On the equivalence of Ising models on ‘small-world’ networks and LDPC codes on channels with memory

Izaak Neri and Nikos S Skantzos

Instituut voor Theoretische Fysica, Katholieke Universiteit Leuven, Celestijnenlaan 200D, B-3001 Leuven, Belgium
E-mail: izaakneri@gmail.com and nikos.skantzos@gmail.com

Received 18 February 2014, revised 13 July 2014
Accepted for publication 22 July 2014
Published 5 September 2014

Abstract
We demonstrate the equivalence between thermodynamic observables of Ising spin-glass models on small-world lattices and the decoding properties of error-correcting low-density parity-check codes on channels with memory. In particular, the self-consistent equations for the effective field distributions in the spin-glass model within the replica symmetric ansatz are equivalent to the density evolution equations for Gilbert–Elliott channels. This relationship allows us to present a belief-propagation decoding algorithm for finite-state Markov channels and to compute its performance at infinite block lengths from the density evolution equations. We show that loss of reliable communication corresponds to a first order phase transition from a ferromagnetic phase to a paramagnetic phase in the spin glass model. The critical noise levels derived for Gilbert–Elliott channels are in very good agreement with existing results in coding theory. Furthermore, we use our analysis to derive critical noise levels for channels with both memory and asymmetry in the noise. The resulting phase diagram shows that the combination of asymmetry and memory in the channel allows for high critical noise levels: in particular, we show that successful decoding is possible at any noise level of the bad channel when the good channel is good enough. Theoretical results at infinite block lengths using density evolution equations are compared with average error probabilities calculated from a practical implementation of the corresponding decoding algorithms at finite block lengths.

Keywords: spin glasses, error correcting codes, replicated transfer matrices, replica method
PACS numbers: 84.40.Ua, 89.70.-a, 02.50.-r
1. Introduction

The use of networks and graphical models has been a very successful tool for the analysis of complex systems in many different disciplines such as biology, sociology, etc [1–3]. Among the various network structures small-worlds play an important role. They allow for both a small distance between nodes and a high clustering coefficient, as observed in much real-world data [4]. Small worlds can be constructed by superimposing a one-dimensional ring structure on a random Erdös–Renyi graph [5].

Ising models on small world networks interpolate between a non-mean field behaviour (imposed by the geometry) and a mean-field behaviour (given by random interactions) [6, 7]. Nikoletopoulos et al developed an analytical method to assess the thermodynamic behaviour of Ising models on small world networks [8, 9]. This is achieved using a replicated transfer matrix method, which combines the transfer matrix method developed to solve one-dimensional (and two-dimensional) Ising models [10] with the replica method for random graphs [11, 12]. The solution of the latter is based on deriving a self-consistent equation of the effective field distribution which, upon solving, determines the thermodynamic free energy and thus thermodynamic observables of the system.

The theory of random graphs appears naturally in the context of error-correcting codes. A person A will encode a message by including redundant information with it. This redundant information consists of a string of additional bits attached to the original message describing the result of bit-wise arithmetic operations among bits of the original message (the ‘parity-checks’). The entire message is then sent through a noisy channel and reaches a person B. By checking the validity of the designated arithmetic operations on the parity checks person B can infer which bits have been corrupted by noise, if the noise level is not too high. The link to random graphs is provided by representing the message bits as nodes on a graph while the parity checks between the message bits are described as edges on this graph. Statistical physics can be used to assess typical properties such as noise thresholds or error-exponents of various encoding–decoding schemes (see for example [13–21] and for a review see [22]).

Given a certain noisy channel, the theoretical maximum rate of information one can send reliably through this channel is known and is given by the so-called ‘capacity’ of the channel (the Shannon limit). Codes approaching channel capacity are therefore, in this perspective, the best one can expect. In 1996 it was discovered that practical and efficient codes exist that can reach this upper limit [23]. These use an encoding–decoding scheme introduced by Gallager [24]. This family of codes, called low-density parity-check codes (LDPC) or Gallager codes, can be represented by a sparse (random) graph (the ‘Tanner’ graph) and is characterized by the distribution of the number of bits per parity-check and the distribution of the number of parity-checks per bit. By optimizing the encoding rate of the code to these degree distributions it has been shown that Shannon’s limit can be reached on a binary erasure channel without memory [25]. It is in general accepted that LDPC can reach the Shannon limit on more general channels such as the binary symmetric channel [26].

In more general situations, the channel noise during subsequent uses of the channel is autocorrelated leading to channels with memory. In [27] Elliott extended a channel introduced by Gilbert [28] to model burst-noise in wireline telephone circuits. The Gilbert–Elliott channel is a specific example of a finite-state Markov channel [29] where the noise depends on the state of the channel at the moment it is used. The state of the channel represents an additional degree of freedom and evolves according to a Markov process. The Gilbert–Elliott
channel can be in two states, both corresponding to a binary symmetric channel, but with different noise levels. Nowadays channels with memory are a topic of active research. They are used in fading channels modelling mobile radio communication; see [30] for a tutorial.

In this work, our purpose is to show the connection between Ising models on small-world networks and low-density parity-check codes on channels with memory. The latter have been studied using belief propagation in [33, 40]. As we show here, the probability of inferring a certain message can be denoted by a Boltzmann distribution of a spin model on a small world hypergraph. The random encoding scheme imposes the random interactions between the message bit-variables (these are the long range interactions) while the memory in the noisy channel introduces a spatial one-dimensional character to the graph (the short-range interactions); see figure 1. The study of the decoding properties of LDPC codes on channels with memory is therefore equivalent to the study of the thermodynamic observables of an Ising model on a small-world hypergraph [31].

Using the replica method for sparse small-world graphs [6, 7, 31, 32] we derive the self-consistent equations for the effective field distribution of the Ising model corresponding to LDPC codes of finite-state Markov channels. In the replica symmetric limit, these equations reduce to the density evolution equations for finite-state Markov chains. The density evolution equations determine the distribution of messages propagated in a belief propagation decoding algorithm for finite–state Markov-modulated channels with asymmetric noise in the limit of infinitely large code words (i.e. infinite block lengths). We compare the belief propagation algorithm developed in this paper with algorithms for LDPC codes presented before in the literature. For channels with a large amount of asymmetry and memory in the noise, the algorithm developed here allows for probabilities of error which are considerably smaller. We show that belief propagation on finite-state Markov-modulated channels develops a critical noise level at infinite block lengths. Solving the density evolution equations using a population dynamics algorithm [42, 43] (i.e. a Monte Carlo method) we determine the critical noise levels for several symmetric and asymmetric finite-state Markov chains. As a benchmark test, we have compared our results against (i) the special case of symmetric finite-state Markov chains (Gilbert–Elliott channel) [33] and (ii) the memoryless

![Figure 1. The LDPC code for a channel with memory can be mapped on an Ising model with a 'small-world' architecture. The parity checks provide the sparse random long-range interaction structure while the memory in the channel provides the regular short-range interaction structure through a Markov chain. The result is a spin model on a small-world graph.](image-url)
asymmetric channels [16, 34]. In both cases we found excellent agreement. We also compare our results on thresholds for reliable communication with the error probabilities calculated with a direct implementation of the corresponding decoding algorithms.

Note that similar results for symmetric channels have been derived using factor-graph analysis in [49]. Using a replica method Mori and Tanaka have also derived the density evolution equations for finite-state Markov channels [41]. In particular they derive critical noise levels for a generalized erasure channel.

The aim of this paper is twofold. Firstly, to provide a link between the statistical physics theory of small-world graphs and the analysis of LDPC codes with memory. We show how the method of replicated transfer matrices developed for small-world graphs can be applied to error correcting codes. The saddle point equations of the free energy of the spin model determine the decoding properties for any values of the channel and decoding parameters. Secondly, we illustrate how to extract from this analysis new results regarding error-correcting codes on finite-state Markov-chains. An example of the latter can be found in the phase diagram of figure 6; introducing asymmetry and memory in the channel allows successful decoding at arbitrary high noise levels of the bad channel. The main focus of this work is the development of a general methodology for finite-state Markov chains using statistical physics.

Our paper is organized as follows: In the next section we provide the definition used in our analysis both with regards to channels with memory (subsection 2.2) and with regards to LPDC (subsection 2.3). The analysis is given in section 3. We start in subsection 3.1 with a discussion of the analogy between small-world architecture and the decoding process of channels with memory. In subsection 3.2 we provide a technical analysis of the model. In section 4 we provide the density evolution equations while in section 5 we present the corresponding decoding algorithm. Results are presented in the ensuing section 6. In subsection 6.1 we compare the decoding algorithm developed in this manuscript with decoding algorithms for LDPC codes taken from the literature. We show the existence of a critical noise level separating a phase where reliable communication with belief propagation is possible from a phase where reliable communication with belief propagation is not possible. In the next subsection 6.2 we compute the critical noise levels for several channels with both asymmetry and memory in the noise. Phase diagrams of these channels are presented. Our concluding remarks are given in section 7.

2. Definitions and model setup

2.1. Communication through a noisy channel

We consider scenarios in which person A wants to communicate reliably a signal \( \hat{\sigma} \) to person B with \( \hat{\sigma} \in \{-1, 1\}^N \). Prior to its communication over the channel, \( \hat{\sigma} \) is encoded to \( \sigma \in \{-1, 1\}^M \) with \( M > N \). The encoding is defined by the map \( \mathcal{G} : \{-1, 1\}^N \rightarrow \{-1, 1\}^M \) while the elements of the image \( \mathcal{C} \) of \( \mathcal{G} \) are the codewords. The encoding procedure consists in adding redundant bits to the message \( \sigma \) sent.

The output of this transmission will be a stream of bits \( \rho \), which is usually different than \( \sigma \); the difference being due to the fact that in real circumstances, the sent signal is affected by a variety of factors such as temperature fluctuations, atmospheric noise, scratches on a memory disk, etc. In a binary symmetric channel we choose to represent these microscopic processes by a single scalar variable \( p \in [0, 1] \) denoting the probability that a bit in the message \( \sigma \) will flip due to channel noise independently of the other bits. In real applications bits can often consist of different physical units and the noise intensity can depend on whether
σᵢ = 1 or σᵢ = −1. An interesting model is therefore the binary asymmetric channel in which an input bit σᵢ = 1 flips due to channel noise with a probability q while a bit σᵢ = −1 flips with a probability p. In the special case where p = 0 we obtain the so-called Z-channel.

2.2. Channels with memory

In this work we consider channels in which the single bits do not flip independently. The motivation for this model originates from a common problem in modern mobile telecommunication systems, namely that the strength of the signal varies over time as a result of the motion of the receiver with respect to the source and the varying number of obstacles that shadow the signal over time. Channels describing communication of attenuated signals are termed ‘fading channels’. Fading channels are modeled by finite-state Markov channels (FSMC) [29]. These channels have fueled significant research activity (for a review on the subject see [30]). In FSMCs there exist a number of different channel states that correspond to the various possible attenuation factors. Each of the states describes a memoryless channel characterized by an error probability, while the transition from one state to another occurs according to a stationary Markov process. As a practical illustration, a receiver may, due to his motion in space, find that channel noise is very low whenever he is in an open area while it becomes problematic in a closed one. In the first case the channel is more appropriately described as a ‘good’ channel with low flip-probability, while in the latter as a ‘bad’ channel with a high flip-probability. Since there are different states in the FMSC the error-probabilities between subsequent uses of the channel are correlated, i.e. there is memory in the channel.

