Analysis of Spatially Coupled Systems using the Potential Functional with Applications to Coding Theory

THÈSE N^{O} 7358 (2016)

PRÉSENTÉE LE 12 DÉCEMBRE 2016
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE THÉORIE DES COMMUNICATIONS
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS
Rafah EL-KHATIB
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Rafah EL-KHATIB

acceptée sur proposition du jury:

Prof. B. Rimoldi, président du jury
Dr N. Macris, Prof. R. Urbanke, directeurs de thèse
Prof. P. Martinez Olmos, rapporteur
Dr L. Schmalen, rapporteur
Prof. A. Shokrollahi, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

Analysis of Spatially Coupled Systems using the Potential Functional with Applications to Coding Theory

Rafah El-Khatib

EPFL - Ecole Polytechnique Fédérale de Lausanne

Thesis No. 7358 (October 2016)

Thesis presented to the faculty of Computer and Communication sciences for obtaining the degree of Docteur ès Sciences

Accepted by the jury:

Nicolas Macris and **Rüdiger Urbanke**
Thesis directors

Pablo M. Olmos
Expert

Laurent Schmalen
Expert

Amin Shokrollahi
Expert

Bixio Rimoldi
President of the jury

Ecole Polytechnique Fédérale de Lausanne, 2016

Dedicated to my loving parents,
Nada and Bassem,

Sans l'encre, la pensée aurait le sens du vide.
Albert Amon

Acknowledgements

Before joining EPFL, and as I was still wondering which labs I wanted to apply to, I received the following piece of advice from someone[1]: "The most important component of your doctoral studies is your advisor." Four years later, I still think that this was the most important piece of advice I was given, and I can confirm that it is true. The PhD experience can be quite painful at times: the insecurities, the "researcher's block", the existential crises[2], … so it's very important to get support, reassurance, and the occasional push from one's advisor. Fortunately for me, I did not have only one, but two, remarkable advisors. I could not begin to imagine how my experience at EPFL would have been if it were not for them.

I first met Nicolas Macris and Ruediger Urbanke during the Information Theory Workshop (ITW) in 2012, when it was being held at EPFL. They were very busy with the workshop but, as usual, generous with their time, so we met to discuss different research projects that I can work on under their supervision. Although we only spoke for around 20 minutes, Ruediger still managed to explain 9 research projects to me, while Nicolas sat discreetly on a comfortable couch in the corner, nodding away with curiosity in his eyes.

Working with Nicolas has inspired me in many ways. I am still surprised at how much personal investment he puts into every student that comes his way, whether the student is pursuing a PhD under his supervision, doing a semester project with him, or even taking one of his courses. Nicolas has a natural gift for explaining things in very simple terms, and making scary-looking equations seem intuitive. His attitude towards a student is versatile and adapted to the student's needs: in Lebanese we have an expression for this, which says that "he tightens and loosens (the rope)".[3] I will always be grateful for what I learned from Nicolas, his passion for science and knowledge, his fatherly guidance, and his efforts in teaching me how to approach research, and more precisely, how to be precise ☺.

My interactions with Ruediger were of a very different nature. One might assume that most of our interactions involved discussions about research, but I preferred to use my precious time around Ruediger to make fun of him. Ruediger has a very good sense of humor, and he is obviously very witty. However, he is also European, which means that he cares about being politically correct. I mostly used this to my advantage to "win" and thoroughly enjoyed seeing his face consequently turning red. Thank you, Rudi, for being the positive person that you are, for being a very

---

[1] Sorry I forgot who you are, but I really value(d) your advice!

[2] Apparently, deciding to become a farmer is a normal symptom.

[3] Please note that the rope is only metaphorical – in this case.

supportive advisor, and most importantly, for tolerating me.

I had thought that the private PhD defense is a very scary experience, but I was proved wrong by my very kind thesis committee. I would like to gratefully thank Pablo Olmos, Laurent Schmalen, and Amin Shokrollahi for reading my thesis, giving me feedback, and making my defense a positive experience in which we were able to openly discuss ideas and research directions. I would also like to thank Bixio Rimoldi, the president of the committee, whose positive outlook and kind smile helped soothen my wrecked nerves.

Our lab space, IPG, has been home to me for around one third of the past four years. I am happy to have met a very diverse but consistently kind group of people throughout my stay. Nothing would have run so smoothly if it weren't for Muriel and Francoise's superpower organizational skills and Damir's patience with my (admittedly, Windows) computer problems. Although Damir killed my beloved and long-awaited cluster jobs, he very nicely provided consolation with homemade cakes and a secret stash of movies. I deeply regret having only met Holly in my last year of studies. I met her one of my worst days at EPFL, a few days before the draft deadline of my thesis, in sweatpants and *awful* hair. Not only did her bubbly personality completely lift my spirits, but she also managed to help me with my thesis despite having very limited time.

IPG has also been home to many other PhD students who have made the "workplace" a "play-place". Thanks Marco, for *trying* to help me dress better, and for approving of my outfit *once* before I defended (yay!); Mani, for helping me with my endless Linux issues; Jean, for being as weird as me and making funny noises although you're a postdoc, this gives me faith in being weird; Young-Jun, for your optimism and making the effort not to breathe too loudly in the office next-door; Karol and Camila, for your kindness and friendship; Andrei, for always teaching me new things and scaring me away from climbing (as per your previous victims); Marc D., for your joyful willingness to answer my math questions and explain bizarre[4] theorems; Hamed, for first introducing me to Nicolas and Ruediger and for your help towards the end of the tunnel; Vahid, for our passionate discussions about Lebanese food and spatially coupled codes (I'm not sure what you were more excited about!); Mohammad K., for three things, your wordly advice, your philosophical opinions, and I forgot the third one; Adrian, for not microwaving my blind turtle (toy); and everyone else, Dewan, Chun Lam, Wei, Saeid, Saeid, Giel, Lyudmila, Mine, Nicholas, and the past and current members of IPG, for the nice discussions and coffee breaks.

The other third of my life during the past four years was mostly spent with friends (granted, not while writing my thesis). I deeply thank Marwa and Abbas, my family in Lausanne. You are the first people I want to be with when I'm in a happy mood, and the only people I want to be with when I'm not. Ghid has been my flatmate, one of my best friends, and my rolemodel in many respects, interchangeably and even simultaneously. Thank you, Ghid and Dan, for the music, all our chats and brunches, for teaching me how to be diplomatic, and for letting me babysit Dalia (yes, I will!). Wherever you go, you will find Lebanese people, or they will find you. Thank you to the Lebanese community in Lausanne, my home away from home, with which I can be completely myself: Farah, Elie, Hiba, Mohamad, Ghofran, Serj, Sahar, Rajai, and Elio. Special thanks go to Baker for listening to

---

[4] Or in your language, exciting!

me complain and for helping me solve problems of every kind almost everyday this past year at precisely 11 am. I met Ajay and Artem on my first day at EPFL, when all three of us arrived late to the welcoming reception. I learned on that day that sometimes, it's okay to arrive late (just kidding, Ajay, don't be late!), because it just happened that I met the two most awesome fellow PhD students. Ajay, thanks for being our reminder not to sweat the small (or big) stuff, and to just take it ee-hee-zay; and Artem, also known as Ken to many at the Karaoke bar, thanks for being our reminder to always, always, par-tay! Emti, Yuko (, and Zeta!), thanks for having the weirdest (by other people's standards, of course) conversations with me and for bringing the cutest chipmunk into the world. Thank you to those happy spirits who pushed me to get off my couch (or joined me there when I was too cold to go out), with whom I shared nights out, celebrations, game nights, concerts, picnics, and trips: Renata (who can make you laugh no matter the occasion), Manos, Katerina, Jose, Fay, Jean-Eudes, Silvia, Helen, and Roman. Special thanks goes to the friends who colored my getaway adventures away from Lausanne: Rasha Salem, Jad Hachem, Sara Kheireddine, Ribal Abi Raad, Haya Farah, and Farah Khouri. Last but certainly not least, thank you Antonio, the person I met with "low probability and high consequence", for teaching me so many things about art, math, philosophy, history, and more controversially, the human nature; no conversation is ever boring with you. Finally, this thesis would not have been possible without the inspiring music of Amy Winehouse, Mashrou' Leila, and Ibrahim Maalouf; I thank you deeply for your art.

It is not easy to find lifetime friends, those with whom nothing changes even after months or years apart, those who don't mind your tantrums and enjoy naming your multiple personalities. I was very fortunate to meet my best friends in my "younger days" back in Lebanon, but very unfortunate in that we have been living on different parts of the globe since. Thank you Philippo (who also responds to "Ibrahim" for some reason), Zahi, Edmond, John, Mira, Dana, Nicolas, Ramzi, Oussama, Lotfi, and Mariam for being who you are. I will carry you in my heart (and cellphone) wherever I go.

No words or actions will be enough to thank my family for the unwavering love and support they have offered with open arms during every phase of my PhD and life. Family is the only constant. Thank you Teta Hana, Teta Nadia, Jeddo Sami, and Jeddo Nizam for being the basin of strength for our family. And thank you Mom and Dad, my sanctuary, and Nayla, my (significantly better) alter ego, for teaching me the meaning of unconditional love.

Abstract

For the past 70 years or so, coding theorists have been aiming at designing transmission schemes with efficient encoding and decoding algorithms that achieve the capacity of various noisy channels. It was not until the '90s that *graph-based codes*, such as low-density parity-check (LDPC) codes, and their associated low-complexity iterative decoding algorithms were discovered and studied in depth. Although these schemes are efficient, they are not, in general, capacity-achieving.[5] More specifically, these codes perform well up to some *algorithmic threshold* on the channel parameter, which is lower than the *optimal (maximum a-posteriori) threshold*.

The gap between the algorithmic and optimal thresholds was finally closed by *spatial coupling*, a generic graph-based technique that constructs large graphs from underlying probabilistic graphical models by replicating these models on a spatial axis and by interconnecting local replicas. In the context of coding, the belief propagation algorithm on spatially coupled codes yields capacity-achieving low-complexity transmission schemes: The gap between the algorithmic and optimal thresholds of such coupled codes disappears as the number of replicas goes to infinity. The reason behind the optimal performance of spatially coupled codes is "seeding" perfect information on the replicas at the boundaries of the coupling chain. This extra information makes decoding easier near the boundaries, and this effect is then propagated into the coupling chain upon iterations of the decoding algorithm, due to the local connections between adjacent replicas of the underlying graph.

Spatial coupling was also applied to various other problems that are governed by low-complexity message-passing algorithms, such as random constraint satisfaction problems, compressive sensing, and statistical physics. Each system has an associated algorithmic threshold (that depends on its message-passing algorithm) and an optimal threshold (that depends on its "optimal solver"). As with coding, once the underlying graphs are spatially coupled, the algorithms for these systems exhibit optimal performance.

In this thesis, we analyze the performance of iterative low-complexity message-passing algorithms on general spatially coupled systems, and we specialize our results in coding theory applications. To do this, we express the evolution of the state of the system (along iterations of the algorithm) in a variational form, in terms of the so-called *potential functional*, in the continuum limit approximation (obtained by first taking the coupling-chain length, and then the coupling-window size, to infinity).

---

[5] The exception is the capacity-achieving code ensemble designed by Luby, Mitzenmacher, Shokrollahi, and Spielman in [1] for transmission over the BEC.

This thesis consists of two parts. In the first part, we consider the *dynamic phase* of the message-passing algorithm, in which the system parameter is between the algorithmic and optimal thresholds, and iterations of the algorithm modify the state of the spatially coupled system. Assuming that the boundaries of the coupled chain are appropriately “seeded”, we find a closed-form analytical formula for the velocity with which the extra information propagates into the chain. We apply this result to coupled irregular LDPC code-ensembles with transmission over general BMS channels and to coupled general scalar systems, such as compressive sensing and the Gaussian approximation. We perform numerical simulations for several applications and show that our formula gives values that match the empirical, observed velocity. This confirms that the continuum limit is an approximation well-suited to the derivation of the formula.

In the second part of this thesis, we consider the *static phase* of the message-passing algorithm; this occurs when the system parameter is set to the optimal threshold and the algorithm can no longer modify the state of the system. We introduce a novel proof technique that employs *displacement convexity*, a mathematical tool from optimal transport, to prove that the potential functional is strictly displacement convex under an alternative structure in the space of probability measures. We hence establish the uniqueness of the state to which the spatially coupled system converges, and we characterize it. We apply this result to the (ℓ, r)-regular Gallager ensemble with transmission over the BEC and to coupled general scalar systems.

Keywords: spatial coupling, sparse graphical models, density evolution, potential functional, continuum limit, displacement convexity, speed of decoding, wave propagation, compressive sensing.

Résumé

Durant les 70 dernières années, le but de la théorie des codes était de concevoir des systèmes de communication et de transmission des données à l'aide d'algorithmes d'encodage et de décodage efficaces permettant d'atteindre la capacité des canaux bruités. Dans les années 90, les chercheurs ont découvert des *codes basés sur des graphes*, comme les codes à contrôle de parité à faible densité (LDPC), et leurs algorithmes de décodage associés, qui sont itératifs et de basse complexité. Bien que ces régimes soient efficaces, ils n'atteignent pas, en général, la capacité[6]. Plus précisément, ces codes fonctionnent très bien jusqu'à un certain *seuil algorithmique* sur le paramètre du canal, qui est inférieur au *seuil optimal (MAP)*.

Les théoriciens des codes ont finalement réussi à combler l'écart entre le seuil algorithmique et le seuil optimal en utilisant le *couplage spatial*, une technique générique à base de graphes qui construit un grand graphe à partir d'un modèle graphique probabiliste de base (non-couplé) en étalant des copies de ce modèle sur un axe spatial et en connectant les copies localement. Dans le contexte des codes correcteurs d'erreurs, l'algorithme de propagation des croyances (BP) sur les codes spatialement couplés produit des systèmes de transmission de basse complexité *et* qui atteignent la capacité: l'écart entre le seuil algorithmique et le seuil optimal de ces codes couplés disparaît quand le nombre de copies tend vers l'infini. Cette performance optimale est attribuée à "l'ensemencement" des informations parfaites sur les copies au niveau des frontières de la chaîne de couplage. Cette information supplémentaire rend le décodage plus facile à proximité des frontières, et cet effet est ensuite propagé dans la chaîne de couplage lors des itérations de l'algorithme de décodage en raison des connections locales entre les copies voisines.

Le couplage spatial a été appliqué non seulement aux systèmes de codage mais aussi à divers autres problèmes qui sont régis par des algorithmes de basse complexité connus sous le nom d'algorithmes de passage de messages, comme les problèmes de satisfaction de contraintes aléatoires, l'acquisition comprimée ou la physique statistique. Pour chacun de ces systèmes il existe un seuil algorithmique associé à l'algorithme de passage de messages et un seuil optimal associé au "solveur optimal". Pour toutes ces applications, les algorithmes de passage de messages présentent des performances optimales lorsque les graphes de base sont spatialement couplés.

Dans cette thèse, nous analysons les performances des algorithmes de passage de messages itératifs et de basse complexité sur des systèmes généraux couplés, et nous spécialisons nos résultats aux applications de codage. Pour ce faire, nous exprimons l'évolution de l'état du système (lors des itérations de l'algorithme) sous une forme

---

[6] avec l'exception des codes conçus par Luby, Mitzenmacher, Shokrollahi et Spielman [1].

variationnelle avec une fonctionnelle que nous appelons *le potentiel*, dans la limite asymptotique de la taille du graphe couplé (qui est obtenue en faisant tendre d'abord la longueur de la chaîne de couplage, et ensuite la taille de la fenêtre de couplage, vers l'infini).

Cette thèse est divisée en deux parties. Dans la première partie, nous considérons la *phase dynamique* de l'algorithme de passage de messages, lorsque le paramètre du système est entre le seuil algorithmique et le seuil optimal, et des itérations de l'algorithme modifient l'état du système couplé. En supposant que les frontières de la chaîne couplée sont convenablement “ensemencées”, nous trouvons une formule analytique pour la vitesse avec laquelle les informations supplémentaires se propagent dans la chaîne de couplage. Nous appliquons ce résultat sur les codes irréguliers à contrôle de parité à faible densité (LDPC) lors de transmissions sur un canal général à entrées binaires sans mémoire et symétriques (BMS). De même, nous considérons aussi des systèmes couplés scalaires généraux comme ceux de l'acquisition comprimée et l'approximation Gaussienne. Nous effectuons des simulations numériques pour plusieurs applications et montrons que la formule donne des valeurs qui correspondent à la vitesse empirique (observée). Cela confirme que la limite asymptotique de la taille du graphe est une approximation bien adaptée à la dérivation de la formule.

Dans la deuxième partie de cette thèse, nous considérons la *phase statique* de l'algorithme de passage de messages, lorsque le paramètre du système est égal au seuil optimal et que l'algorithme ne réussit plus à modifier l'état du système couplé. Nous introduisons *la convexité de déplacement*, outil mathématique emprunté à la théorie du transport, pour prouver que le potentiel est strictement convexe par rapport à une structure alternative dans l'espace de fonctions de densité de probabilité. Cela établit l'existence et l'unicité de l'état vers lequel le système couplé converge. Nous caractérisons cet état et nous appliquons ce résultat à l'ensemble de codes Gallager réguliers lors de transmissions sur le canal d'effacements (BEC) ainsi que sur des systèmes couplés scalaires généraux.

Mots clés: couplage spatial, modèles graphiques probabilistes dilués, équations d'évolution de densité, potentiel, la limite de grande taille (continue), convexité de déplacement, vitesse de décodage, propagation d'onde, l'acquisition comprimée.

Contents

Acknowledgements **iii**

Abstract **vii**

Résumé **ix**

Contents **xi**

List of Figures **xiv**

1 Introduction **1**
- 1.1 The Coding Problem 1
- 1.2 Channel Encoding via Low-Density Parity-Check Codes 5
- 1.3 Channel Decoding via Message-Passing 7
  - 1.3.1 Bit-wise MAP Decoding 7
  - 1.3.2 Factor Graphs and Marginalization 8
  - 1.3.3 Belief Propagation Equations 10
  - 1.3.4 Simplification of Belief Propagation Equations 12
- 1.4 Tools for Performance Analysis 16
  - 1.4.1 Two Basic Simplifications 17
  - 1.4.2 The LDPC(λ, ρ) Configuration Model 18
  - 1.4.3 Density Evolution on the BEC 19
  - 1.4.4 Density Evolution on BMS Channels 22
  - 1.4.5 The Potential Function 26
- 1.5 Spatial Coupling 30
  - 1.5.1 Spatially Coupled Configurations and Threshold Saturation . . 30
  - 1.5.2 Coupled Density Evolution Equations 38
  - 1.5.3 Coupled System Potential Functionals 41
  - 1.5.4 Thresholds 42
- 1.6 The Continuum Limit 44
- 1.7 Displacement Convexity 48
- 1.8 Main Contributions 50

2 Decoding Velocity for Irregular Codes on the BMS **55**
- 2.1 Introduction 55
- 2.2 Preliminaries 58
  - 2.2.1 Single System 58

2.2.2 Spatially Coupled System 59
2.2.3 Phenomenological Observations 61
2.3 Continuum Limit and Main Result 63
2.3.1 Continuum Limit 63
2.3.2 Statement of Main Result 64
2.4 Derivation of Main Result 65
2.4.1 Density Evolution as Gradient Descent 65
2.4.2 Final Steps of the Derivation 66
2.5 Applications and Numerical Experiments 68
2.5.1 Binary Erasure Channel (BEC) 68
2.5.2 Gaussian Approximation (GA) 74
2.6 Application to Scaling Laws for Finite-Length Coupled Codes 77
2.7 Appendix 79
2.7.1 Derivation of Equation (2.21) 79
2.7.2 Derivation of Equation (2.32) 80
2.7.3 Derivation of Expression (2.59) 80

3 Propagating Wave Velocity for General Scalar Systems 83
3.1 Introduction 83
3.2 Preliminaries 85
3.2.1 Phenomenological Observations 87
3.3 Continuum Limit and Main Result 87
3.3.1 Continuum Limit 87
3.3.2 Statement of Main Result 87
3.4 Derivation of Main Result 88
3.5 Applications and Numerical Experiments 89
3.5.1 Generalized LDPC (GLDPC) Codes 89
3.5.2 Compressive Sensing 90

4 Displacement Convexity for (ℓ, r)-regular Codes on the BEC 95
4.1 Introduction 95
4.2 Preliminaries 97
4.2.1 Single System 97
4.2.2 Spatially Coupled System 98
4.3 Continuum Limit and Main Results 99
4.3.1 Continuum Limit 99
4.3.2 Main Results 101
4.4 Existence of Minimizing Profile 102
4.4.1 Preliminaries 103
4.4.2 The Direct Method 108
4.5 Displacement Convexity 109
4.5.1 Discussion on Displacement Convexity 109
4.5.2 Displacement Convexity of the Single-Potential 110
4.5.3 Displacement Convexity of the Interaction-Potential 110
4.6 Strict Displacement Convexity 112
4.6.1 Explicit Expressions of Kernel Function 112
4.7 Appendix 114
4.7.1 Proof of Lemma 4.11 114

4.7.2 Proof of Lemma 4.12 114
4.7.3 Proof of Lemma 4.13 116

5 Displacement Convexity for General Scalar Systems 119
5.1 Introduction 119
5.2 Preliminaries 121
5.2.1 Single System 121
5.2.2 Spatially Coupled System 123
5.2.3 Discussion 127
5.3 Rearrangements 129
5.4 Existence of Minimizers 133
5.5 Displacement Convexity 134
5.6 Fixed Points and Minimizers 138
5.7 Strict Displacement Convexity 140
5.8 Illustrations 144
5.8.1 LDPC Code Ensembles on the BEC 144
5.8.2 Generalized LDPC Codes 144
5.8.3 The Gaussian Approximation 146
5.8.4 Compressive Sensing 147
5.9 Appendix 148
5.9.1 Integrability 148
5.9.2 Basic Bounds 150
5.9.3 Rearrangement 151
5.9.4 Minimizers 152
5.9.5 Second Derivative 152

6 Conclusions and Further Directions 155
6.1 Conclusions 155
6.2 Further Directions 157

Bibliography 161

List of Figures

1.1 A communication system with source and channel encoding and decoding 2
1.2 The channel coding problem on a communication system. 2
1.3 A Tanner graph 6
1.4 A factor graph 9
1.5 A Tanner graph with channel information 12
1.6 The single system potential function of the $(3,6)$-regular Gallager code with transmission over the BEC(ϵ) 27
1.7 Types and constellations in the random construction of spatially coupled ensembles 33
1.8 Copies of the uncoupled Tanner graph are placed on the spatial axis of coupling. 34
1.9 A $(3,6)$-regular “randomly constructed” spatially coupled code with window size $w = 2$ 34
1.10 The protograph of a standard $(3,6)$-regular ensemble. 35
1.11 Copies of the protograph are placed on the spatial axis of coupling. . . . 35
1.12 A protograph-based $(3,6)$-regular spatially coupled code with window size $w = 3$. 35
1.13 A two-sided profile and its one-sided counterpart 37
1.14 Displacement interpolation 49

2.1 The transient phase in the dynamic phase of decoding 62
2.2 The decoding wave (or soliton) for the coupled $(3,6)$-regular code on the BEC 63
2.3 Velocities for the coupled $(3,6)$-regular Gallager code on the BEC 72
2.4 Velocities for the coupled $(4,6)$-regular Gallager code on the BEC 72
2.5 Velocities for a coupled irregular LDPC code on the BEC 73
2.6 Velocities for another coupled irregular LDPC code on the BEC 73
2.7 The wave propagation phenomenon within the framework of the Gaussian approximation. 75
2.8 Velocities for the coupled $(3,6)$-regular Gallager code within the framework of the Gaussian approximation. 77

3.1 The propagating wave (or soliton) for the compressive sensing problem . 85
3.2 Velocities for the coupled GLDPC code on the BEC or BSC 91
3.3 The single potential function for compressive sensing 91
3.4 Velocities for the coupled compressive sensing problem 93

4.1 An alternative form of the single potential function for the $(3, 6)$-regular Gallager code on the BEC, at the MAP threshold 98

5.1 A generic example of EXIT-like curves 125
5.2 A profile x and its saturated version $\lfloor x \rceil_K$. 128
5.3 Simple example of an increasing rearrangement for step functions. 130
5.4 Illustration of level sets used in the proof of Lemma 5.13 132
5.5 The EXIT curves for the $(3, 6)$-regular LDPC ensemble with transmission over the BEC(ϵ_{MAP}). 145
5.6 The single potential $\phi(u, v)$ for the $(3, 6)$-regular LDPC ensemble with transmission over the BEC(ϵ_{MAP}). 145
5.7 The EXIT curves for the GLDPC code with $n = 15$, $e = 3$ with transmission over the BEC(ϵ_{MAP}). 146
5.8 The alternative area potential $\tilde{A}(v)$ for the GLDPC code with $n = 15$, $e = 3$ with transmission over the BEC(ϵ_{MAP}). 146

Introduction

1

The fundamental problem addressed by coding theory is the design of efficient and reliable schemes for the transmission of information over different noisy channels. In this chapter, we introduce the transmission model that we consider and the notions that enable us to describe our contributions. We briefly discuss the historical development of coding theory and formulate the coding problem in Section 1.1. We also mention source and channel encoding and decoding and formulate the problem of communication over noisy binary-input memoryless channels.

We describe low-density parity-check (LDPC) codes and how they are constructed in Section 1.2. LDPC codes are typically decoded by using a low-complexity message-passing algorithm called belief propagation. In Section 1.3 we describe such decoding algorithms, and in Section 1.4 we show the tools that we will use to analyze them, specifically the density evolution equations and the potential functions.

LDPC codes can further be *spatially coupled*, yielding a family of codes that has been proved to exhibit excellent performance in the sense of error-correction. We will describe how spatially coupled LDPC codes are constructed and summarize their main properties in Section 1.5. We also describe the "threshold saturation" phenomenon that explains the capacity-achieving performance of coupled codes, along with the corresponding coupled density evolution equations and potential functions. We then consider the coupled system in the large-system size and continuum limits described in Section 1.6. It is in those limits that the derivations become more tractable and we carry out our analyses.

Finally, in Section 1.8, we give a summary of our main contributions. These contributions are not restricted to coding (see Chapters 3 and 5). However, we limit the scope of the introduction to that of the field of coding theory, as our work was initially inspired by applications in this context and for clarity.

