HAL
open science

# Fast Simulation for the Performance Evaluation of LDPC Codes using Fast Flat Histogram Method

Shahkar Kakakhail, Sylvain Reynal, David Declercq, Vincent Heinrich

## To cite this version:

Shahkar Kakakhail, Sylvain Reynal, David Declercq, Vincent Heinrich. Fast Simulation for the Performance Evaluation of LDPC Codes using Fast Flat Histogram Method. IEEE Sarnoff Symposium, 2008, Princeton, United States. pp.1-5, 10.1109/SARNOF.2008.4520050 . hal-00512276

**HAL Id: hal-00512276**

**https://hal.science/hal-00512276v1**

Submitted on 29 Aug 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fast Simulation for the Performance Evaluation of LDPC Codes using Fast Flat Histogram Method

S. Kakakhail[†*], S. Reynal[†], D. Declercq[†], V. Heinrich[*]
†ETIS/ENSEA/UCP/CNRS UMR-8051
95014 Cergy-Pontoise, France
kakakhail,reynal,declercq@ensea.fr
*STMicroelectronics
Crolles, France
vincent.heinrich@st.com

## Abstract

*The quality of transmission in digital communication systems is usually measured by Frame Error Rate (FER). The time taken by standard Monte Carlo (MC) simulation to estimate the FER increases exponentially with the increase in Signal-to-Noise Ratio (SNR). In this correspondence, we present an Adaptive Importance Sampling (AIS) technique inspired by statistical physics called Fast Flat Histogram (FFH) method to evaluate the performance of LDPC codes with a reduced simulation time. The FFH method employs Wang Landau algorithm based on a Markov Chain Monte Carlo (MCMC) sampler and we managed to decrease the simulation time by a factor of 13 to 173 for LDPC codes with block lengths up to 2640 bits.*

## 1 Introduction

In digital communication systems, the quality of transmission system is usually measured by Frame Error Rate (FER) i.e., the ratio between the decoded frames that contain errors to the total number of frames sent through the system. With the advancement in the code design and better decoders, it has become very important to gauge the performance of the system at very low error rates. On the one hand, there are novel applications operating at very low error probability and standard Monte Carlo (MC) simulation takes extremely long to gauge their performance. On the other hand, new codes and decoders compete for a better performance in the error floor region where the performance evaluation is curbed and it is quite difficult to study the system properties.

The LDPC codes make a class of error correcting codes which are graphically represented by Tanner graph [1]. In the high Signal-to-Noise Ration (SNR) region, the probability of error is dominated by decoder failures which do not correspond to erroneous codewords [2]. This is due to the convergence of decoder towards pseudo-codewords also known as Trapping Sets (TS) [3]. Since the errors in the high SNR region are dominated by TS, a lot of work is oriented to study the LDPC codes performance in terms of TS. The main problem with the methods that aim at reducing simulation time based on the graph structure is that the identification of TS becomes extremely cumbersome if the graph connectivity is irregular or the code is long. In addition, the TS are not easily generalizable when the errors in the channel are produced owing to the presence of Additive White Gaussian Noise (AWGN) and an iterative soft decision decoding is employed.

Importance Sampling (IS) [4] is a technique where the probability density of events is biased in a controlled way (based on a biasing function) in order to get the important events more frequently. For linear block codes, IS was first used in [5] where a block length of 96 was used making use of the code structure to produce noise events more frequently. A recent approach [6] also makes use of both graph structure and IS. Multiple error impulses are applied on specific nodes in the graph to tease out the dominant TS. Biasing function is calculated based on these TS and IS is then employed by producing noise events in the dominant TS regions to get errors more frequently and FERs are calculated.

Another approach to simulate the error probability, which is not dependent on graph topologies, is the advanced version of IS known as Adaptive Importance Sampling (AIS). In AIS, the probability density function is biased in a controlled way during multiple iterations thus making it possible to visit the tails of the noise distribution.

The system response (the output of the BP decoder in our case) is evaluated and stored in the process. An example of AIS technique, which does not take into account the graph structure of LDPC codes, has been successfully employed in [2] where the authors use a Dual Adaptive Importance Sampling (DAIS) technique using Multicanonical approach [7] based on Berg's recursion equations [8]. DAIS has been tested on a regular very short LDPC code of 96 bits.