Memory effects can be modeled using the transformation σ → ρ where the output of the channel ρ ∈ {−1, 1}^M depends on the input through the channel state variable s ∈ {−1, 1}^M:

\[ P(\rho|s, \sigma) = \prod_{n=1}^{M} \left( P_{\text{chan}}(\rho_n|\sigma_n, s_n) \right) P_{\text{state}}(s). \]

The probability of the states P_{\text{state}}(s) follows a Markov process

\[ P_{\text{state}}(s) = P_{\text{state}}(s_1) \prod_{n=1}^{M-1} \mathcal{W}[s_{n+1}|s_n]. \]

Depending on the definition of the Markov process and the channel noise one obtains different FSMCs. We consider the most simple case of two-state Markov-modulated binary channels in which the channel can have two states s ∈ S = \{G, B\}, which usually are referred to as the bad state s = B and the good state s = G. Let us be more concrete, and write the following rather general formula for the noise of two-state Markov-modulated binary channels:

\[ P_{\text{chan}}(\rho_i = −\sigma_i|s_i, \sigma_i) = \begin{cases} q_B, & s_i = B, \sigma_i = 1, \\ p_B, & s_i = B, \sigma_i = −1, \\ q_G, & s_i = G, \sigma_i = 1, \\ p_G, & s_i = G, \sigma_i = −1, \end{cases} \]

for which we take \((p_B + q_B) > (p_G + q_G)\). The Markov process is determined by the transition probability matrix \(\mathcal{W}\) given by
This matrix describes the transition probability from state $s$ to $s'$ as $\mathcal{W}[s'|s] = \mathcal{W}_{s,s'}$. Here $g$ corresponds to the transition probability from state $B$ to $G$, and $b$ the transition probability from state $G$ to $B$ (figure 2). We define the memory $\mu_\ell$ at time step $\ell$ of the Markov process as

$$\mu_\ell = \mathcal{W}[s_\ell = s|s_0 = s] - \mathcal{W}[s_\ell = s|s_0 \neq s].$$

From (4) we find $\mu_\ell = (1 - g - b)^\ell \equiv \mu^\ell$ with the time index $\ell = 1, 2, \ldots$ and $\mu \in [-1, 1]$. For $\mu = 0$ we obtain $b = 1 - g$, hence the two columns of the transition matrix are identical and the channel has no memory. For $\mu > 0$ we have persistent memory: the probability of remaining in a given state is higher than the steady-state probability of being in that state. For $\mu < 0$ we have an oscillatory memory. We also define the good-to-bad ratio $r = \frac{g}{b}$. The FSMCs we consider are determined by the 6-tuple $T = (\mu, r, p_B, q_B, p_G, q_G)$. The Gilbert–Elliott channel [27, 28] corresponds to the subset of symmetric channels $T_{\text{GEC}} = (\mu, r, p_B, p_B, p_G, p_G)$. We also consider asymmetric Markov channels $T_{\text{AS}} = (\mu, r, q_B, q_B, q_B, q_G)$ with $\kappa \in ]0, 1[$ for which the chance to flip a value of one is different than the chance to flip a zero value. We define as well Markov $Z$-channels $T_Z = (\mu, r, 0, q, q, 0)$ (see also [40]). We illustrate the different channels in figure 2.

2.3. Low-density parity-check codes

For memoryless channels low-density parity-check codes (along with Turbo codes) are known to reach close to Shannon’s limit [23–26, 35]. In FSMCs the memory content in the channel results in an increased information capacity, compared to its memoryless counterpart [36]. It is therefore of practical interest to extend Turbo codes and LDPC codes to FSMCs [33, 38].

The set of codewords $C$ of an LDPC code is described by its parity check matrix $H$ through:
\[ C = \left\{ \sigma \in \{-1, 1\}^M | H^* \sigma = 1 \right\} \text{ with } (H^* \sigma)_j = \prod_{j=1}^{M} \sigma_j^H_j, \]

for all \( i = 1, \ldots, M - N \). In LDPC codes \( H_j \) is a sparse, random matrix of dimension \((M - N) \times M\) and with elements \( H_j \in \{0, 1\}\). A \((C,K)\)-regular LDPC-code contains \( K \) non-zero elements per row and \( C \) non-zero elements per column.

### 3. Statistical mechanics analysis

The aim of the statistical mechanics analysis we present is to compute the critical channel noise above which decoding fails. This is achieved by calculating the thermodynamic observables of a related spin model on a hypergraph (which is determined by the parity check matrix as well as the channel).

We start by providing some brief background information on the statistical physics approach to this problem. In the second subsection we derive the free energy of the model which determines the relevant decoding properties.

#### 3.1. The analogy with the thermodynamics of spin glasses on small-world graphs

The decoding problem describing interactions between parity checks and codeword variables can be mapped to large frustrated systems of interacting particles [39]. Using statistical mechanics methods the performance of Gallager, MacKay-Neal and Turbo codes over binary-symmetric, -asymmetric, or real-valued channels have been analyzed; for a review see [22]. The main interest in this approach lies in the generating function \( f \) of the \( a \ posteriori \) probability distribution of codewords \( \sigma \) given the channel’s output \( \rho \) and the parity check matrix \( H \). This is similar to the free energy of spin models with hyperspin interactions as analyzed in [46]. This model contains a ferromagnetic phase, a spin glass phase (also referred to as the suboptimal phase) and a paramagnetic phase. For sufficiently low noise levels the system is in a ferromagnetic phase. When the noise in the system increases, a metastable phase arises, which is the paramagnetic phase. The analogy with error-correcting codes is made upon identifying reliable communication with the ferromagnetic phase. When the suboptimal metastable state (i.e. the paramagnetic phase) appears, it will disturb reliable communication: decoding algorithms will be trapped in the basin of attraction of this metastable state. Increasing the noise even further makes the free energy of the metastable state so low that reliable communication of the coded word using any decoding algorithm becomes impossible. All these decoding limits lie below the Shannon limit and are dependent on the particular code used.

The ‘free energy’ can be calculated with the replica method for finitely connected systems [22], which was originally developed to solve the Viana–Bray model of spin glasses [11, 13, 43]. The free energy is expressed in terms of an order parameter which is a distribution of ‘effective fields’ (with the additional replica symmetric assumption). The typical fraction of correctly decoded bits is given in terms of an integral over this distribution. The statistical physics approach for the computation of the critical noise levels is different from that of the information theoretic approach: the two approaches connect elegantly at the level of the distribution of effective fields which within replica symmetry turns out to be identical to the density evolution that describe the messages propagated in the decoding algorithm (see e.g. [37]).

In the presence of memory in the channel the traditional replica method approach will not work [11, 12, 43]. The main difficulty is illustrated in figure 1. We see that the analogy with spin systems on a random graph with infinite-range interactions is not entirely correct anymore. The system now contains an additional set of degrees of freedom which represents the state of...
the channel (if the state is unchanged we are in a memoryless channel). The transition from one state to another is described by some predefined transition probability. It is exactly this Markovian process that induces nearest-neighbour interactions among our microscopic degrees of freedom corresponding to a small-world architecture in the correlation structure of the bits. This small-world architecture renders a conventional mean-field analysis with the replica method insufficient to study the system. Therefore, we consider here the replicated transfer matrix method which combines the one-dimensional transfer matrix method for one-dimensional spin models with the replica method for spin models on random graphs.

3.2. Application of the replicated transfer matrix method

We illustrate now the required modifications to the replica method. Similar computations can be found in the more abstract works [31, 32] on small-world hyper spin models. For more mathematical details on our derivation we refer to appendix A.

The starting point is the generating function $f$ of the a posteriori probability distribution $P(\sigma | \rho, \mathcal{H})$ of the codeword $\sigma$ given the channel’s output $\rho$ and the parity check matrix $\mathcal{H}$:

$$f(\rho, \mathcal{H}) \equiv -\lim_{M \to \infty} M^{-1} \log \sum_{\sigma} P(\sigma | \rho, \mathcal{H}).$$

(7)

The a posteriori distribution can be used to express the properties of the bitstream received by person B at the end of the channel. We analyze the properties of this distribution in the limit $M \to \infty$.

Using Bayes’ rule and (1) we obtain

$$P(\sigma | \rho, \mathcal{H}) = \frac{P(\rho | \sigma)P(\sigma | \mathcal{H})}{P(\rho | \mathcal{H})} = Z^{-1} P_{\text{init}}(\sigma) \delta_{\mathcal{H} | \sigma} \sum_{s} P_{\text{state}}(s) P_{\text{chan}}(\rho | s, \sigma),$$

(8)

with $P(\sigma | \mathcal{H}) \sim P_{\text{init}}(\sigma) \delta_{\mathcal{H} | \sigma}$ the initial probability distribution of the codewords and $Z$ a normalization constant. We will consider an unbiased source of codewords sent by person A such that $P_{\text{init}}(\sigma) = Z^{-M}$. The distribution $P_{\text{chan}}(\rho | s, \sigma)$ gives the a priori probability distribution of the output $\rho$ given the state vector $s$ and the codeword $\sigma$, equation (3). The Kronecker delta $\delta_{\mathcal{H} | \sigma}$ constrains the summation only to those codewords that obey the parity check equation (6).

Averaging the generating function over the ensemble of parity-check matrices, true-states, true-codewords and outputs gives

$$-\bar{F} = \lim_{M \to \infty} \frac{1}{M} \sum_{\mathcal{H}} \left[ \sum_{\sigma} \delta_{\mathcal{H} | \sigma} \sum_{s} P_{\text{state}}(s) \sum_{\rho} P_{\text{chan}}(\rho | s, \sigma) \right] \times \log \left( \sum_{\sigma, s} P_{\text{state}}(s) \delta_{\mathcal{H} | \sigma} P_{\text{chan}}(\rho | s, \sigma) \right),$$

(9)

plus irrelevant constant terms. The probability distribution of the parity-check matrices $P(\mathcal{H})$ of a $(C, K)$-regular code can be written in terms of a tensor with $K$ indices whose elements are in $\{0, 1\}$. The probability that an element of the tensor is 1 equals $C^{(K-1)M+1}$ and the sum of its elements equals $C$ for all of its indices (see e.g. [16, 22]). We will denote by $\sigma^0, s^0 \in \{ -1, 1 \}^M$ the true codeword and true channel state vectors respectively that were realized during the signal communication. The averaging of the free energy $\bar{F}$ will be done using the replica trick $\langle \log Z \rangle = \lim_{n \to \infty} \frac{1}{n} \log \langle Z^n \rangle$. This means that the spin
variables $\sigma = \{\sigma_1, \ldots, \sigma_M\}$ and $s = \{s_1, \ldots, s_M\}$ will be replicated $n$ times (to $\{\sigma^1, \ldots, \sigma^n\}$ for $\sigma$ and similarly for $s$).

Let us point out that in equation (9) we see the combination of short- and long-range interactions that lead to an equivalent small-world network. Inside the logarithm we find the density $P_{\text{state}}(s)$, which describes one-dimensional nearest-neighbour interactions of the state variables; i.e. it is equal to $P_{\text{state}}(s) = P_{\text{state}}(s_1) \prod_{n=1}^{M-1} W[s_{n+1}/s_n]$. On the other hand, the variables $s$ and $\sigma$ are coupled in a sparse and random fashion via the parity checks imposed by the delta function. In the absence of the channel-state variables the above expression would be seen as a spin system on a mean-field graph.

Performing the summations over $s$, results, for $M \to \infty$, in a saddle point integral whereby the free energy at the saddle point is given by

$$
\tilde{f} = \lim_{n \to 0} \frac{1}{n} \text{extr}_{P, \rho} F\left(P(\sigma, \sigma), \hat{P}(\sigma, \sigma)\right),
$$

with $F$ the exponent of the saddle point integral. The extremization is taken over the order parameter functions $P(\sigma, \sigma)$ and $\hat{P}(\sigma, \sigma)$. These represent the usual order parameter functions describing finite connectivity systems, see for instance [12], with $\sigma = (\sigma^1, \ldots, \sigma^n) \in \{-1, 1\}^n$ originating from the replication of the dynamic codeword-variables while $\sigma \in \{-1, 1\}$ stems from the inclusion of the quenched true codeword in the order function. Note that in order to avoid unnecessary extra notation we denote in what follows by $\sigma$, $s$ the replicated spin variables $\sigma = (\sigma^1, \ldots, \sigma^n)$ and $s = (s^1, \ldots, s^n)$ (and thus the same notation for which we used $\sigma^1, \ldots, \sigma^M$ before). Whenever necessary we will make the site index explicit.