1.1 The Coding Problem

In 1948, Claude Shannon published the groundbreaking article "A Mathematical Theory of Communication" [2], in which he revolutionized communication theory

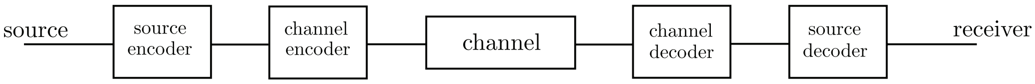

Figure 1.1: The abstracted overview of communication systems that we consider, updated with channel encoding and decoding blocks.

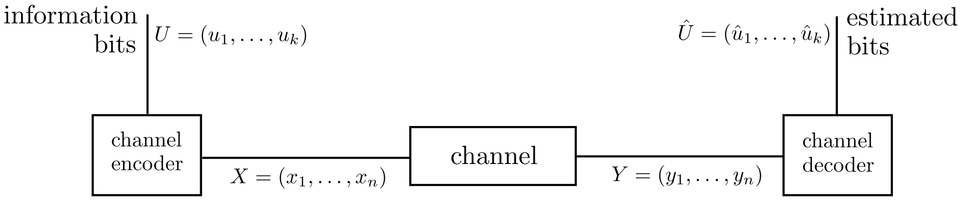

Figure 1.2: The channel coding problem on a communication system.

and created the field of information theory. In this work, Shannon established the notion of information entropy as a measure of the uncertainty in a message, and he addressed the problem of designing codes that ensure efficient and reliable transmission of information. The field of information theory studies the theoretical limits involved in a communication system at which reliable transmission is achievable, whereas the field of *coding theory* studies system and code constructions that are able to achieve the said limits.

An abstracted overview of the scheme for the transmission of information over a communication system is depicted in Figure 1.1, where we consider a noisy binary-input memoryless channel. The source produces a message from a finite set of possible messages and, ultimately, wants to send the message through the channel to the receiver. In order to do this, the source has to first transform the message from its original alphabet to the binary alphabet so that it is compatible with the channel. This transformation is termed *source encoding*. Typically, in order to make the transmission efficient, the encoding scheme attempts to minimize the average length of the encoded messages according to the probability distribution over the source messages. This is also called *data compression*; two examples of such schemes are Huffman coding [3] and Lempel-Ziv compression [4, 5]. The inverse transformation, *source decoding*, is done at the receiver side to recover the original message (in the original alphabet).

The noisy channel might erase, delete, or modify the incoming binary sequence, or even insert new bits into it. To increase the reliability of transmission, we append *redundancy bits* to the source-encoded binary sequence. The additional bits contain information about the original binary sequence and can thus help the receiver recover it in case it was distorted. This is called *channel encoding*; at the channel output, we perform *channel decoding* to remove the redundancy bits. This thesis is concerned with a low-complexity iterative channel-decoding scheme, that is called belief propagation when channel encoding is done using low-density parity-check (LDPC) codes. We consider binary-input memoryless channels defined below.

Definition 1.1. *A binary-input channel is said to be memoryless (and without feedback) if the probability of receiving a bit y_i at the channel output depends only on the corresponding transmitted bit x_i and the channel is completely described by its input* *and output alphabets and the conditional probability distribution* $p_{Y|X}(y_i|x_i)$*, where* X *and* Y *denote the random variables of the bit input and bit output, respectively. In other words, if we assume that the length-*n *codeword* $\underline{x}$ *is transmitted one bit at a time, then the noise realization at one time instant is independent of its realization at the previous time instant. A memoryless channel (without feedback) that takes* $\underline{x}$ *as an input codeword and outputs* $\underline{y}$ *satisfies*

$$p_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x}) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i). \tag{1.1}$$

In this thesis, we consider the "channel encoder - channel - channel decoder" block shown in Figure 1.2. The source is modeled as a random variable $\underline{u} = (u_1, \ldots, u_k)$ that takes values in the set of binary length-k vectors $\{0,1\}^k$ uniformly at random. The message $\underline{u}$ is encoded into a length-n binary *codeword* $\underline{x} = (x_1, \ldots, x_n)$ in the set $\{0,1\}^n$ using a channel-encoding scheme in order to improve the reliability of recovery at the receiver side. The set of codewords obtained from encoding the set of source messages is called the *codebook*. The code is characterized by the rate, defined as the ratio of the number of information bits to the number of message bits $R = k/n$. The binary codeword is sent through a noisy memoryless channel, that in turn outputs a corrupted version $\underline{y} = (y_1, \ldots, y_n)$ of it.[1] We note here that the elements of $\underline{y}$ are not necessarily binary. For instance, when we transmit a bit in $\{0,1\}$ on the binary erasure channel of parameter $\epsilon \in [0,1]$ (BEC(ϵ)), it is erased with probability ϵ and left unchanged with probability $1-\epsilon$. Therefore, the output alphabet is $\{0,1,?\}$ where "?" symbolizes an erasure. Once the vector $\underline{y}$ is received at the channel output, we perform channel decoding on it to obtain the message $\underline{\hat{u}} = (\hat{u}_1, \ldots, \hat{u}_k)$ that estimates $\underline{u}$.

According to Shannon's noisy-channel theorem [2], for every rate R that is strictly less than the capacity C, there exists a code and a decoder such that the probability of the block-code error vanishes for large blocklengths. The goal of coding theory is to find such code constructions with computationally practical encoding and decoding complexities. The maximum a-posteriori (MAP) decoder, though optimal in the sense of error correction, is computationally impractical and thus intractable. Ever since the publication of Shannon's revolutionary article, coding theorists have been working on the design of error-correcting codes that can achieve the capacity of various noisy channels and that are also efficient, reliable, and of low complexity.

The first family of codes developed for this purpose consists of algebraic coding schemes and algorithms that make use of algebraic and combinatorial tools. We mention some linear block codes such as Hamming codes [6], Reed-Muller codes [7,8], Reed-Solomon codes [9], and Bose-Chaudhuri-Hocquenghem (BCH) codes [10, 11]. Some efficient decoding algorithms were devised for these codes, such as Berlekamp's algorithm [12], also used by Massey [13], Sudan's list-decoding algorithm for RS codes [14] and its improvements by Guruswami and Sudan [15] and by Koetter and Vardy [16].

In 1955, Elias introduced convolutional codes as an alternative to block codes [17]; they offered the advantage of efficient maximum-likelihood soft-decision decod-

---

[1] In this manuscript, we assume that the channel does not perform any deletions or insertions, thus $\underline{x}$ and $\underline{y}$ have the same lengths.

ing. Some of the most popular decoding algorithms were developed by Wozencraft [18] and Viterbi [19]. Later on in the twentieth century, lattice codes were developed as structured codes for Gaussian channels and networks; they were shown to be capacity-achieving [20–23].

The field of error-correcting codes was revolutionized in the '90s, however, with the emergence of graph-based codes, such as turbo codes that were proposed by Berrou and Glavieux in 1996 [24] and low-density parity-check (LDPC) codes. The main advantage of graph-based codes is that we can decode them using low-complexity *iterative decoding algorithms*. LDPC codes and iterative decoding were first developed in Gallager's thesis in 1963 [25] but were not implemented due to the limited computational resources that existed at the time. They were rediscovered three decades later by Neil and MacKay [26–28], and independently by Spielman [29,30].

LDPC codes are constructed from sparse graphical models and have been shown to exhibit very good performance under the low-complexity sum-product decoding algorithm [31,32]. However, this performance can only be attained up to a maximum noise level on the channel, or a *threshold*, lower than the threshold up to which the MAP decoder performs very well. In other words, for some levels of noise on the channel, the average performance of an LDPC code ensemble, when decoding is done using the sum-product iterative algorithm, is not as good as that when decoding is done using the MAP decoder. As the MAP decoder is too computationally expensive and intractable, coding theorists sought to close the gap between the MAP threshold and the iterative decoding threshold by designing better iterative decoding schemes.

A first step in the direction of designing codes that exhibit excellent performance under low-complexity decoding algorithms, such as the sum-product algorithm, *up to the capacity* for transmission over the BEC, was taken by Luby, Mitzenmacher, Shokrollahi, and Spielman in 1998 [1, 33]; they introduced *irregular LDPC codes*. The idea is to optimize the degree distributions of the nodes of the underlying graphical model so that the performance is improved [34–36]. Using this technique, Chung, Richardson and Urbanke were able to achieve performance within 0.0045dB of the Shannon capacity when transmission takes place over the binary-input additive white Gaussian noise channel [37].

This brings to light the two most recent code constructions, polar codes,[2] proposed by Arikan in 2009 [38], and spatially coupled codes. The first "version" of spatially coupled codes was termed *terminated convolutional LDPC codes* and was introduced by Lentmaier, Fettweis, Zigangirov, and Costello in 2009 [40]. More recently, however, the term *spatially coupled codes* was coined by Kudekar, Richardson, and Urbanke [41] to describe such codes whose construction is less constrained, and this includes all the previously derived variants. Polar codes and spatially coupled codes are both provably capacity-achieving on binary-input memoryless symmetric-output (BMS) channels and have low decoding complexities. However, spatially coupled codes have two advantages over polar codes: The original construction of polar codes is not universal[3] and its performance converges more slowly to the

---

[2] Polar codes are capacity-achieving under successive cancellation decoding [38,39], not iterative decoding.

[3] The universality of a code means that one and the same code ensemble is good for a whole class of channels, assuming that the receiver has knowledge of the channel. In fact, Hassani and Urbanke propose polar-like codes that are universal [42].

asymptotic desirable one [43].

A spatially coupled LDPC code is constructed by locally connecting replicas of the underlying (uncoupled) LDPC code, such that the original degree distribution on the nodes is preserved. The intuition behind the relationship between the two codes (uncoupled and coupled) is the following: As the connections are locally the same from the nodes' persepective, the two codes share fundamental quantities, such as the MAP threshold. In fact, one of the attractive properties of spatially coupled codes is the so-called *threshold saturation* property that means that the noise threshold, up to which coupled codes perform very well under the BP algorithm, is pushed or saturated up to the MAP threshold. Despite the fact that spatially coupled codes are universally capacity-achieving under the low-complexity BP algorithm, they are still not prevalent in practice because this excellent performance is only demonstrated when the blocklength is significantly large.

The construction technique of spatial coupling is not restricted to coding [44,45]. Since its introduction, it has been shown to approach the capacity regions of many coding systems [46–51], but also to exhibit excellent performance in several other applications such as compressive sensing [52–54], random constraint satisfaction problems [55–58], and a coupled Curie-Weiss (toy) model [59, 60]. Our results in Chapter 3 and 5 include such scalar systems.

1.2 Channel Encoding via Low-Density Parity-Check Codes

In this thesis, we consider channel encoding using the (n, k)-LDPC code with $n > k$. This adds $n - k$ redundancy bits to each source message $\underline{u} = (u_1, \dots, u_k)$, which yields the encoded message $\underline{x} = (x_1, \dots, x_n)$.

LDPC codes form a class of linear block codes that can be represented in matrix or graphical form. An LDPC code can be graphically represented by a bipartite graph, called a Tanner graph; it consists of two types of nodes: n *variable nodes* that contain the channel-encoded bits, and m *check nodes* that impose constraints on their adjacent variable nodes. We consider binary inputs and perform the operations in $\mathbb{F}_2$. We show a Tanner graph in Figure 1.3 and write the constraints imposed by the check nodes near them. For LDPC codes, a check node forces the sum of the bits on its adjacent variable nodes to be zero in $\mathbb{F}_2$.

The matrix representation H of an LDPC code is an $m \times n$ adjacency matrix where the entry $H_{i,j}$ at row i and column j is 1 if there is an edge between the i^{th} check node and the j^{th} variable node, and 0 otherwise. The adjacency matrix corresponding to the Tanner graph in Figure 1.3 is shown below.

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

The first part of the name of this code is "low-density" because the matrix H is sparse (the numbers of 1's per row and column are small) or, equivalently, because the Tanner graph is sparse (the degrees of variable and check nodes are small). By looking at the graph in Figure 1.3 and its adjacency matrix, we can already see that these conditions can be valid when the size of the graph is large.

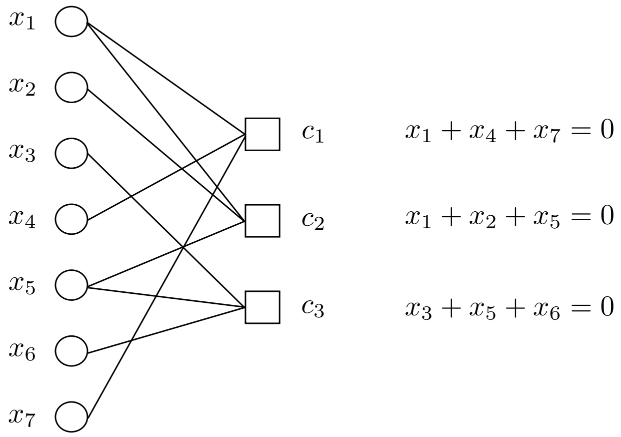

Figure 1.3: An example of a Tanner graph, with variable nodes $x_1, \ldots, x_7$ denoted by circles and check nodes $c_1, \ldots, c_3$ denoted by squares. The conditions imposed by check nodes are written on their right.

Due to the nature of the constraints imposed by the check nodes on their adjacent variable nodes, the second part of the name of these codes is "parity-check". More specifically, the constraint imposed by a check node on its neighboring variable nodes is that the sum of their bit values in $\mathbb{F}_2$ should equal 0. This is why we can also call the adjacency matrix *the parity-check matrix*.

There are various algorithms with which suitable LDPC codes can be constructed, such as that suggested by Gallager in 1963 and those proposed by MacKay in 1996. Such algorithms are concisely summarized in [27].

Encoding the message bits into channel-encoded bits is roughly done by copying the k message bits onto some subset of the n final bits (typically the first k bits) and by filling out the remaining ones so that the parity checks are satisfied. We formalize this process as follows. Given the parity-check matrix H, elementary row operations can be performed to rewrite it in the form $[-P^T | I_{n-k}]$, where I_{n-k} denotes the $(n-k) \times (n-k)$ identity matrix and $-P^T$ is the leftmost remainder of the matrix (and T denotes the matrix transpose operator). We can then define the *generator matrix* of the codebook as $G = [I_k | P]$ where I_k denotes the $k \times k$ identity matrix. Finally, to obtain the codewords in the codebook associated with this parity check matrix, we multiply all length-k strings with G.

We show how the generator matrix of the example in Figure 1.3 is found. We first perform elementary row operations on the parity-check matrix H to write it in the form $[-P^T | I_3]$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix},$$

where the first equality is obtained by adding the second and third rows, and storing the result in the third row; and the second equality is obtained by rearranging the rows. The generator matrix is thus

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The drawback in using this approach is that it requires operations involving the entire parity-check matrix and can thus be of high computational complexity. However, more efficient methods have been developed in practice that make use of the sparsity of the matrix or the structure of the graph (if it is known).

1.3 Channel Decoding via Message-Passing

Despite the desirable results that MAP decoding can provide for decoding LDPC codes, this technique remains computationally impractical. Fortunately, the low-complexity message-passing algorithm, called the *belief propagation* algorithm in the context of coding, can be employed to approximate the MAP decoding rule.

We formalize the MAP decoding rule for bit-wise error correction in Section 1.3.1, and introduce factorization and marginalization in Section 1.3.2. We derive the approximative message-passing rules corresponding to the MAP decoding rule in Section 1.3.3. These rules are then simplified and scheduled in Section 1.3.4.

1.3.1 Bit-wise MAP Decoding

Consider transmission over a binary-input memoryless channel described by the conditional probability distribution

$$p_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x}) = \prod_{i=1}^{n} p_{Y_i|X_i}(y_i|x_i),$$

where we denote by $\underline{x} = (x_1, \ldots, x_n)$ the channel input (or codeword), with $x_i \in \{0,1\}$, and $\underline{y} = (y_1, \ldots, y_n)$ the channel output (or observation).

We assume that the source bits are encoded using a linear block code described by its parity-check matrix, and that the codewords are chosen uniformly at random from the codebook $\mathcal{C}$. To obtain the MAP estimate $\hat{x}_i^{\text{MAP}}$ of the bit x_i, $i = 1, \ldots, n$, given the channel output $\underline{y}$, we refer to the bit-wise MAP decoding rule

$$\begin{aligned} \hat{x}_i^{\text{MAP}}(\underline{y}) &= \operatorname{argmax}_{x_i \in \{0,1\}} p_{X_i|\underline{Y}}(x_i|\underline{y}) \\ &= \operatorname{argmax}_{x_i \in \{0,1\}} \sum_{\sim x_i} p_{\underline{X}|\underline{Y}}(\underline{x}|\underline{y}), \end{aligned}$$

where the notation $\sum_{\sim x_i}$ is used to indicate a summation over all components of $\underline{x}$ except x_i. In the last step, we use the law of total probability $p_{A_i}(a_i) = \sum_{\sim a_i} p_{\underline{A}}(a_1, \ldots, a_i, \ldots, a_n)$, with $\underline{A} = (A_1, \ldots, A_n)$. This computation is very expensive because it involves a sum over all value combinations of $n - 1$ bits and thus an exponential (in n) number of terms. We can use Bayes' rule to rewrite the MAP estimate as

$$\begin{aligned}\hat{x}_i^{\mathrm{MAP}}(\underline{y}) &= \operatorname{argmax}_{x_i \in \{0,1\}} \sum_{\sim x_i} p_{\underline{Y}|\underline{X}}(\underline{y}|\underline{x}) p_{\underline{X}}(\underline{x}) \\ &= \operatorname{argmax}_{x_i \in \{0,1\}} \sum_{\sim x_i} \Big(\prod_{j=1}^{n} p_{Y_j|X_j}(y_j|x_j) \Big) \mathbb{1}_{\{\underline{x} \in \mathcal{C}\}}, \qquad (1.2)\end{aligned}$$

where in the last step we have used the property (1.1) of memoryless channels and the assumption that the codewords in the codebook $\mathcal{C}$ have a uniform prior.

The MAP decoder is *optimal*, with respect to the bit-error rate, because it minimizes the bit-error probability under random errors incurred by the channel. However, the calculation of the MAP estimate involves a sum over $\mathcal{O}(2^n)$ terms, where n is the blocklength. Therefore, this decoder is too computationally expensive and impractical. This led to the design and use of *message-passing* as an efficient alternative. Such iterative algorithms are of *low complexity*, but they are *suboptimal* in the sense that they cannot decode up to the MAP threshold of the code ensemble and that the bit-error probability is not minimized.

The rules for message-passing algorithms are derived using the Tanner graph of the LDPC code, and their solution is exact when the graph is a tree. In addition, it has been shown that these algorithms still work well on graphs that are not trees. The intuition behind this is that if the parity-check matrix is sparse and at the same time very large, the graph, from the perspective of any randomly selected node, is locally tree-like. However, the algorithms do not work as well on general graphs as they do on trees. This is the “price to pay” for applying algorithms designed for trees on graphs that are not trees.

In order to show how message-passing can be employed to solve the optimization problem above, we will show how the expression in (1.2) is related to the Tanner graph of the LDPC code and describe the messages that are to be exchanged between the graph’s nodes so that decoding takes place. We derive the messages assuming that the Tanner graph is a tree.

1.3.2 Factor Graphs and Marginalization

The marginalization problem describing the MAP decoder estimate $\hat{x}_i^{\mathrm{MAP}}$ involves a sum over $\mathcal{O}(2^n)$ terms and is thus computationally expensive. In this section, we show how we can split this problem into smaller subproblems and how this can be used to design a message-passing algorithm on a tree structure.

Consider the graph in Figure 1.3. We write the parity-check constraints near the check nodes; these are the conditions that the bits $x_1, \dots, x_7$ must satisfy so that $\underline{x} = (x_1, \dots, x_7)$ is a codeword. We recall that operations are performed in $\mathbb{F}_2$ and write the code membership function for this graph as follows,

$$\begin{aligned} f(x_1, \dots, x_7) = \mathbb{1}_{\{\underline{x} \in \mathcal{C}\}} &= \mathbb{1}_{\{x_1+x_4+x_7=0\}} \mathbb{1}_{\{x_1+x_2+x_5=0\}} \mathbb{1}_{\{x_3+x_5+x_6=0\}} \\ &= f_1(x_1, x_4, x_7)\, f_2(x_1, x_2, x_5)\, f_3(x_3, x_5, x_6), \end{aligned}$$

where we have defined the following *factors*

$$\begin{cases} f_1(x_1, x_4, x_7) &= \mathbb{1}_{\{x_1+x_4+x_7=0\}}, \\ f_2(x_1, x_2, x_5) &= \mathbb{1}_{\{x_1+x_2+x_5=0\}}, \\ f_3(x_3, x_5, x_6) &= \mathbb{1}_{\{x_3+x_5+x_6=0\}}. \end{cases}$$

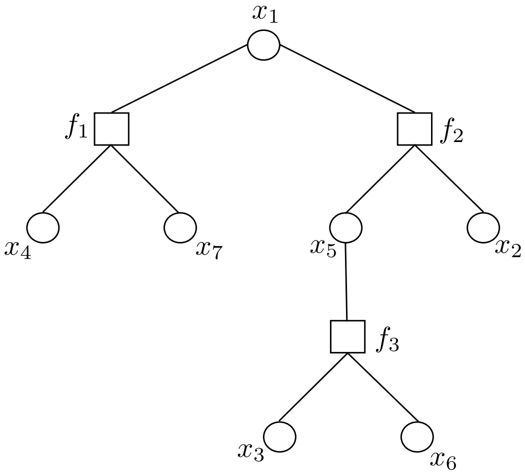

Figure 1.4: We draw the factor graph corresponding to the Tanner graph in Figure 1.3 when marginalization is over x_1. We denote by f_i the factor node or check node that we previously denoted by c_i, $i = 1, 2, 3$.

To solve the marginalization problem shown in (1.2) over x_i, by using this definition of the code membership function, we could make a brute force summation over the values of all the variable nodes in the graph except x_i. Such a computation can be quite costly, especially when the number of variables is large. This is the basis for the idea of *factorization*, which splits the summation into smaller ones.

Consider the example shown in Figure 1.3. To visualize the marginalization over $x_i = x_1$ for example, we can pull out the variable node x_1, which yields the tree shown in Figure 1.4. We call this tree the *factor graph* [32] specifically because it demonstrates the factorization structure of the marginal of its root, and we call the squares (representing the parity-check nodes) the *factor nodes* for reasons that will shortly become clear. We slightly abuse notation by defining $f(x_1) \triangleq \sum_{\sim x_1} f(x_1, \ldots, x_7)$ to describe the marginalization over x_1 and write this as

$$f(x_1) = \sum_{\sim x_1} f_1(x_1, x_4, x_7) f_2(x_1, x_2, x_5) f_3(x_3, x_5, x_6) \tag{1.3}$$

$$= \sum_{x_4, x_7} f_1(x_1, x_4, x_7) \sum_{x_3, x_5, x_6} f_3(x_3, x_5, x_6) \sum_{x_2} f_2(x_1, x_2, x_5). \tag{1.4}$$

We describe the general factorization rules that permit us to compute the marginal of a generic function g with respect to some variable z. This marginal is defined as

$$g(z) = \sum_{\sim z} g(z, \ldots). \tag{1.5}$$

We can automatically factor $g(z, \ldots)$ as follows

$$g(z, \ldots) = \prod_{k=1}^{K} [g_k(z, \ldots)],$$

where g_k, $k = 1, \ldots, K$ are the factors of the function g such that the variable z appears in *each* of the factors, whereas each other variable appears in *only one* factor. For the example shown in Figure 1.4, we can specialize this rule to the following (with $z = x_1$)

$$f(x_1, \dots, x_7) = [f_1(x_1, x_4, x_7)]\,[f_3(x_3, x_5, x_6)\, f_2(x_1, x_2, x_5)],$$

with $K = 2$, $g_1(z, \dots) = f_1(x_1, x_4, x_7)$, and $g_2(z, \dots) = f_3(x_3, x_5, x_6)\, f_2(x_1, x_2, x_5)$. To see this graphically, one can think of this factorization as follows. The integer K is the number of subtrees with x_1 as a parent node and g_k, $k = 1, \dots, K$ are the marginals of these subtrees.

At this point, we can find the marginal $g(z)$ from (1.5). We first make an observation from (1.4). We have transformed a summation of the product of marginals in (1.3) to the product of summations of marginals in (1.4). In fact, this is always possible by using the distributive law when the graph is not fully connected because every marginal will contain a subset of the components of $\underline{x}$. Therefore, for the generic function g we can write ((2.7) in [61])

$$g(z) = \sum_{\sim z} \prod_{k=1}^{K} [g_k(z, \dots)] = \prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z, \dots) \Big]. \tag{1.6}$$

So far, we have applied this rule to the root of the factor tree. However, it can be applied recursively on the subtrees, yielding marginals that are easier to compute. The rule can be thought of as follows: The marginal of a variable node is the product of the marginals of its children in the factor graph.[4]

We can also derive a general rule for the computation of the marginal of a check (or factor) node in the factor graph, by looking at the factorization equations. We first define the *kernel* h of the marginal g_k at a check node as the node's own factor or membership function. By carrying out the recursive factorization steps, we can deduce that the rule for the marginal $g_k(z, \dots)$ at a check node is the following

$$g_k(z, \dots) = h_k(z, z_1, \dots, z_J) \prod_{j=1}^{J} [h_j(z_j, \dots)], \tag{1.7}$$

where $h_k(z, z_1, \dots, z_J)$ is the kernel of the factor node and $h_j(z_j, \dots)$, $j = 1, \dots, J$, are the factors. Here again we can determine the factors by following certain rules. The variable z appears only in the kernel, and every other component z_j can appear in at most two places: possibly the kernel and in at most one of the factors $h_j(z_j, \dots)$. This rule can be understood as follows: The marginal at a check node is obtained by multiplying its kernel with the marginals obtained from its children. Finally, and due to the definition of the marginal in (1.5), we obtain the marginal at this check node by summing over all the components except its parent z. This gives ((2.9) from [61])

$$\sum_{\sim z} g_k(z, \dots) = \sum_{\sim z} h_k(z, z_1, \dots, z_J) \prod_{j=1}^{J} [\sum_{\sim z_j} h_j(z_j, \dots)], \tag{1.8}$$

1.3.3 Belief Propagation Equations

The two rules found in Section 1.3.2 can be applied recursively to the rest of the factor tree, breaking down the computation of the marginal of the root $f(x_1)$ into

---

[4] As we consider bipartite graphs, the children of variable nodes are necessarily check nodes.

smaller computations of less complex marginals. The complexity of the computation is therefore significantly reduced. We can compute the marginals of the parts shown in (1.4) and combine them using the rules we have just described. This gives rise to a message-passing algorithm that can be applied on the factor graph in Figure 1.4. The algorithm starts at the leaves of the tree, that are initialized to some values (as we will see). Messages (the smaller marginals) are then transmitted up the tree, and whenever a node has received messages from all of its children, it processes and combines them, then passes the result up to its parent.

