$$

where the factor $1+\gamma$ appears since we are looking at two lengths simultaneously and the second one has an extra factor γ.

Let $\lambda_1 \geq \cdots \geq \lambda_{2m+1}$ be the eigenvalues of M and $e_1, \ldots, e_{2m+1}$ the (generalized) eigenvectors. Since M describes the evolution of a (conditional) probability we have $\lambda_1 = 1$. We will see shortly that the contribution which stems from this eigenvalue will cancel with the term proportional to x^2.

The term that is of interest to us is associated to the second eigenvalue λ_2. Generically this second eigenvalue must be degenerated, i.e., it must have geometric multiplicity larger than one. The most common case is that λ_2 has multiplicity 2 but only one associated eigenvector. In this case e_2 is the associated eigenvector and e_3 is the associate generalized eigenvector which fulfills the equation $(M - \lambda_2 I)e_3^T = e_2^T$. To be specific, assume for the following that indeed λ_2 is a degenerated eigenvalue of multiplicity two and that λ_4 to λ_{2m+1} are of multiplicity one.

Let us expand out the initial condition in terms of the eigenvectors of the matrix M. Then we have

$$\mathrm{M}^{l}(\underline{c}^{l,0})^{T} = \mathrm{M}^{l}\sum_{i=1}^{2m+1} c_i e_i^T = \sum_{i=1}^{2m+1} c_i\lambda_i^l e_i^T + lc_3\lambda_2^{l-1}e_2^T.$$

So we can write (5) as

$$\begin{aligned}&(\mathtt{1}-1)\gamma^{2l}\left(\left(\sum_{j=1}^{2m+1} c_j\lambda_j^l e_j^T + lc_3\lambda_2^{l-1}e_2^T\right)^{T}\mathrm{K}\right. \\ &\left.\left(\sum_{i=1}^{2m+1} c_i\lambda_i^l e_i^T + lc_3\lambda_2^{l-1}e_2^T\right) - x^2(1+\gamma)\right). \qquad (6)\end{aligned}$$

In principle this gives $(2m+1)^2$ terms but most of them are either zero or are of smaller order than the dominant term. E.g., it must be true that

$$c_1^2 e_1 \mathrm{K} e_1^T = x^2(1+\gamma),$$

since otherwise the variance would be infinite. Indeed, this condition is true for the examples we present in the next section. Also recall that by our assumption the EXIT-like function was regular. Some thought shows that this implies that $\mathcal{V}$ must grow inversely like $(1-\gamma\lambda_2(x))^2$. This in turn implies that for a given l the dominant contribution of all pairs at this distance must behave like $l\gamma^l\lambda_2(x)^l$. It follows that $e_1\mathrm{K}e_i^T = 0$ for all $i > 1$ since otherwise we would have a term with a growth rate larger than what we need. In a similar manner we must have $e_2\mathrm{K}e_2^T = 0$ since such a term would give a contribution of the form $l^2\gamma^l\lambda_2(x)^l$, again implying a too large growth rate. All these conditions can be checked for a given case and they are fulfilled for the examples we present below.

The dominant term is of the form $e_2\mathrm{K}e_3^T$. It gives a contribution of the correct form $l\gamma^l\lambda_2(x)^l$ and there are exactly two of them. If we insert this into (6) then we get

$$\mathcal{V} = \lim_{x\to x^\star} \frac{(\mathtt{1}-1)c_3^2}{2\lambda_2} e_2\mathrm{K}e_3^T \frac{1}{(1-\gamma\lambda_2)^2}\ (1+O(x-x^\star)).$$

In the last step we have made use of the fact that $\gamma\lambda_2$ tends to 1 as we approach the threshold. Therefore the sought after constant equals $\xi = (\mathtt{1}-1)c_3^2 e_2\mathrm{K}e_3^T/(2\lambda_2)$. Due to numerical issues, it is in general not easy to compute the (generalized) eigenvectors of $\lim_{x\to x^\star}\mathrm{M}$. We can avoid this by computing ξ in the following way:

$$\xi = \frac{(\mathtt{1}-1)\underline{c}^{l,0}\mathrm{K}\prod_{i=1,i\neq 3}^{2m+1}(\mathrm{M}-\lambda_i\mathrm{I})(\underline{c}^{l,0})^T}{2\lambda_2\prod_{i=1,i\neq 2,3}^{2m+1}(\lambda_2-\lambda_i)}.$$

We can also use the following technique in order to approximate ξ. First, note that for increasing l we have

$$(\mathrm{M}-\mathrm{I}\lambda_2)\mathrm{M}^l c^T \approx (1-\lambda_2)c_1 e_1^T + c_3\lambda_2^l e_2^T.$$

Since the multiplication with M kills the contribution of the all terms $\mu_i e_i^T$, $i \geq 4$. Therefore,

$$\xi \approx \frac{(\mathtt{1}-1)}{2\lambda_2^{l+1}(1-\lambda_2)}\underline{c}^{l,0}\mathrm{K}(\mathrm{M}-\mathrm{I})(\mathrm{M}\ \lambda_2\mathrm{I})\mathrm{M}^l(\underline{c}^{l,0})^T. \qquad (7)$$

This approximation is convenient, since it only requires quantities which are easily computable at the threshold. In particular, we do not need to compute the complete set of (generalized) eigenvectors and we do not even need to compute any eigenvalues except λ_2.