The exponent $F$ reaches a minimum at the values $\left(P(\sigma, \sigma), \hat{P}(\sigma, \sigma)\right)$ that satisfy the saddle point equations:

$$
\hat{P}(\sigma, \sigma) = \prod_{(\sigma_1, \sigma_2, \ldots, \sigma_{K-1})} P(\sigma_1, \sigma_2) \times \delta(\sigma_1 \sigma_2 \cdots \sigma_{K-1} \sigma_1, 1) \prod_{a=1}^{n} \delta(\sigma_a^1 \sigma_a^2 \cdots \sigma_{K-1} \sigma_a, 1),
$$

$$
P(\sigma, \sigma) = \frac{\text{Tr} \left[ V^{N-1}(\hat{P}) Q(\sigma, \sigma; \hat{P}) \right]}{\text{Tr} \left[ V^N(\hat{P}) \right]},
$$

where ‘Tr’ denotes a trace of matrices over the replicated channel state variables $s$, with

$$
\left\langle s, s^0 | Q(\sigma, \sigma; \hat{P}) | s', (s')^0 \right\rangle = \left( \hat{P}(\sigma, \sigma) \right)^{n} \left[ W[ (s^0) | s^0 ] \times \prod_{a=1}^{n} W[ (s')^a | s_a^a ] \right] \left( \prod_{a=1}^{n} P_{\text{chan}}(\rho | s^a, \sigma_a^a) \right)_{\rho|s^0,\sigma}.
$$

For notational economy we introduced in the above the abbreviation over (3):

$$
\langle f(\rho) \rangle_{\rho|s^0,\sigma} = \sum_{\rho} f(\rho) P_{\text{chan}}(\rho | s^0, \sigma). \nonumber
$$

Note that while the summations over the replicated codeword variables $\{s_i\}_{i=1}^M$ have been performed by reducing the graph into a single-site problem, the summations over the replicated channel-state variables $\{s_i\}_{i=1}^M$ are written as a trace over a matrix product (12) (for which the matrix $V$ is raised to the power $M$). This constitutes the key difficulty in our problem as we are dealing with the $(2^n + 1) \times (2^n + 1)$ replicated transfer matrix:
\[ \langle s, s' | V (\tilde{\rho}) | s', (s')' \rangle = \sum_{\sigma, \sigma'} \mathcal{W} [ (s^0)' | s^0] \prod_{\alpha=1}^{\tilde{n}} \mathcal{W} [ (s')^\alpha | s^\alpha] \\
\times \left( \tilde{\rho} (\sigma, \sigma') \right)^C \left( \prod_{\alpha=1}^{\tilde{n}} P_{\text{chan}} (\rho | s^\alpha, \sigma^\alpha) \right) .\]

To proceed further we now have to make an assumption with regard to the structure of the replica space. The simplest, the replica symmetric ansatz, assumes that all replicas \( \alpha = 1, \ldots, n \) are invariant to permutations:

\[ P (\sigma, \sigma) = 2^{1-K} \int dh W (h | \sigma) \prod_\alpha \frac{e^{h a^\alpha}}{2 \cosh (h)} , \]  \hspace{1cm} (14)

\[ \tilde{\rho} (\sigma, \sigma) = 2^{K-1} \int du Z (u | \sigma) \prod_\alpha \frac{e^{h a^\alpha}}{2 \cosh (u)} , \]  \hspace{1cm} (15)

for some (yet unknown) densities \( W, Z \). The prefactors \( 2^{-1/K} \) and \( 2^{-(K-1)/K} \) ensure that the distributions \( W \) and \( Z \) solving the ‘density evolution’ equations are normalized \( (\int dh W (h | \sigma) = \int du Z (u | \sigma) = 1) \).

For the left- and right-eigenvectors \( L (s, s), R (s, s) \) of \( V \) we now assume a similar ‘replica-symmetric’ form:

\[ \langle s', s' | R \rangle = \sum_s P_R (s') \int dx \Phi_R (x | s') e^{y \sum a^\alpha} , \]  \hspace{1cm} (16)

\[ \langle L | s, s \rangle = \sum_s P_L (s) \int dy \Phi_L (y | s) e^{y \sum a^\alpha} , \]  \hspace{1cm} (17)

for some (yet unknown) densities \( \Phi_R, \Phi_L \). The form of the above two equations follows the replica symmetric ansatz as introduced in \cite{8, 9}. Note that in addition to the assumptions in \cite{8, 9} we presented the densities \( P_R, P_L \) that are the right- and left-eigenvectors of \( \mathcal{W} \), respectively:

\[ P_R (s_0) = \sum_{s'_0} \mathcal{W} [ s'_0 | s_0] P_R (s'_0) , \]  \hspace{1cm} (18)

\[ P_L (s'_0) = \sum_{s_0} \mathcal{W} [ s'_0 | s_0] P_L (s_0) . \]  \hspace{1cm} (19)

The fact that replica symmetry is imposed (in terms of products over replica indices) allows us to take the remaining trace in (12) and take the limit \( n \to 0 \) in equation (10).

### 4. Density evolution equations

We present now the self-consistent equations for the densities \( W (h | \sigma), Z (u | \sigma), M (\zeta | \sigma), \Phi_R (x | s) \) and \( \Phi_L (x | s) \) of the effective fields which extremize the free energy (10) within the replica symmetric ansatz. Following similar computations as in \cite{8, 31}, we derive in the limit \( n \to 0 \) the closed, self-consistent equations.
\[ W(h|\sigma) = \int\left( \prod_{r=1}^{C-1} du_r Z(u_r|\sigma) \right) \int d\zeta M(\zeta|\sigma) \delta \left[ h - \zeta - \sum_{r=1}^{C-1} u_r \right], \quad (20) \]

\[ F(\xi|\sigma) = \int\left( \prod_{r=1}^{C-1} du_r Z(u_r|\sigma) \right) \delta \left[ \xi - \sum_{r=1}^{C} u_r \right]. \quad (21) \]

and also

\[ M(\xi|\sigma) = 2 \sum_{s'=1}^{1} \sum_{r'=1}^{1} P_R(\xi'|s') \mathcal{W}\left( (s')|s' \right) P_L(s') \]

\[ \times \int dx dy \Phi_L(y|s') P_R(x'|s') \sum_{\rho=\pm 1} P_{\text{chan}}(\rho|\sigma, s') \]

\[ \times \delta \left[ \zeta - \frac{1}{2} \sum_{r=1}^{1} \tau \sigma \log \left( \sum_{s'=1}^{1} e^{(s'+sy)} \mathcal{W}[s'|s] P_{\text{chan}}(\rho|\tau, s) \right) \right]. \quad (22) \]

\[ Z(u|\sigma) = \sum_{(s')=1}^{1} \frac{\delta(s_1 \cdots s_{K-1}|\sigma, 1)}{2^{K-2}} \int \prod_{\ell=1}^{K-1} dh_{\ell} W(h_{\ell}|\sigma_{\ell}) \]

\[ \times \delta \left[ u - \text{atanh} \prod_{\ell=1}^{K-1} \tanh (h_{\ell}) \right]. \quad (23) \]

\[ \Phi_R(x'|s') = \sum_{(s')=1}^{1} \mathcal{W}\left( (s')|s' \right) \sum_{r=1}^{1} \frac{1}{2} \int d\xi F(\xi|\tau) \int dx \Phi_R(x'|s') \]

\[ \times \sum_{\rho=\pm 1} P_{\text{chan}}(\rho|\tau, s') \]

\[ \times \delta \left[ x - \frac{1}{2} \sum_{s'=1}^{1} s \log \left( \sum_{s'=1}^{1} \mathcal{W}[s'|s] \frac{e^{s'x'}}{2 \cosh(x')} \right) \right] \]

\[- \frac{1}{2} \sum_{s'=1}^{1} s \log \left( \sum_{\sigma=\pm 1} \frac{e^{-\sigma s'}}{2 \cosh \xi} P_{\text{chan}}(\rho|\sigma, s) \right) \]. \quad (24) \]

\[ \Phi_L(x'|s') = \sum_{s'=1}^{1} \mathcal{W}\left( (s')|s' \right) \frac{P_L(s')}{P_L(s')} \sum_{r=1}^{1} \frac{1}{2} \int d\xi F(\xi|\tau) \int dx \Phi_L(x'|s') \]

\[ \times \sum_{\rho=\pm 1} P_{\text{chan}}(\rho|\tau, s') \]

\[ \times \delta \left[ x - \frac{1}{2} \sum_{s'=1}^{1} s' \log \left( \sum_{s'=1}^{1} \mathcal{W}[s'|s] \frac{e^{s'x'}}{2 \cosh(x')} \right) \right] \]

\[- \left( \sum_{\sigma=\pm 1} \frac{e^{\sigma s'}}{2 \cosh \xi} P_{\text{chan}}(\rho|\sigma, s) \right) \]. \quad (25) \]

We refer to appendix A for details on the derivation.
The equations (20)–(25) are the density evolution equations for binary asymmetric two-state Markov channels. These equations are a generalization of the density evolution equations for symmetric Gilbert–Elliott channels [33], as well as a generalization of density evolution for asymmetric memoryless channels [16, 34]. The equations (20)–(25) describe the evolution of the densities of messages of a decoding algorithm propagating along a tripartite graph, in the limit of infinite block lengths $M \to \infty$. The graph consists of a chain of channel-state nodes connected to symbol-variable nodes and these in turn to parity-check nodes. The tripartite graph has three different sets of vertices: the set $V_{\text{code}}$ of variable nodes, the set $V_{\text{pc}}$ of parity-check nodes and the set $V_{\text{chan}}$ of channel-state nodes; see figure 3. Due to the presence of memory there are 6 types of messages propagating according to:

| Message | From | To |
|---------|------|----|
| $h_{i \rightarrow a}$ | $i \in V_{\text{code}}$ | $a \in V_{\text{pc}}$ |
| $u_{a \rightarrow i}$ | $a \in V_{\text{pc}}$ | $i \in V_{\text{code}}$ |
| $\zeta_{c \rightarrow i}$ | $c \in V_{\text{chan}}$ | $i \in V_{\text{code}}$ |
| $\xi_{i \rightarrow c}$ | $i \in V_{\text{code}}$ | $c \in V_{\text{chan}}$ |
| $x_{R,c \rightarrow c+1}$ | $c \in V_{\text{chan}}$ | $c+1 \in V_{\text{chan}}$ |
| $x_{L,c \rightarrow c-1}$ | $c \in V_{\text{chan}}$ | $c-1 \in V_{\text{chan}}$ |

The update schedule for these messages on single-graph instances (the so-called ‘message-passing’ equations) [40] correspond to the arguments of the delta functions in the density evolution equations (20)–(25). In the next section we describe with more detail the corresponding belief propagation decoding algorithm. However, before presenting the belief propagation algorithm we introduce some notations which describe the connectivity properties of the tripartite factor graph.

Every variable node $i$ is connected to $C$ check nodes $a \in \partial_i$. Analogously, every check node $a$ is connected to $K$ variable nodes $i \in \partial_a$, defining the set of neighbouring check nodes $\partial_a$. The neighbourhoods $\partial_i$ and $\partial_a$ define the usual bipartite factor graph used in belief propagation algorithms of memoryless channels. The memory introduces an additional chain of channel nodes $c$. To each variable node $i$ we associate a channel node $c \in \partial_i$, for which we also have $i \in \partial_c$. The channel nodes $c$ have three neighbouring nodes: the corresponding variable node $i \in \partial_c$ but also two channel nodes which we label with respect to $c$ as $c-1$ and
Two exceptions are what we call the first channel node and the last channel node, which have only one neighbouring channel node denoted, respectively, by \( c + 1 \) and \( c - 1 \).