Each component x_j, $j = 1\ldots,n$, of $\underline{x}$ can be estimated by finding the corresponding marginal $f(x_j)$. We can either redraw a factor tree for each marginal, or simply consider the Tanner graph with messages that simultaneously flow on edges in both directions during the message-passing algorithm. We choose the latter and formalize the message-passing rules as follows. We denote by $\mu_{i\to c}$ a message transmitted from a variable node to a check node and by $\hat{\mu}_{c\to i}$ a message transmitted from a check node to a variable node.[5] We denote by ∂ the neighbors, in the sense that $x_{\partial a}$ denotes the neighbors in $\underline{x}$ of a, and $b\in\partial i\backslash a$ denotes all neighbors b of i (or x_i) except a. Then, the message-passing equations are

$$\mu_{i\to a}(x_i) = \prod_{b\in\partial i\backslash a} \hat{\mu}_{b\to i}(x_i), \tag{1.9}$$

$$\hat{\mu}_{a\to i}(x_i) = \sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\backslash i} \mu_{j\to a}(x_j), \tag{1.10}$$

where f_a denotes the kernel of a. Whenever a node receives messages from all its neighbors, it performs the appropriate marginalization step. This produces a new estimate for the components x_j that are used in the next iteration of the message-passing algorithm.

What remains is the initialization at the leaves. Assume that the leaf is a variable node x_j that sends the initial message $\mu_{j\to a}(x_j)$ to its parent check node a. The marginal at a is (for example) $f(x_k) = \sum_{\sim x_k} f(x_j,x_k)\mu_{j\to a}(x_j)$ where we have applied the check node processing rule in (1.10). However, we also know that $f(x_k) = \sum_{\sim x_k} f(x_j,x_k)$, hence we can deduce that the initial message from a leaf, when it is a variable node, is the constant function 1. Similarly, we can find that the initial message from a leaf node, when it is a check node, is its own kernel.

The message-passing algorithm we described is called the *belief propagation* (BP) algorithm in the context of coding. As the nature of the computations involved in the BP messages is not complex, and due to the fact that factorization breaks the main problem into smaller subproblems, implementing the BP algorithm is of low complexity. However, the update equations (1.9) and (1.10) were derived for factor graphs that are trees. Surprisingly, they have been known to provide good estimates $\underline{\hat{x}}$ for the input vector $\underline{x}$, even when the factor graph is not a tree.[6] The price to pay in this case, however, is that the BP algorithm decodes perfectly (in the limit of infinite blocklength n) on such graphs up to a certain level of channel noise that we call the *BP threshold*; and this is lower than the level up to which the MAP decoder decodes perfectly (again, when $n\to+\infty$), that we call the *MAP threshold*.

---

[5] We use the letters a, b, c, … to denote check nodes (to be associated with f_a, f_b, f_c, …) and the letters i, j, k, … to denote variable nodes (to be associated with x_i, x_j, x_k, …).

[6] Due to the sparsity and large size of the factor graph, the computational tree from the perspective of any variable node is locally tree-like.

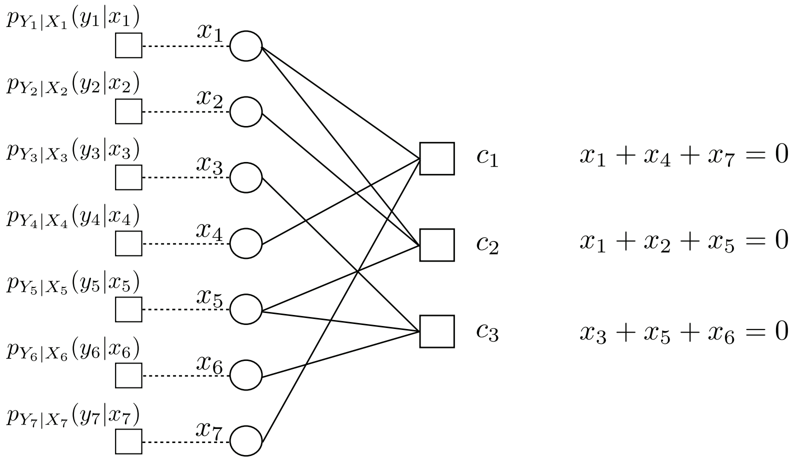

Figure 1.5: We draw the Tanner graph shown in Figure 1.3 with appended "channel check nodes" to represent the channel information.

Spoiler alert! Fortunately, this impediment can be overcome by spatial coupling that combines both advantages: Spatially coupled LDPC codes can be decoded up to the MAP threshold using the low-complexity BP decoding algorithm. We cover this in detail in Section 1.5.

1.3.4 Simplification of Belief Propagation Equations

In this section and for the purpose of simplifying calculations, we assume that the source emits bits X_i that take values in $\{-1,+1\}$ instead of the traditional binary domain $\{0,1\}$. The map we consider is

$$\begin{cases} 0 \rightarrow +1, \\ 1 \rightarrow -1. \end{cases}$$

By looking at the BP equations in (1.9) and (1.10), it might be thought that we have not taken the channel information into account. In fact, the equations are correct, but the Tanner and factor graphs we have shown so far are incomplete. To account for the channel information, we append to every variable node X_i a *factor leaf node* that provides the i^{th} channel realization, the length-2 vector $(p_{Y_i|X_i}(y_i|+1), p_{Y_i|X_i}(y_i|-1))$. This is shown in Figure 1.5, where we write this vector as $p_{Y_i|X_i}(y_i|x_i)$ as shorthand. The channel can equivalently be represented by the so-called log-likelihood ratio (LLR) that serves as a sufficient statistic and that we define below.

Definition 1.2 (Channel Log-Likelihood Ratio (LLR))**.** *For any binary-input channel, the log-likelihood ratio (LLR) is defined as*

$$l(y) = \ln\left(\frac{p_{Y|X}(y|+1)}{p_{Y|X}(y|-1)}\right).$$

The LLR is a random variable as it depends on the *random* channel output y. In Section 1.4.1, we will see that once we consider *symmetric channels*, we can assume that the all-+1 codeword was transmitted (or the all-zero codeword in the traditional binary domain). Therefore, we consider the distribution $\mathtt{c}_{\mathtt{h}}(l)$ of the LLR under the assumption that +1 is transmitted, where $\mathtt{h}$ denotes the entropy of the channel (think of it here as a noise level parameter, we formally define it in Section 1.4.4) and $l = l(y)$ denotes the LLR variable.[7] Elsewhere, this is called the L-density [61]. We give the precise expressions of this distribution for the binary erasure channel (BEC), the binary symmetric channel (BSC), and the binary-input additive white Gaussian noise (BIAWGN) channel below.

Example 1.3 (Binary Erasure Channel)**.** *The BEC(ϵ) erases a bit with probability $\epsilon \in [0,1]$ and leaves it unchanged with probability $1-\epsilon$. The input alphabet is $\{-1,+1\}$ and the output alphabet is $\{-1,+1,?\}$, where "?" denotes the erasure symbol. For an input bit x and an output bit y, the channel transition probabilities are thus*

$$\begin{cases} p_{Y|X}(+1|-1) & = p_{Y|X}(-1|+1) = 0, \\ p_{Y|X}(+1|+1) & = p_{Y|X}(-1|-1) = 1-\epsilon, \\ p_{Y|X}(?|+1) & = p_{Y|X}(?|-1) = \epsilon. \end{cases}$$

The LLR distribution of the BEC(ϵ) can be written as

$$\mathtt{c}_{\epsilon}(l) = \epsilon\delta_0(l) + (1-\epsilon)\delta_{\infty}(l). \tag{1.11}$$

Example 1.4 (Binary Symmetric Channel)**.** *The BSC(p) flips a bit ($+1 \to -1$ or $-1 \to +1$) with probability p and leaves it unchanged with probability $1-p$. The input and output alphabets are $\{-1,+1\}$. For an input bit x and an output bit y, the channel transition probabilities are thus*

$$\begin{cases} p_{Y|X}(+1|+1) & = p_{Y|X}(-1|-1) = 1-p, \\ p_{Y|X}(+1|-1) & = p_{Y|X}(-1|+1) = p. \end{cases}$$

The LLR distribution of the BSC(p) can be written as

$$\mathtt{c}_p(l) = p\delta_{\ln\frac{p}{1-p}}(l) + (1-p)\delta_{\ln\frac{1-p}{p}}(l). \tag{1.12}$$

Example 1.5 (Binary-Input Additive White Gaussian Noise Channel)**.** *The BIAWGN(σ) channel adds Gaussian noise to the input bit. The input alphabet is $\{-1,+1\}$ and the output alphabet is $\mathbb{R}$. For an input bit x and an output bit y, the channel transition probability is given by*

$$p_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-x)^2}{2\sigma^2}},$$

where $\sigma^2 \in [0,+\infty)$ is the noise variance.

The LLR distribution for the BIAWGN(σ) can be written as

$$\mathtt{c}_{\sigma}(l) = \sqrt{\frac{1}{8\pi\sigma^{-2}}}e^{-\frac{(l-2\sigma^{-2})^2}{8\sigma^{-2}}}. \tag{1.13}$$

---

[7] We note that, throughout the thesis, the characters l and h are typically used to represent an LLR variable, whereas the character $\mathtt{h}$ is to be interpreted as an entropy.

In Section 1.3.3, we have seen how to compute a marginal $\mu(x_i)$ at some variable node x_i, where the bit x_i takes values in $\{-1,+1\}$. By applying the BP rule in (1.9), we can write the messages $\mu_{i\to a}$ from the i^{th} variable node to check node a and the messages $\hat{\mu}_{a\to i}$ from check node a to the i^{th} variable node as

$$\mu_{i\to a}(+1) = \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(+1), \qquad \hat{\mu}_{a\to i}(+1) = \sum_{\sim x_i} f(+1,\dots) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j),$$

$$\mu_{i\to a}(-1) = \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(-1), \qquad \hat{\mu}_{a\to i}(-1) = \sum_{\sim x_i} f(-1,\dots) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j).$$

We can convert this two-dimensional information message into a one-dimensional real-valued message using the corresponding LLRs. This yields

$$h_{i\to a} = l_i + \sum_{b\in\partial i\setminus a} \hat{h}_{b\to i},$$

where l_i denotes the LLR that the variable node x_i receives from the channel and

$$\hat{h}_{b\to i} = \ln\left(\frac{\hat{\mu}_{b\to i}(+1)}{\hat{\mu}_{b\to i}(-1)}\right).$$

This message is equivalent to the BP message at a variable node shown in (1.9).

Remark 1.6. *Throughout the thesis, we will use the characters l and h to denote LLRs. Typically, we use l_i to denote the channel LLR (which remains unchanged before and during the BP algorithm), $h_{i\to a}$ to denote the LLR message sent from variable node x_i to check node a during the BP algorithm, and $\hat{h}_{a\to i}$ to denote the LLR message sent from check node a to variable node x_i during the BP algorithm. That is, at iteration zero of the BP algorithm, $h_{i\to a} = l_i$.*

It is more difficult to find the message equivalent to that in (1.10) that corresponds to the BP message at a check node. We start by rewriting the kernel of a check node a as

$$f(x,\dots) = \mathbb{1}_{\{\prod_{j\in\partial a\setminus i} x_j = x\}}. \tag{1.14}$$

The indicator function of the kernel contains a product over the x_j's, whereas in the previous subsection, it contained a summation. This is because we previously considered bits in $\mathbb{F}_2$, whereas here the domain is $\{-1,+1\}$. Then, using the BP rule in (1.10), we write the ratio of BP messages on the check node side as

$$\hat{r}_{a\to i} = \frac{\hat{\mu}_{a\to i}(+1)}{\hat{\mu}_{a\to i}(-1)} = \frac{\sum_{\sim x_i} f(+1,\dots) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)}{\sum_{\sim x_i} f(-1,\dots) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)}.$$

We use the definition of the kernel in (1.14) and divide the numerator and denominator by $\mu_{j\to a}(-1)$ to obtain

$$\hat{r}_{a\to i} = \frac{\sum_{\sim x_i : \prod_{j\in\partial a\setminus i} x_j = +1} \prod_{j\in\partial a\setminus i} \frac{\mu_{j\to a}(x_j)}{\mu_{j\to a}(-1)}}{\sum_{\sim x_i : \prod_{j\in\partial a\setminus i} x_j = -1} \prod_{j\in\partial a\setminus i} \frac{\mu_{j\to a}(x_j)}{\mu_{j\to a}(-1)}}.$$

We rewrite the inner fraction as

$$\frac{\mu_{j\to a}(x_j)}{\mu_{j\to a}(-1)} = \begin{cases} 1, & \text{if } x_j = -1, \\ r_{j\to a}, & \text{if } x_j = +1. \end{cases}$$

$$= \begin{cases} r_{j\to a}^{(1+x_j)/2}, & \text{if } x_j = -1, \\ r_{j\to a}^{(1+x_j)/2}, & \text{if } x_j = +1. \end{cases}$$

We can therefore rewrite the ratio $\hat{r}_{a\to i}$ as

$$\hat{r}_{a\to i} = \frac{\sum_{\sim x_i:\prod_{j\in\partial a\setminus i} x_j=+1} \prod_{j\in\partial a\setminus i} r_{j\to a}^{(1+x_j)/2}}{\sum_{\sim x_i:\prod_{j\in\partial a\setminus i} x_j=-1} \prod_{j\in\partial a\setminus i} r_{j\to a}^{(1+x_j)/2}}.$$

We further write the ratio in one more alternative form,

$$\hat{r}_{a\to i} = \frac{\prod_{j\in\partial a\setminus i}(r_{j\to a}+1) + \prod_{j\in\partial a\setminus i}(r_{j\to a}-1)}{\prod_{j\in\partial a\setminus i}(r_{j\to a}+1) - \prod_{j\in\partial a\setminus i}(r_{j\to a}-1)}. \tag{1.15}$$

To see why this is true, remark that

$$\begin{aligned} &\prod_{j\in\partial a\setminus i}(r_{j\to a}+1) \\ &\quad = 1 + \sum_{j\in\partial a\setminus i} r_{j\to a} + \sum_{j,k\in\partial a\setminus i:j\neq k} r_{j\to a}r_{k\to a} + \sum_{j,k,l\in\partial a\setminus i:j\neq k\neq l} r_{j\to a}r_{k\to a}r_{l\to a} + \dots, \\ &\prod_{j\in\partial a\setminus i}(r_{j\to a}-1) \\ &\quad = -1 + \sum_{j\in\partial a\setminus i} r_{j\to a} - \sum_{j,k\in\partial a\setminus i:j\neq k} r_{j\to a}r_{k\to a} + \sum_{j,k,l\in\partial a\setminus i:j\neq k\neq l} r_{j\to a}r_{k\to a}r_{l\to a} - \dots, \end{aligned}$$

where we have used the slightly bad notation $j \neq k \neq l$ to indicate that $j \neq k$, $j \neq l$, and $k \neq l$ (due to space limitations). In words, the coefficients involved in the expansion of $\prod_{j\in\partial a\setminus i}(r_{j\to a}+1)$ are all positive and equal to +1, and for $\prod_{j\in\partial a\setminus i}(r_{j\to a}-1)$, the products in the expansion that consist of d terms such that $J-d$ is odd have the negative coefficients -1, and the remaining products have the positive coefficients +1. Thus, carrying out the addition and subtraction of these products, for the numerator and denominator, respectively, gives

$$\prod_{j\in\partial a\setminus i}(r_{j\to a}+1) + \prod_{j\in\partial a\setminus i}(r_{j\to a}-1) = 2 \sum_{x:\prod_{j\in\partial a\setminus i} x_j=+1} \prod_{j\in\partial a\setminus i} r_{j\to a}^{(1+x_j)/2},$$

$$\prod_{j\in\partial a\setminus i}(r_{j\to a}+1) - \prod_{j\in\partial a\setminus i}(r_{j\to a}-1) = 2 \sum_{x:\prod_{j\in\partial a\setminus i} x_j=-1} \prod_{j\in\partial a\setminus i} r_{j\to a}^{(1+x_j)/2}.$$

This yields the expression we show in (1.15).

We are now ready to derive the simplified BP message processing rule at a check node. We divide the numerator and denominator in (1.15) by $\prod_{j\in\partial a\setminus i}(r_{j\to a}+1)$ and obtain

$$\hat{r}_{a\to i} = \frac{1 + \prod_{j\in\partial a\setminus i} \frac{r_{j\to a}-1}{r_{j\to a}+1}}{1 - \prod_{j\in\partial a\setminus i} \frac{r_{j\to a}-1}{r_{j\to a}+1}}$$

We can rewrite this equation as

$$\frac{\hat{r}_{a\to i}-1}{\hat{r}_{a\to i}+1}=\prod_{j\in\partial a\setminus i}\frac{r_{j\to a}-1}{r_{j\to a}+1}. \tag{1.16}$$

By the definition of the LLR, $\hat{h}_{a\to i}=\ln\hat{r}_{a\to i}$, or equivalently, $\hat{r}_{a\to i}=\exp(\hat{h}_{a\to i})$. Thus, $(\hat{r}_{a\to i}-1)/(\hat{r}_{a\to i}+1)=\tanh(\hat{h}_{a\to i}/2)$. We can thus rewrite (1.16) in terms of the LLRs as

$$\tanh(\hat{h}_{a\to i}/2)=\prod_{j\in\partial a\setminus i}\tanh(h_{j\to a}/2).$$

This yields the simplified BP equation at a check node a

$$\hat{h}_{a\to i}=2\tanh^{-1}\Big(\prod_{j\in\partial a\setminus i}\tanh(h_{j\to a}/2)\Big). \tag{1.17}$$

So far, we have mentioned that the message-passing algorithm is iterative, but we have not formalized this. In fact, different schedules for passing messages might yield different performances. In our analysis, we consider the *flooding* or *parallel* schedule, where the variable and check nodes use the messages obtained in the previous iteration in order to compute the new messages. So, at every iteration, all variable nodes send messages to their neighboring checks, then all check nodes send messages to their neighboring variables. We choose this schedule mainly because it is convenient for analysis. As for the initial condition, the only information the graph possesses at first is the information from the channel. Hence, the factor nodes of the channel send their LLR messages. We represent this in the form of "channel check nodes" that we append to variable nodes. These check nodes send their associated LLR messages at the first iteration of the algorithm. As for the remaining nodes of the graph, they are initially set to send "neutral" messages. In the language of LLRs, this translates to the message 0.

We summarize the above discussion by writing the system of scheduled simplified BP equations. We denote by $h^{(t)}_{i\to a}$ the LLR sent from a variable node to a check node at iteration $t\in\mathbb{N}$ and by $\hat{h}^{(t)}_{a\to i}$ that sent from a check node to a variable node at iteration $t\in\mathbb{N}$. Then, with the channel LLRs l_i for variable x_i and once the iterative process is initialized with $h^{(0)}_{i\to a}=\hat{h}^{(0)}_{a\to i}=0$, the LLRs are updated according to the following BP equations

$$\begin{cases} h^{(t)}_{i\to a} &= l_i+\sum_{b\in\partial i\setminus a}\hat{h}^{(t)}_{b\to i},\\ \hat{h}^{(t+1)}_{a\to i} &= 2\tanh^{-1}\Big(\prod_{j\in\partial a\setminus i}\tanh(h^{(t)}_{j\to a}/2)\Big).\end{cases} \tag{1.18}$$

1.4 Tools for Performance Analysis

Before analyzing the performance of the BP decoder on LDPC codes, we mention two simplifications that facilitate the analysis in Section 1.4.1. We then define the configuration model of LDPC codes that we consider in Section 1.4.2. We derive the density evolution (DE) equations associated with such code ensembles in the framework where the blocklength is very large. To instill the intuition behind this formulation, we start by writing the DE equations when transmission takes place over the BEC in Section 1.4.3, then generalize the equations to the case when transmission takes place over binary memoryless symmetric (BMS) channels in Section 1.4.4. Finally, we present the variational equivalent of the DE equations, called the *potential function*, in Section 1.4.5.

1.4.1 Two Basic Simplifications

The first simplification that we make is the "restriction to the all-(+1) codeword" (or the all-zero codeword in the traditional binary domain). In other words, it is sufficient for our analysis to assume that the input to the channel is the all-(+1) codeword (and this codeword is as good as any other one). The advantage of this assumption is the simplicity with which one can count the number of errors in the channel output and the decoder output: it is simply the number of bits that are not +1. For this restriction to be valid, we need two kinds of symmetry to hold, namely channel symmetry and decoder symmetry, defined below.

Definition 1.7 (Channel Symmetry [61]). *We say that a channel is symmetric (more precisely, output-symmetric) if*

$$p_{Y|X}(y|+1) = p_{Y|X}(-y|-1), \tag{1.19}$$

where $y \in \mathbb{R}$ is the output bit and X, Y are the random variables describing the input and output bits.

Definition 1.8 (Decoder Symmetry [62]). *A decoder is said to be symmetric if*

- *At any check node, the magnitude of the outgoing message is only a function of the magnitudes of the incoming messages, and its sign is equal to the product of the signs of these messages.*
- *At any variable node, the sign of the outgoing message is reversed if the sign of each incoming message is reversed.*

In this thesis, we restrict our work to binary-input memoryless symmetric-output channels, thus the first condition is satisfied by assumption. By looking at the BP equations, we can see that the BP decoder is symmetric, thus the second condition is also satisfied.

The second simplification we make is "concentration", or the restriction to the analysis of the ensemble average performance, and it is valid when the blocklength is large. More specifically, instead of analyzing the performance of individual codes, we assess that of the ensemble average. This motivates coding theorists to design and construct code ensembles that have a good *average* performance.

The idea behind the concentration simplification is that the performance of an individual code is, with high probability, arbitrarily close to that of the ensemble average. More precisely, the two performances are arbitrarily close for all except an exponentially - in the blocklength - small fraction of codes. To express this formally, let $P_{BP,b}(\mathcal{C}, t)$ denote the average bit-error probability for a code $\mathcal{C}$ chosen uniformly at random from a Gallager ensemble, when the BP decoder has been run for t iterations. We consider transmission over BMS channels. Then, for any given $\delta > 0$, there exists an $\alpha > 0$ (that depends on δ and the degree distribution of the code ensemble) such that

$$\mathbb{P}\{|P_{BP,b}(\mathcal{C},t) - \mathbb{E}[P_{BP,b}(\mathcal{C},t)]| > \delta\} \le e^{-\alpha n},$$

where the expectation $\mathbb{E}$ is over the code ensemble.

1.4.2 The LDPC(λ, ρ) Configuration Model

In this thesis, we often use the ensemble of bipartite graphs LDPC(n, λ, ρ) (or special cases of it), where n denotes the blocklength and λ and ρ describe the degree distributions of the variable and check nodes, respectively. This ensemble is also called the *configuration model*.

The degree distributions are defined as follows. Let λ_i denote the fraction of edges in a Tanner graph that connect to variable nodes of degree i, and ρ_i denote the fraction of edges that connect to check nodes of degree i. Alternatively, we can think of λ_i or ρ_i as the probability that an edge chosen uniformly at random is connected to a variable node or a check node, respectively, of degree i. Then λ and ρ are the variable and check *degree distributions from an edge perspective*, respectively, and can be expressed as

$$\lambda(x) = \sum_i \lambda_i x^{i-1}, \quad \rho(x) = \sum_i \rho_i x^{i-1}.$$

Equivalently, we can look at the variable and check degree distributions *from a node perspective* that are denoted by L and R, respectively. We define L_i as the fraction of variable nodes of degree i and R_i the fraction of check nodes of degree i. Then,

$$L(x) = \sum_i L_i x^i, \quad R(x) = \sum_i R_i x^i.$$

The two notions of degree distributions are related by

$$\begin{aligned} \lambda(x) &= \frac{L'(x)}{L'(1)}, & L(x) &= \frac{\int_0^x \mathrm{d}z\, \lambda(z)}{\int_0^1 \mathrm{d}z\, \lambda(z)}, \\ \rho(x) &= \frac{R'(x)}{R'(1)}, & R(x) &= \frac{\int_0^x \mathrm{d}z\, \rho(z)}{\int_0^1 \mathrm{d}z\, \rho(z)}. \end{aligned}$$

The average variable degree ℓ_{avg} and the average check degree r_{avg} are then defined as

$$\ell_{\text{avg}} = L'(1) = \frac{1}{\int_0^1 \mathrm{d}z\, \lambda(z)}, \qquad r_{\text{avg}} = R'(1) = \frac{1}{\int_0^1 \mathrm{d}z\, \rho(z)},$$

and the *design rate* of the code is

$$r(\lambda, \rho) = 1 - \frac{\ell_{\text{avg}}}{r_{\text{avg}}}.$$

We note that the design rate is in general not equal to the “true rate” of a code, but approaches it when the blocklength becomes large. In fact, it is a lower bound on the true rate.

As we will see, we consider the code ensembles in the limit when the blocklength goes to infinity (so that the underlying graph structure is tree-like); we henceforth omit the argument n when we describe the configuration model.

Example 1.9 (The (ℓ, r)-regular Gallager ensemble)**.** *The (ℓ, r)-regular Gallager ensemble is defined as the LDPC(λ, ρ) ensemble when the degree distributions take the form $\lambda(x) = x^{\ell-1}$, $\rho(x) = x^{r-1}$, with $\ell < r$. Equivalently, the variable nodes in this ensemble have degree ℓ (have ℓ outgoing edges) and its check nodes have degree r.*

1.4.3 Density Evolution on the BEC

In order to analyze the performance of the BP decoder on the LDPC(λ, ρ) Gallager ensemble, we can track the evolution of the bit-error probability as iterations of the algorithm are executed. In this section, we demonstrate this analysis on the (ℓ, r)-regular Gallager ensemble and then on more general LDPC(λ, ρ) ensembles when transmission takes place over the BEC in order to instill the correct intuition. In Section 1.4.4, we generalize this analysis to general LDPC(λ, ρ) ensembles with transmission over BMS channels. This generalization is not conceptually difficult but heavier in terms of notation.