AIS is especially interesting for its genericity in the sense that it can be adapted to any stationary memoryless channel (AWGN, BSC, etc.), to any type of decoder (Gallager B, BP, BCJR, etc.) and to any class of codes (regular and irregular LDPC codes, Turbocodes, etc.). In this paper, we will focus on LDPC codes used on AWGN channel decoded with standard Belief Propagation (BP). We propose to bypass the limitations of DAIS by using another AIS approach inspired by statistical physics called Fast Flat Histogram (FFH) method. We will show in particular that our method is still robust for relatively large codeword lengths up to 2640 coded bits.

The rest of the paper is organized as follows: Section 2 describes the FFH method detailing how it is applied to regular and irregular LDPC codes. Section 3 gives a comparison between DAIS and FFH showing that unlike DAIS, FFH is not dependent on histogram entries and is thus easily extendible to any codelength owing to a different update procedure. Section 4 gives the results for some typical test codes and quasi-cyclic codes from IEEE 802.11 standard using AWGN channel and an iterative soft decision decoder with BP in the probability domain. Section 5 concludes our work.

# 2 Fast Flat Histogram Method

## 2.1 Rationale

In statistical physics, the Fast Flat Histogram method [9] employing Wang Landau algorithm [10, 11] was introduced to estimate the density of states $g(E)$, i.e., the number of all possible states for an energy level $E$ of the system. The algorithm is based on the observation that if a random walk in energy space is performed by flipping spins randomly for a spin system and the probability to visit a given energy level $E$ is proportional to the reciprocal of the density of states $1/g(E)$, then a flat histogram is generated for the energy distribution. This is accomplished by modifying the estimated density of states in a systematic way to produce a flat histogram over the allowed range of energy and simultaneously making the density of states converge to the true value.

## 2.2 Description

The basic skeleton of our technique is the same as that in DAIS [2], that is we aim at increasing the number of events in the tails of the probability density function (pdf) by sampling from a biased pdf [12]. However, our technique is somewhat different and solves some issues confronted by DAIS as will be explained in section 3.

We recall the notations from [2]. Let $\Gamma$ be the $n$-dimensional probability space of the noise in the $n$ bits of a codeword. The noise vector $\mathbf{z} = (z_1, z_2, ..., z_n)$ is a multivariate Gaussian with joint pdf $\rho(\mathbf{z}) = \prod_{l=1}^{n} \rho_l(z_l)$. The transmitted bit vector is represented by $\mathbf{b} = (b_1, b_2, ..., b_n)$ and $\mathbf{y} = (y_1, y_2, ..., y_n)$ represents the received codeword. The algorithm is controlled by a scalar control Quantity $V$ given as $V(\mathbf{z}) = \left[\frac{1}{n}\sum_{l=1}^{n}[H(q_l z_l)z_l]^2\right]^{1/2}$ where $q_l = (-1)^{b_l}$ while $b_l$ is the transmitted bit in the $l$th position and $H(x) = 1$ if $x < 0$ and $H(x) = 0$ otherwise. $V(\mathbf{z})$ is contructed such that a noise component $z_l$ contributes to $V$ only if it may produce a bit error at the input to the decoder.

Given a range $[V_{min}, V_{max}]$ for $V$, $\Gamma$ is partitioned into $L$ subsets $\Gamma_k = \{\mathbf{z} \in \Gamma \mid V_{k-1} \leq V(\mathbf{z}) < V_k\}$ , where $V_k = Vmin + k\Delta V$, $1 \leq k \leq L$ and $\Delta V = V_k - V_{k-1} = (V_{max} - V_{min})/L$ is the width of each bin in the partition of $[V_{min}, V_{max}]$. The number of bins depends on the code length and on the signal-to-noise ratio. We observe that an optimized number of bins is obtained by $L = 10^{\frac{1}{\sigma}} \times \frac{n}{10}$ where $\sigma$ represents the standard deviation corresponding to the $E_b/N_0$ value.