5. Message-passing based decoding algorithm

The decoding process aims at inferring the codeword sent \( \sigma \in \mathcal{C} \) from the output word \( \rho \). We adapt a decoding algorithm which can be considered as a particular version of belief propagation: we compute the marginals of the conditional distribution of codewords given the received output word. The decoding algorithm is also a message-passing algorithm which propagates messages along a tripartite graph. The distributions of the messages sent are given by the derived density evolution equations (20)–(25), in the limit of infinite block lengths \( M \to \infty \).

The core of the algorithm is similar to the sum-product algorithm used to decode LDPC-codewords on memoryless channels. It consists of two types of messages: \( h_{ia}^{(t)} \) from a variable node \( i \) to a check node \( a \) and \( u_{ai}^{(t)} \) from a check node \( a \) to a variable node \( i \). These messages are calculated iteratively for \( t \geq 1 \):

\[
 h_{ia}^{(t+1)} = r_{ia}^{(t)} + \sum_{b \in \partial a \setminus i} u_{bi}^{(t)},
\]

and \( h_{i\rightarrow a}^{(0)} = \zeta_{i 
rightarrow a}^{(0)} \). The index \( t \) indicates the iteration step. The set \( \partial a \) contains the \( K \) variable nodes corresponding to the \( a \)-th parity check node, the set \( \partial_i \) contains the \( C \) parity check nodes corresponding to the \( i \)-th variable node and the channel node \( c \in \partial_i \), as defined in the previous section. After convergence of the above message-passing schedule, we calculate

\[
 h_{i} = \zeta_{i 
rightarrow 1}^{(t)} + \sum_{b \in \partial_i} u_{bi}^{(t)},
\]

with again \( c \in \partial_i \). To estimate the codeword bit sent we use \( \hat{\sigma}_i = \text{sign}(h_i) \).

The value of the message \( \zeta_{i 
rightarrow 1}^{(t)} \) depends on the channel considered. For memoryless channels it is given by the a priori log-likelihood for codeword bit \( \sigma_i \) given the received bit \( \rho_i \). For a binary symmetric memoryless channel [22, 44, 45]:

\[
 \zeta_{i 
rightarrow 1}^{(t)} = \zeta_i = \frac{\rho_i}{2} \log \left( \frac{1 - \rho}{\rho} \right),
\]

and for a binary asymmetric memoryless channel [16, 34]:

\[
 \zeta_{i 
rightarrow 1}^{(t)} = \zeta_i = \begin{cases} 
 \frac{1}{2} \log \left( \frac{1 - q}{p} \right) 
 & \rho_i = 1 \\
 -\frac{1}{2} \log \left( \frac{1 - p}{q} \right) 
 & \rho_i = -1.
\end{cases}
\]

We refer to the message-passing algorithm corresponding to the equations (26)–(28) and equation (29) as BP-BSC. Analogously, we name the algorithm corresponding to equations (26)–(28) and (30) BP-BAC.
For channels with memory the message $\zeta_{c\to i}^{(t)}$ from the channel node $c$ to the neighbouring variable node $i \in \partial c$ has to be calculated iteratively and is becoming an integral part of the message-passing algorithm. We have for $t \geq t_0$:

$$
\sum_{s} x_{c^{(t)}} = \sum_{s} \exp \left[ \sum_{s'} \exp \left[ s' x_{c^{(t)}} + s y_{c^{(t)}} \right] W[s'] P_{\text{chan}}(\rho, \tau, s) \right] W[s] P_{\text{chan}}(\rho, \tau, s).
$$

(31)

The message $\zeta_{c\to i}^{(t)}$ depends on the channel node messages $x_{c^{(t)}}$ and $y_{c^{(t)}}$. These are given by the iterative expressions for $t \geq t_0$:

$$
\begin{align*}
\sum_{s} x_{c^{(t+1)}} & = \frac{1}{2} \sum_{s} \exp \left[ \sum_{s'} \exp \left[ s' x_{c^{(t)}} + s y_{c^{(t)}} \right] W[s'] P_{\text{chan}}(\rho, \tau, s) \right] W[s] P_{\text{chan}}(\rho, \tau, s), \\
\sum_{s} y_{c^{(t+1)}} & = \frac{1}{2} \sum_{s} \exp \left[ \sum_{s'} \exp \left[ s' x_{c^{(t)}} + s y_{c^{(t)}} \right] W[s'] P_{\text{chan}}(\rho, \tau, s) \right] W[s] P_{\text{chan}}(\rho, \tau, s).
\end{align*}
$$

(32)

(33)

with $y_{c^{(t)}} = x_{c^{(t)}} + 1 + c = 1/2 \log (g/b)$. To have a well-defined set of equations we need an expression for $x_{c^{(t)}}$ and $y_{c^{(t)}}$. If we assume that the Markov-modulated channel is in a stationary state throughout its use, we can set $x_{c^{(t)}} = 1/2 \log (g/b)$. If we assume that the Markov-modulated channel is in a stationary state throughout its use, we can set $x_{c^{(t)}} = 1/2 \log (g/b)$.

Finally we have an update rule for the message $\zeta_{i\to c}^{(t)}$ from the variable node $i$ to the channel node $c$, for $t \geq 1$:

$$
\zeta_{i\to c}^{(t)} = \sum_{u \in \partial c} u_{i\to c}^{(t)}.
$$

(34)

In all of the above equations we have the relationship $i \in \partial c$ and $c \in \partial i$.

We name the full message-passing algorithm, consisting of equations (26)–(28) and equations (31)–(34) given the asymmetric Markov-modulated channel $P_{\text{chan}}(\rho, \tau, s)$ in equation (3), BP-AsymMM. For a symmetric Markov-modulated channel $P_{\text{chan}}(\rho, \tau, s) = P_{\text{chan}}(\rho, \sigma, s)$ we recover the message-passing algorithms for Gilbert–Elliott channels [35, 40], which we call BP-GEC. Note that for a memoryless channel $P_{\text{chan}}(\rho, \sigma, s) = P_{\text{chan}}(\rho, \sigma)$ we recover the BP-BSC and BP-BAC algorithms, as expected.

In the next section we continue evaluating the decoding algorithms presented.

6. Results

In this section we evaluate the performance of the sum-product algorithms for regular LDPC codes on Markov modulated channels. In a first subsection we determine the probability of error using belief propagation decoding algorithms on Markov modulated channels. We show that the derived density evolution determine the performance of the sum-product in the limit of infinite block lengths $M \to \infty$. Solving the density evolution equations we show the presence of a transition from a phase at low noise, where reliable communication with belief propagation is possible, to a phase at high noise, where reliable communication with the decoding algorithms presented is not possible. We evaluate in the subsequent subsection more carefully the different decoding regimes using density evolution.
6.1. Probability of error

We recall that the decoding algorithm consists of propagating messages along a tripartite factor graph (see figure 3) which leads to estimates for the log-likelihood ratios $h_i$ of the marginals of the codeword bits. The sent codeword bit $\sigma_i$ is estimated by

$$\hat{\sigma}_i = \text{sign}(h_i).$$

Performance of the decoding algorithm is evaluated through the probability of error

$$P_e \equiv \sum_{\sigma} \sigma \equiv -\log \left( \frac{1 - \sigma \hat{\sigma}}{2} \right).$$

which is the fraction of bits which have been incorrectly decoded. We say that reliable communication is possible when $\langle P_e \rangle \to 0$ for $M \to \infty$. The brackets represent an average over many uses of the channel (i.e. an average over codewords, channel noise and parity check matrices). We also compute the standard deviation $\sigma[P_e]$, which is the square root of the variance of $P_e$ over repeated uses of the channel.

To evaluate $\langle P_e \rangle$ we have randomly generated 100 parity check matrices $\mathcal{H}$ with connectivity parameters $C = 3$, $K = 4$ and fixed codeword block length $M$. For each parity check matrix we have randomly generated 100 codewords $\sigma$ such that $\langle \sigma \rangle = 0$. The codewords $\sigma$ are converted into output words $\rho$ using a channel with memory parameters $\mu = 0.90$, good-to-bad ratio $r = g/b = 1$ and a degree of asymmetry $\kappa = p_G/q_G = p_b/q_b = 0.01$. Note that to simplify the presentation of our results the two channel states have here the same $\kappa$.

Given the received output words $\rho$ and the parity check matrix $\mathcal{H}$ we calculate the estimates $\hat{\sigma}_i$ of the codewords bit using the belief propagation algorithms presented in section 5. The average of the probability of error $\langle P_e \rangle$ is computed over repeated uses of the channel and over the different parity check matrices.

In figure 4 we compare the average probability of error $\langle P_e \rangle$ for the decoding algorithms BP-AsymMM, BP-BAC and BP-BSC at codeword lengths $M = 600$, at fixed noise levels of the good channel $(p_G + q_G)/2 = 0.05$ or $(p_G + q_G)/2 = 0.1$ and at varying noise levels $(p_b + q_b)/2$ of the bad channel. Note that the BP-BSC algorithm is run with a noise parameter $p = (p_G + q_G + p_b + q_b)/4$ and BP-BAC is run for noise values $q = (q_b + q_G)/2$ and $p = (p_G + p_b)/2$. From figure 4 one can appreciate that BP-AsymMM has the smallest error probability and therefore allows us to decode messages at higher noise levels. This is particularly true for large enough values of the memory $\mu$ and asymmetry $\kappa$ in the channel noise. The algorithm BP-GEC is not shown in the figures, but numerical results show that at high degrees of asymmetry it does not perform much better than BP-BSC in terms of $\langle P_e \rangle$.

In the figure 4 we also compare finite block length values $\langle P_e \rangle$ with infinite block length results computed from the density evolution equations (20)–(25). To compute the average probability of error from the density evolution equations we consider the distribution of the single bit marginals of the codewords given the output word:

$$\tilde{\mathcal{W}}(h|\sigma) = \int \prod_{r=1}^{C} du_r \mathcal{Z}(u_r|\sigma) \int d\zeta M(\zeta|\sigma) \delta \left[ h - \zeta - \sum_{r=1}^{C} u_r \right],$$

with $\mathcal{Z}$ and $M$ the distributions solving equations (20)–(25). The average probability of error follows as

$$\langle P_e \rangle = 2^{-1} \sum_{\sigma = \pm 1} \int dh \tilde{\mathcal{W}}(h|\sigma) \text{sign}(h).$$

Results in figure 4 show a first order transition in $\langle P_e \rangle$ from a value $\langle P_e \rangle = 0$ at $q < q_c$ to a value $\langle P_e \rangle > 0$ at $q > q_c$. Density evolution results show therefore a phase transition from a
regime at which reliable communication is possible \( q < q_c \) to a regime for which reliable communication using the belief propagation decoding algorithm is not possible. From the results in figure 4 it follows that the transition is not clearly visible from the decoding results presented at finite block lengths. We believe this is due to strong finite size effects in the probability of error \( \langle P_e \rangle \) close to the transition. Finite size effects in channels with memory are expected to be larger than in channels without memory, due to the correlations in the noise variables. Therefore, larger fluctuations are present in the probability of error (with respect to the different channel uses).

To show that the deviations between the results from density evolution and the BP-AsymMM decoding algorithm are indeed due to finite size effects, we present in figure 5 the probability of error \( \langle P_e \rangle \) for increasing values of \( M = 600, 1200, 2400 \). To have an experimental validation of the density evolution equations, we should observe a convergence of the BP-AsymMM probability of error \( \langle P_e \rangle \) towards the values calculated from the density evolution equations. These results indicate that the density evolution equations (20)–(25) determine the probability of error of BP-AsymMM at infinite block lengths \( M \to \infty \).