We first briefly describe the concept of a computation graph, which is the basis for the tree-type analysis we perform. At every iteration of the BP algorithm, every node in the Tanner graph of a code receives "incoming" messages (beliefs) from its neighbors and sends back an "outgoing" message to each of them. Consider a randomly selected node in the graph; let us say it is a variable node. The outgoing belief it sends to a neighbor depends on all its other incoming beliefs, as well as its own initial belief (based on the channel observation). Similarly, the beliefs sent to this variable node from its neighboring check nodes were computed by these check nodes by using the beliefs they had received from their neighboring variable nodes. By unraveling this "dependency graph", we obtain the so-called *computation graph*. This graph increases in size as more iterations of the BP algorithm are performed.

It is not difficult to see that, for a small graph, we are bound to create a cycle as we apply more iterations of BP; thus the computation graph is not a tree. However, it has been shown that for a fixed number of iterations, and as the blocklength goes to infinity, the graph is a tree with probability one. This is explained in more detail in [61]. In our framework, we first take the blocklength, then the number of iterations, to infinity. In this setting, the underlying graph is with high probability tree-like and we can apply the formalism below.

As we have seen, the BP equations describe the messages exchanged between variable and check nodes in the Tanner graph. In the framework of large blocklengths, the computation graph, as seen from the perspective of some node, is tree-like. Therefore, the incoming messages for this node can be assumed to be independent.

Consider transmission over the BEC with erasure parameter ϵ (we can write this as BEC(ϵ)). Notice the following: The BP decoder cannot decode *incorrectly* because all the "wrong" bits are erasures and not flips. As the decoder is not "fooled" by any bit values, it will either be able to decode correctly, or it will not be able to decode at all.

We denote by $x^{(t)}$ (resp. $y^{(t)}$) the probability that the message sent from a variable node to a check node (resp. a check node to a variable node) at iteration t is an erasure. Then $y^{(t)}$ can be computed as a function of $x^{(t)}$ as

$$y^{(t)} = 1 - (1 - x^{(t)})^{r-1}. \tag{1.20}$$

This is obtained by applying the parity-check rule at the check node side in the framework of large blocklengths. Recall that the message sent from a check node to a variable node is a function of the former's incoming messages on all the *other* edges. Then, applying the parity-check rule, we can observe that the message emitted by the check node is an erasure if any of its incoming messages (on the other edges) are erasures, in which case it cannot compute its parity. Otherwise, it would have been able to decode this variable node by computing its parity (based on the other incoming messages) and deducing this variable node's correct value. Now, the probability that any of the check node's incoming messages are erasures is the complement of the probability that none of them sends an erasure. We assume that those probabilities are independent and identically distributed when the blocklength n goes to infinity. This yields the formula shown in (1.20).

Similarly, the value of $x^{(t+1)}$ from a variable node to a check node at the next iteration is computed in terms of the messages incoming from the other check nodes as

$$x^{(t+1)} = \epsilon(y^{(t)})^{\ell-1}. \tag{1.21}$$

This is because a variable sends a neighboring check node an erasure message if it was erased by the channel (and this occurs with probability ϵ) *and* all its *other* neighboring check nodes also sent it erasure messages (and each of these messages is sent with probability $y^{(t)}$). We finally obtain the formula in (1.21) because we assume that the incoming messages are independent.

We call these equations the *density evolution* (DE) equations. In the case of transmission over the BEC, the densities are simply real-valued probabilities. However, when transmission takes place over a BMS channel, the density can be infinite-dimensional, as we explain in Section 1.4.4.

We rewrite the DE equations for the (ℓ, r)-regular Gallager ensemble when transmission takes place over the BEC(ϵ) for future reference

$$\begin{cases} y^{(t)} &= g(x^{(t)}; \epsilon), \\ x^{(t+1)} &= f(y^{(t)}; \epsilon), \end{cases} \tag{1.22}$$

with initial condition $x^{(0)} = 1$, $g(x; \epsilon) = 1 - (1 - x)^{r-1}$, and $f(y; \epsilon) = \epsilon y^{\ell-1}$, yielding

$$\begin{cases} y^{(t)} &= 1 - (1 - x^{(t)})^{r-1}, \\ x^{(t+1)} &= \epsilon(y^{(t)})^{\ell-1}, \end{cases} \tag{1.23}$$

with initial condition $x^{(0)} = 1$.

Remark 1.10. *The redundancy in writing the DE equations in this chapter is done on purpose. In fact, some of the results we present in this thesis are not restricted to coding and treat more general systems. We therefore write the more general formulation of the DE equations in* (1.22) *to encompass such cases.*

We can rewrite the above system of equations as a recursion in the probability x as

$$x^{(t+1)} = f(g(x^{(t)}; \epsilon); \epsilon), \tag{1.24}$$

with initial condition $x^{(0)} = 1$ and again $g(x; \epsilon) = 1 - (1-x)^{r-1}$, and $f(y; \epsilon) = \epsilon y^{\ell-1}$. Alternatively, we can write the recursion in the probability y as

$$y^{(t+1)} = g(f(y^{(t)}; \epsilon); \epsilon), \tag{1.25}$$

with initial condition $y^{(0)} = 1$.

In this thesis, we will use both DE formulations – depending on what we need. To differentiate between them, we call the system of equations in (1.22) and (1.23) the *DE equations* and those in (1.24) and (1.25) the *DE recursions*.

It is interesting to find the fixed points of the DE equations, to which these equations converge and at which more iterations of the BP algorithm no longer results in change. Let us first check that there exist fixed points to the above DE equations. We recall that $(x^{(t)}, y^{(t)}) \in [0,1]^2$ and observe that the update functions $f(y; \epsilon) = \epsilon y^{\ell-1}$ and $g(x; \epsilon) = 1 - (1-x)^{r-1}$ are non-decreasing in y and x, respectively. Furthermore, by applying the first iteration, we find that $x^{(1)} < x^{(0)}$ and $y^{(1)} < y^{(0)}$, thus the DE iterations decrease the values of $(x^{(t)}, y^{(t)})$. Therefore, using the monotone convergence theorem, we can see that the DE equations decrease the values of $(x^{(t)}, y^{(t)})$ until they converge to some limit values $(x^{(\infty)}, y^{(\infty)})$. As the functions f and g are continuous, then these limit points are solutions to the DE equations. (This should not be too surprising as a decoding algorithm is expected to decrease the erasure probabilities!)

If the DE equations converge to the trivial fixed point $(x^{(\infty)}, y^{(\infty)}) = (0,0)$, then the BP algorithm has successfully decoded all erasures caused by the BEC channel. Otherwise, it has become blocked with some strictly positive fraction of bits that it has not been able to decode. In fact, this depends on the initial probability of erasure of the channel ϵ or, in other words, the amount of noise in the channel. There exists a *threshold* for ϵ below which perfect decoding is possible (and $(x^{(\infty)}, y^{(\infty)}) = (0,0)$) and above which there remains a strictly positive fraction of undecoded bits.[8] We call this value of the channel parameter the *BP threshold* and define it formally below.[9]

Definition 1.11 (BP threshold)**.** *The BP threshold* ϵ_{BP} *of the* (ℓ, r)*-regular LDPC code ensemble is defined by*

$$\epsilon_{BP} \triangleq \sup\{\epsilon \in [0,1] \mid x^{(\infty)} = 0\}, \tag{1.26}$$

where $x^{(\infty)}$ *is the limit value of the recursion in* (1.24)*.*

In comparison, the threshold of noise ϵ up to which the (optimal) MAP decoder can decode perfectly is defined below.

---

[8] We say perfect decoding here in the sense that all errors are corrected. This is only true in the limit of infinite blocklength. When the latter is finite, we still get an error floor for values of ϵ that are lower than the BP threshold.

[9] This definition will be generalized in Section 1.4.4 for general BMS channels.

Definition 1.12 (MAP threshold)**.** *The MAP threshold* ϵ_{MAP} *of the* (ℓ, r)*-regular LDPC code ensemble is defined by*

$$\epsilon_{MAP} \triangleq \inf\{\epsilon \in [0,1] \mid \liminf_{n\to+\infty} \frac{1}{n}\mathbb{E}[\underline{X}|\underline{Y}(\epsilon)] > 0\}, \tag{1.27}$$

where n *is the blocklength,* $\underline{X}$ *is the vector of random variables pertaining to the channel input (codeword),* $\underline{Y}$ *is the vector of random variables pertaining to the channel output (observation), and the expectation* $\mathbb{E}[\cdot]$ *is over the mentioned LDPC ensemble.*

In general for an LDPC(λ, ρ) Gallager ensemble, the BP threshold is lower than the MAP threshold of the ensemble. However, as we will see in Section 1.5, the spatially coupled construction of LDPC codes results in the so-called *threshold saturation* phenomenon that makes the BP threshold equal to the MAP threshold in the limit $n \to +\infty$, for any degree distribution.

For the LDPC(λ, ρ) Gallager ensemble, when transmission takes place over the BEC(ϵ), the update equations take the form $f(y;\epsilon) = \epsilon\lambda(y)$ and $g(x;\epsilon) = 1 - \rho(1 - x)$. We write the corresponding DE equations and DE recursion below for future reference and to facilitate the transition into the next chapter. The DE equations read

$$\begin{cases} y^{(t)} & = 1 - \rho(1 - x^{(t)}), \\ x^{(t+1)} & = \epsilon\lambda(y^{(t)}). \end{cases} \tag{1.28}$$

We can rewrite this in the form of a recursion in x as

$$x^{(t+1)} = \epsilon\lambda(1 - \rho(1 - x^{(t)})). \tag{1.29}$$

The recursion in terms of y is

$$y^{(t+1)} = 1 - \rho(1 - \epsilon\lambda(y^{(t)})). \tag{1.30}$$

1.4.4 Density Evolution on BMS Channels

In Section 1.4.3, we derived the DE equations for the case when transmission takes place over the BEC. These equations describe the evolution of the probabilities $x^{(t)}$ and $y^{(t)}$ that variable and check nodes, respectively, emit erasure messages at iteration t of the BP algorithm. We can observe from (1.23) and (1.28) that the required such updates are polynomial equations in these probabilities.

For the more general case of transmission over BMS channels, instead of tracking the probabilities of erasure messages $x^{(t)}$ and $y^{(t)}$, we look at the evolution of the *probability distributions of the LLRs* $h_{i\to a}$ and $\hat{h}_{a\to i}$ described in Section 1.3.4. In addition, the "channel parameter" is no longer the scalar parameter ϵ but a *channel distribution* or, equivalently, the distribution of the channel LLR l_i (corresponding to the channel realization for variable x_i). The corresponding DE equations will therefore be of integral form and will make use of different operations on distributions.

We consider distributions that are measures on the extended real numbers $\bar{\mathbb{R}}$. Moreover, we consider "symmetric" such measures $\mathtt{x}$ on $\bar{\mathbb{R}}$ satisfying

$$\mathtt{x}(h) = e^{2h}\mathtt{x}(-h), \tag{1.31}$$

for all $h \in \bar{\mathbb{R}}$ where h is interpreted as an LLR. These distributions will be interpreted as the probability distributions of the random variables that describe the LLRs that play the role of the messages exchanged in the BP algorithm or the observations from the channel in the LLR domain. As we consider symmetric-output channels, the messages exchanged during the BP algorithm are also symmetric. We thus implicitly assume that all distributions we consider henceforth are symmetric.

Definition 1.13 (Entropy Functional)**.** *The linear* entropy functional H *describes the average Shannon entropy of the BP marginal probability distribution and is defined as*

$$H(\mathtt{x}) = \int \mathrm{d}\mathtt{x}(h) \log_2(1 + e^{-2h}). \tag{1.32}$$

This entropy functional can be interpreted as a measure of the uncertainty in a bit, once we have run the BP decoder. To understand the expression given in (1.32), recall that the BP marginal probability distribution for a bit $x_i \in \{-1, +1\}$ is equal to

$$p_{X_i|H_i}(x_i|h_i) = \frac{e^{h_i x_i}}{e^{h_i} + e^{-h_i}}, \tag{1.33}$$

where h_i is the LLR associated with x_i computed by BP. At this point, we can write the *average* Shannon entropy of the bit given the LLR h_i, where the average is over the distribution of the LLR, as

$$\begin{aligned} - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) p_{X_i|H_i}(1|h) \log_2 p_{X_i|H_i}(1|h) \\ - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) p_{X_i|H_i}(-1|h) \log_2 p_{X_i|H_i}(-1|h). \end{aligned}$$

To see that this sum is equal to the expression in (1.32), we make the change of variables $h = -h$ in the second integral to obtain

$$\begin{aligned} - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) p_{X_i|H_i}(1|h) \log_2 p_{X_i|H_i}(1|h) \\ - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(-h) p_{X_i|H_i}(-1| - h) \log_2 p_{X_i|H_i}(-1| - h) \end{aligned}$$

Using the symmetry properties in (1.19) and (1.31), this becomes

$$\begin{aligned} - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) p_{X_i|H_i}(1|h) \log_2 p_{X_i|H_i}(1|h) \\ - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) e^{-2h} p_{X_i|H_i}(1|h) \log_2 p_{X_i|H_i}(1|h) \end{aligned}$$

Finally, we plug in the explicit expression of the conditional probability distribution for a bit in (1.33) to obtain

$$\begin{aligned} &= - \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) \log_2 \Big(\frac{e^h}{e^h + e^{-h}} \Big) \Big[\frac{e^h}{e^h + e^{-h}} + \frac{e^{-h}}{e^h + e^{-h}} \Big] \\ &= \int_{\mathbb{R}} \mathrm{d}h \mathtt{x}(h) \log_2 \big(1 + e^{-2h} \big). \end{aligned}$$

We recall the simplified BP equation that describes the evolution of the LLR $h_{i\to a}^{(t)}$ from a variable node to a check node at iteration t,

$$h_{i\to a}^{(t)} = h_i + \sum_{b\in\partial i\setminus a} \hat{h}_{b\to i}^{(t)},$$

where h_i denotes the LLR from the channel to variable x_i and $\hat{h}_{b\to i}^{(t)}$ denotes the LLR from check node b to variable x_i at iteration t. This equation makes use of the summation of LLRs. We therefore define below the probability distribution of an LLR that is obtained by summing other LLRs.

Definition 1.14. *Let h_1 and h_2 be two independent random variables with distributions $\mathtt{x}_1(h_1)$ and $\mathtt{x}_2(h_2)$. Then, the distribution of their sum $h = h_1 + h_2$ is defined by*

$$(\mathtt{x}_1 \circledast \mathtt{x}_2)(h) = \int_{\mathbb{R}^2} \mathrm{d}h_1 \mathrm{d}h_2\, \mathtt{x}_1(h_1)\mathtt{x}_2(h_2)\, \delta(h - (h_1 + h_2)). \tag{1.34}$$

We can also write this as

$$(\mathtt{x}_1 \circledast \mathtt{x}_2)(E) = \int \mathrm{d}h\, \mathtt{x}_2(h)\mathtt{x}_1(E - h), \tag{1.35}$$

for any measurable set $E \in \mathbb{R}$.

The second simplified BP equation we derive in Section 1.3.4 describes the evolution of the LLR $\hat{h}_{a\to i}$ from a check node to a variable node. At iteration $t+1$, we can write it as

$$\hat{h}_{a\to i}^{(t+1)} = 2\tanh^{-1}\Big(\prod_{j\in\partial a\setminus i} \tanh(h_{j\to a}^{(t)}/2)\Big).$$

Due to the calculation required in the equation above, we define the following convolution operator.

Definition 1.15. *Let h_1 and h_2 be two independent random variables with distributions $\mathtt{x}_1(h_1)$ and $\mathtt{x}_2(h_2)$. Then, the distribution of $h = \tanh^{-1}(\tanh(h_1)\tanh(h_2))$ is defined by*

$$(\mathtt{x}_1 \boxast \mathtt{x}_2)(h) = \int_{\mathbb{R}^2} \mathrm{d}h_1 \mathrm{d}h_2\, \mathtt{x}_1(h_1)\mathtt{x}_2(h_2)\, \delta(h - \tanh^{-1}(\tanh(h_1)\tanh(h_2))). \tag{1.36}$$

We can also write this as

$$(\mathtt{x}_1 \boxast \mathtt{x}_2)(E) = \int \mathrm{d}h\, \mathtt{x}_2(h)\mathtt{x}_1\Big(2\tanh^{-1}\Big(\frac{\tanh(E/2)}{\tanh(h/2)}\Big)\Big), \tag{1.37}$$

for any measurable set $E \in \mathbb{R}$.

Each operation taken separately is associative, commutative, and linear, so that

$$\mathtt{x}_1 \circledast \mathtt{x}_2 = \mathtt{x}_2 \circledast \mathtt{x}_1, \qquad \mathtt{x}_1 \boxast \mathtt{x}_2 = \mathtt{x}_2 \boxast \mathtt{x}_1, \tag{1.38}$$

$$(\mathtt{x}_1 \circledast \mathtt{x}_2) \circledast \mathtt{x}_3 = \mathtt{x}_1 \circledast (\mathtt{x}_2 \circledast \mathtt{x}_3), \qquad (\mathtt{x}_1 \boxast \mathtt{x}_2) \boxast \mathtt{x}_3 = \mathtt{x}_1 \boxast (\mathtt{x}_2 \boxast \mathtt{x}_3). \tag{1.39}$$

However when they are taken together there is no distributive or associative law in the sense that $\mathtt{x}_1 \circledast (\mathtt{x}_2 \boxast \mathtt{x}_3) \neq (\mathtt{x}_1 \circledast \mathtt{x}_2) \boxast \mathtt{x}_3$ and $\mathtt{x}_1 \boxast (\mathtt{x}_2 \circledast \mathtt{x}_3) \neq (\mathtt{x}_1 \boxast \mathtt{x}_2) \circledast \mathtt{x}_3$.

We will also use the so-called *duality rules* [63]

$$\begin{cases} H(\mathtt{x} \circledast \mathtt{y}) + H(\mathtt{x} \boxast \mathtt{y}) = H(\mathtt{x}) + H(\mathtt{y}), \\ H(\mathtt{x} \circledast \mathtt{a}) + H(\mathtt{x} \boxast \mathtt{a}) = H(\mathtt{a}), \\ H(\mathtt{a} \circledast \mathtt{b}) + H(\mathtt{a} \boxast \mathtt{b}) = 0, \end{cases} \tag{1.40}$$

where $\mathtt{x}, \mathtt{y}$ are measures on $\bar{\mathbb{R}}$ and $\mathtt{a}$, $\mathtt{b}$ are differences of probability measures $a = \mathtt{x}_1 - \mathtt{x}_2$, $b = \mathtt{x}_3 - \mathtt{x}_4$, where $\mathtt{x}_i$, $i = 1, 2, 3, 4$, are measures on $\bar{\mathbb{R}}$.

We denote by Δ_0 the Dirac mass at 0 and by Δ_∞ that at $+\infty$ and observe the following

$$\Delta_0 \circledast \mathtt{x} = \mathtt{x}, \qquad \Delta_\infty \circledast \mathtt{x} = \Delta_\infty, \tag{1.41}$$

$$\Delta_0 \boxast \mathtt{x} = \Delta_0, \qquad \Delta_\infty \boxast \mathtt{x} = \mathtt{x}. \tag{1.42}$$

In words, the Dirac mass Δ_0 at 0 is the identity of the operator $\circledast$ and the annihilator of the operator $\boxast$, whereas the Dirac mass Δ_∞ at $+\infty$ is the identity of the operator $\boxast$ and the annihilator of the operator $\circledast$.

We consider a family of BMS channels whose distribution $\mathtt{c}_{\mathtt{h}}(h)$ is parametrized by the channel entropy $H(\mathtt{c}_{\mathtt{h}}) = \mathtt{h}$. Then, we can write the DE equations for LDPC(λ, ρ) Gallager ensembles over these channels as

$$\begin{cases} \mathtt{y}^{(t)} &= \rho^{\boxast}(\mathtt{x}^{(t)}), \\ \mathtt{x}^{(t+1)} &= \mathtt{c}_{\mathtt{h}} \circledast \lambda^{\circledast}(\mathtt{y}^{(t)}), \end{cases} \tag{1.43}$$

where by $\rho^{\boxast}(\mathtt{x}) = \sum_r \rho_r \mathtt{x}^{\boxast(r-1)}$ and $\lambda^{\circledast}(\mathtt{x}) = \sum_\ell \lambda_\ell \mathtt{x}^{\circledast(\ell-1)}$, with $\mathtt{x}^{\boxast(r-1)} = \mathtt{x} \boxast \cdots \boxast \mathtt{x}$ (multiplication $r-1$ times) and $\mathtt{x}^{\circledast(\ell-1)} = \mathtt{x} \circledast \cdots \circledast \mathtt{x}$ (multiplication $\ell - 1$ times).

Remark 1.16. *In this section we consider BP messages that are distributions, as opposed to the scalar BP messages that we consider in Section 1.4.3. The analysis pertaining to distributions is not generalized to systems beyond coding in this thesis. Therefore, to avoid superfluous notation, we will not provide more general notation, as we did for the scalar case in* (1.22)*.*

We can alternatively write the DE recursion in terms of the variable node output distribution $\mathtt{x}$ as

$$\mathtt{x}^{(t+1)} = \mathtt{c}_{\mathtt{h}} \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{x}^{(t)})), \tag{1.44}$$

with initial condition $\mathtt{x}^{(0)} = \Delta_0$. Equivalently, we can take the (perhaps) more natural initial condition $\mathtt{x}^{(0)} = \mathtt{c}_{\mathtt{h}}$.

Just as in the case of transmission over the BEC(ϵ), it can be seen that applying the DE recursion on a distribution yields a less degraded one,[10] and that the recursion converges to a limiting distribution $\mathtt{x}^{(\infty)}$. The proofs are quite technical and would require the introduction of many technical notions that we do not use in the thesis, hence we omit them and instead refer the interested reader to [63].

We now redefine the BP and MAP thresholds in the more general setting of transmission over BMS channels.

---

[10] Degradation is an important concept used to compare distributions and can be expressed in different ways, one of which is the following. We say that a distribution $\mathtt{x}_1$ is degraded with respect to the distribution $\mathtt{x}_2$ if $\int \mathrm{d}h\, \mathtt{x}_1(h) f(|\tanh(h/2)|) \geq \int \mathrm{d}h\, \mathtt{x}_2(h) f(|\tanh(h/2)|)$ for all concave non-increasing functions $f : [0,1] \to \mathbb{R}$. We say that $\mathtt{x}_1$ is strictly degraded with respect to $\mathtt{x}_2$ if the inequality is strict.

Definition 1.17 (BP threshold (BMS)). *The BP threshold* $\mathtt{h}_{BP}$ *of an LDPC*(λ,ρ) *code ensemble when transmission takes place over a BMS channel of distribution* $\mathtt{c}_{\mathtt{h}}$ *is defined by*

$$\mathtt{h}_{BP} \triangleq \sup\{\mathtt{h} \in [0,1] \mid \mathtt{x}^{(\infty)} = \Delta_\infty\}, \tag{1.45}$$

where $\mathtt{x}^{(\infty)}$ *is the limit value of the recursion in* (1.44).

Definition 1.18 (MAP threshold (BMS)). *The MAP threshold* $\mathtt{h}_{MAP}$ *of an LDPC*(λ,ρ) *code ensemble when transmission takes place over a BMS channel of distribution* $\mathtt{c}_{\mathtt{h}}$ *is defined by*

$$\mathtt{h}_{MAP} \triangleq \inf\{\mathtt{h} \in [0,1] \mid \liminf_{n\to+\infty} \frac{1}{n}\mathbb{E}[H(\underline{\mathtt{x}}|\underline{\mathtt{y}}(\mathtt{c}_{\mathtt{h}}))] > 0\}, \tag{1.46}$$

where n *is the blocklength,* $\underline{\mathtt{x}}$ *is the channel input,* $\underline{\mathtt{y}}$ *is the channel output,* $H(\underline{\mathtt{x}}|\underline{\mathtt{y}}(\mathtt{c}_{\mathtt{h}}))$ *is the conditional Shannon entropy of the input given by the channel observations, and the expectation* $\mathbb{E}[\cdot]$ *is over the LDPC code ensemble.*

1.4.5 The Potential Function

In Sections 1.4.3 and 1.4.4, we derived DE equations that enable us to assess the performance of an LDPC code ensemble when the blocklength is large. In general, the fixed points of these equations can be viewed as the stationary point equations of a function that is typically called the "potential function". We can view the potential function as an "average form" of the Bethe free energy [64] of the code's graphical model. In the context of statistical mechanics, it is essentially the "replica free energy functional" [65] or, equivalently, the negative of the trial entropy, also called the replica-symmetric free entropy [66–68]. A pedagogical explanation of the precise connection between the Bethe free energy and the potential function in the context of coding can be found in the appendix of [63].

It is already recognized that this variational formulation is a powerful tool to analyze DE updates under suitable initial conditions [55,63,69–74]. There are several ways to define a potential function; all definitions share the property that their stationary point equations are the fixed points of the DE equations of the system. In this section, we list a few such definitions that we will use throughout the thesis. As will be seen, we find that different expressions are convenient for different analyses.

The potential function plays a key role in the derivation of our results. In fact, the real "stars of the show" are the potential *functionals* of the associated spatially coupled systems as considered in the large system size limit (that we formalize in Sections 1.5 and 1.6). In Chapters 2 and 3, we rewrite the (continuous) coupled system potential in terms of the associated DE equations, then we use functional derivatives to compute the number of bits decoded per iteration of the BP algorithm for a spatially coupled code. In Chapters 4 and 5, we use a tool from optimal transport called "displacement convexity" (that we define in Section 1.7) to analyze the (continuous) potential functionals of spatially coupled systems and to prove that, under some conditions, they are "displacement convex" in the space of probability measures.