Let $P_k$ be the probability of selecting a realization $\mathbf{z}$ from $\rho$ such that $\mathbf{z} \in \Gamma_k$ [12, 8]. Then,

$$P_k = \int_{\Gamma} \chi_k(\mathbf{z}) \frac{\rho(\mathbf{z})}{\rho^*(\mathbf{z})} \rho^*(\mathbf{z}) d\mathbf{z} \approx \frac{1}{N} \sum_{i=1}^{N} \chi_k(\mathbf{z}^{*,i}) \frac{\rho(\mathbf{z}^{*,i})}{\rho^*(\mathbf{z}^{*,i})} \tag{1}$$

where $\rho^*(\mathbf{z})$ is a positive biasing pdf, $\chi_k = 1$ if $\mathbf{z} \in \Gamma_k$ and $\chi_k(\mathbf{z}) = 0$ otherwise. $\mathbf{z}^{*,i}$ are $N$ random sample points in $\Gamma$ selected according to the pdf $\rho^*(\mathbf{z})$. The variance of the estimate of (1) is zero if the optimal biasing pdf $\rho^*_{opt}(\mathbf{z}) = \chi_k(\mathbf{z})\rho(\mathbf{z})/P_k$ is used. However, $\rho^*_{opt}(\mathbf{z})$ depends on $P_k$ which is initially unknown. In standard IS, one uses physical intuition to guess a biasing pdf that is close to $\rho^*_{opt}$. Like DAIS, the FFH method instead iterates over a sequence of biasing pdfs $\rho^{*,j}$ that approach $\rho^*_{opt}$. We define $\rho^{*,j}$ for $j$th iteration by $\rho^{*,j}(\mathbf{z}) = \rho(\mathbf{z})/(c^j P_k^j)$ where $k$ is such that $\mathbf{z} \in \Gamma_k$ is satisfied. The quantities $P_k^j$ satisfy $P_k^j > 0$ and $\sum_{k=1}^{M} P_k^j = 1$ and $c^j$ is an unknown constant that ensures $\int_{\Gamma} \rho^{*,j}(\mathbf{z}) d\mathbf{z} = 1$. The vector $P_k$ completely determines the bias and is initialized with $1/L, \forall k = 1, ..., L$.

By employing Metropolis algorithm [13], we produce a random walk of *samples* $\mathbf{z}^{*,i}$ whose pdf equals $\rho^{*,j}(\mathbf{z})$. We consider a Markov chain of transitions consisting of small steps in the noise space. Each transition goes from $\mathbf{z}^{*,i} = \mathbf{z}_a^* \in \Gamma_{k_a}$ to $\mathbf{z}_b^* = (\mathbf{z}_a + \epsilon\Delta\mathbf{z}) \in \Gamma_{k_b}$ where $\Delta\mathbf{z}$ is random and symmetric, i.e., it does not favor any direction in $\Gamma$ and the transition is accepted with probability $\pi_{ab}$. $\epsilon$ here is the perturbation constant. If a transition from $\mathbf{z}^{*,i}$ to $\mathbf{z}_b^*$ is accepted, we set $\mathbf{z}^{*,i+1} = \mathbf{z}_b^*$, else we set $\mathbf{z}^{*,i+1} = \mathbf{z}^{*,i} = \mathbf{z}_a^*$. The ratio $\pi_{ab}/\pi_{ba}$ equals $\rho^{*,j}(\mathbf{z}_b^*)/\rho^{*,j}(\mathbf{z}_a^*)$ which is the *detailed balance equation* that ensures that the limiting (stationary) pdf for infinitely many steps of this random walk is $\rho^{*,j}$ [13].

We consider the perturbation of the noise component in each bit $z_{a,l}^*$ of $\mathbf{z}_a^*$ separately and accept it or reject it independently with the probability $\min[\rho(z_{b,l}^*)/\rho(z_{a,l}^*), 1]$. We pick each perturbation $\Delta z_l$ from a zero mean symmetric pdf. We obtain a trial state $\mathbf{z}_b^*$ in which only some of the components are different from their previous values in $\mathbf{z}_a^*$. Then we compute $k_b$, the bin corresponding to $\mathbf{z}_b^*$ and finally accept the step from $\mathbf{z}_a^*$ to $\mathbf{z}_b^*$ with the probability $\min[P_{k_a}^j)/P_{k_b}^j), 1]$. The compound transition probability thus becomes

$$\pi_{ab} = \left\{ \prod_{l=1}^{n} \min\left[\frac{\rho(z_{b,l}^*)}{\rho(z_{a,l}^*)}, 1\right] \right\} \min\left[\frac{P_{k_a}^j}{P_{k_b}^j}, 1\right] \qquad (2)$$