Figure 4. Average probabilities of error for different belief propagation algorithms are compared with the derived density evolution equations. Belief propagation algorithms decode codewords generated from LDPC-codes on finite-state Markov-modulated channels with asymmetric noise. The channel parameters are: memory \( \mu = 0.9 \), good-to-bad ratio \( r = g/b = 1 \) and asymmetry ratio \( \kappa = p_G/q_G = p_B/q_B = 0.01 \). Source words of length \( N = 200 \) are encoded by a \((C, K) = (3, 4)\) parity check matrix into codewords of length \( M = 600 \). Results are averages over \( 1e+4 \) generated code words from \( 1e+2 \) randomly generated parity check matrices (thus for \( 1e+2 \) codewords each). The algorithms considered are given by the message-passing algorithm presented in section 5. Density evolution results are obtained from the (20)–(25) in section 4, and thus evaluate the BP-AsymMM probability of error in the limit \( M \to \infty \). These equations are solved with a Monte Carlo algorithm with a sample size of \( 1e+5 \) fields per distribution (see also [42, 43]). Note that all curves presented evaluate the probability of error for the same finite-state Markov-modulated channel using different decoding algorithms.
Comparing the \( \langle P \rangle \) curves at increasing values of \( M \) we find some indications of a phase transition being present at values \( M \to \infty \), as predicted by density evolution. The noise value at which the \( \langle P \rangle \)-lines cross gives a reasonable approximation of the critical threshold \( q_c \) computed from density evolution (indicated by the vertical dashed line). Moreover, results presented are consistent with a critical noise level in the limit \( M \to \infty \) at the \( q_c \) value computed from the density evolution equations. We also present in figure 5 the standard deviation \( \sigma \) in the probability of error. The curves develop a maximum value at \( q \approx q_c \) for \( M \to \infty \), a result which is also consistent with the presence of a first order phase transition in \( \langle P \rangle \) at infinite block lengths.

Figure 5. BP-AsymMM results for the average probability of error \( \langle P \rangle \) and its standard deviation \( \sigma \) as a function of the noise \((p_g + q_g)/2\). The channel parameters are the same as in figure 4 with \((p_g + q_g)/2 = 0.1\). Averages are over \( 1 \times 4 \) unbiased codewords and \( 1 \times 2 \) parity check matrices, as in figure 4. The LDPC-code is, just as in figure 4, a (3,4)-code but now with three different codeword lengths: \( M = 600, 1200, 2400 \). The critical noise level \( q_c \) at infinite block lengths is presented as a dashed line (computed from density evolution). Although BP-AsymMM results show strong finite size effects for \( q \sim q_c \), results do indicate a convergence towards the density evolution results for \( M \to \infty \) (presented by the diamond green markers). The standard deviation \( \sigma \) peaks at a value \( q \sim q_c \) which is consistent with the presence of a first order phase transition in \( \langle P \rangle \) at infinite block lengths.

6.2. Critical noise levels

In the previous subsection we have shown that reliable communication fails at a critical noise level. This transition is of first order in the probability of error. Failure of reliable communication corresponds to a spinodal ferromagnetic to paramagnetic phase transition in the language of spin models on a small world graph. Indeed, loss of reliable communication at \( q = q_c \) is equivalent to a transition from a ferromagnetic solution with effective field distribution \( W(h|\sigma) = \delta(h - \infty) \) at values \( q < q_c \) to a paramagnetic solution at values \( q > q_c \). This transition can be characterized by an order parameter \( \rho_b = \int dh W(h) \text{sign}(h) \). The value \( \rho_b = 1 \) corresponds to perfect decoding (ferromagnetic phase) while \( \rho_b < 1 \) describes decoding failure (due to the spin glass or paramagnetic attractor; see for instance the discussion in [17]). We now calculate the critical noise levels \( q_c \) for various channels by numerically solving the density evolution equations (e.g. through population dynamics [42, 43]). The initial condition for solving the density evolution equations is such that the distributions \( W(h|\sigma) = \delta(h) \), \( Z(u|\sigma) = \delta(u) \) while \( \Phi_L(x|s) = \Phi_R(x|s) = \delta(x - 1/2 \log (g/h)) \). This initial condition
implies that the decoder has no a priori knowledge of the message sent (which is unbiased) and corresponds to the initial condition in the belief propagation decoding algorithm.

Solving the density evolution equations we thus find the decoding thresholds for BP-AsymMM in the parameter space \( \left( \frac{1}{2}(p_B + q_B), \frac{1}{2}(p_G + q_G) \right) \) for different values of the asymmetry \( \kappa \in [0, 1] \) (the degree of asymmetry is assumed to be the same in both channels). For \( \kappa = 0 \) we obtain the Z-channel. For all symbols the memory equals \( \mu = 0.90 \) while the good-to-bad ratio is \( r = 1 \). The dotted line represents the memoryless threshold \( \mu = 0 \). The star symbol represents the point where \( p_B = p_G \) and is a result quoted in [17]. We see that as the level of asymmetry increases successful decoding can occur even for large noise values of the ‘bad’ channel, as long as the noise in the ‘good’ channel remains relatively low.

In figure 6 we present critical noise levels for the channel \( T_{AS} \) in which there exist two Z-type states: \( (p_B, q_B) = (0, q) \) and \( (p_G, q_G) = (q, 0) \) (hence the terms ‘good’ vs ‘bad’ are not
very meaningful here) and with a good-to-bad ratio $r = 1$, as defined in [40]. This type of configuration can model ‘burst-error’ channels where a very large number of consecutive bits appear corrupted while the corruption is selective with regards to the input symbol. We show results in the $(\mu, q)$ space for Gallager $(C, K) = (3, 4)$ and $(4, 6)$ codes. The lower dashed line corresponds to twice the critical noise level of a memoryless symmetric channel $q_{BS}$, where $q_{BS}$ is the critical noise level a regular $(C, K)$ Gallager code on a memoryless binary-symmetric channel. The fact that at $\mu = 0$ the channel consists of two complementary $Z$-type states connected through a Markov chain without memory, and therefore, with an equal transition probability $b = g = 1/2$ at each time step between the two states. Since the ‘good’ and ‘bad’ channel here are complementary $Z$-channels, the finite-state Markov channel becomes in fact symmetric.

The upper dashed line corresponds to the critical noise level of a memoryless $Z$-channel [16], which is reached by the finite-state Markov channel at $\mu = -1$; the transition probabilities then become $b = g = 1$ at $\mu = -1$. The channel state thus oscillates deterministically between the two states. We thus recover a deterministic channel whose decoding properties are similar to the $Z$-channel.

We note as well that figure 5 is symmetric with respect to the $\mu = 0$ axis, a property that also follows from the density evolution equations.
7. Conclusions

We have shown how the statistical mechanics of small-world networks can be used as a tool to analyze an interesting application, namely the performance of decoding algorithms on binary channels with both memory and asymmetry in the noise levels. The main motivation of studying error-correcting codes on channels with memory and with asymmetry is the fact that their capacity outperforms that of traditional codes on memoryless channels. While both effects have been studied separately (for asymmetry in [16, 34] and for memory in [33]), their combined effect has not been studied.

We have studied the limits of reliable communication by representing the decoding problem as a random graph with interacting elements. Similarly to earlier work on memoryless channels [22], the symbol variables (the ‘spins’) are the nodes of the graph while the parity checks the edges. The memory process introduces, however, a new element, namely channel-state variables which are coupled to each other with nearest-neighbour interactions. The resultant architecture is thus a small-world network. The thermodynamic observables of Ising models on such networks can be calculated using a replicated transfer matrix technique [8]. The main objective of this work is to illustrate this link and provide the statistical physics methodology to derive the relevant density evolution equations within the replica symmetric ansatz.

With the density evolution equations we have shown the combined effect of memory and asymmetry in a phase diagram which shows the critical noise levels for reliable communication using belief propagation and unbiased codewords. We find that the combination of both memory and asymmetry in the channel improves the critical noise levels considerably. In particular, the bad and the good channel can have a compensating effect: in some cases (e.g. a Z-channel) successful decoding can be possible even for arbitrarily high levels in the bad channel as long as the noise in the good channel is sufficiently low. As a benchmark of our work, we find that the density evolution equations that follow from the analysis reproduce very well the special limiting cases of the Gilbert–Elliott channel [33, 40] or the memoryless binary-asymmetric channel [16, 34].

Our statistical physics method is quite general and can be applied to various cases. It allows us to derive critical threshold values of decoding algorithms for which the noise values of the channels are not known (which is sometimes referred to in the statistical physics literature as finite-temperature decoding [50]). The calculation of free energies can provide limits on ‘thermodynamic’ or MAP decoding [51]. The method can be applied as well to a variety of multi-state error-correcting codes, such as multi-symbol, Gaussian-, non-Markovian or intersymbol-interference channels. Another interesting future direction would be the inclusion of replica symmetry-breaking effects (along the lines of [47, 48]) which might correct the critical noise levels we present here. This is likely to pose new challenges from both an analytical and a numerical viewpoint due to the spatial structure of the factor graph.

Acknowledgments

We would like to thank Bastian Wemmenhove who participated in the initial stages of this work and Toshiyuki Tanaka for useful correspondence. IN is grateful to Désiré Bollé for his guidance through the preparation of the PhD dissertation, when this work was realized.
Appendix A. Replica analysis

A.1. Saddle point equations in replica space

To compute the disorder-averaged free energy our starting point is equation (9) which upon using the gauge transformations \( \rho_i \rightarrow \rho_i \delta_{i0} \) and \( \sigma_i \rightarrow \sigma_i \delta_{i0} \) is modified to:

\[
- \mathcal{F} = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{\mathcal{H}} P(\mathcal{H}) \sum_{s^0} P_{\text{state}}(s^0) \sum_{\sigma^0} P_{\text{chan}}(\rho \sigma^0 | s^0 \sigma^0) \delta_{\mathcal{H}}[\sigma^0] \\
\times \log \left( \sum_{\sigma,s} P_{\text{state}}(s) \delta_{\mathcal{H}}[\sigma] P_{\text{chan}}(\rho \sigma^0 | s, \sigma^0) \right),
\]

(A.1)

In order to calculate the average over the logarithm we use the replica trick \( \langle \log Z \rangle = \lim \log Z \). This implies that the dynamic variables \( (\sigma, s) \) will be replicated \( n \) times to \( \{ (\sigma^1, s^1), (\sigma^2, s^2), \ldots, (\sigma^n, s^n) \} \). We will represent the probability \( P_{\text{chan}} \) sometimes as

\[
P_{\text{chan}}(\rho | s, \sigma) = \prod_{j=1}^{M} \frac{e^{\rho(h(s_j)\sigma_j + g(s_j))}}{N(s_1, \sigma_1)},
\]

where \( N(s, \sigma) \) corresponds to the normalization \( N(s, \sigma) = 2 \cosh(h(s)\sigma + g(s)) \). In what follows, averages over this distribution will be denoted with the abbreviation:

\[
\langle f(\rho) \rangle_{s, \sigma} \equiv \sum_{\rho \in [0,1]} \frac{e^{\rho(h(s)\sigma + g(s))}}{N(s, \sigma)} f(\rho).
\]

In expression (A.1) we need to determine the average over the disorder variables \( \{ \mathcal{H}, s^0, \sigma^0, \rho \} \). The summation over the \( \mathcal{H} \) matrices can be done as in the memoryless case, see e.g. [15, 16]. The only difference now is that the ratios \( h_t \) and \( g_t \) depend on the state variables of the channel \( \{ s_l \} \). The probability distribution of the tensor \( \mathcal{H} \), viz.,

\[
P(\mathcal{H}) = \frac{1}{\mathcal{M}} \prod_{\{l_1, j_2, \ldots, j_K\}} \frac{C(K - 1)!}{M^{K-1}} \delta \left( H_{l_1 j_2 \ldots j_K} - 1 \right)
\]

\[
+ \left( 1 - \frac{C(K - 1)!}{M^{K-1}} \right) \delta \left( H_{l_1 j_2 \ldots j_K} \right)
\]

\[
\times \prod_{k=1}^{M} \delta \left( \sum_{\{l_1, j_2, \ldots, j_K\}|j_k = l} H_{l_1 j_2 \ldots j_K} - C \right).
\]

(A.2)

The quantity \( \mathcal{M} \) is the normalization constant, which equals to

\[
\mathcal{M} = e^{-MC + M \log \frac{C}{C_t}}.
\]

The notation \( \langle j_1, j_2, \ldots, j_K \rangle \) denotes an unordered set of \( K \) indices and \( \prod_{\{j_1, j_2, \ldots, j_K\}} \) is the product all unordered sets of \( K \) indices \( \{ j_1, j_2, \ldots, j_K \} \) with \( j_1 = 1, \ldots, M \) (this is equivalent to a product over all \( K \)-tuples such \( j_1 < j_2 < \cdots < j_K \)).