We refer to the potential functions in this section as "single potentials" as compared to the "coupled potentials" that we later define for spatially coupled LDPC

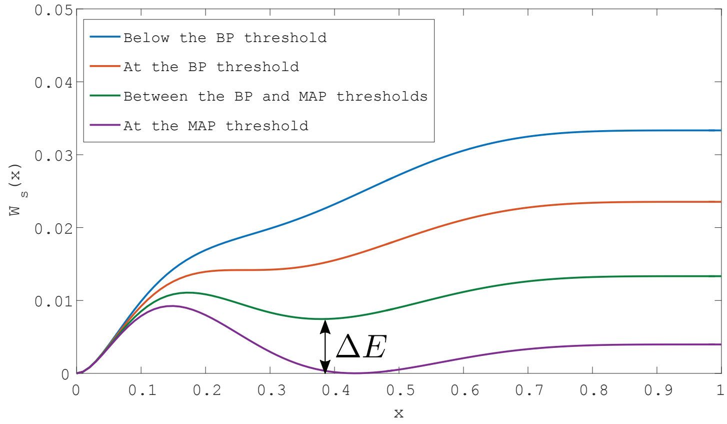

Figure 1.6: We plot the potential function (Definition 1.21) of the $(3,6)$ regular LDPC for the BEC(ϵ) for several values of the erasure probability ϵ. Note that $x=0$ is always a trivial stationary point. For $\epsilon<0.43=\epsilon_{\mathsf{BP}}$ the potential function is strictly increasing and $x=0$ is the only minimum. At $\epsilon_{\mathsf{BP}}=0.43$ a horizontal inflexion point develops. For all $\epsilon\in[\epsilon_{\mathsf{BP}},\epsilon_{\mathsf{MAP}})$, we have a second non-trivial minimum and so the energy gap ΔE is defined and strictly positive. At $\epsilon_{\mathsf{MAP}}=0.488$ the trivial and non-trivial minima are at the same height and ΔE vanishes.

ensembles. We start by writing the single potentials we use when transmission takes place over the BEC; that is, when the messages exchanged between variable and check nodes are *scalar*. To recover the fixed points of the DE equations or one of the DE recursions from these potentials, it is enough to set their derivatives - with respect to one of the probabilities x or y - to zero.

The definition below will be used in Chapter 3, concerned with the speed of decoding when the DE equations involve scalar quantities.

Definition 1.19 (Single Potential in One Scalar Variable [71])**.** *The single-system potential function associated with the DE recursion in* (1.29) *for the LDPC(λ,ρ) Gallager ensemble with transmission on the BEC(ϵ) is defined as*

$$U_s(x;\epsilon)=xg(x;\epsilon)-G(x;\epsilon)-F(g(x;\epsilon);\epsilon), \tag{1.47}$$

where

$$f(x;\epsilon)=\epsilon\lambda(x),\qquad g(x;\epsilon)=1-\rho(1-x),$$
$$F(x;\epsilon)=\int_0^x \mathrm{d}u\, f(u;\epsilon),\qquad G(x;\epsilon)=\int_0^x \mathrm{d}u\, g(u;\epsilon).$$

The definitions below will be used in Chapters 4 and 5 in the context of displacement convexity, where the DE equations involve scalar quantities.

Definition 1.20 (Area Potential in One Scalar Variable [70])**.** *The area potential for the LDPC(λ,ρ) Gallager ensemble with transmission on the BEC(ϵ) can be defined as*

$$A(y;\epsilon)=\int_0^y \mathrm{d}u\,(g^{-1}(u;\epsilon)-f(u;\epsilon)), \tag{1.48}$$

where $f(u;\epsilon)=\epsilon\lambda(u)$, *and* $g(u;\epsilon)=1-\rho(1-u)$.

We note that the area potential $A(y;\epsilon)$ is equal to the single potential $U_s(x;\epsilon)$ in Definition 1.19 if we set $y = g(x;\epsilon)$.

Definition 1.21 (Alternative Area Potential in One Scalar Variable [75]). *Alternatively, the area potential for the LDPC(λ,ρ) Gallager ensemble with transmission on the BEC(ϵ) can be defined as*

$$\tilde{A}(x;\epsilon) = \int_0^x \mathrm{d}u\,(f^{-1}(u;\epsilon) - g(u;\epsilon)), \tag{1.49}$$

where again $f(u;\epsilon) = \epsilon\lambda(u)$, and $g(u;\epsilon) = 1 - \rho(1-u)$.

We note that the alternative area potential $\tilde{A}(x;\epsilon)$ is equal to the single potential $U_s(x;\epsilon)$ in Definition 1.19 *at the fixed points of DE.*

The last two definitions of the potential can be obtained by replacing $x = f(y;\epsilon)$ and $y = g(x;\epsilon)$, respectively, in the following potential function that is expressed in terms of both DE quantities x and y.

Definition 1.22 (Single Potential in Two Scalar Variables [70]). *In terms of both DE variables x and y, the potential for the LDPC(λ,ρ) Gallager ensemble with transmission on the BEC(ϵ) can be defined as*

$$\phi(x,y;\epsilon) = \int_0^x \mathrm{d}u\, f^{-1}(u;\epsilon) + \int_0^y \mathrm{d}u\, g^{-1}(u;\epsilon) - xy, \tag{1.50}$$

where $f(u;\epsilon) = \epsilon\lambda(u)$, and $g(u;\epsilon) = 1 - \rho(1-u)$.

Finally, we give the single potential we use when the DE equations involve probability distributions (instead of scalar probabilities). In our work, we only use the single potential in terms of one variable, the variable node output distribution $\mathtt{x}$, that we define below.

Definition 1.23 (Single Potential in One Variable [76]). *The potential for the LDPC(λ,ρ) Gallager ensemble with transmission over a BMS channel of distribution $\mathtt{c}$ can be defined as*

$$\begin{aligned} W_s(\mathtt{x};\mathtt{c}) = \frac{1}{R'(1)} H(R^{\boxast}(\mathtt{x})) + H(\rho^{\boxast}(\mathtt{x})) - H(\mathtt{x} \boxast \rho^{\boxast}(\mathtt{x})) \\ - \frac{1}{L'(1)} H(\mathtt{c}_{\mathtt{h}} \circledast L^{\circledast}(\rho^{\boxast}(\mathtt{x}))), \end{aligned} \tag{1.51}$$

where H denotes the entropy functional defined in (1.32)*.*

We note that the potential $W_s(\mathtt{x};\mathtt{c})$ above gives back the potential $U_s(x;\epsilon)$ in Definition 1.19 if we specialize it to a system with scalar messages ($\mathtt{x} \to x$) and a scalar channel parameter ($\mathtt{c} \to \epsilon$).

The fixed point form of the DE recursion (1.44) is obtained by setting to zero the functional derivative of the potential functional $W_s(\mathtt{x};\mathtt{c}_{\mathtt{h}})$ defined above with respect to $\mathtt{x}$. In other words $\mathtt{x} = \mathtt{c}_{\mathtt{h}} \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{x}))$ is equivalent to

$$\lim_{\gamma\to 0} \frac{1}{\gamma}(W_s(\mathtt{x}+\gamma\eta;\mathtt{c}_{\mathtt{h}}) - W_s(\mathtt{x};\mathtt{c}_{\mathtt{h}})) = 0, \tag{1.52}$$

where η is a difference of two probability measures [63].

We will repeat and redefine these potentials appropriately whenever needed in the thesis chapters.

Remark 1.24. *In the remainder of the thesis, we omit the subscript* $\mathtt{h}$ *from* $\mathtt{c_h}$ *and the argument* h *from* $\mathtt{x}(h)$ *and* $\mathtt{c_h}(h)$ *most of the time. This is because we will need a subscript (resp. an argument)* z *that represents the position along the spatial coupling chain in the discrete (resp. continuous) case. This will be made clear in Section 1.5.*

The BP and MAP thresholds, $\mathtt{h}_{\text{BP}}$ and $\mathtt{h}_{\text{MAP}}$, can be obtained from the analysis of the stationary points of the potential function. We demonstrate this with the example shown in Figure 1.6, where we plot the "single potential in one scalar variable" (Definition 1.21) for the $(3,6)$-regular Gallager ensemble with transmission over the BEC(ϵ), for several values of ϵ.

We can track the evolution of the probability x in the recursion (1.24) by using the plot of the potential as follows. We consider the initial condition to always be $x^{(0)} = 1$. As the DE recursion decreases the value of x along iterations of BP, we can see from the plot that the value of x will keep decreasing until it reaches the first stationary point. Let us first consider the case when the channel parameter is lower than the BP threshold $\epsilon < \epsilon_{\text{BP}}$. In this case, x will converge to the stationary point $x^{(\infty)} = 0$. When $\epsilon = \epsilon_{\text{BP}}$, a non-trivial stationary point appears and thus x converges to a strictly positive limiting value $x^{(\infty)} > 0$. That is, we define the BP threshold as the smallest value of the channel parameter ϵ at which perfect decoding (in the limit $n \to +\infty$) is no longer possible and $x^{(\infty)} > 0$. Clearly, if we increase the noise level of the channel even more, for $\epsilon > \epsilon_{\text{BP}}$, the recursion will still converge to a positive limit value $x^{(\infty)} > 0$. We observe graphically that for higher values of ϵ, the value of $x^{(\infty)}$ also increases, but its potential value $U_s(x^{(\infty)};\epsilon)$ decreases. At a certain value of ϵ, we obtain a stationary point $x^{(\infty)}$ at which $U_s(x^{(\infty)};\epsilon) = U_s(0;\epsilon)$. We call this parameter the *potential threshold* ϵ_{pot}, which we formalize below. As explained in Section 1.5.4, the potential threshold is equal to the MAP threshold for a large class of ensembles.

For values of ϵ in $[\epsilon_{\text{BP}}, \epsilon_{\text{pot}}]$, we define the *energy gap* ΔE as the difference in the (single) potential at the non-trivial (undesirable) fixed point and the trivial (desirable) one. In the chapters where we use this quantity, namely Chapters 2 and 3, we assume that we have exactly *two* stable DE fixed points (equivalently, two minima for the single potential). Thus, the energy gap is simply the difference in the single potential at these values. In fact, in these chapters, we consider the *dynamic behavior* of spatially coupled ensembles, when the DE equations non-trivially update distributions. This behavior occurs precisely when $\Delta E(\mathtt{c_h}) > 0$.

Definition 1.25 (Potential Threshold (BMS))**.** *Consider an LDPC(λ,ρ) ensemble and a family of BMS channels* $\mathtt{c_h}$ *parametrized by their entropies* $H(\mathtt{c_h}) = \mathtt{h}$*. Then, the potential threshold is defined as*

$$\mathtt{h}_{pot} = \sup\{\mathtt{h} \in [0,1] \,|\, \Delta E(\mathtt{c_h}) > 0\}, \tag{1.53}$$

where $\Delta E(\mathtt{c_h})$ *is the energy gap obtained when the channel distribution is* $\mathtt{c_h}$.

For the specific case of transmission over the BEC, the potential threshold is defined as $\epsilon_{\text{pot}} = \sup\{\epsilon \in [0,1] \,|\, \Delta E(\epsilon) > 0\}$, where $\Delta E(\epsilon)$ is the difference in the single potential of the system (between the non-trivial and the trivial stable fixed points) when the channel parameter is equal to ϵ.

1.5 Spatial Coupling

Spatial coupling was first introduced by Felstrom and Zingangirov in the context of LDPC codes [77]. Spatially coupled codes have been shown to be capacity-achieving on BMS channels under BP decoding. The capacity-achieving property is due to the "threshold saturation" of the BP threshold of the coupled system towards the MAP threshold of the uncoupled code ensemble [43, 63].

Spatial coupling has also been applied to several other problems besides coding [44,45], such as compressive sensing [52–54], random constraint satisfaction problems [55–57], and a coupled Curie-Weiss (toy) model [59, 60].

In this section, we briefly describe the protograph construction of LDPC codes, as well as their random construction. Although we only use the latter in this thesis, we briefly describe the former as it is used in practice and provides some advantages over random ensembles. We also explain the threshold saturation phenomenon and provide the density evolution equations and potential functionals that correspond to spatially coupled codes. Finally, we describe and compare the thresholds involved in spatially coupled codes, as well as their underlying uncoupled systems, namely the BP, MAP, potential, and area thresholds.

1.5.1 Spatially Coupled Configurations and Threshold Saturation

In the 1980's, Tanner had the idea of "unwrapping" (or "flattening out") cyclic block codes and terminating (or fixing) the boundaries [78,79]. The resulting convolutional structure not only exhibited very good performance, but also paved the way to many variants of convolutional LDPC code constructions that finally led to the design of universally capacity-achieving spatially coupled codes.

The analysis of convolutional LDPC ensembles via density evolution was introduced by Sridharan, Lentmaier, Costello and Zigangirov [45, 80], in the case when transmission takes place over the BEC, and then generalized by the same authors [44, 45] for transmission over general channels.

The thresholds determined by these results and analyses seemed to exhibit the (now proven) "threshold saturation" property of convolutional LDPC codes; it equates the BP threshold of those coupled codes with the MAP threshold of the underlying (uncoupled) ensemble [81]. This property was first observed numerically by G. Liva and then formulated as a conjecture by Lentmaier and Fettweis. At approximately the same time, this attractive property was proved rigorously for transmission over the BEC by Kudekar, Richardson, and Urbanke [41, 69, 70]. Shortly afterwards, another proof technique was introduced by Yedla, Jian, Nguyen, and Pfister [71] to establish the same result. Threshold saturation was also afterwards proved for transmission over BMS channels by Kudekar, Richardson, and Urbanke [43] as well as (using a different, more concise technique), Kumar, Young, Macris, and Pfister [63]. In fact, it has further been shown by Giurgiu, Macris, and Urbanke [82] that the MAP threshold of the underlying (uncoupled) code ensemble is equal to that of the associated spatially coupled code ensemble for a large class of codes. We elaborate further on this in Section 1.5.4.

In this section, we describe two constructions of spatially coupled codes: the "random construction" that we use throughout the thesis and the "protograph construction". The idea behind the excellent performance of both constructions is the same and is caused by "seeding". They are also both obtained by taking say, $2L_c+1$ copies of the uncoupled code, erasing all the original edges and reconnecting the copies locally. The precise way in which these systems are coupled depends on the construction we use.

Spatially coupled codes are capacity-achieving under the low-complexity BP algorithm: They exhibit excellent performance up to thresholds higher than the BP threshold of the underlying system. However, they have smaller rates than their associated underlying codes. This rate-loss effect becomes negligible when the number of copies becomes large. Protograph-based spatially coupled codes show a better trade-off regarding rate, thresholds, and blocklength than random-based coupled codes. However, the latter is easier to analyze. For instance, both code constructions exhibit excellent performance experimentally, but threshold saturation has only been proved rigorously for the random construction.

We end this section by defining the notion of a "profile" that describes the error distributions along a spatially coupled system and that is a key object in our work. We present the one-sided and two-sided profiles, the former of which we use in our derivations.

Remark 1.26. *A small remark on terminology: We use "single", "uncoupled", or "underlying" code to describe the "original" LDPC code, such as the one described in Section 1.2 and "spatially coupled" or simply "coupled" code to refer to the codes we describe below.*

The Random Construction of Spatially Coupled Codes

We describe the construction of the spatially coupled LDPC code ensemble that we use in this thesis; it is referred to as the (ℓ, r, L_c, w) ensemble (when the underlying code is regular) or the random construction ensemble (more generally). We then explain how this code is terminated at the boundaries so that the threshold saturation phenomenon occurs.

We start with an informal description of the spatially coupled ensemble and then formalize it.

Consider a single LDPC code ensemble described by its variable and check node degree distributions (from a node perspective) L and R, respectively. For instance, consider the following distributions

$$\begin{aligned}L(x) &= 0.3x^2 + 0.6x^3 + 0.1x^5,\\ R(x) &= x^4.\end{aligned}$$

This code ensemble satisfies the following property: A fraction 0.3 of its variable nodes has degree 2, a fraction 0.6 has degree 3, and the rest (a fraction 0.1) has degree 5, and all its check nodes have degree 4. We note that the number of variable sockets (the total number of edges emanating from the variable node side) should be equal to the number of check sockets (the total number of edges emanating from the check node side). That is, if N and M denote the numbers of variable and check nodes, respectively, then for the example above we require that $N \times (0.3 \times 2 + 0.6 \times 3 + 0.1 \times 5) = M \times (1 \times 4)$. This condition applies in general and not only to this example.

The corresponding spatially coupled random code ensemble is constructed by making several, say $2L_c + 1$, copies or replicas of the single system on the spatial positions $z = -L_c, \dots, L_c$, and by reconnecting them locally. The number of replicas is called the *coupling chain length* and the maximum size of the local connections (the largest distance between two replicas that share edges) is called the *window size* that can also be thought of as a smoothing parameter. We say that an edge connects a "variable node socket" to a "check node socket" (or vice-versa). We erase the edges from all replicas but keep their "empty" (unconnected) sockets in place and reconnect them as follows. Consider a variable node socket (or outgoing edge) at position z of the coupling chain $-L_c, \dots, L_c$. Connect this socket to a check node socket chosen uniformly and independently at random in the range of positions $z, \dots, z+w-1$. This guarantees a very important property: The degree distributions of variables and checks at each position of the coupling chain are the same as those of the uncoupled code ensemble, with the additional property that local replicas are connected (we discuss this in more detail below). We note that the labeling of the coupling axis (start and end points) is arbitrary and can be modified for convenience.

The formalism we describe below is adapted from [41] so that it applies to irregular LDPC(λ, ρ) ensembles.

Consider an instance of the irregular LDPC(λ, ρ) code ensemble with $\hat{N}$ variable nodes and $\hat{M}$ check nodes, and a realization of the degree distribution imposed by the ensemble. We erase all edges in the code instance but keep the sockets associated with variables and checks in place. We will eventually reconnect them.

Place $2L_c + 1$ copies of the code instance on positions $z = -L_c, \dots, L_c$. Later, we will define the numbers N and M of variable and check nodes at each position. For each variable node v, with d_v sockets, we define the two following quantities. The *type* t_v of v is the w-tuple $t_v = ((t_v)_0, \dots, (t_v)_{w-1})$ where $(t_v)_j$ is equal to the total number of edges emanating from v to check nodes at position $z = i + j$. A *constellation* k_v of v is the d_v-tuple $k_v = ((k_v)_0, \dots, (k_v)_{d_v-1})$ where $(k_v)_l$ is equal to the position difference $j = z - i$ between v and the check connected to it via its l^{th} socket. A small example to demonstrate these two definitions is shown in Figure 1.7. We can write the relation between the type $\tau_v(k_v)$ and the constellation k_v of a variable node as $\big(\tau_v(k_v)\big)_j = \sum_{l=0}^{d_v-1} \mathbb{1}_{(k_v)_l = j}$.

Then, as we want to choose the position of each edge independently, we impose a uniform distribution on the set of all constellations. We call p_v the distribution for the variable node v and define it as

$$p_v(t_v) = \frac{|\{k_v : \tau_v(k_v) = t_v\}|}{w^{d_v}}.$$

Then the distribution p for a *set of types* $t = \{t_0, \dots, t_{\hat{N}-1}\}$ on the $\hat{N}$ variable nodes is defined as

$$p(t) = L(p_v(t_v)),$$

where L is the variable degree distribution from a node perspective. Now we choose the number of variable nodes per position N so that $N/\hat{N} \in \mathbb{N}$ and $Np_v(t_v) \in \mathbb{N}$ for all types t_v. For each position z on the coupling chain, we choose $Np_v(t_v)$ variables with edges assigned according to type t_v. Finally, for each variable v, we use a random permutation on the sockets chosen uniformly from the set of all permutations on d_v letters in order to map a type to a constellation. The number of resulting check nodes per position M satisfies $M/\hat{M} = N/\hat{N}$.

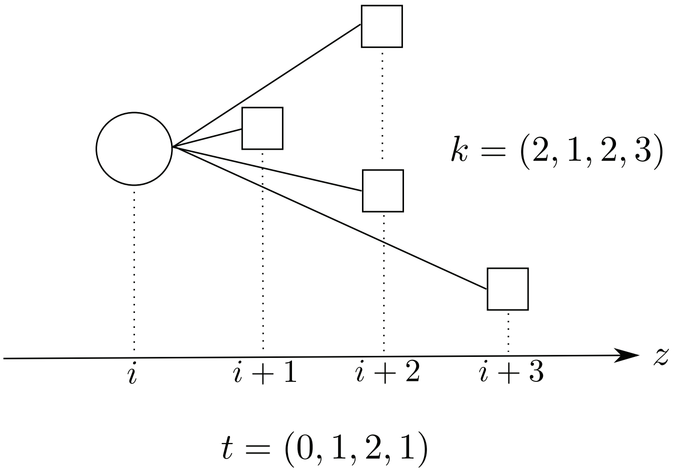

Figure 1.7: We show an example of the type and constellation of a variable node when the window size is $w = 4$. The node is located at position i and is connected to four check nodes at positions $i+1$, $i+2$, and $i+3$. Then the type corresponding to how many check nodes it is connected to at each position $i, \dots, i+3$ is $t = (0,1,2,1)$ and the constellation that shows for each socket, the difference in the distance between the variable and its connected check node is $k = (2,1,2,3)$.

You might have noticed a "flaw" in the construction above: At the boundaries, the original degree distributions of the underlying code ensemble cannot be preserved. To see this, think of the (ℓ, r)-regular code ensemble where all variable nodes have degree ℓ and check nodes degree r. We construct the edges from the variable node perspective.[11] Therefore, from every variable node at position z, we connect its ℓ sockets to check node sockets chosen uniformly and independently at random on positions $z, \dots, z+w-1$. Then, at the left boundary, the check nodes in the first $w-1$ positions $-L_c, \dots, -L_c+w-1$ will not be reached by "enough" variable node sockets to have a degree of r. At the right boundary, the variable nodes on the last $w-1$ positions $L_c-w+1, \dots, L_c$ will seek to connect to check nodes at positions $L_c, \dots, L_c+w-1$, but the coupling chain only extends to position L_c.

Therefore, we modify the boundaries of the coupled code in a way that is convenient for us. On the left side, we add *auxiliary variables* on positions $z = -L_c-w+1, \dots, -L_c-1$ (and their corresponding check nodes) so that the check nodes on positions $z = -L_c, \dots, -L_c+w-1$ have degree r, and on the right side, we add *auxiliary check nodes* on positions $z = L_c+1, \dots, L_c+w-1$ (and their corresponding variable nodes) so that the variable nodes on positions $z = L_c-w+1, \dots, L_c$ have degree ℓ. Although the auxiliary nodes do not have regular degrees, we are more interested in "regularizing" the degrees of the nodes on the original positions $z = -L_c, \dots, L_c$. In Figure 1.9 we show a spatially coupled $(3,6)$-regular code with spatial length L and window size $w = 2$. We have two auxiliary systems, one at each boundary, denoted by dashed lines and lighter colors.

Spatially coupled codes exhibit threshold saturation because of *seeding*. This is done by fixing, to some predetermined value, the $w-1$ auxiliary variable nodes on the left boundary and the $w-1$ auxiliary variable nodes on the right boundary.

---

[11] This is not obligatory, but we choose to fix this technique here for clarity.

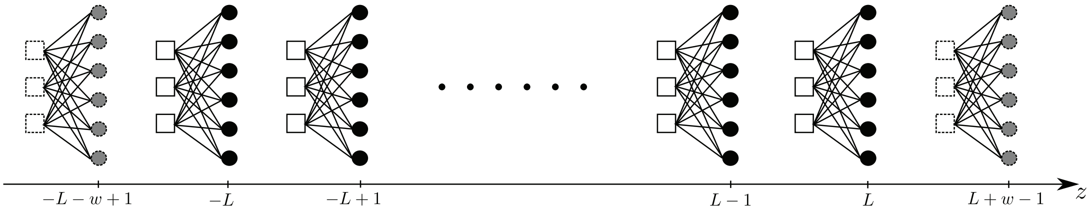

Figure 1.8: In order to construct a "random" spatially coupled code with window size $w = 2$, we first place $2L_c+2w-1$ copies of the randomly constructed LDPC (uncoupled) Tanner graph on the spatial positions $z = L_c-w+1, \dots, L_c+w-1$. The leftmost $w-1 = 1$ and the rightmost $w - 1 = 1$ copies are the auxiliary codes, whose variable nodes are fixed to some predetermined values; we draw these codes with dashed borders. We note that this is not a good example since the uncoupled graphs are fully connected (whereas LDPC codes are sparse), but we use it for demonstration purposes.

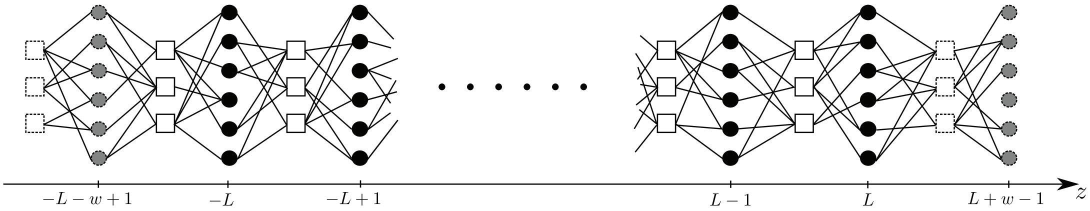

Figure 1.9: To coupled the copies of the uncoupled code, we first erase all the edges, then reconnect them randomly within a window size, which is $w = 2$ in our example, such that the degree distributions of the original (not auxiliary) copies are preserved. The variable nodes at the boundaries are colored in gray and are fixed to some predetermined values known to both the sender and the receiver; they constitute the "seed" that makes decoding easier at the boundaries.

For instance, the values of these bits can all be zeros. This "extra information" and the graph structure are known to both the sender and the receiver. With this modification, the constraints on the check nodes connected to the fixed variable nodes become easier to solve (as they have fewer unknowns to solve for). Therefore, decoding becomes easier on the boundaries of the coupled chain and more variables on the boundaries are decoded. Once this occurs, then other check nodes connected to these recently decoded variables become easier to solve. This makes larger the boundary effect with extra information, and the effect of the originally fixed auxiliary bits moves inwards into the chain. The propagation of information from the boundaries into the middle of the chain eventually leads to a completely decoded graph *when the channel parameter is lower than the MAP threshold*. With the uncoupled code, complete decoding was possible only when the parameter is lower than the BP threshold. This is precisely why we say that spatially coupled codes exhibit the threshold saturation phenomenon.

We note that the addition of auxiliary check nodes decreases the design rate of the code. We recall that the rate is defined as $r(\lambda, \rho) = 1 - \ell_{\text{avg}}/r_{\text{avg}}$. Due to the auxiliary check nodes with few connections, the average check node degree r_{avg} will

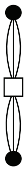

Figure 1.10: The protograph of a standard $(3,6)$-regular ensemble.