The Asymptotically Optimal Acceptance Rate AOAR $\alpha \triangleq$(number of accepted steps)/(total number of steps) for a Metropolis algorithm for target distributions with IID components is 0.234 [14]. The perturbation constant $\epsilon$ is adjusted so as to keep $\alpha$ close to this value. The noise realizations are recorded in the histogram $H^{*,j}$ where $H_k^{*,j} = \sum_{i=1}^{N} \chi_k(\mathbf{z}^{*,i})$ is the number of $\mathbf{z}^{*,i}$ in iteration $j$ that fall into $\Gamma_k$. To keep a record of errors in bin $k$, we produce an error histogram $G_k^{*,j}$. $P_k$ is updated *on the fly* such that when $k$ bin is visited, $P_k$ is modified by the refinement parameter $f > 1$, i.e. $P_k \to P_k \cdot f$ [10, 11]. In practice, we have to use the log domain $\ln P_k \to \ln P_k + \ln f$ in order to fit all possible $P_k$ into double precision numbers. If the random walk rejects a possible move and stays in the same bin $k$, we modify the same $P_k$ with the modification factor to keep the detailed balance equation in equilibrium.

The histogram $H_k^{*,j}$ is checked after about each $L \times 10$ Monte Carlo (MC) sweeps. When the histogram is flat (flatness criterion is the same as in [10, 11]), the modification factor is reduced to a finer one using the function $f_{j+1} = \sqrt{f_j}$ ($f_{init} = e = 2.7182818$), the histogram is reset and the next iteration of random walk is started where $P_k$ are now modified with the finer modification factor. We continue doing so until the histogram is flat again and then we begin the next Wang-Landau (WL) iteration with a finer $f$ and so on. We stop the random walk when the change from one WL iteration to the other is "quite small". The above detailed random walk can also be carried out in a parallel fashion by dividing the range $[V_{min}, V_{max}]$ ito $W$ partitions and then exploring each partition separately, combining the results in the end.

It is extremely important to determine the optimum $[V_{min}, V_{max}]$ interval since the accuracy and speed of the simulation depends heavily on it. Our aim is to explore the whole of probabilty space $\Gamma$ using random walk [15]. $[V_{min}, V_{max}]$ is initialized to $[0, 1]$ and this interval is divided into L bins. Now the random walk is performed to determine the optimum $[V_{min}, V_{max}]$ interval. The value of P is updated for every Markov Chain transition during the walk. After a number of steps (we use $L \times E_b/N_0$ steps), the walk is ceased and the farthest bins on either side are detected which were approached by the random walk. These two bins on either side determine the optimum $[V_{min}, V_{max}]$ interval.

Let $P_{err}$ be the probability that a received word with noise realization $\mathbf{z}$ selected from $\rho$ leads to an error and $P_{err,k}$ the probability that $\mathbf{z}$ leads to an error *and* falls into bin $k$. Then

$$P_{err,k} = P_{err|k} P_k \qquad (3)$$

$$P_{err} = \sum_{k=1}^{M} P_{err,k} \qquad (4)$$

where $P_{err|k}$ is the conditional probability of an error *given* that $\mathbf{z}$ falls into bin $k$. We can approximate $P_{err|k} \approx P_{err|k}^{j_{max}} = \sum_{j=1}^{j_{max}} G_k^{*,j} / \sum_{j=1}^{j_{max}} H_k^{*,j}$ after $j_{max}$ iterations where $j_{max}$ is the iteration when $f$ gets very close to 1 and we stop further refinement of the modification factor. Using (3) and (4), we get $P_{err}$.

# 3 Comparison between DAIS and FFH Method

In this section, we explain in detail the main difference between FFH method and DAIS of [2]. We also point out why the FFH method partially solves the limitations confronted by DAIS.

DAIS is using Berg's recursion equations which are explained as follows:

To update $P_k^j$ at the end of iteration $j$, $P_1^{j+1}$ is initially set to an arbitrary positive value. Then, the recursion

equations are [12, 8]

$$P_{k+1}^{j+1} = \frac{P_k^{j+1} P_{k+1}^j}{P_k^j} \left( \frac{H_{k+1}^{*,j}}{H_k^j} \right)^{\hat{g}_k^j} \tag{5}$$

where

$$\hat{g}_k^j = \frac{g_k^j}{\sum_{l=1}^j g_k^l}, \qquad g_k^l = \frac{H_k^{*,l} H_{k+1}^{*,l}}{H_k^{*,l} + H_{k+1}^{*,l}} \tag{6}$$

where in addition, $\hat{g}_k^j = 0$ if $g_k^j = 0$ and $g_k^l = 0$ if $H_k^{*,l} + H_{k+1}^{*,l} = 0$. The exponent $0 \leq \hat{g}_k^j \leq 1$ hence depends on all previous iterations. Finally, $P_k^{j+1}$ is normalized so that $\sum_{k=1}^L P_k^{j+1} = 1$.