After replication of the argument \( Z = \sum_{\sigma} P_{\text{state}}(s) \delta_{\mathcal{H}}[\sigma] P_{\text{chan}}(\rho \sigma^0 | s, \sigma^0) \) in the logarithm of equation (A.1), we perform the summation over the ‘quenched’ variables \( H_{(j_1, j_2, \ldots, j_K)} \) with \( H_{(j_1, j_2, \ldots, j_K)} = 1 \) indicating the presence of a parity-check constraint between the variables \( \{ j_1, j_2, \ldots, j_K \} \) and \( H_{(j_1, j_2, \ldots, j_K)} = 0 \) indicating the absence of such a parity-check constraint. To
perform the summation it is necessary to introduce a set of \( M \) auxiliary variables \( \omega_l \), with \( l = 1 \ldots M \), through the integral representation of the Kronecker delta function in the definition (A.2) of \( P(\mathcal{H}) \):

\[
\delta \left( \sum_{\{i_1, i_2, \ldots, i_K\} : \sum_{l=1}^{M} \omega_l = l} H_{\{i_1, i_2, \ldots, i_K\}} - C \right) \\
= \int_{0}^{2\pi} \frac{d\omega_l}{2\pi} \exp \left[ i\omega_l \left( \sum_{\{i_1, i_2, \ldots, i_K\} : \sum_{l=1}^{M} \omega_l = l} H_{\{i_1, i_2, \ldots, i_K\}} - C \right) \right].
\]

We can now sum up the variables \( H_{\{i_1, i_2, \ldots, i_K\}} \). After replication, we factor out the functions independent of the graph variables \( \mathcal{H} \) these graph variables and have:

\[
\sum_{\mathcal{H}} P(\mathcal{H}) \delta_{\mathcal{H}} \left[ \sigma^0 \right] \prod_{a=1}^{n} \delta_{\mathcal{H}} \left[ \sigma^a \right] \\
= \int \left( \prod_{l=1}^{M} \frac{d\omega_l}{2\pi} \right) e^{-i\sum_{l=1}^{M} \omega_l C} \\
\times \exp \left[ \frac{C}{M^{K-1}} \sum_{\{i_1, i_2, \ldots, i_K\}} \left( e^{i\sum_{a=0}^{K} \prod_{m=1}^{K} \delta_{\mathcal{H}} \left( \prod_{m=1}^{K} \sigma^a_m ; 1 \right) - 1} \right) \right] \\
= e^{-MC} \int \left( \prod_{l=1}^{M} \frac{d\omega_l}{2\pi} \right) e^{-i\sum_{l=1}^{M} \omega_l C} \\
\times \exp \left[ \frac{MC}{K} \left( \sum_{\sigma, \sigma'} e^{i\omega(a, \sigma) \delta(a, \sigma')} \right) \right] \prod_{m=1}^{K} \delta_{\mathcal{H}} \left( \prod_{m=1}^{K} \sigma^a_m ; 1 \right) \\
= \prod_{a=1}^{n} \delta_{\mathcal{H}} \left( \prod_{m=1}^{K} \sigma^a_m ; 1 \right),
\]

with the summation \( \sum_{\{i_1, i_2, \ldots, i_K\}} = \sum_{i_1}^{M} \sum_{i_2}^{M} \ldots \sum_{i_K}^{M} = 1 \) and \( \sigma = (\sigma^1, \sigma^2, \ldots, \sigma^n) \).

Equation (A.4) implies naturally the introduction of the order-parameter function \( P(\sigma, \sigma) \):

\[
P(\sigma, \sigma) \equiv \frac{1}{M} \sum_{i=1}^{M} e^{i\omega_i} \delta_{\sigma_i, \sigma} \delta_{\sigma_a, \sigma} \]

with \( \sigma = (\sigma^1, \sigma^2, \ldots, \sigma^n) \). The order parameter (A.5) is a non-normalized function due to the auxiliary variable \( \omega_i \) which has been introduced to deal with the constraint imposed by a fixed number \( C \) of \( K \) parity-check constraints per bit. The order parameter \( P(\sigma, \sigma) \) would be normalized for a code where a number of \( M - N, K \) parity-check constraints, are distributed randomly over the bits. For such codes the distribution \( P(\mathcal{H}) \) is given by the expression (A.2) without the Kronecker delta on the second line. In that case there is no need to introduce the auxiliary variables \( \omega_l \) through equation (A.3). We also note that unlike similar computations on small-worlds (e.g. [31]) here it is necessary to include the quenched variable \( \sigma^i \) into the order parameter.
We introduce explicitly the order parameters $P(\sigma, \sigma)$ through a delta function expression
\[ 1 = \prod_{\sigma, \sigma} \frac{dP(\sigma, \sigma) d\hat{P}(\sigma, \sigma)}{2\pi} \exp \left[ \int \frac{\sum_{\sigma, \sigma} \hat{P}(\sigma, \sigma) \left( P(\sigma, \sigma) - \frac{1}{M} \sum_{\sigma} e^{\delta_{\sigma, \sigma} \delta_{\hat{\sigma}, \sigma}} \right)}{2\pi} \right] = 1. \] (A.6)

After introduction of these integrals, we can perform the summation over the quenched and non-quenched parity bit variables $\sigma \{ i \}$, with $i = 1, \ldots, N$ and $\alpha = 0, \ldots, N$ as well the integral over the auxiliary variables $\omega$ with $i = 1, \ldots, N$. The expression for $Z$ reduces to a saddle point problem (as usual in free energy calculations of spin models on graphs [31]), viz.:

\[ \langle Z' \rangle = \int \prod_{\sigma, \sigma} dP(\sigma, \sigma) d\hat{P}(\sigma, \sigma) \exp \left[ M \Psi \left( P(\sigma, \sigma), \hat{P}(\sigma, \sigma) \right) \right]. \]

with $M \to \infty$ and

\[ \Psi \left( P(\sigma, \sigma), \hat{P}(\sigma, \sigma) \right) = \frac{C}{K} \sum_{(\sigma, \alpha)} \prod_{i=1}^{K} P(\sigma, \sigma) \delta(\sigma_1, \ldots, \sigma_K; 1) \prod_{\alpha=1}^{n} \delta(\sigma_1, \sigma_2, \ldots, \sigma_K, 1) + \sum_{(\sigma, \sigma)} P(\sigma, \sigma) \hat{P}(\sigma, \sigma) - \frac{C}{K} \log(M) + \Phi \left( P(\sigma, \sigma), \hat{P}(\sigma, \sigma) \right). \]

Note that in order to avoid unnecessary extra notation we have denoted by $\sigma$, $s$ the replicated spin variables $\sigma = (\sigma_1, \ldots, \sigma^a)$ and $s = (s_1, \ldots, s^a)$ (and thus the same notation for which we have used $\omega_i, \ldots, \omega_N$ before). The quantity $\Phi(P, \hat{P})$ contains the main difficulty in this computation, namely the trace over the replicated channel state variables $\omega$. We use $(\tau)^N = (\tau_1, \ldots, \tau_N)$ and $(\tau)^a = (\tau_1, \ldots, \tau_N)$:

\[ \Phi \left( P(\sigma, \sigma), \hat{P}(\sigma, \sigma) \right) = \lim_{M \to \infty} \frac{1}{M} \log \left( \sum_{(\sigma, \alpha)} \sum_{(s_1)} \sum_{(s^a)} \prod_{i=1}^{M} \left( P(\sigma, \sigma) \prod_{i=1}^{M} W\left(s^a_{i+1} | s^a_i \right) \right) \right) \]

\[ \times \prod_{i=1}^{M} \left( \frac{-\hat{P}(\sigma, \sigma)}{C!} \right) \left( \prod_{\alpha} \exp \left[ \rho_i \left( h(s^a_i) \sigma_i^a + g(s^a_i) \sigma_i \right) \right] \right) \]

\[ \lim_{M \to \infty} \frac{1}{M} \log \left( \sum_{(s_1)} \sum_{(s^a)} \sum_{(s^b)} \sum_{(s^c)} \prod_{i=1}^{M} \left( \text{init}[s_1, \sigma_i, \sigma_i^0, \sigma_i^0] \right) \prod_{M} \left( s^b_{M}, \sigma_{M}, s^0_{M} \right) \right) \]

\[ \times \left( \prod_{i=1}^{M-1} \left( \text{init}[s_{i+1}, \sigma_{i+1}, \sigma_i^0, \sigma_i^0] \right) \right). \]

The non-trivial spin summations are in the last line of the above equation, where we introduced the matrix:
\[ \langle s, \sigma, s^0, \sigma^0 | T \langle \hat{P} | s', \sigma', (s^0)', (\sigma^0)' \rangle \]
\[ = W \left[ \langle s^0 | s^0 \rangle \prod_\alpha W \left[ (s')^\alpha | s^\alpha \right] (-\hat{P} (\sigma, \sigma))^C \right] \times \left( \prod_\alpha \frac{\exp \left[ \rho (h(s^\alpha)\sigma^\alpha + g(s^\alpha)) \right]}{M (s^\alpha, \sigma^\alpha)} \right)_{\rho \mid s^0, \sigma}, \quad (A.7) \]

\[ \langle \text{init} | s_1, \sigma_1, s_1^0, \sigma_1^0 \rangle = P_{\text{state}} (s_1^0) \prod_\alpha P_{\text{state}} (s_1^\alpha), \quad (A.8) \]

\[ \langle s_M, \sigma_M, s_M^0, \sigma_M^0 | \text{end} \rangle \]
\[ = (-\hat{P} (\sigma_M, \sigma_M))^C \left( \prod_\alpha \frac{\exp \left[ \rho (h(s_M^\alpha)\sigma_M^\alpha + g(s_M^\alpha)) \right]}{M (s_M^\alpha, \sigma_M^\alpha)} \right)_{\rho \mid s^0, \sigma} \quad (A.9) \]

We use now that \( \langle s, \sigma, s^0, \sigma^0 | T \langle \hat{P} | s', \sigma', (s^0)', (\sigma^0)' \rangle \) does not depend on \( \sigma' \) and \( (\sigma^0)' \) to define

\[ \langle s, s^0 | V | s', (s^0)' \rangle \equiv \sum_{\sigma, \sigma^0} \langle s, \sigma, s^0, \sigma^0 | T \langle \hat{P} | s', \sigma', (s^0)', (\sigma^0)' \rangle \].