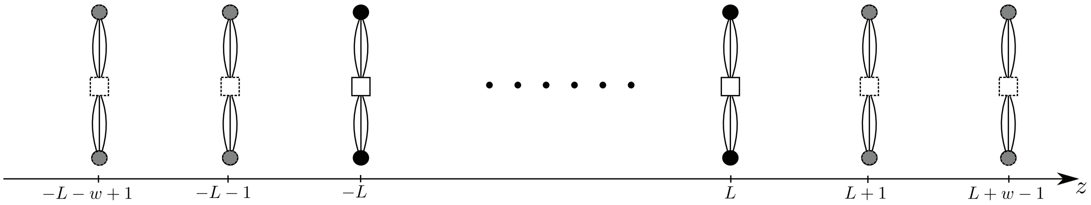

Figure 1.11: We place $2L_c + 2w - 1$ copies of the protograph on the spatial positions $z = -L_c - w + 1, \ldots, L_c + w - 1$. The leftmost $w - 1$ and the rightmost $w - 1$ copies are the auxiliary codes, whose variables nodes are fixed to some predetermined values; we draw them with dashed borders.

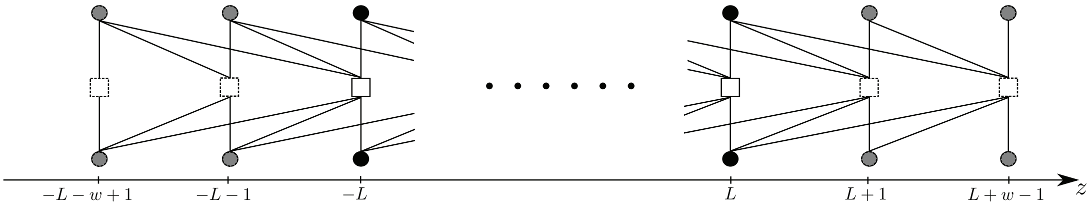

Figure 1.12: The edges of the protographs are spread to their adjacent codes. Since the window size is $w = \ell = 3$, then for each variable node at position z, $\ell - 1$ edge bundles are connected to check nodes at positions $z+1, \ldots, z+w-1$ (one bundle per position). The auxiliary variables (in gray) are fixed to predetermined values; thus, they serve as the seed and initiate decoding at the boundaries.

decrease, which in turn decreases the rate.[12] However, spatially coupled systems are typically analyzed and implemented with large blocklengths, in which case the effect of the rate loss becomes negligible.

The Protograph Construction of Spatially Coupled Codes

Protograph-based spatially coupled ensembles exhibit excellent performance and experimentally demonstrate threshold saturation. Such codes are implemented in practice on chips. There are several variants of protograph-based coupled codes, as well as several ways to describe each of them [41, 83–89]. Here, we describe the simple example of the (ℓ, r)-regular protograph-based spatially coupled code with $\ell = 3$, $r = 6$, and window size $w = \ell = 3$.

A protograph is a Tanner graph with a relatively small number of nodes [85, 90]. We first consider a protograph of a standard $(3,6)$-regular ensemble, that is

---

[12] The auxiliary variable nodes do not affect the design rate because they are fixed.

represented by two variable nodes, with 3 outgoing edges each, and 1 check node with 6 outgoing edges, as shown in Figure 1.10. We "lift" the protograph by a factor of M, so that there are M identical such protographs stacked on top of each other; this means that we now have a total of $2M$ variable nodes and M check nodes. We call every overlapping stack of M edges an edge "bundle". Within each bundle, we make a random permutation on the edges. This yields a larger $(3,6)$-regular code with the same rate as the original protograph.

In order to construct the corresponding protograph-based spatially coupled code, we first place $2L_c + 1$ copies of the lifted protograph on the spatial positions $z = -L_c, \ldots, L_c$. Then, to couple the copies, we spread out the edges. More specifically, out of the $\ell = 3$ bundles outgoing from a variable node at position z on the coupling axis, $\ell - 1 = 2$ edge bundles are connected to the check nodes at positions $z+1, z+2$ (1 bundle to each destination position).

As with the random construction of coupled codes, this creates degree irregularities at the boundaries: the check nodes at the left boundary have less than 6 outgoing bundles, and the variable nodes at the right boundary have less than 3 outgoing bundles. We there add auxiliary protographs at the boundaries, on $z = -L_c - w + 1, \ldots, -L_c - 1$ and $z = L_c + 1, \ldots, L_c + w - 1$. This regularizes the "original" protographs on the coupling axis. The variable nodes of the auxiliary codes are fixed to some predetermined values; this is the "seed" that causes easier decoding at the boundaries.

The coupling procedure can alternatively be seen as follows. We first place the codes (including the auxiliary ones) on the spatial axis $z = -L_c - w + 1, \ldots, L_c + w - 1$, as shown in Figure 1.11. Then, we connect every $w = 3$ adjacent copies, as shown in Figure 1.12.

The ensemble of protograph-based coupled codes is then the set of codes obtained from permutations on the edges within each bundle. To obtain a particular instance of the ensemble, we pick a permutation from the set of all permutations uniformly at random and connect the variable nodes to the check nodes according to this permutation.

One-Sided and Two-Sided Profiles

We have so far seen two constructions of spatially coupled codes, the protograph-based and the random ensembles. In both cases, we spread out $2L_c + 1$ replicas of the underlying uncoupled code on the positions $z = -L_c, \ldots, L_c$, and locally connect them within a window size w. In the remainder of this thesis, we consider the random constructions of spatially coupled codes. In order to track the behavior of belief propagation on such codes, we define the so-called "profile" below.

At every iteration of the BP algorithm, the variable and check nodes of the spatially coupled graph exchange messages that are described by a set of coupled DE iterative equations. We call the solution to the DE equations the "decoding profile" x (resp. y) in the scalar case and $\mathtt{x}$ (resp. $\mathtt{y}$) in the case of higher dimensions. A profile is a vector of distributions of the BP log-likelihood estimates of the bits along the spatial axis of coupling; they can be thought of as "error distributions". More precisely, let the integer $z \in \{-L_c, \ldots, L_c\}$ denote the position along the spatial direction of the graph construction (on which the replicas are spread). Then the z^{th} component of $\mathtt{x}$, call it $\mathtt{x}_z$, denotes the distribution the BP log-likelihood estimate

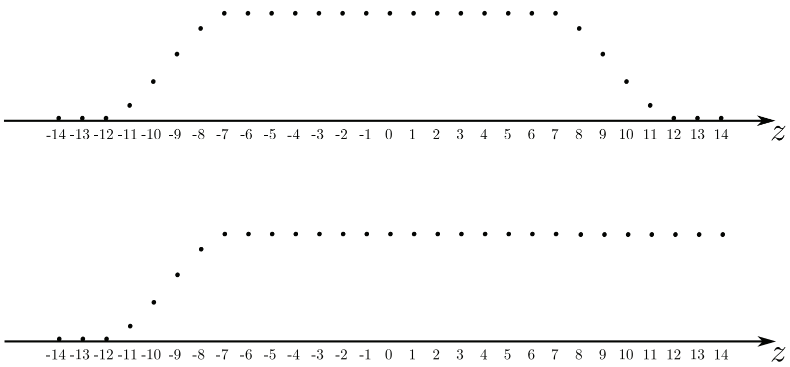

Figure 1.13: We plot the profiles x_z on the vertical axis as a function of the spatial coordinate z. On the top, we draw the two-sided profile and on the bottom, its associated one-sided profile.

at the z^{th} position. In the special case of the BEC, for instance, this component is reduced to the usual scalar erasure probability $0 \le x_z \le 1$ at position z along the spatial axis of coupling.

Spatially coupled codes perform well and are capacity achieving due to the "threshold saturation" phenomenon. In terms of the decoding profile of a *spatially coupled* code, as long as the channel noise is below the MAP threshold, the profile converges to the all-Δ_∞ vector after enough iterations of the BP algorithm, where Δ_∞ is the Dirac mass at infinite log-likelihood (i.e., perfect knowledge of the bits). In the special case of the BEC, the all-Δ_∞ vector corresponds to a vector of scalar erasure probabilities driven to zero by DE iterations. Whereas, the probability distributions of the log-likelihoods of bits of the corresponding *uncoupled* code only converge to Δ_∞ when the channel noise is below the BP threshold (which is lower than the MAP threshold).

The threshold saturation phenomenon is made possible due to "seeding" at the boundaries of the spatially coupled code. Seeding means that we fix the bits at the boundaries so that the probability distributions of their log-likelihoods are Δ_∞. This facilitates BP decoding near the boundaries, and this effect is propagated along the rest of the coupled chain. The minimum size of the seed that guarantees the propagation of the decoding effect is of the same order as the size w of the coupling window (however an exact determination of the minimum possible such size is an still an interesting open question). We denote by $\hat{\mathtt{x}}$ the decoding profile. Then, the initial condition on the profile is $\hat{\mathtt{x}}_z = \Delta_\infty$ (perfect information) at the boundaries $z = -L_c, \dots, -L_c + w - 1$ and $z = L_c - w + 1, \dots, L_c$, and $\hat{\mathtt{x}}_z = \Delta_0$ (only information from the channel) elsewhere. In the special case of transmission over the BEC(ϵ), we have $\hat{x}_z = 0$ when $z = -L_c, \dots, -L_c + w - 1$ or $z = L_c - w + 1, \dots, L_c$, and $\hat{x}_z = \epsilon$ elsewhere. We call this the "two-sided profile" because we have seeding on both boundaries.

The two-sided profile described above is perfectly symmetric. That is why we *define the associated one-sided profile* $\mathtt{x}$ below.

$$\mathtt{x}_z = \hat{\mathtt{x}}_z, \qquad z = -L_c, \dots, 0, \tag{1.54}$$

$$\mathtt{x}_z = \mathtt{x}_0, \qquad z = 1, \dots, L_c, \tag{1.55}$$

where $\mathtt{x}_0$ denotes the distribution $\mathtt{x}_z = \hat{\mathtt{x}}_z$ at $z = 0$. We show an example of a two-sided profile and its associated one-sided profile in Figure 1.13, when $L_c = 14$.

We note that it is equivalent to work with either profile, in the sense that the behavior is the same (up to symmetry). In our work, we use the one-sided profile because it is more convenient in the context of displacement convexity (see Section 1.7). In Chapters 2 and 3, however, when we calculate the number of bits decoded per iteration of BP, we note that the number we find is half of that we would have obtained had the analysis involved the two-sided profile (which is the "real" profile in practice).

In the remainder of the thesis, we redefine the range of positions z on which we spread the replicas of the underlying code (which was previously $z = -L_c, \dots, L_c$). We set it to $z = -w + 1, \dots, L_c$ and seed at the boundary $z = -w + 1, \dots, 0$.

1.5.2 Coupled Density Evolution Equations

Transmission over the BEC

In this section, we write the system of DE equations as well as the DE recursion associated with the spatially coupled LDPC(λ, ρ) code ensemble when transmission takes place over the BEC(ϵ). The main difference between the equations we write here and those in Section 1.4.3 is that here the updated erasure probabilities are obtained by average other probabilities (from the previous BP iteration) over a window.

Spatially coupled codes are constructed in such a way that replicas within a certain window size w are dependent. In the random construction of coupled systems that we consider, every edge outgoing from a variable node at position z on the coupling chain is connected to a randomly selected check node at positions $z, z + 1, \dots, z + w - 1$. Similarly, every edge outgoing from a check node at position z is connected to a randomly chosen variable node at positions $z - w + 1, \dots, z$.[13] Therefore, to decode a variable node at position z, we must use the information in all the system replicas on which it depends. More specifically, the recursive DE equation in the error probability of a variable node at position z depends on the error probabilities of check nodes on positions $j = z, \dots, z + w - 1$, which in turn depend on the error probabilities of variable nodes on positions $k = j - w + 1, \dots, j$. We formalize this when we write the spatially coupled version of DE equations below.

We denote by $\tilde{x}_z^{(t)}$ and $\tilde{y}_z^{(t)}$ the variable node output and check node output probabilities, respectively, at position $z = -w + 1, \dots, L_c$ on the spatial axis and at time $t \in \mathbb{N}$. We also define a general coupling window function that we will use below.

Definition 1.27 (General Coupling Window Function)**.** *We define a general coupling window function $w(\cdot)$ that satisfies $w(z) > 0$ for $z \in [-W, +W]$, where $W \leq +\infty$ and $w(z) = 0$ elsewhere. We also impose the condition that $\int_{\mathbb{R}} \mathrm{d}z \, w(z) = 1$. Then, we define the normalized function*

$$w_W(z) = \frac{w(z)}{\frac{1}{2W} \sum_{j=-W}^{W} w\left(\frac{j}{W}\right)}.$$

---

[13] We assume the probability that one such edge is connected to one such "destination node" is uniform (unless explicitly stated otherwise).

Remark that $\frac{1}{2W}\sum_{j=-W}^{W} w\big(\frac{j}{W}\big) = 1$ *and that that as* $W \to +\infty$*, we have* $w_W(z) \to w(z)/\int_{\mathbb{R}} \mathrm{d}z\, w(z) = w(z)$*.*

Then, the system of DE equations takes the form

$$\begin{cases} \tilde{y}_z^{(t)} & = g\Big(\sum\limits_{j=-\infty}^{+\infty} a_{\tilde{z},z} x_z^{(t)}; \epsilon_z\Big), \\ \tilde{x}_z^{(t+1)} & = f\Big(\sum\limits_{i=-\infty}^{+\infty} a_{z,\tilde{z}} y_z^{(t)}; \epsilon_z\Big), \end{cases} \tag{1.56}$$

where $g(x;\epsilon) = 1-\rho(1-x)$, $f(y;\epsilon) = \epsilon\lambda(y)$, and the coupling window coefficients $a_{z,\tilde{z}}$ satisfy $a_{z,\tilde{z}} = \frac{1}{2W} w_W\big(\frac{\tilde{z}-z}{W}\big)$. The initial conditions are written below.

If we consider the uniform window function, this yields the equations $\tilde{y}_z^{(t)} = g\Big(\frac{1}{w}\sum_{j=0}^{w-1} \tilde{x}_{z-j}^{(t)}; \epsilon_z\Big)$, $\tilde{x}_z^{(t+1)} = f\Big(\frac{1}{w}\sum_{i=0}^{w-1} \tilde{y}_{z+i}^{(t)}; \epsilon_z\Big)$. For the LDPC$(\lambda,\rho)$ code ensemble, this becomes

$$\begin{cases} \tilde{y}_z^{(t)} & = 1-\rho\Big(1-\frac{1}{w}\sum\limits_{j=0}^{w-1} \tilde{x}_{z-j}^{(t)}\Big), \\ \tilde{x}_z^{(t+1)} & = \epsilon_z\lambda\Big(\frac{1}{w}\sum\limits_{i=0}^{w-1} \tilde{y}_{z+i}^{(t)}\Big). \end{cases}$$

For this system of equations $\epsilon_z = \epsilon$, for $z = 1,\dots,L_c+w-1$ and $\epsilon_z = 0$ for $z = -w+1,\dots,0$. Furthermore we fix the left boundary to $\tilde{x}_z^{(t)} = \tilde{y}_z^{(t)} = 0$ for $z = -w+1,\dots,0$, for all $t \in \mathbb{N}$. These conditions express perfect information at the left boundary, which is what enables seeding and decoding along the chain of coupled codes. The initial conditions for the remainder of the coupling chain are $\tilde{x}_z^{(0)} = \tilde{y}_z^{(0)} = 1$ for $z = 1,\dots,L_c+w-1$.

Alternatively, we can write the DE recursion in $\tilde{x}_z^{(t)}$ as

$$\tilde{x}_z^{(t+1)} = f\left(\frac{1}{w}\sum_{i=0}^{w-1} g\Big(\frac{1}{w}\sum_{j=0}^{w-1} \tilde{x}_{z+i-j}^{(t)}; \epsilon_z\Big); \epsilon_z\right). \tag{1.57}$$

For the LDPC(λ,ρ) code ensemble, this becomes

$$\tilde{x}_z^{(t+1)} = \epsilon_z\lambda\left(1-\frac{1}{w}\sum_{i=0}^{w-1}\rho\Big(1-\frac{1}{w}\sum_{j=0}^{w-1}\tilde{x}_{z+i-j}^{(t)}\Big)\right). \tag{1.58}$$

with the same initial conditions on $\tilde{x} = (\tilde{x}_{-w+1},\dots,\tilde{x}_{L_c+w-1})$ as above.

It will be convenient to work with a smoothed version of the profile $\tilde{x}_z^{(t)}$, namely $x_z^{(t)} = \frac{1}{w}\sum\limits_{i=0}^{w-1}\tilde{x}_{z-i}^{(t)}$, where $x_z^{(t)}$ denotes the check node input probability at position $z = -w+1,\dots,L_c$ on the spatial axis and at time $t \in \mathbb{N}$. Then, using this change of variables, the DE recursion can be written as

$$x_z^{(t+1)} = \frac{1}{w}\sum_{i=0}^{w-1} f\left(\frac{1}{w}\sum_{j=0}^{w-1} g\big(x_{z-i+j}^{(t)}; \epsilon_z\big); \epsilon_{z-i}\right). \tag{1.59}$$

For the LDPC(λ,ρ) code ensemble with transmission over the BEC(ϵ), the recursion in (1.58) can be rewritten as

$$x_z^{(t+1)} = \frac{1}{w}\sum_{i=0}^{w-1}\epsilon_{z-i}\lambda\left(1-\frac{1}{w}\sum_{j=0}^{w-1}\rho\big(1-x_{z-i+j}^{(t)}\big)\right). \tag{1.60}$$

We repeat the initial and boundary conditions for completeness. We have $\epsilon_z = \epsilon$, for $z = 1, \dots, L_c + w - 1$ and $\epsilon_z = 0$ for $z = -w + 1, \dots, 0$. We fix the left boundary to $x_z^{(t)} = 0$ for $z = -w + 1, \dots, 0$, for all $t \in \mathbb{N}$. The initial condition for the iterations is $x_z^{(0)} = 1$ (or ϵ) for $z = 1, \dots, L_c + w - 1$.

In Chapter 4, we find it more convenient to work with the coupled DE recursion (and potential) as expressed in terms of the variable node input probability $y_z^{(t)}$ at position $z = -w + 1, \dots, L_c$ and time $t \in \mathbb{N}$. We express the DE recursion as

$$y_z^{(t+1)} = 1 - \rho\left(1 - \frac{1}{w}\sum_{i=0}^{w-1} \epsilon_{z-i}\lambda\Big(\frac{1}{w}\sum_{j=0}^{w-1} y_{z-i+j}^{(t)}\Big)\right), \tag{1.61}$$

with $\epsilon_z = \epsilon$, for $z = 1, \dots, L_c + w - 1$ and $\epsilon_z = 0$ for $z = -w + 1, \dots, 0$, and the boundary $y_z^{(t)}$ fixed to 0 for $z = -w + 1, \dots, 0$, for all $t \in \mathbb{N}$. On the remainder of the coupling chain, the initial condition is $y_z^{(0)} = 1$ for $z = 1, \dots, L_c + w - 1$.

Transmission over BMS Channels

When transmission takes place over general BMS channels, the messages exchanged by the nodes are probability distributions of LLRs. The DE equations and recursions described in Section 1.4.4 are thus modified accordingly.

We denote by $\tilde{\mathtt{x}}_z^{(t)}$ and $\tilde{\mathtt{y}}_z^{(t)}$ the variable node output and check node output distributions, respectively, at position $z = -w + 1, \dots, L_c$ on the spatial axis and at time $t \in \mathbb{N}$. For simplicity we take a uniform coupling window. The system of DE equations thus takes the form

$$\begin{cases} \tilde{\mathtt{y}}_z^{(t)} &= \rho^{\boxast}\Big(\frac{1}{w}\sum\limits_{j=0}^{w-1} \tilde{\mathtt{x}}_{z-j}^{(t)}\Big), \\ \tilde{\mathtt{x}}_z^{(t+1)} &= \mathtt{c}_z \circledast \lambda^{\circledast}\Big(\frac{1}{w}\sum\limits_{i=0}^{w-1} \tilde{\mathtt{y}}_{z+i}^{(t)}\Big). \end{cases}$$

For this system of equations $\mathtt{c}_z = \mathtt{c}$, for $z = 1, \dots, L_c + w - 1$ and $\mathtt{c}_z = \Delta_\infty$ for $z = -w + 1, \dots, 0$. Furthermore we fix the left boundary to $\tilde{\mathtt{x}}_z^{(t)} = \tilde{\mathtt{y}}_z^{(t)} = \Delta_\infty$ for $z = -w + 1, \dots, 0$, for all $t \in \mathbb{N}$. These conditions express perfect information at the left boundary, which is what enables seeding and decoding along the chain of coupled codes. The initial conditions for the remainder of the coupling chain are $\tilde{\mathtt{x}}_z^{(0)} = \tilde{\mathtt{y}}_z^{(0)} = \Delta_0$ for $z = 1, \dots, L_c + w - 1$.

Alternatively, we can write the DE recursion in $\tilde{\mathtt{x}}_z^{(t)}$ as

$$\tilde{\mathtt{x}}_z^{(t+1)} = \mathtt{c}_z \circledast \lambda^{\circledast}\left(\frac{1}{w}\sum_{i=0}^{w-1} \rho^{\boxast}\Big(\frac{1}{w}\sum_{j=0}^{w-1} \tilde{\mathtt{x}}_{z+i-j}^{(t)}\Big)\right). \tag{1.62}$$

Here again $\mathtt{c}_z = \mathtt{c}$, for $z = 1, \dots, L_c + w - 1$ and $\mathtt{c}_z = \Delta_\infty$ for $z = -w + 1, \dots, 0$. Furthermore, we fix the left boundary to $\tilde{\mathtt{x}}_z^{(t)} = \Delta_\infty$ for $z = -w + 1, \dots, 0$, for all $t \in \mathbb{N}$. These conditions express perfect information at the left boundary, which is what enables seeding and decoding along the chain of coupled codes. The initial condition (1.62) is $\tilde{\mathtt{x}}_z^{(0)} = \Delta_0$ for $z = 1, \dots, L_c + w - 1$.

It is convenient to work with a smoothed version of the profile $\tilde{\mathtt{x}}_z^{(t)}$, namely $\mathtt{x}_z^{(t)} = \frac{1}{w}\sum\limits_{i=0}^{w-1} \tilde{\mathtt{x}}_{z-i}^{(t)}$, where $\mathtt{x}_z^{(t)}$ denotes the check node input probability distribution.

Then, using this change of variables, (1.62) can be rewritten as

$$\mathtt{x}_z^{(t+1)} = \frac{1}{w}\sum_{i=0}^{w-1} \mathtt{c}_{z-i} \circledast \lambda^{\circledast}\left(\frac{1}{w}\sum_{j=0}^{w-1} \rho^{\boxast}\left(\mathtt{x}_{z-i+j}^{(t)}\right)\right). \tag{1.63}$$

This is the DE formulation we use throughout our work when we consider LDPC(λ, ρ) code ensembles with transmission over general BMS channels. We repeat the initial and boundary conditions for completeness. We have $\mathtt{c}_z = \mathtt{c}$, for $z = 1, \ldots, L_c + w - 1$ and $\mathtt{c}_z = \Delta_\infty$ for $z = -w+1, \ldots, 0$. We fix the left boundary to $\mathtt{x}_z^{(t)} = \Delta_\infty$ for $z = -w+1, \ldots, 0$, for all $t \in \mathbb{N}$. The initial condition for the iterations is $\mathtt{x}_z^{(0)} = \Delta_0$ (or $\mathtt{c}$) for $z = 1, \ldots, L_c + w - 1$.

1.5.3 Coupled System Potential Functionals

The DE equations and recursions in Section 1.5.2 can be expressed as the stationarity conditions of some *coupled potential functionals*. We list some of the potentials that we use in our work below.

Definition 1.28 (Coupled Potential in One Scalar Variable [71]). *The potential functional associated with the DE recursion in* (1.59) *is*

$$U_c(\underline{x}) = \sum_{z=-w+1}^{L_c} \left\{ x_z g(x_z; \epsilon) - G(x_z; \epsilon) - F\Big(\frac{1}{w}\sum_{i=0}^{w-1} g(x_{z+i}; \epsilon_z); \epsilon_z\Big) \right\}, \tag{1.64}$$

where $\underline{x} = (x_{-w+1}, \ldots, x_{L_c+w-1})$ *is the so-called* decoding profile *or simply the* profile.

For the LDPC(λ, ρ) code ensemble, with transmission over the BEC(ϵ), the potential above becomes

$$U_{\text{BEC}}(\underline{x}) = \sum_{z=-w+1}^{L_c} \left\{ \frac{1}{R'(1)}(1 - R(1 - x_z)) - x_z \rho(1 - x_z) \right.$$
$$\left. - \frac{\epsilon_z}{L'(1)} L\Big(1 - \frac{1}{w}\sum_{i=0}^{w-1} \rho(1 - x_{z+i})\Big) \right\}, \tag{1.65}$$

where $\underline{x} = (x_{-w+1}, \ldots, x_{L_c+w-1})$. The fixed point form of (1.60) is equivalent to $\frac{\mathrm{d}}{\mathrm{d}\underline{x}} U_{\text{BEC}}(\underline{x}) = 0$.

Alternatively, we can express the potential functional in (1.65) in terms of the variable node input probability $y_z^{(t)}$ as[14]

$$U_{\text{BEC}}(\underline{y}) = \sum_{z=-w+1}^{L_c} \left\{ y_z + \frac{1}{R'(1)}\Big(1 - (1 - y_z)\rho^{-1}(1 - y_z)\Big) \right.$$
$$\left. - (1 - (1 - y_z)\rho^{-1}(1 - y_z)) - \frac{\epsilon_z}{L'(1)} L\Big(\frac{1}{w}\sum_{i=0}^{w-1} y_{z+i}\Big) \right\}, \tag{1.66}$$

where $\underline{y} = (y_{-w+1}, \ldots, y_{L_c+w-1})$ is also called the *decoding profile* or *profile*. This is associated with the DE recursion in (1.61).

---

[14] We abuse notation by keeping the same label U_{BEC} but consistently use different arguments as a sufficient indicator to differentiate between the two potentials.

In Chapter 5 we make use of a potential functional that is expressed in terms of the two profiles x and y when we consider general scalar systems with a general coupling window. Using the general window function in Definition 1.27, we can express the coupled potential in x and y for general scalar systems as follows.