The histograms are accumulated linearly so obviously the value $\hat{g}_k^j$ will decrease with larger histograms rendering the update process of $P_k$ very slow. Intrinsically, Berg's equations converge to the optimum value of $P_k$ without sampling enough smaller noise realizations leading to errors. To overcome this problem of undersampling, one has to launch another simulation constrained in a different $V$ range.

The above mentioned problem is solved by FFH method employing Wang-Landau Algorithm since the dynamic update of $P_k$ is independant of the histogram values. Modifying $P_k$ at each step of the random walk allows one to approach the optimum value of $P_k$ in a very quick and efficient manner. The noise samples leading to errors are accumulated with enough statistics hence there is no need to carry out another simulation. FFH method thus works for powerful codes having low error floors at low SNR and for codes having large block-lengths.

# 4 Simulation Results

For all our test-benches, Sum-Product (Belief Propagation in probability domain) decoding algorithm has been used. BPSK modulation is employed using symmetric signal levels of +1 and -1 for logical 0s and 1s respectively. An all zeros codeword is transmitted since the code is linear and the noise is symmetric.

Our test benches comprise of three codes obtained from [16] and three quasi-cyclic LDPC codes used in IEEE 802.11 standard. The simulation results are shown in Figs. 1 & 2. The first remark is that our simulation results stick to the MC curves for all codes and for all FERs. This was though expected since the FFH method implements an unbiased estimation of FER. The efficiency of the FFH method as compared to standard MC can be measured by the simulation gain [4] i.e., the ratio of codewords simulated in MC

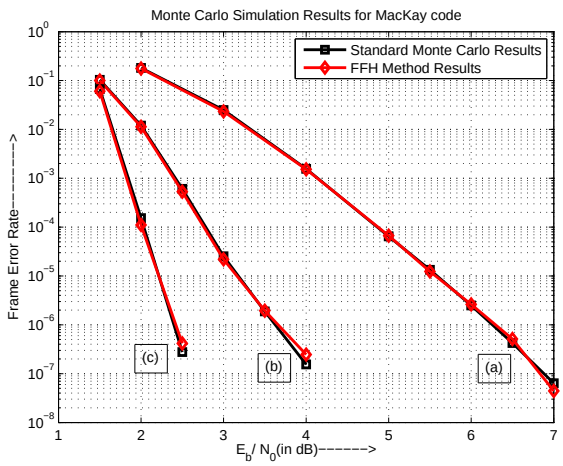


Figure 1: Results for MacKay codes (a) $n$ = 120, $R$ = 0.47 regular (b) $n$ = 504, $R$ = 0.5 irregular (c) $n$ = 2640, $R$ = 0.5 regular

simulation to those simulated in FFH method. The simulation gain for all the codes in test-benches is given in Table 1. The gain is quite impressive and the result is a considerable reduction in simulation time. They are determined for the highest $E_b/N_0$ for the particular code as indicated by the values within parenthesis in column 1. The codewords simulated in case of standard MC simulation are for 100 erroneous frames. Our current experience suggests that the simulation gain increases with decreasing FER but the dependence of FFH on the number of codewords or code

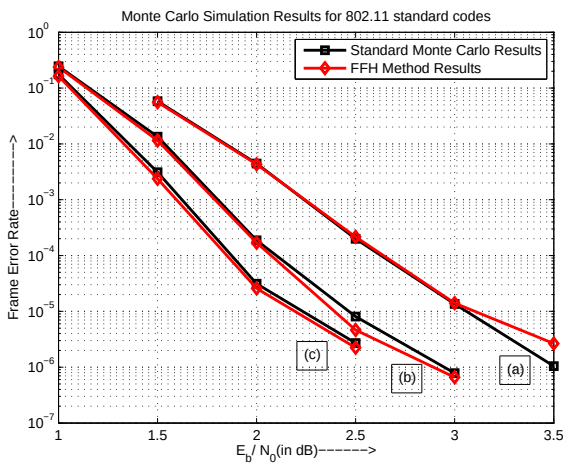


Figure 2: Results for IEEE 802.11 standard Quasi-cyclic irregular codes (a) $n$ = 648, $R$ = 0.5 (b) $n$ = 1296, $R$ = 0.5, (c) $n$ = 1944, $R$ = 0.5