We then get

\[ \langle s, s^0 | V | s', (s^0)' \rangle = \sum_{\sigma, \sigma^0} W \left[ \langle s^0 | s^0 \rangle \prod_\alpha W \left[ (s')^\alpha | s^\alpha \right] (-\hat{P} (\sigma, \sigma))^C \right] \times \left( \prod_\alpha \frac{\exp \left[ \rho (h(s^\alpha)\sigma^\alpha + g(s^\alpha)) \right]}{M (s^\alpha, \sigma^\alpha)} \right)_{\rho \mid s^0, \sigma}. \quad (A.10) \]

The saddle point equations become

\[ \hat{P} (\sigma, \sigma) = -C \sum_{(\sigma_1, \sigma_2, \ldots, \sigma_{K - 1})} \Pi_{\alpha = 1}^{K - 1} P (\sigma_\alpha, \sigma_\alpha) \times \delta (\sigma, \sigma_1 \sigma_2 \cdots \sigma_{K - 1}, 1) \Pi_{\alpha = 1}^{K - 1} \delta (\sigma_{\alpha 2}^{\alpha 2} \cdots \sigma_{K - 1}^{\alpha 1} \sigma_{\alpha 1}^{\alpha 1}, 1) \]

\[ P (\sigma, \sigma) = -\frac{\text{Tr} \left[ V^{M - 1} (\hat{P}) Q (\sigma, \sigma; \hat{P}) \right]}{\text{Tr} \left[ V^M (\hat{P}) \right]} \]

where we used that for indecomposable channels we have \( \lim_{M \to \infty} \langle \text{init} | V^M (\hat{P}) | \text{end} \rangle = \lim_{M \to \infty} \text{Tr} \left[ V^M \right] \).
\[
\langle s, s^0 | Q (\tau, \tau; \hat{P}) | s', (s^0)' \rangle = -\frac{1}{(C-1)!} \left( -\hat{P} (\tau, \tau) \right)^{C-1} \\
\times W \left[ (s^0)' | s^0 \right] \prod_a W \left[ (s')^a | s^a \right] \left\langle \Pi_a \exp \left[ \frac{\rho \left( h(s^a) r^a + g(s^a) \tau \right)}{M(s^a, \tau^a)} \right] \right\rangle_{\rho | s^0, \tau}.
\]

We perform the transformation \( \hat{P}(\sigma) \rightarrow -C\hat{P}(\sigma) \) such that

\[
\hat{P}(\sigma, \sigma) = \sum_{(\sigma_0, \sigma_1, \ldots, \sigma_{K-1}, \sigma_K)} \prod_{i=1}^{K-1} P(\sigma_i, \sigma_{i+1}) \\
\times \delta(\sigma_1 \sigma_2 \cdots \sigma_{K-1}, 1) \prod_{a=1}^{K} \delta(\sigma^a_1 \sigma^a_2 \cdots \sigma^a_K, 1)
\]

\[
P(\sigma, \sigma) = \frac{\text{Tr} \left[ V^{M-1} \left( \hat{P} \right) Q (\sigma, \sigma; \hat{P}) \right]}{\text{Tr} \left[ V^M \left( \hat{P} \right) \right]}, \tag{A.11}
\]

with the elements of the transfer matrix \( V \) given by:

\[
\langle s, s^0 | V \left( \hat{P} \right) | s', (s^0)' \rangle = \sum_{\sigma, \sigma'} W \left[ (s^0)' | s^0 \right] \prod_a W \left[ (s')^a | s^a \right] \left\langle \Pi_a \exp \left[ \frac{\rho \left( h(s^a) \sigma^a + g(s^a) \sigma \right)}{M(s^a, \sigma^a)} \right] \right\rangle_{\rho | s^0, \sigma}, \tag{A.12}
\]

\[
\langle \text{init} | s_1, \sigma_1, s^0_1, \sigma^0_1 \rangle = P_{\text{state}} (s^0_1) \prod_a P_{\text{state}} (s^a_1)
\]

\[
\langle s_M, \sigma_M, s^0_M | \text{end} \rangle = \left( \hat{P} (\sigma_M) \right)^C \left\langle \exp \left[ \frac{\rho \left( \sum_a h(s^a_M) \sigma^a_M + g(s^a_M) \right)}{} \right] \right\rangle_{\rho | s^0_M},
\]

while the elements of the matrix \( Q \) are given by

\[
\langle s, s^0 | Q (\tau, \tau; \hat{P}) | s', (s^0)' \rangle = \left( \hat{P} (\tau, \tau) \right)^{C-1} \\
\times W \left[ (s^0)' | s^0 \right] \prod_a W \left[ (s')^a | s^a \right] \\
\times \left\langle \Pi_a \exp \left[ \frac{\rho \left( h(s^a) r^a + g(s^a) \tau \right)}{N(s^a, \tau^a)} \right] \right\rangle_{\rho | s^0, \tau}.
\]
Putting everything together, equation (A.1) becomes

\[
\Psi\left(P(\sigma, \sigma), \hat{P}(\sigma, \sigma)\right) = -C \left( \sum_{(\sigma, \sigma)} P(\sigma, \sigma) \hat{P}(\sigma, \sigma) - 1 \right) \\
+ \frac{C}{K} \left( \sum_{(\sigma, \sigma), \ldots, (\sigma, \sigma, \ldots)} \prod_{i=1}^{K} P(\sigma, \sigma) \delta(\sigma_1 \sigma_2 \cdots \sigma_K, 1) \prod_{\alpha=1}^{\mu} \delta(\sigma_1^\alpha \sigma_2^\alpha \cdots \sigma_K^\alpha, 1) - 1 \right) \\
+ M^{-1} \log \left( \sum_{s^0} \left\langle \text{init}|s_i, s_i^0\right\rangle \left( \prod_{i=1}^{M-1} \left\langle s_i, s_i^0|V(\hat{P}))[s_{i+1}, s_i^0]\right\rangle \left\langle s_N, s_N^0|\text{end}\right\rangle \right) \right).
\]

A.2. Replica symmetry ansatz in the saddle point equations

We will now show how the density evolution equations follow from the saddle point equations. To this aim, we introduce a replica symmetric ansatz

\[
P(\sigma, \sigma) = 2^{-1} K \int dh W(h|\sigma) \prod_\alpha \frac{e^{\beta h \alpha}}{2 \cosh(h)}, \quad (A.14)
\]

\[
\hat{P}(\sigma, \sigma) = 2^{-1} K \int du Z(u|\sigma) \prod_\alpha \frac{e^{\beta u \alpha}}{2 \cosh(u)}. \quad (A.15)
\]

We recall that \(\sum_{\sigma} P(\sigma, \sigma) \neq 1\) as the order parameter is not normalized. Using (A.15) allows us to rewrite the expression of \(V\) in (A.10):

\[
\left\langle s, s^0, V(\hat{P})|s', (s')^0\right\rangle = W\left( (s')^0|s^0\right) \prod_\alpha W\left( (s')^\alpha|s^\alpha\right) \\
\times \sum_{\sigma} \left( \hat{P}(\sigma, \sigma) \right)^C \left\langle \Pi_\alpha \exp\left[ \frac{\rho \left( h\left( s^\alpha\right) \sigma^\alpha + g\left( s^\alpha\right) \sigma^\alpha \right) }{N\left( s^\alpha, \sigma^\alpha\right) } \right] \right\rangle_{\rho|\rho, \sigma} \\
= 2^{-C-1} \sum_{\tau} W\left( (s')^0|s^0\right) \prod_\alpha W\left( (s')^\alpha|s^\alpha\right) \\
\times \left\langle \prod_{\tau=1}^{C} du_{\tau} Z(u_{\tau}|\tau) \left\langle \exp\left[ 2^{-1} \sum_\alpha s^\alpha \right] \right\rangle \sum_s \log \left( \sum_\sigma \prod_\alpha \frac{e^{\alpha s\sigma}}{2 \cosh u_{\tau} N(s, \sigma)} \right) \right\rangle_{\rho|\rho, \tau}, \quad (A.16)
\]
and similarly for the expression of $Q$ in (A.13):

$$
\langle s, s^0|Q\left(\mathbf{r}, \mathbf{r}; \hat{P}\right)|s', (s')^0\rangle = 2^{-\rho}e^{-V_s} \prod_s W[\langle s'\rangle]|s'\rangle
$$

$$
\times \int \prod_{\tau=1}^{\rho} du_\tau \, Z(u_\tau|\tau) \left\langle \exp \left( \frac{1}{2} \sum_s \delta^{\rho} \left( \frac{e^{\phi_h(e^{\phi_h} + g_s(\tau))}}{2 \cosh u_\tau} \right) \right) \right\rangle.
$$

(A.17)

Upon substitution of (A.14) and (A.15) in (A.11) and taking the limit $n \to 0$ we find

$$
2^{-\frac{K-1}{K}} Z(u|\tau) = 2^{-\frac{K-1}{K}} \sum_{\tau_1, \ldots, \tau_{K-1}} \delta(\tau_1 \cdots \tau_{K-1}|\tau; 1) \int \prod_{\ell=1}^{K-1} dh_\ell \, W(h_\ell|\tau_\ell)
$$

$$
\delta\left( u - 2^{-1} \sum_{\sigma} \sum_{\tau_1, \ldots, \tau_{K-1}} \prod_\ell \frac{e^{\phi_h(\sigma_\ell)}}{2 \cosh(h_\ell)} \delta(\sigma_1 \cdots \sigma_{K-1}; 1) \right)
$$

$$
= 2^{-\frac{K-1}{K}} \sum_{\tau_1, \ldots, \tau_{K-1}} \delta(\tau_1 \cdots \tau_{K-1}|\tau; 1) \int \prod_{\ell=1}^{K-1} dh_\ell \, W(h_\ell|\tau_\ell) \delta\left( u - \text{atanh}\left( \prod_\ell \tanh(h_\ell) \right) \right).
$$

The above equation gives us our first density evolution equation (23).

To derive the other density evolution equation we will need to make some further ansätze regarding the eigenvectors of the replicated transfer matrices.

### A.3. Replica symmetry ansatz in the replicated transfer matrices

We proceed in the derivation of the remaining density evolution equations by decomposing $V(\hat{P})$ in left and right eigenvectors:

$$
V(\hat{P}) = \sum_{n=0}^{2^{\rho-1}-1} \lambda_n \left| R_n \right\rangle \left\langle L_n \right|.
$$

(A.18)

We write for the largest eigenvalue $\lambda_0$ and its corresponding eigenvectors $\langle L_0 | = \langle L |$ and $| R_0 \rangle = | R \rangle$. The largest eigenvalue fulfills the following eigenvalue equations

$$
\sum_{s', s} \langle s, s| V|s', s'\rangle \langle s', s'| R \rangle = \lambda_0 \langle s, s| R \rangle,
$$

(A.19)

$$
\sum_{s, s'} \langle L| s, s \rangle \langle s, s'| V|s', s'\rangle = \lambda_0 \langle L| s', s' \rangle.
$$

(A.20)

Because of the Perron–Frobenius theorem we can suggest for the eigenvectors $| R \rangle$ and $\langle L |$ following replica symmetric ansätze

$$
\langle s', s'| R \rangle = \sum_{s'} P_R(s') \int dx \, \Phi_R(x|s') e^{\frac{\lambda_0}{2}} \sum_{s} \langle s|x \rangle
$$

$$
\langle L| s, s \rangle = \sum_{s} P_L(s) \int dy \, \Phi_L(y|s) e^{\frac{\lambda_0}{2}} \sum_{s'} \langle s'| y \rangle.
$$
Normalization gives us \[ \sum_{s, s_0} \langle s, s_0 | V | s', s'_0 \rangle \{s', s'_0| R \} \]

\[ = \sum_{s, s_0} P_R(s'_0) W \left[ (s'_0^s) | 0 \right] \Pi_a W \left[ (s^a)' \mid s^a \right] \]

\[ \times \sum_{\tau} 2^{-12(\frac{r-1}{k})+1} \int \Pi_{r=1}^C du_r z(\mu_r|\tau) \]

\[ \times \left\{ \exp \left[ 2^{-1} \sum_a s^a_s \sum_s \log \left( \sum_{\alpha} \Pi_{r} \alpha \frac{e^{i\mu_r \alpha}}{2 \cosh u_r} N(s, \sigma) \right) \right] \right\}_{\rho|s'_0, \tau} \]

\[ \times \int dx \Phi_R(x|s'_0) \frac{e^{i\sum_s s^a_s}}{(2 \cosh(x))^{\rho}}. \]