V. Applications

A. Gallager Algorithm A

Consider transmission over a BSC and decoding using Gallager's algorithm A [7]. As discussed in [3] the threshold of most regular ensembles for this case is determined by a fixed point right at the beginning of the decoding process. A derivation of the scaling parameter for this case is contained in [3]. For the $(3,3)$-regular ensemble the threshold is determined by a "regular" fixed point. Further, one can check that in this case the EXIT-like function has the required shape (with a negative second derivative at the critical point). The threshold is $\epsilon^\star \approx 0.223$. If we specialize the generic computation of α to this case we get $\xi \approx 0.0942$ and $\alpha \approx 0.7847$. Figure 3 compares our prediction with simulations. The match is quite good but we would like to add a word of caution: for this case the codes exhibit a significant error floor. Since the scaling law only applies to large scale failures we consider expurgated ensembles. For the lengths considered, the separation between which errors are due to the error floor and which are part of large scale errors is not evident. Therefore, in order to truly gage the quality of the approximation we first have to find the contribution of the error floor and to compare the total curve.

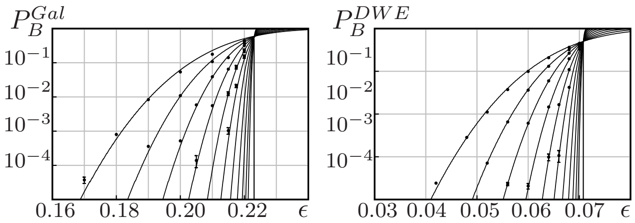

Fig. 3. Left: Block error probability of a (3,3)-code decoded with Gallager A algorithm: $\epsilon^{\mathrm{Gal}} \approx 0.2230$. Comparison of the block error probability determined via simulations with the prediction given by the scaling law with $\alpha \approx 0.7847$. The lengths are n = 1024, 2048, 4096, 8192, 16384 and 32768. Right: Block error probability of a (3,6)-code decoded with the decoder with erasures: $\epsilon^\star = 0.07076$. Comparison of the block error probability determined via simulations with the prediction given by the scaling law with $\alpha = 1.04 \pm 2$. The lengths are n = 1024, 2048, 4096, 8192, 16384 and 32768.

B. Decoder With Erasures

We consider transmission over a BSC and decoding using the Decoder with Erasures [7]. For the $(3,6)$-regular ensemble and the weight sequence $w(1) = 2,\ w(i) = 1$ if $i > 1$, one can check that the EXIT-like curve has the required form to apply our computation. The threshold is $\epsilon^\star \approx 0.07076$. Our computation gives $\xi \approx 0.01313$ and $\alpha \approx 1.04 \pm 2$. Figure 3 compares the prediction on the block error probability with some simulation points. The match is again quite good.

Acknowledgment

J. E. has been supported in part by the NCCR-MICS Center of the Swiss National Science Foundation under grant number 5005-67322.

References

[1] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, *Finite-length scaling for iteratively decoded LDPC ensembles*, in Proc. 41th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, 2003.

[2] A. Amraoui, A. Montanari, and R. Urbanke, *Finite-length optimization of iteratively decoded LDPC ensembles.* submitted to IEEE IT, 2005.

[3] J. Ezri, A. Montanari, and R. Urbanke, *Finite-length scaling for Gallager a*, in 44th Allerton Conf. on Communication, Control, and Computing, Monticello, IL, Oct. 2006.

[4] M. E. Fisher, *Proceedings of the Enrico Fermi school, Varenna, Italy, 1970, course n. 51*, in Critical Phenomena, International School of Physics Enrico Fermi, Course LI, edited by M. S. Green, (Academic, New York, 1971), 1971.

[5] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann, *Practical loss-resilient codes*, in Proceedings of the 29th annual ACM Symposium on Theory of Computing, 1997, pp. 150–159.

[6] A. Montanari, *Finite-size scaling of good codes*, in Proc. 39th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, 2001.

[7] T. Richardson and R. Urbanke, *The capacity of low-density parity check codes under message-passing decoding*, IEEE Trans. Inform. Theory, 47 (2001), pp. 599–618.