Definition 1.29 (Coupled Potential in Two Scalar Variables)**.** *The two-profile potential functional for scalar systems is defined as*

$$W_2(\underline{x},\underline{y}) = \sum_{z=-L_c}^{L_c} \left\{ \int_0^{y_z} \mathrm{d}u\, g^{-1}(u) + \int_0^{x_z} \mathrm{d}v\, f^{-1}(v) - \sum_{\tilde{z}=-\infty}^{+\infty} a_{z,\tilde{z}} x_{\tilde{z}} y_z \right\}, \tag{1.67}$$

where $\underline{x}$ *and* $\underline{y}$ *are the probability profiles and the coupling window coefficients* $a_{z,\tilde{z}}$ *satisfy* $a_{z,\tilde{z}} = \frac{1}{2W} w_W\big(\frac{\tilde{z}-z}{W}\big)$.

The solutions of spatially coupled DE equations (1.56) are given by the stationary points of the potential functional W_2 of x and y. This can be checked by setting the functional derivatives of the potential, with respect to each of x and y, to zero.

Definition 1.30 (Coupled Potential in One Variable [63])**.** *The potential functional associated with the DE recursion in* (1.63) *for the case of transmission over general BMS channels is defined as*

$$\begin{aligned} W(\underline{\mathtt{x}}) = \sum_{z=-w+1}^{L_c} \Bigg\{ & \frac{1}{R'(1)} H(R^{\boxast}(\mathtt{x}_z)) + H(\rho^{\boxast}(\mathtt{x}_z)) - H(\mathtt{x}_z \boxast \rho^{\boxast}(\mathtt{x}_z)) \\ & - \frac{1}{L'(1)} H\Big(\mathtt{c}_z \circledast L^{\circledast}\Big(\frac{1}{w} \sum_{i=0}^{w-1} \rho^{\boxast}(\mathtt{x}_{z+i})\Big)\Big)\Bigg\}. \end{aligned} \tag{1.68}$$

where $\underline{\mathtt{x}} = (\mathtt{x}_{-w+1}, \dots, \mathtt{x}_{L_c+w-1})$.

The fixed point form of (1.63) is equivalent to $\lim_{\gamma\to 0}(W(\underline{\mathtt{x}}+\gamma\underline{\eta}) - W(\underline{\mathtt{x}}))/\gamma = 0$ for $\underline{\eta} = (\eta_{-w+1}, \dots, \eta_{L_c+w-1})$ where η_i are differences of probability measures.

1.5.4 Thresholds

In Shannon's 1948 revolutionary article, he defined the theoretical limit on the channel noise up to which reliable transmission of information is possible. We call this limit the *Shannon threshold.* So far, we have defined the BP threshold $\mathtt{h}_{\mathrm{BP}}$, the MAP threshold $\mathtt{h}_{\mathrm{MAP}}$, and the potential threshold $\mathtt{h}_{\mathrm{pot}}$ in Definitions 1.17, 1.18, and 1.25, respectively, for (uncoupled) LDPC(λ,ρ) code ensembles when transmission takes place over BMS channels. In this section, we further define the area threshold and the BP and MAP thresholds that correspond to spatially coupled such systems. We then describe the relationship between all these values, for coupled code ensembles, as well as their underlying ensembles. As usual, we consider the framework in which the blocklength n is very large.

The MAP threshold is the limit on the channel noise up to which the (optimal) MAP decoder can decode perfectly. This threshold is lower than the Shannon threshold, but approaches it as the degrees of the LDPC code are increased. The MAP decoder, however, assumes that we have infinite resources and computational power. It is therefore impractical due to physical and computational limitations.

The BP decoder, however, is of low complexity and is quite suitable for sparse graphical models such as LDPC codes. Unsurprisingly, the BP threshold is lower than the MAP threshold when the code is not spatially coupled. Moreover, for these (uncoupled) codes, the BP threshold decreases as the node degrees are increased, moving farther away from the Shannon threshold.

As we have already mentioned, spatial coupling combines the benefit of low-complexity decoding and high thresholds. Spatially coupled codes can decode perfectly up to the MAP threshold using the low-complexity BP decoder. More formally, we define the BP and MAP thresholds of spatially coupled LDPC(λ, ρ) code ensembles when transmission takes place over BMS channels below.

Definition 1.31 (Coupled BP threshold (BMS)). *The BP threshold* $\mathtt{h}^c_{BP}$ *of a spatially coupled LDPC code with transmission on BMS channels is defined by*

$$\mathtt{h}^c_{BP} \triangleq \sup\{\mathtt{h} \in [0,1] \mid \mathtt{x}_z^{(\infty)} = \Delta_\infty \ \forall\ z \in \{-w+1, \dots, L_c\}\}, \tag{1.69}$$

where $\mathtt{x}_z^{(\infty)}$ *is the limit value of the distribution at position* z *of the coupling chain, governed by the recursion in* (1.63).

Definition 1.32 (Coupled MAP threshold (BMS)). *The MAP threshold* $\mathtt{h}^c_{MAP}$ *of a spatially coupled LDPC code with transmission on BMS channels is defined by*

$$\mathtt{h}^c_{MAP} \triangleq \inf\{\mathtt{h} \in [0,1] \mid \liminf_{n \to +\infty} \frac{1}{n} \mathbb{E}[H(\underline{\mathtt{X}} | \underline{\mathtt{Y}}(\mathtt{c}(\mathtt{h})))] > 0\}, \tag{1.70}$$

where n *is the blocklength of the entire spatially coupled code; that is,* $n = NL_c$ *where* L_c *is the number of copies of the underlying code (or the length of the coupling chain) and* N *is the number of variable nodes per copy, and the limit* $n \to +\infty$ *is taken by first taking* $N \to +\infty$ *and then taking* $L_c \to +\infty$*. The vector of random variables* $\underline{\mathtt{X}}$ *(resp.* $\underline{\mathtt{Y}}$*) represents the channel input (resp. output),* $H(\underline{\mathtt{X}} | \underline{\mathtt{Y}}(\mathtt{c}(\mathtt{h})))$ *is the conditional Shannon entropy of the input given by the channel observations, and the expectation* $\mathbb{E}[\cdot]$ *is over the coupled LDPC code ensemble.*

Threshold saturation for irregular LDPC(λ, ρ) code ensembles and transmission over BMS channels was proven in [63] using *potential functionals*. More concretely, it was shown that the coupled system BP threshold is equal to the potential threshold of the underlying system; that is,

$$\mathtt{h}^c_{\mathrm{BP}} = \mathtt{h}_{\mathrm{pot}}.$$

It was also shown in [63] that if the graph does not contain degree-2 variable nodes, the MAP threshold of the underlying system is lower than its potential threshold and thus satisfies

$$\mathtt{h}_{\mathrm{MAP}} \le \mathtt{h}_{\mathrm{pot}}.$$

Furthermore, it was shown in [82] that for LDPC codes with regular check nodes of even degree, the MAP thresholds of the spatially coupled code and its underlying system are equal. That is,

$$\mathtt{h}_{\mathrm{MAP}} = \mathtt{h}^c_{\mathrm{MAP}}.$$

Combining this with the optimality of the MAP decoder ($\mathtt{h}^c_{\mathrm{BP}} \le \mathtt{h}^c_{\mathrm{MAP}}$), we obtain

$$\mathtt{h}_{\mathrm{MAP}} \le \mathtt{h}_{\mathrm{pot}} = \mathtt{h}^c_{\mathrm{BP}} \le \mathtt{h}^c_{\mathrm{MAP}} = \mathtt{h}_{\mathrm{MAP}}, \tag{1.71}$$

for these codes.

In [43], the proof technique for threshold saturation did not use potential functions as was done in [63]. Instead, the authors used GEXIT functions and yet another threshold called the *area threshold* $\mathtt{h}_{\mathrm{A}}$ to prove that for (ℓ, r)-regular LDPC ensembles, the BP threshold of a coupled code saturates to the MAP threshold of the underlying system universally on the class of BMS channels. In particular, it is shown in [43] that

$$\mathtt{h}_{\mathrm{A}} = \mathtt{h}^c_{\mathrm{BC}},$$

where the area threshold for such systems is defined as

Definition 1.33 (Area Threshold [63])**.** *The area threshold* $\mathtt{h}^c_{\hat{A}}$ *of an* (ℓ, r)*-regular LDPC code with transmission on BMS channels is defined by*

$$\mathtt{h}_{\hat{A}} \triangleq \sup\{\mathtt{h} \in [0,1] \mid \hat{A}(\mathtt{x}^{(\infty)}, \ell, r) \le 0\}, \tag{1.72}$$

and $\hat{A}$ *is the area under the corresponding GEXIT curve defined by*

$$\hat{A}(\mathtt{x}, \ell, r) = H(\mathtt{x}) + (\ell - 1 - \frac{\ell}{r})H(\mathtt{x}^{\boxast r}) - (\ell - 1)H(\mathtt{x}^{\boxast (r-1)}).$$

We note that this area is not equal to those defined in Definitions 1.20 and 1.21. It can be seen that at the fixed point $\mathtt{x}^{(\infty)}$, the area defined above is exactly the negative of the single potential in Definition 1.23. This immediately implies that

$$\mathtt{h}_{\mathrm{pot}} \le \mathtt{h}_{\mathrm{A}}.$$

Combining this with the results in (1.71), we conclude that

$$\mathtt{h}_{\mathrm{A}} = \mathtt{h}_{\mathrm{pot}},$$

and therefore

$$\mathtt{h}_{\mathrm{MAP}} = \mathtt{h}_{\mathrm{pot}} = \mathtt{h}_{\mathrm{A}},$$

for (ℓ, r)-regular LDPC codes with even check node degree and without degree-2 variable nodes.[15]

1.6 The Continuum Limit

The continuum limit is the setting obtained by *first* taking the length of the coupling chain L_c to be very large ($L_c \to +\infty$) and *then* taking the window size w to be very large ($w \to +\infty$). As the name suggests, this makes the quantities we work

---

[15] In fact, it has been proven rigorously that $\mathtt{h}_{\mathrm{MAP}} = \mathtt{h}_{\mathrm{A}}$ for (ℓ, r)-regular LDPC codes with even check node degree. For more general codes and transmission over BMS channels, this is still a conjecture called the Maxwell conjecture [91].

with continuous rather than discrete (their natural setting) when the appropriate normalizations are made. We will use this setting throughout the thesis because it is well-adapted to the techniques we need, such as taking functional derivatives in Chapters 2 and 3, and using the tool of displacement convexity in Chapters 4 and 5. The continuum limit has already been considered for the special case of the BEC in [69, 70, 72, 73].

To avoid overloading the notation, and because we use only the continuum limit in our derivations (and not the discrete formulations), we keep the same symbols for the profiles, the spatial position, and the channel distribution when we move from the discrete to the continuous space. We thus denote by $x(\cdot,\cdot)$ (resp. $\mathtt{x}(\cdot,\cdot)$) and $y(\cdot,\cdot)$ (resp. $\mathtt{y}(\cdot,\cdot)$) the *continuous* profiles of distributions defined by $x(\frac{z}{w},t) \equiv x_z^{(t)}$ (resp. $\mathtt{x}(\frac{z}{w},t) \equiv \mathtt{x}_z^{(t)}$) and $y(\frac{z}{w},t) \equiv y_z^{(t)}$ (resp. $\mathtt{y}(\frac{z}{w},t) \equiv \mathtt{y}_z^{(t)}$), and then replace $\frac{z}{w} \to z$ so that the new variable z is the continuous variable on the spatial axis, $z \in \mathbb{R}$.

For general scalar systems with control parameter ϵ, the DE recursion (1.59) becomes

$$x(z,t+1) = \int_0^1 \mathrm{d}u\, f\Big(\int_0^1 \mathrm{d}s\, g\big(x(z-u+s,t);\epsilon\big);\epsilon\Big). \tag{1.73}$$

The natural boundary conditions are $x(z,t) \to 0$, $z \to -\infty$ and $x(z,t) \to x_{\mathrm{BP}}$, $z \to +\infty$. The left limit results from "seeding" at the left boundary of the original, discrete system; and the right limit results from the empirically observed "height" obtained by the DE recursion of the original system after a few "transient" iterations of the recursion. A general analysis of this recursion, with applications that are not limited to coding, is considered in Chapter 3.

The fixed points of the continuous recursion above are the stationary points of a potential functional. The potential is an integral over the spatial direction $z \in \mathbb{R}$ and in general we must subtract a "reference energy" in order to get a convergent result. Essentially any static reference profile, here called $x_0(z)$ that satisfies the boundary conditions $x_0(z) \to x_{\mathrm{good}}$, $z \to -\infty$ and $x_0(z) \to x_{\mathrm{bad}}$, $z \to +\infty$ will do the job. For concreteness, we can take a Heaviside-like profile $x_0(z) = x_{\mathrm{good}}$, $z < 0$, $x_0(z) = x_{\mathrm{bad}}$, $z \geq 0$ (note that $x_0(z)$ does not depend on time). The potential functional is then defined as

$$\Delta\mathcal{U}_c(x) = \int_{\mathbb{R}} \mathrm{d}z \Big\{ P_c(z,x) - P_c(z,x_0) \Big\}, \tag{1.74}$$

where $P_c(z,x)$ is a z-dependent functional of x equal to

$$P_c(z,x) = x(z,t)g(x(z,t);\epsilon) - G(x(z,t);\epsilon) - F\Big(\int_0^1 \mathrm{d}u\, g(x(z+u,t);\epsilon);\epsilon\Big). \tag{1.75}$$

In the case of transmission over the BEC, the DE recursion (1.60) becomes

$$x(z,t+1) = \epsilon \int_0^1 \mathrm{d}u\, \lambda\Big(1 - \int_0^1 \mathrm{d}s\, \rho\big(1 - x(z-u+s,t)\big)\Big). \tag{1.76}$$

The initial condition at $t = 0$ is given by a profile $x(z,0)$ that interpolates between the two limiting values of the boundary condition, namely $x(z,0) \to 0$ when $z \to -\infty$ and $x(z,0) \to x_{\mathrm{BP}}$ when $z \to +\infty$, where x_{BP} is the non-trivial fixed point value of the underlying system. (The reasons for this initialization are made clear in Chapter 2.)

The fixed points of the continuous recursion above are the stationary points of a potential functional. As before, we must subtract a “reference energy” from the potential so that it is well-defined. We consider for instance the static reference profile $x_0(z)$ defined by $x_0(z) = 0$, $z < 0$, $x_0(z) = x_{\mathrm{BP}}$, $z \geq 0$. The potential functional is then defined as

$$\Delta\mathcal{W}_{\mathrm{BEC}}(x) = \int_{\mathbb{R}} \mathrm{d}z \Big\{ P_x(z,x) - P_x(z,x_0) \Big\}, \tag{1.77}$$

where $P_x(z,x)$ is a z-dependent functional of x equal to

$$\begin{aligned} P_x(z,x) = \frac{1}{R'(1)}\Big(1 - R(1-x(z,t))\Big) - x(z,t)\rho(1-x(z,t)) \\ - \frac{\epsilon}{L'(1)} L\Big(1 - \int_0^1 \mathrm{d}u\, \rho(1-x(z+u,t))\Big). \end{aligned} \tag{1.78}$$

Similarly, the DE recursion (1.61) in the continuum limit becomes

$$y(z,t+1) = 1 - \rho\Big(1 - \epsilon \int_0^1 \mathrm{d}u\, \lambda\Big(\int_0^1 \mathrm{d}s\, y(z-u+s,t)\Big)\Big). \tag{1.79}$$

Similarly, the initial condition at $t = 0$ satisfies $y(z,0) \to 0$ when $z \to -\infty$ and $y(z,0) \to y_{\mathrm{BP}}$ when $z \to +\infty$, where y_{BP} is the non-trivial fixed point value of the underlying system. We consider a profile y_0 defined by $y_0(z) = 0$, $z < 0$, $y_0(z) = y_{\mathrm{BP}}$, $z \geq 0$. Then, the associated potential functional can be written as

$$\Delta\mathcal{W}_{\mathrm{BEC}}(y) = \int_{\mathbb{R}} \mathrm{d}z \Big\{ P_y(z,y) - P_y(z,y_0) \Big\}, \tag{1.80}$$

where $P_y(z,y)$ is a z-dependent functional of y equal to

$$\begin{aligned} P_y(z,y) =& y(z,t) + \frac{1}{R'(1)}\Big(1 - (1-y(z,t))\rho^{-1}(1-y(z,t))\Big) \\ & - \Big(1 - (1-y(z,t))\rho^{-1}(1-y(z,t))\Big) - \frac{\epsilon}{L'(1)} L\Big(\int_0^1 \mathrm{d}u\, y(z+u,t)\Big). \end{aligned} \tag{1.81}$$

We next consider the case of general scalar systems and the general coupling window $w(\cdot)$ in Definition 1.27. Then the windowed average becomes a convolution of the profiles x and y with the window function w. We express such convolutions $x \circledast w(z) = \int_{\mathbb{R}} \mathrm{d}\tau\, x(\tau)w(z-\tau)$ as $x^w(z)$ as shorthand. Similarly, we write $y \circledast w$ as y^w. The continuous version of the DE equations (1.56) then becomes

$$\begin{cases} y(z,t) & = g(x^w(z,t)), \\ x(z,t+1) & = f(y^w(z,t)). \end{cases} \tag{1.82}$$

The initial conditions at $t = 0$ are given by profiles that interpolate between the two fixed point values. More precisely, if x_+ (resp. y_+) and x_- (resp. y_-) are the desirable and undesirable fixed points of the profiles x (resp. y) of the underlying system, then the initial conditions are set to $x(z,0) \to x_+$ (resp. $y(z,0) \to y_+$) for $z \to -\infty$ and $x(z,0) \to x_-$ (resp. $y(z,0) \to y_-$) for $z \to +\infty$.

The potential functional corresponding to the system of DE equations above is the continuum version of that in (1.67). We consider two profiles x_0 and y_0 that are Heaviside-like and satisfy $x_0(z) = x_-$, $z < 0$, $x_0(z) = x_+$, $z \geq 0$, $y_0(z) = y_-$, $z < 0$, $y_0(z) = y_+$, $z \geq 0$. Then, the potential is expressed as

$$\Delta \mathcal{W}_2(x, y) = \int_{\mathbb{R}} \mathrm{d}z \left\{ P_2(z, x, y) - P_2(z, x_0, y_0) \right\}, \tag{1.83}$$

where $P_2(z, x, y)$ is a z-dependent functional of x and y defined as

$$P_2(z, x, y) = \int_0^{y(z)} \mathrm{d}u \, g^{-1}(u) + \int_0^{x(z)} \mathrm{d}v \, f^{-1}(v) - x^w(z) y(z).$$

The solutions of the spatially coupled DE equations (1.82) are given by the stationary points of the potential functional $\mathcal{W}_2$ of x and y. This can be checked by setting the functional derivatives of the potential with respect to each of x and y to zero.

For the case of transmission over the BMS channel, we consider a *continuous channel distribution* and we denote by $\mathtt{c}(z)$ the channel distribution at the continuous spatial position $z \in \mathbb{R}$. The DE equation (1.63) then takes the form

$$\mathtt{x}(z, t+1) = \int_0^1 \mathrm{d}u \, \mathtt{c}(z-u) \circledast \lambda^{\circledast}\Big(\int_0^1 \mathrm{d}s \, \rho^{\boxast}\big(\mathtt{x}(z-u+s, t)\big)\Big). \tag{1.84}$$

The initial condition at $t = 0$ is given by a profile $\mathtt{x}(z, 0)$ that interpolates between the two limiting values of the boundary condition, namely $\mathtt{x}(z, 0) \to \Delta_\infty$ when $z \to -\infty$ and $\mathtt{x}(z, 0) \to \mathtt{x}_{\mathrm{BP}}$ when $z \to +\infty$.

We call $\Delta \mathcal{W}(\mathtt{x})$ the potential functional of the coupled system in the continuum limit obtained from (2.8). As before, we subtract a "reference energy" so that the potential is well-defined. We denote by $\mathtt{x}_0(z)$ the static reference profile defined by $\mathtt{x}_0(z) = \Delta_\infty$, $z < 0$, $\mathtt{x}_0(z) = x_{\mathrm{BP}}$, $z \geq 0$. The potential functional is thus defined as

$$\Delta \mathcal{W}(\mathtt{x}) = \int_{\mathbb{R}} \mathrm{d}z \left\{ P(z, \mathtt{x}) - P(z, \mathtt{x}_0) \right\}, \tag{1.85}$$

where $P(z, \mathtt{x})$ is a z-dependent functional of $\mathtt{x}$ equal to

$$\begin{aligned} P(z, \mathtt{x}) = \frac{1}{R'(1)} H(R^{\boxast}(\mathtt{x}(z,t))) + H(\rho^{\boxast}(\mathtt{x}(z,t))) - H(\mathtt{x}(z,t) \boxast \rho^{\boxast}(\mathtt{x}(z,t))) \\ - \frac{1}{L'(1)} H\Big(\mathtt{c}(z) \circledast L^{\circledast}\Big(\int_0^1 \mathrm{d}s \, \rho^{\boxast}(\mathtt{x}(z+s, t))\Big)\Big). \end{aligned} \tag{1.86}$$

In our work, we find it convenient to split the continuous potential functional into two components: the "single system potential functional" and the "interaction functional". The latter, that we also call the interaction potential for brevity, expresses the contribution of the "cross-connections" of replicas due to spatial coupling. In other words, this term disappears if evaluated at a constant $\mathtt{x}(z) = \mathtt{x}$. The remaining term is called the single potential for brevity and expresses the contribution of the entire system as if spatial coupling were not made and every replica had edges only between its nodes. In fact, we can write the single potential functional as an integral over the spatial component z of some integrand. This integrand is simply the potential of the underlying uncoupled code ensemble. This is easily seen by recognizing that the usual DE equation for the variable input erasure probability is recovered by setting the derivative of this single potential to zero. We show the explicit expressions of the single and interaction potentials when we use them in Chapters 2-5, because we write them in different ways (for convenience) in different chapters.

1.7 Displacement Convexity

Displacement convexity can be very useful in functional analysis. It goes back to McCann [92] and plays an important role in the theory of optimal transport [93]. It was used in [94] and [95] to study a functional governing a spatially coupled Curie-Weiss model, that bears close similarities with the coding theory model studied here (see [59]). In this section, we give a quick introduction to the tool of displacement convexity.

Recall first that the usual notion of convexity of a generic functional $\mathcal{F}(y)$ on a generic space $\mathcal{X}$ means that for all $x_0, x_1 \in \mathcal{X}$ and $\lambda \in [0,1]$,

$$\mathcal{F}((1-\lambda)x_0 + \lambda x_1) \leq (1-\lambda)\mathcal{F}(x_0) + \lambda\mathcal{F}(x_1).$$

The correct setting for defining displacement convexity is the space of probability measures. For measures over the real line, we can conveniently define the displacement interpolation in terms of the cumulative distribution functions (cdf's) associated with the measures. This is the simplest setting and the one that we adopt here.

In the following, we assume that the functions x_0 and x_1 belong to the space of right-continuous cdf's of some underlying measures $\mathrm{d}x_0$ and $\mathrm{d}x_1$, respectively, over the real line. That is, they are non-decreasing with left and right limits of 0 and 1, respectively. The definitions we give below can be adjusted for scaled cdf's or for the equivalent analysis with probability measures.

The inverse $x^{-1}(z)$ of a cdf x is defined almost everywhere and has unique and well-defined left and right limits, $x^{-1}(z_-)$ and $x^{-1}(z_+)$, respectively. In our work we use the definition of the right-continuous inverse that is defined for all $x_* \in (0,1)$, namely

$$z(x_*) \triangleq \inf\{\, z \,|\, x(z) > x_* \}.$$

Similarly, a cdf x can be obtained from its inverse z_* using the relation

$$x(z_*) \triangleq \inf\{\, x \,|\, z(x) > z_* \}.$$

For any two right-continuous cdf's x_0, x_1, we consider z_0, z_1 their respective inverses under the maps defined above. Then for any $\lambda \in [0,1]$, the interpolated profile x_λ is defined as follows,

$$z_{x,\lambda} = (1-\lambda)z_0(x) + \lambda z_1(x), \tag{1.87}$$

$$x_\lambda(z) = \inf\{\, x \,|\, z_{x,\lambda} > z \}. \tag{1.88}$$

In words, the difference in interpolation under the alternative structure is that the linear interpolation is applied on the inverse of the functions of interest, and the effect of such an interpolation is then mapped back into the space of functions.

The displacement convexity of a functional $\mathcal{F}(x)$ on the space of right-continuous cdf's simply means that the following inequality holds

$$\mathcal{F}(x_\lambda) \leq (1-\lambda)\mathcal{F}(x_0) + \lambda\mathcal{F}(x_1), \tag{1.89}$$

for any two right-continuous cdf's x_0, x_1 and all $\lambda \in [0,1]$. Strict displacement convexity means that this inequality is strict as long as x_0 and x_1 are distinct and $\lambda \in (0,1)$.

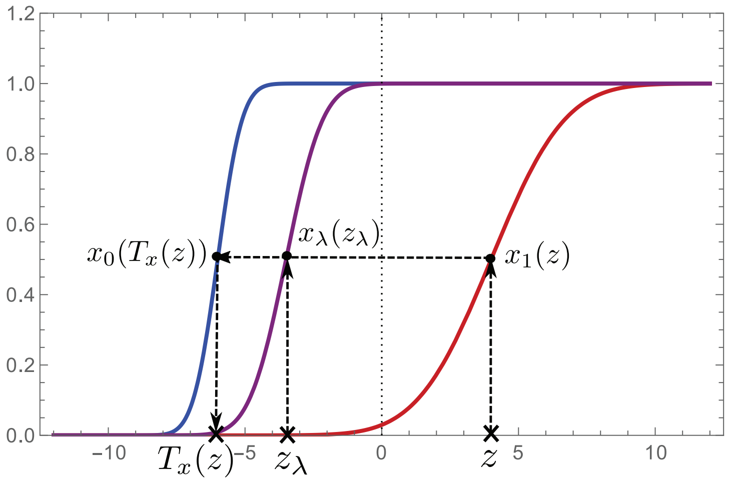

Figure 1.14: Monotonic profiles x_0 and x_1, the map $T_x(z)$, $z_{x,\lambda} = (1-\lambda)z + \lambda T_x(z)$, and the interpolant x_λ. Here $\lambda = 1/4$.

In our analysis of displacement convexity in Chapters 4 and 5, we also make use of the notion of the *pushforward map*, described below.