Requiring that the above equation equals to

\[ \lambda_0 P_R(s_0) \int dx \Phi_R(x|s_0) \frac{e^{i\sum_s s^a_s}}{(2 \cosh(x))^{\rho}} \]

results in

\[ \lambda_0 P_R(s_0) \Phi_R(x|s_0) = \sum_{s'} P_R(s'_0) W \left[ s'_0 | s_0 \right] \]

\[ \sum_{\tau} 2^{-12(\frac{r-1}{k})+1} \int \Pi_{r=1}^C du_r z(\mu_r|\tau) \int dx' \Phi_R(x'|s'_0) \sum_{\rho} P_{\text{chan}}(\rho|\tau, s_0) \]

\[ \delta \left( x - \frac{1}{2} \sum_{s} \log \left( \sum_{s'} W[s'|x] \frac{e^{s^a_{s'}}}{2 \cosh(x)} \right) - \frac{1}{2} \sum_{s} \log \left( \sum_{\alpha} \Pi_{r} \alpha \frac{e^{i\mu_r \alpha}}{2 \cosh u_r} N(s, \sigma) \right) \right). \]

If we integrate over \( x \) we recover

\[ \lambda_0 P_R(s_0) = 2^{C\frac{r-1}{k}+1} \sum_{s'_0} P_R(s'_0) W \left[ s'_0 | s_0 \right] \]

Because of the normalization condition we have \( \lambda_0 = 2^{C\frac{r-1}{k}+1} \) and \( P_R(s_0) = \sum_{s'} P_R(s'_0) W \left[ s'_0 | s_0 \right] \). The distribution \( \Phi_R(x|s_0) \) fulfills thus

\[ \Phi_R(x|s_0) = \sum_{s'_0} W \left[ s'_0 | s_0 \right] \]

\[ \sum_{\tau} \frac{1}{2} \int \Pi_{r=1}^C du_r z(\mu_r|\tau) \int dx' \Phi_R(x'|s'_0) \sum_{\rho} P_{\text{chan}}(\rho|\tau, s_0) \]

\[ \delta \left( x - \frac{1}{2} \sum_{s} \log \left( \sum_{s'} W[s'|x] \frac{e^{s^a_{s'}}}{2 \cosh(x)} \right) - \frac{1}{2} \sum_{s} \log \left( \sum_{\alpha} \Pi_{r} \alpha \frac{e^{i\mu_r \alpha}}{2 \cosh u_r} N(s, \sigma) \right) \right) \]

which is the density evolution equation (24).
For the left eigenvector equation we have analogously

$$P_L(s'_0) = \sum_{s_0} W(s'_0|s_0) P_L(s_0)$$

and

$$\Phi_L(x|s'_0) = \sum_{s_0} \frac{W(s'_0|s_0) P_L(s_0)}{P_L(s'_0)}$$

$$\sum_{r=1}^{C} \frac{1}{2} \int \prod_{r=1}^{C} du_r \, Z(u_r|\tau) \int dx' \Phi_L(x'|s_0) \sum_{\rho} P_{\text{chan}}(\rho\sigma|\sigma, s_0)$$

$$\delta \left( x - \frac{1}{2} \sum_{s'_0} \log \left( \sum_{x} W(s'|x) \frac{e^{y's}}{2 \cosh(x')} \left( \sum_{\sigma} \prod_{r} \frac{e^{y u_r \sigma}}{2 \cosh u_r} \frac{e^{\rho(h_{\sigma}(s)+g_{\sigma}(s))}}{N(s, \sigma)} \right) \right) \right)$$

which is the density evolution equation (25).

The density evolution equation for $W$ can be derived as follows. We take the limit $n \to 0$ of the self-consistent equation (A.11) describing $P$ by using equation (A.17) for the matrix elements of $Q$ and retaining only the largest eigenvalue $\lambda_0$ in the decomposition of $V$ in eigenvectors. Hence, we are left with the expression

$$P(\sigma, \sigma) = \frac{\text{Tr}[V^{-1}Q(\sigma, \sigma)]}{\text{Tr}[V]} = \frac{\sum_{x, s, s'} \langle L|x_0, s \rangle \langle s_0, s|Q(\sigma, \sigma)|s_0, s' \rangle \langle s_0, s'|R \rangle}{\lambda \sum_{x, s_0} \langle L|x_0, s \rangle \langle s_0, s|R \rangle}$$

$$= \sum_{x, s_0, s'} \int dx \, P_L(s_0) P_R(s'|s_0) \Phi_L(x|s_0) \Phi_R(x|s') e^{y \sum_{a} \alpha_a} e^{y' \sum_{a} \alpha_a}$$

$$\frac{2^{-C-1} K^{-1}}{2^{-C} K^{-1} + 1} W \left( \left( s_0 \right)^y | e^0 \right) \Pi_{\alpha} W \left( \left( s' \right)^y | e^0 \right)$$

$$\times \int \prod_{r=1}^{C-1} du_r \, Z(u_r|\sigma) \exp \left[ 2^{-1} \sum_{a} \alpha_a \right]$$

$$\times \sum_{s} \log \left( \prod_{r} \frac{e^{y u_r \sigma}}{2 \cosh u_r} \frac{e^{\rho(h_{\sigma}(s)+g_{\sigma}(s))}}{N(s, \sigma)} \right)_{\rho, \sigma}. \tag{A.21}$$

If we substitute in (A.21) the replica symmetric ansatz for $P(\sigma)$ and take $n \to 0$ we recover the density evolution equation for $W$.
\[ W(h|\sigma) = 2 \sum_{s_0,s_0'} \mathcal{P}_h(s'|s_0) W\left(\sum_{s_0'}|s_0'\rangle \right) \mathcal{P}_l(s_0) \]

\[ \times \left( \int dx dy \, \mathcal{F}_u \left( y | s_0 \right) \mathcal{F}_u \left( x | s_0' \right) \int \prod_{\tau=1}^{C-1} du_{\tau} Z(u_{\tau}|\sigma) \right) \sum_{\rho} \mathcal{P}_{\text{chan}}(\rho \sigma | \sigma, s_0) \]

\[ \delta \left[ h - \sum_{r=1}^{C-1} u_r - \frac{1}{2} \sum_{\tau} \log \left( \sum_{s,s'} \exp(s'x + sy) W[s'|s] \frac{\exp[\rho(h(s)\tau + g(s)\sigma)]}{N(s, \sigma \tau)} \right) \right] \]

References

[1] Newman M 2010 Networks: An Introduction (Oxford: Oxford University Press)
[2] Dorogovtsev S N and Mendes J F F 2003 Evolution of Networks: From Biological Networks to the Internet and WWW (Oxford: Oxford University Press)
[3] Newman M, Barabási A and Watts D J 2006 The Structure and Dynamics of Networks (Princeton, NJ: Princeton University Press)
[4] Watts D J and Strogatz S H 1998 Nature 393 440
[5] Erdős P and Rényi A 1959 Publ. Math. 6 290
[6] Barrat A and Weigt M 2000 Eur. Phys. J. B 13 547–60
[7] Herrero C P 2002 Phys. Rev. E 65 066110
[8] Nkoletoypoulou T, Coolen A C C, Pérez Castillo I, Skantzos N S, Hatchett J P L and Wemmenhove B 2004 J. Phys. A: Math. Gen. 37 6455
[9] Nkoletoypoulou T and Coolen A C C 2004 J. Phys. A: Math. Gen. 37 8433
[10] Baxter R J 1982 Exactly Solved Models in Statistical Mechanics (New York: Academic)
[11] Viana L and Bray A J 1985 J. Phys. C: Solid State Phys 18 3037
[12] Monasson R 1998 J. Phys. A: Math. Gen. 31 513
[13] Vicente R, Saad D and Kabashima Y 1999 Phys. Rev. E 60 5352
[14] Montanari A and Sours N 2000 Eur. Phys. J. B 18 107
[15] Murayama T, Kabashima Y, Saad D and Vicente R 2000 Phys. Rev. E 62 1577
[16] Neri I, Skantzos N S and Bollé D 2008 J. Stat. Mech. P10018
[17] Tanaka T and Saad D 2003 J. Phys. A: Math. Gen. 36 11143
[18] Skantzos N S, van Mourik J, Saad D and Kabashima Y 2003 J. Phys. A: Math. Gen. 36 11131
[19] Mora T and Rivoire O 2006 Phys. Rev. E 74 056110
[20] Montanari A 2005 IEEE Trans. Inform. Theory 51 3221
[21] Migliorini G and Saad D 2006 Phys. Rev. E 73 026122
[22] Kabashima Y and Saad D 2004 J. Phys. A: Math. Gen. 37 R1
[23] MacKay D J C and Neal R M 1996 Electron. Lett. 32 1645
[24] Gallager R G 1963 Low Density Parity Check Codes (Cambridge, MA: MIT Press)
[25] Luby M G, Mitzenmacher M, Shokrollahi M A and Spielman D A 2001 IEEE Trans. Inform. Theory 47 569
[26] Richardson T J, Shokrollahi M A and Urbanke R L 2001 IEEE Trans. Inform. Theory 47 619
[27] Elliott E O 1963 Bell Syst. Tech. J. 42 1977
[28] Gilbert E N 1960 Bell Syst. Tech. J. 39 1253
[29] Wang H S 1995 IEEE Trans. Vehicular Technol. 44 163–71
[30] Saleh P, Kennedy R, Rapajic P and Shams R 2008 IEEE Signal Proc. Mag. 25 57
[31] Bolle D, Heylen R and Skantzos N S 2006 Phys. Rev. E 74 056111
[32] Bolle D and Heylen R 2008 Phys. Rev. E 77 046104
[33] Eckford A W, Kschischang F R and Pasupathy S 2005 IEEE Trans. Inform. Theory 51 3872
[34] Wang C, Kulkarni S R and Poor H V 2005 IEEE Trans. Inform. Theory 51 4216
[35] Berrou C, Glavieux A and Thitimajshima P 1993 Proc. IEEE Int. Conf. Conf. 2 1064
[36] Mushkin M and Bar-David I 1989 IEEE Trans. Inform. Theory 35 1277
[37] Richardson T J and Urbanke R L 2001 IEEE Trans. Inform. Theory 47 599
[38] Garcia-Frias J and Villasenor J D 2002 IEEE Trans. Commun. 50 357
[39] Sourlas N 1989 Nature 339 693
[40] Eckford A W 2004 Low-density parity-check codes for Gilbert–Elliott and Markov-modulated channels PhD Thesis University of Toronto
[41] Mori R and Tanaka T 2011 Statistical mechanical analysis of low-density parity-check codes on general Markov channel SITA 2011 (Ousyuku, Iwate, Japan Nov.29–Dec.2) arXiv:1110.1930v1
[42] Abou-Chacra R, Anderson P W and Thouless D J 1973 J. Phys. C: Solid State Phys. 6 1734–52
[43] Mézard M and Parisi G 2001 Eur. Phys. J. B 20 217
[44] Mézard M and Montanari A 2009 Information, Physics and Computation (Oxford: Oxford University Press)
[45] MacKay D 1999 IEEE Trans. Inform. Theory 45 399–431
[46] Franz S, Mézard M, Ricci-Tersenghi F, Weigt M and Zecchina R 2001 Europhys. Lett. 55 465
[47] Wemmenhove B, Nikoletopoulos T and Hatchett J P L 2005 J. Stat. Mech. P11007
[48] Huang H 2014 arXiv:1407.0779
[49] Colavolpe G 2006 IEEE Trans. Wireless Commun. 5 1757
[50] Ruján P 1993 Phys. Rev. Lett. 70 2968
[51] Montanari A 2005 IEEE Trans. Inform. Theory 51 3221