| Code ($E_b/N_0$) | MC Codewords (100 Err. Fr.) | FFH Codewords | Simul. Gain |
|---|---|---|---|
| (120,0.5,reg) (7) | 3594801754 | 63636462 | 56.49 |
| (504,0.5,irr) (4) | 1274520408 | 12081492 | 105.49 |
| (2640,0.5,reg) (2.5) | 1445433333 | 8312500 | 173.88 |
| (648,0.5,irr) (3.5) | 161136000 | 6465688 | 24.92 |
| (1296,0.5,irr) (3) | 324935483 | 6645033 | 48.9 |
| (1944,0.5,irr) (2.5) | 78796296 | 6002175 | 13.13 |

Table 1: Simulation Gain

length is unknown at this time and is a subject of continuing research.

# 5 Conclusions

In this paper, we have presented the use of Fast Flat Histogram Method employing Wang Landau Algorithm in an adaptive importance sampling framework to enhance the simulation speed of FER calculation. Sufficient statistical accuracy is ensured through different parameters. Extention to other types of error correcting codes is possible using this framework and will be reported in future work.

# 6 Acknowledgements

The authors wish to thank Chad A. Cole and R. Holzlohner for valuable discussions related to this work.

# References

[1] R. M. Tanner. "A recursive approach to low complexity codes". *IEEE Trans. Inf. Theory*, vol 27, Sept 1981.

[2] R. Holzlohner, A. Mahadevan, C. Menyuk, J. Morris, and J. Zweck. "Evaluation of the very low BER of FEC codes using dual adaptive importance sampling". *IEEE Comm. Letters*, vol 2, Feb 2005.

[3] T. Richardson. "Eror floors of LDPC codes". *Allerton Conference*, 2001.

[4] R. Srinivasan. *"Importance Sampling - Applications in Communications and Detection"*. Springer-Verlag, 2002.

[5] B. Xia and W. E. Ryan. "On importance sampling for linear block codes". *Proc. IEEE International Conference on Communications (ICC 03)*, pages 2904–2908, 2003.

[6] C. A. Cole, S. G. Wilson, E. K. Hall, and T. R. Giallorenzi. "A general method for finding low error rates of LDPC codes". *Submitted to IEEE Trans. Inform. Theory*, May 2006.

[7] B. A. Berg and T. Neuhaus. "The multicanonical ensemble: a new approach to simulate first-order phase transitions". *Phys. Rev. Lett.*, vol 68:9–12, 1992.

[8] B. A. Berg. "Algorithmic aspects of multicanonical Monte Carlo simulations". *Nucl. Phys. Proc. Suppl.*, vol 63:982–984, 1998.

[9] S. Reynal and H. T. Diep. "Fast flat histogram algorithm for generalized spin models". *Phys. Rev.*, E 72(56710), 2005.

[10] F. Wang and D. P. Landau. "Determining the density of states for classical statistical models: a random walk algorithm to produce a flat histogram". *Phys. Rev.*, E 64(056101), 2001.

[11] F. Wang and D. P. Landau. "Efficient, multiple range random walk algorithm to calculate the density of states". *Phys. Rev. Lett.*, 86(2050), 2001.

[12] R. Holzlohner and C. R. Menyuk. "Use of multicanonical Monte Carlo simulations to obtain accurate bit error rates in optical communications systems". *Opt. Lett.*, vol 28:1894–1896, Oct 2003.

[13] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. M. Teller, and E. Teller. "Equation of state calculations by fast computing machines". *J. Chem. Phys.*, vol 21:1087–1092, 1953.

[14] G. O. Roberts, A. Gelman, and W. R. Gilks. "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms". *Ann. Appl. Probab.*, 7:110–20, 1997.

[15] W. Krauth. "Introduction to Monte Carlo Algorithms". *Advances in computer simulation, lectures held at Eotvos summer school in Budapest, Hungary*, 1996.

[16] David MacKay. Encyclopedia of sparse graph codes. http://www.inference.phy.cam.ac.uk/mackay/codes/EN/C.