Consider two cdf's x_0 and x_1, and assume x_0 is continuous. We can define the map $T_x : \mathbb{R} \to \mathbb{R}$ as

$$T_x(z) = x_1^{-1}(x_0(z)). \tag{1.90}$$

The map T_x can be seen as a pushforward map for measures from $\mathrm{d}x_0$ to $\mathrm{d}x_1$. This is expressed as $\mathrm{d}x_1 = T_x\#\mathrm{d}x_0$ that means that

$$\int \mathrm{d}x_1(z)\, h(z) = \int \mathrm{d}x_0(z)\, h(T_x(z)),$$

for any function h such that the integral is well-defined. Then, denoting by id the identity map, the interpolant x_λ is the cdf of the measure $\mathrm{d}x_\lambda$ defined by

$$\mathrm{d}x_\lambda = ((1-\lambda)id + \lambda T_x)\#\mathrm{d}x_0.$$

We have

$$\int \mathrm{d}x_\lambda(z)\, h(z) = \int \mathrm{d}x_0(z)\, h((1-\lambda)z + \lambda T_x(z)),$$

whenever the integral is defined. In particular, if h is convex (in the usual sense) then this shows convexity in λ of the integral due to the following,

$$\begin{aligned} &\int \mathrm{d}x_0(z)\, h((1-\lambda)(z) + \lambda T_x(z)) \\ &\qquad \le (1-\lambda)\int \mathrm{d}x_0(z)\, h(z) + \lambda \int \mathrm{d}x_0(z)\, h(T_x(z)) \\ &\qquad = (1-\lambda)\int \mathrm{d}x_0(z)\, h(z) + \lambda \int \mathrm{d}x_1(z)\, h(z). \end{aligned}$$

The graphical construction of the interpolant x_λ is illustrated in Figure 1.14. Graphically, T_x finds the position $\tilde{z} = T_x(z)$ on the z-axis so that $x_1(\tilde{z}) = x_0(z)$ for some given z. Consider the linear interpolation between points on $\mathbb{R}$, $z_{x,\lambda} =$

$(1-\lambda)z+\lambda T_x(z)$. The displacement interpolant x_λ is defined so that the following equality holds for all $\lambda \in [0,1]$

$$x_\lambda(z_{x,\lambda}) = x_0(z). \tag{1.91}$$

This is equivalent to the relation in (1.88) when x_0 is continuous.

In the case where x_0 is discontinuous, we have to be more careful in the definition. At points of discontinuity of x_0 the map $T_x(z)$ should not be single-valued. As we work in one dimension, this issue is easily circumvented by using the definition in (1.88). For such cases, instead of referring to the definition of the interpolant $x_\lambda(z_{x,\lambda})$ in (1.91), we use the following system of equations

$$x_\lambda^{-1}(u) = (1-\lambda)x_0^{-1}(u) + \lambda x_1^{-1}(u), \tag{1.92}$$

$$x_\lambda(z) = \inf\{u \mid x_\lambda^{-1}(u) > z\}, \tag{1.93}$$

which gives a right-continuous interpolant. Correspondingly, if x is an interpolating non-decreasing profile then, under appropriate regularity of h, we can write

$$\int_{\mathbb{R}} \mathrm{d}x(z)\, h(z) = \int_0^1 \mathrm{d}u\, h(x^{-1}(u)),$$

and we have

$$\int \mathrm{d}x_\lambda(z)\, h(z) = \int_0^1 \mathrm{d}u\, h((1-\lambda)x_0^{-1}(u) + \lambda x_1^{-1}(u)).$$

With this in mind we will continue to use the notation $T_x(z)$ when the above interpretation should be understood.

1.8 Main Contributions

In this thesis, we analyze the solutions of message-passing algorithms on spatially coupled systems by using variational techniques, in the context of coding theory (Chapters 2 and 4) and general scalar systems (Chapters 3 and 5). We consider the systems in the framework of large blocklengths, which enables us to express the performance of the message-passing algorithms in terms of density evolution (DE), or DE-like, equations. We further consider the system in the continuum limit by taking the coupling chain length L_c to infinity (first) and the coupling window size w to infinity (second); this enables us to write the DE equations in the continuum limit, which makes the derivations more tractable. Throughout the thesis, we use the variational equivalent of the DE equations, the (continuous) "potential functional", for which the stationary points are the fixed points of the DE equations. In order to analyze the performance of message-passing algorithms on coupled systems, we can track the evolution of the DE equations or, equivalently, apply gradient descent on their associated potential functional. In our work, we consider only coupled systems for which the uncoupled DE equations have a unique "non-trivial" stable fixed point (or, equivalently, for which the uncoupled potential function has a unique non-trivial minimum).

In Chapter 2, we consider spatially coupled irregular LDPC(λ,ρ) code ensembles, where transmission takes place over general BMS channels. We consider the system in the *dynamic phase*, defined as the decoding phase in which applying iterations of the BP algorithm results in decoding more bits (or, more generally, non-trivially updating the messages exchanged in the system). Due to the "threshold saturation" property of spatially coupled codes, we know that the dynamic phase occurs for all parameters of the channel below the MAP threshold because the BP algorithm can successfully decode the system for all such channel parameters.

We express the originally *discrete* DE recursion in terms of the variable node output distributions; we denote by $\mathtt{x}_z^{(t)}$ the distribution at position z along the spatial axis of coupling and at iteration $t \in \mathbb{N}$. By plotting the *vector of distributions* $\underline{\mathtt{x}}$ (where the z^{th} component contains the distribution at the z^{th} position on the coupling axis) at different iterations of the BP algorithm, we observe that the vector (or "decoding profile") exhibits a solitonic behavior. More specifically, it appears to have a fixed shape and to move (away from the boundaries) at a constant speed. This phenomenon is observed when the channel parameter is between the BP and MAP thresholds (which constitutes part of the dynamic phase); it is called the *wave propagation phenomenon*. In the context of coding on the BEC, for example, this means that the vector of erasure probabilities along the spatial axis of coupling appears to exhibit this solitonic behavior. Our main contribution in this chapter is to find a formula for the velocity of the wave for such coupled codes, assuming that the profile does indeed exhibit the solitonic behavior. To do this, we first consider the continuum limit that enables us to express the DE recursion and the potential functional in the continuous space. The variable node output-distribution at position z and iteration t is then denoted by $\mathtt{x}(z,t)$. The assumption about the solitonic behavior can thus be expressed as $\mathtt{x}(z,t) = \mathtt{X}(z - vt)$.[16] We make algebraic manipulations to write the potential functional in terms of the decoding profile, and we extract a formula for the velocity from this relation. As (possibly infinite-dimensional) distributions are involved in the calculations, we reduce the complexity of the formula by applying the Gaussian approximation that approximates distributions with symmetric Gaussian distributions. Furthermore, we use our results for the velocity to estimate parameters involved in the scaling law for finite-length spatially coupled codes with transmission over the BEC [96]. We perform numerical experiments to verify our results.

In Chapter 3, we consider spatially coupled general scalar systems that are governed by a DE-like scalar recursion, *during the dynamic phase*. In this more general context, the dynamic phase is defined as that in which the governing recursion non-trivially updates the system; this occurs when the *system parameter* is lower than some threshold that we call the *potential threshold*.[17] As in Chapter 2, we conjecture that the profile along the coupling chain exhibits a solitonic behavior upon iterations of the recursion, when the system parameter is between the algorithmic threshold[18] and the potential threshold. We find an analytical formula for the velocity of this soliton. The derivation of the formula resembles that of Chapter 2 and involves expressing the (continuous) potential functional in terms of the (continuous) profile.

---

[16] We slightly abuse notation by keeping the same labels in the discrete and continuous settings.

[17] The reason behind the naming "potential threshold" is related to properties of the potential function, described in Section 1.4.5.

[18] The algorithmic threshold is analogous to the BP threshold in the context of coding; here, it depends on the application-specific message-passing algorithm.

To verify our results, we provide numerical simulation results for applications such as LDPC codes over the BEC, compressive sensing, and generalized LDPC codes over the BEC and the binary symmetric channel (BSC).

In Chapters 2 and 3, we do not prove that the shape of the propagating profile is fixed, nor do we characterize it. However, we observe using numerical simulations that the shape is an increasing "kink", and its limits are the trivial fixed point at $z \to -\infty$ and the non-trivial fixed point at $z \to +\infty$.

In Chapter 4, we consider the spatially coupled (ℓ, r)-regular LDPC Gallager ensemble with transmission over the BEC *at/during the static phase*. In the context of spatially coupled codes, the static phase occurs at the MAP threshold, the channel parameter at which the BP algorithm gets blocked and is no longer able to completely decode the system; thus, the decoding profile does not change upon iterations of the BP algorithm. In other words, the velocity of the wave vanishes. In this chapter, we characterize this fixed point profile and prove that it is unique (up to translation on the spatial axis of coupling[19]). To do this, we introduce a new tool for the analysis of spatially coupled systems, namely the concept of *displacement convexity*, borrowed from optimal transport. This tool considers convexity with respect to an alternative structure in the space of probability measures. In short, a functional $\mathcal{F}$ is "traditionally convex" if $\mathcal{F}(x_\lambda(\cdot)) \le (1-\lambda)\mathcal{F}(x_0(\cdot)) + \lambda\mathcal{F}(x_1(\cdot))$, for all $\lambda \in [0,1]$, when the interpolant $x_\lambda(\cdot)$ is a linear combination of the two functions $x_0(\cdot)$ and $x_1(\cdot)$, so that $x_\lambda(z) = (1-\lambda)x_0(z) + \lambda x_1(z)$ for all $z \in \mathbb{R}$. Whereas, $\mathcal{F}$ is displacement convex if the same inequality holds, but $x_\lambda(\cdot)$ is defined (roughly) as the distribution obtained by first taking the linear interpolation of the inverses of the two distributions $x_0(\cdot)$ and $x_1(\cdot)$ and then by taking the inverse of this interpolation back into the space of probability distributions. This is shown in Figure 1.14 and described in more detail in Section 1.7.

Although the coupled potential functional is *not* convex in the traditional sense, we prove, by using displacement convexity, that it is displacement convex in the space of probability distributions. To do this, we observe that it is possible to view the decoding profile as a cumulative distribution function. Hence,we use truncation and rearrangement inequalities to restrict the space of minimizers of the potential functional to profiles that are increasing from a finite limit at $-\infty$ to a finite limit at $+\infty$. We then show that the potential functional is strictly displacement convex (in the space of probability measures) and characterize its minimum. In the language of coding, we prove that, at the MAP threshold, the decoding profile is static and the DE equations converge to a fixed point solution that is unique, up to translation.

In Chapter 5, we generalize the analysis done in the preceding chapter so that it encompasses general spatially coupled scalar systems (not restricted to coding) that are governed by a system of scalar equations. We again consider these systems *at/during the static phase*. For such general systems, the static phase is defined as the system parameter at which the governing (DE-like) equations no longer non-trivially update the system or, equivalently, the potential threshold. The analysis

---

[19] As opposed to the dynamic setting where translation would refer to a change made on the profile due to the message-passing algorithm, here for the static case it simply means that the origin of the profile is arbitrary. Once the origin is fixed, however, the shape is strictly unique.

in this chapter is more general than that in Chapter 4 in the sense that it is more far-reaching and that it establishes the strict displacement convexity of the coupled potential functionals under milder assumptions on the underlying system. In fact, we find that the convexity result in this chapter applies to such systems that satisfy the strictly positive gap condition [69,70] and some mild conditions on the coupling-window function. Therefore, for general spatially coupled scalar systems, we also characterize the unique fixed-point solution of the DE equations at the static phase.

In Chapters 4 and 5, we consider spatially coupled systems during the static phase of the message-passing algorithm, in the case when the underlying system has a unique non-trivial stable fixed point. During this phase, the velocity studied in the previous chapters *is zero* by definition. In this case, we *prove* that the shape of the solution to the DE equations is unique. That is, the assumptions made in the first two chapters for the dynamic phase (fixed shape and constant velocity) are established for the static phase.

Finally, *in Chapter 6*, we offer a summary of our main contributions, some concluding remarks, and further directions that can be pursued based on our work.

Decoding Velocity for Irregular Codes on the BMS

2

2.1 Introduction

We consider spatially coupled LDPC(λ, ρ) code ensembles, where transmission takes place over general binary-input memoryless symmetric-output (BMS) channels.[1] To obtain such a code from its underlying (uncoupled) ensemble, we spread copies of the latter on the spatial axis of coupling and connect them locally within a certain window size. Spatially coupled codes have been proved to be universally capacity-achieving due to the threshold saturation phenomenon [43, 63]. That is, it has been proved that the belief propagation (BP) threshold of the coupled code-ensemble *saturates* towards the maximum a-posteriori (MAP) threshold of the underlying uncoupled ensemble. This phenomenon can be attributed to the "seeding" that is done at the boundary of a coupled code: once we "seed" perfect information at the copies at the boundaries of the coupling chain and run the BP algorithm, the extra information at the boundaries will propagate into the chain, due to the local connections between adjacent copies of the underlying codes. In this chapter, we find an analytical formula for the "speed" at which this extra information is propagated into the coupling chain, assuming this speed is a constant. We express this more formally below.

We construct a spatially coupled LDPC(λ, ρ) code by taking $L_c + w$ copies of the underlying code and coupling every w adjacent underlying systems by means of a uniform window function.[2] In order to track the performance of the BP algorithm on the coupled code, we express the evolution of the variable node output distri-

---

[1] The content of this chapter is based on the following previous work [76, 97].
[76] R. El-Khatib, N. Macris, *"The velocity of the decoding wave for spatially coupled codes on BMS channels,"* International Symposium of Information Theory (ISIT) 2016, Barcelona, Spain, IEEE pp. 2119–2123.
[97] R. El-Khatib, N. Macris, *"The velocity of the propagating wave for spatially coupled systems with applications to LDPC codes,"* in preparation (2016) to be submitted to the IEEE Transactions on Information Theory.

[2] It is straightforward to generalize the analysis to include more general coupling-windows; we use the uniform window function to alleviate notation.

butions along iterations of the algorithm using a density evolution (DE) recursion. The solution to the DE recursion, called the *decoding profile* $\mathtt{x}$, is a vector of distributions of the BP log-likelihood estimates of the corresponding variable nodes along the spatial axis of coupling; we can think of them as "error distributions". More precisely, let the integer $z = -w+1, \ldots, L_c$ denote the position along the spatial direction of the graph construction (on which the copies are spread). Then the z^{th} component of $\mathtt{x}$, call it $\mathtt{x}_z$, denotes the distribution the BP log-likelihood estimate at the z^{th} position. In the special case of the binary erasure channel (BEC), this component is reduced to the usual scalar erasure-probability $0 \le x_z \le 1$ at position z along the spatial axis of coupling.

As we mentioned, the threshold saturation phenomenon is made possible due to seeding at the boundaries of the spatially coupled code, which is done by fixing the bits in (some number of) copies at the boundaries so that the probability distributions of their log-likelihoods are Δ_∞. This facilitates BP decoding near these copies (due to the connections between adjacent copies), and this effect is propagated along the rest of the coupled chain. The minimum size of the seed that guarantees the propagation of the decoding effect is of the same order of the size w of the coupling window; however, an exact determination of the minimum possible such size is still an interesting open question. Threshold saturation can alternatively be viewed in terms of the decoding profile of the coupled code: As long as the channel noise is below the MAP threshold, the profile converges to the all-Δ_∞ vector after enough iterations of the BP algorithm, where Δ_∞ is the Dirac mass at infinite log-likelihood (i.e., perfect knowledge of the bits). In the special case of the BEC, the all-Δ_∞ vector corresponds to a vector of scalar erasure probabilities driven to zero by DE iterations. In comparison, the probability distribution of the log-likelihoods of bits of the corresponding *uncoupled* code only converge to Δ_∞ when the channel noise is below the BP threshold *that is lower than the MAP threshold*.

We consider the case where the channel noise is between the BP and the MAP thresholds, and the underlying uncoupled ensemble has a *unique non-trivial stable BP fixed point* that blocks decoding.[3] In this case, it has empirically been observed that the decoding profile behaves like a *solitonic decoding wave* after a certain number of transient iterations of the BP algorithm that we call the transient phase. We show the transient phase in Figure 2.1 and the wave propagation phenomenon in Figure 2.2. The *soliton* is characterized by a *fixed shape* that seems independent of the initial condition and that has a *constant traveling velocity* that we denote by v. The phenomenology of the soliton is discussed in more detail in Section 2.2.3. The main purpose of this chapter is to derive a formula for the velocity of the soliton for irregular LDPC codes, where transmission takes place over *general BMS channels*.

The decoding wave has recently been studied for coding when transmission takes place over the BEC. In [69, 70], the authors prove that the solitonic wave solution exists and derive bounds on its velocity. However, the independence of the unique shape of the wave from the initial conditions remains an open question. In [98], more complex coupled systems are studied, where it is possible to have more than one non-trivial stable BP fixed point; in their work, other bounds on the velocity of the soliton are provided. The solitonic behavior has also been studied for the coupled

---

[3] The trivial fixed point is the all-Δ_∞ vector (or profile) whereas a non-trivial BP fixed point is a profile to which the BP algorithm converges and that is not trivial.

Curie-Weiss toy model in [53], where a formula for the velocity of the soliton, as well as an approximation for it, are derived and tested numerically.

In order to derive a formula for the velocity of the decoding wave, we consider the spatially coupled system in the continuum limit $L_c \gg w \gg 1$; this makes the derivations quite tractable. In fact, the assumption that the soliton indeed appears can be strictly true only in an asymptotic limit of a very large chain length and a large iteration number (or time); we conjecture that the formula is exact in this limit. However, even when we consider finite systems, we can see from numerical simulations that the continuum approximation of the (originally discrete) system does not compromise the exactness of our formula "too much": The results of the formula are very close to the empirical results for the velocity. It is worth noting that in Chapter 3, we also find a formula for the velocity of the propagating wave in the context of general scalar systems (not restricted to coding), for which we use the same tools and assumptions.

The formula for the velocity of the wave greatly simplifies, when we consider transmission over the BEC, because the decoding profile reduces to a scalar vector of erasure probabilities. For transmission over general BMS channels, we also simplify the analysis by applying the Gaussian approximation [37]. This consists of approximating the DE densities and the channel distribution by suitable "symmetric" Gaussian densities. The analysis then reduces to that of a one-dimensional scalar system, for which the technical difficulty is similar to that of the BEC case. We thus obtain a more tractable formula for the velocity and compare its numerical predictions with the empirical value of the velocity for finite coupling length L_c and window size w. A good agreement is found on practically the whole range of values between the BP threshold and the MAP threshold, even for small values of w.

It is of theoretical, as well as practical, interest to have a hold on the analytical expression of the velocity of the wave. The velocity is also related to other fundamental quantities that describe a coding system, such as the finite-size scaling law that predicts the error probability of finite-length spatially coupled codes. In [96], the scaling law for a finite-length spatially coupled (ℓ, r, L_c) code, where transmission takes place over the BEC, was derived. Involved in this scaling law are parameters that can be estimated using the value of the velocity of the decoding wave. Using the results of our derived formula, we provide reasonably good estimates of these parameters.

The formula for the velocity also makes use of the *potential functional* introduced and used in a series of works [55, 59, 63, 71, 99]. The potential is a "variational formulation" of the message-passing algorithm on coding systems: its stationary points are the fixed points of the DE equations described by this algorithm. The potential has been used to prove threshold saturation for several applications in [63, 100, 101]. The formalism in [63] is heavily used in the present chapter.

This chapter is organized as follows. In Section 2.2, we introduce a few preliminary notions that we will need and review the phenomenology of the solitonic wave. In Section 2.3, we formulate the continuum limit and state our main formula for the velocity of the soliton on general BMS channels. In Section 2.4, we present the derivation of the formula. In Section 2.5, we present comparisons with numerical experiments. These concern transmission over BEC channels, as well as general BMS channels in the so-called Gaussian approximation. Finally, in Section 2.6, we discuss a possible application of our formula to scaling laws for finite-size ensembles.

2.2 Preliminaries

We consider (almost) the same setting as in [63] and adopt most of the notation introduced in that work. We summarize the preliminaries we need for this chapter, that are explained in more detail in Chapter 1.

We denote by $M(\bar{\mathbb{R}})$ the space of probability measures $\mathtt{x}$ on the extended real numbers $h \in \bar{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$. Here, $h \in \bar{\mathbb{R}}$ should be interpreted as a "log-likelihood variable". We call the measure $\mathtt{x}$ *symmetric* if $\mathtt{x}(h) = e^{2h}\mathtt{x}(-h)$.

We recall that the *entropy functional* $H : \mathcal{M} \to \mathbb{R}$, that maps a finite probability measure from $M(\bar{\mathbb{R}})$ to a real number, is defined as

$$H(\mathtt{x}) = \int \mathrm{d}\mathtt{x}(h) \log_2(1 + e^{-2h}) \tag{2.1}$$

Note that this is a linear functional. Linearity is used in an important way to compute the entropy of convex combinations of measures (which also yields a probability measure). But we also compute the "entropy" associated to differences of measures by setting $H(\mathtt{x}_1 - \mathtt{x}_2) \equiv H(\mathtt{x}_1) - H(\mathtt{x}_2)$. In other words, the entropy functional is extended in an obvious way to the space of signed measures.

In the remainder of the chapter, we use the Dirac masses $\Delta_0(h)$ and $\Delta_\infty(h)$ at zero and infinite likelihood, with entropies $H(\Delta_0) = 1$ and $H(\Delta_\infty) = 0$, respectively.

We also use the standard variable-node and check-node convolution operators $\circledast$ and $\boxast$ for log-likelihood ratio message distributions involved in the DE equations [61]. We explain these operators and their properties in detail in Section 1.4.4. In short, for $\mathtt{x}_1$, $\mathtt{x}_2 \in M(\bar{\mathbb{R}})$, the usual convolution $\mathtt{x}_1 \circledast \mathtt{x}_2$ is the density of $h = h_1 + h_2$, and $\mathtt{x}_1 \boxast \mathtt{x}_2$ is the density of $h = 2\tanh^{-1}(\tanh \frac{h_1}{2} \tanh \frac{h_2}{2})$.

We make use of the so-called *duality rules* that we repeat here for convenience

$$\begin{cases} H(\mathtt{x} \circledast \mathtt{y}) + H(\mathtt{x} \boxast \mathtt{y}) = H(\mathtt{x}) + H(\mathtt{y}), \\ H(\mathtt{x} \circledast \mathtt{a}) + H(\mathtt{x} \boxast \mathtt{a}) = H(\mathtt{a}), \\ H(\mathtt{a} \circledast \mathtt{b}) + H(\mathtt{a} \boxast \mathtt{b}) = 0, \end{cases} \tag{2.2}$$

where $\mathtt{x}, \mathtt{y} \in M(\bar{\mathbb{R}})$ and $\mathtt{a}$, $\mathtt{b}$ are differences of probability measures $a = \mathtt{x}_1 - \mathtt{x}_2$, $b = \mathtt{x}_3 - \mathtt{x}_4$, $\mathtt{x}_i \in M(\bar{\mathbb{R}})$, $i = 1, 2, 3, 4$.

2.2.1 Single System

Consider an (uncoupled) LDPC(λ, ρ) code ensemble with transmission over the BMS channel. Here $\lambda(y) = \sum_\ell \lambda_\ell y^{\ell-1}$ and $\rho(y) = \sum_r \rho_r y^{r-1}$ are the usual edge-perspective variable-node and check-node degree distributions. The node-perspective degree distributions L and R are defined by $L'(y) = L'(1)\lambda(y)$ and $R'(y) = R'(1)\rho(y)$, respectively. Moreover, consider communication over a family of BMS channels whose distribution $\mathtt{c}_{\mathtt{h}}(h)$ in the log-likelihood domain is parametrized by the channel entropy[4] $H(\mathtt{c}_{\mathtt{h}}) = \mathtt{h}$.

Let $\mathtt{x}^{(t)}$ denote the variable node output distribution of the BP algorithm at iteration $t \in \mathbb{N}$. We can track the average behavior of the BP decoder by means of

---

[4] In the literature, this quantity is often denoted by $\mathtt{c}(\mathtt{h})$.

the DE iterative equations that we write as a recursion in terms of the variable node output distribution as follows,

$$\mathtt{x}^{(t+1)} = \mathtt{c}_{\mathtt{h}} \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{x}^{(t)})), \tag{2.3}$$

with initial condition $\mathtt{x}^{(0)} = \Delta_0$ (equivalently, we can take the more natural initial condition $\mathtt{x}^{(0)} = \mathtt{c}_{\mathtt{h}}$).

There are two thresholds of interest for us. The first one is the algorithmic threshold. For a family of BMS channels whose channel distributions $\mathtt{c}_{\mathtt{h}}(h) : \mathbb{R} \to M(\bar{\mathbb{R}})$ are ordered by degradation and parametrized by their entropy $H(\mathtt{c}_{\mathtt{h}}) = \mathtt{h}$, the algorithmic or BP threshold is defined to be

$$\mathtt{h}_{\mathrm{BP}} = \{\mathtt{h} \in [0,1] : \mathtt{x} = \mathtt{c}_{\mathtt{h}} \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{x})) \implies \mathtt{x} = \Delta_\infty\}.$$

The second threshold corresponds to optimal (MAP) decoding

$$\mathtt{h}_{\mathrm{MAP}} = \{\mathtt{h} \in [0,1] : \liminf_{n\to\infty} \frac{1}{n}\mathbb{E}[H(X^n|Y^n(\mathtt{h}))] > 0\},$$

where $H(X^n|Y^n(\mathtt{h}))$ is the conditional Shannon entropy of the input given by the channel observations and $\mathbb{E}$ is the expectation over the code ensemble.

The potential functional $W_s(\mathtt{x})$ of the "single" or uncoupled system is

$$W_s(\mathtt{x}) = \frac{1}{R'(1)}H(R^{\boxast}(\mathtt{x})) + H(\rho^{\boxast}(\mathtt{x})) - H(\mathtt{x} \boxast \rho^{\boxast}(\mathtt{x})) - \frac{1}{L'(1)}H(\mathtt{c} \circledast L^{\circledast}(\rho^{\boxast}(\mathtt{x}))). \tag{2.4}$$

The fixed point form of the DE equation (2.3) is obtained by setting to zero the functional derivative of $W_s(\mathtt{x};\mathtt{c})$ with respect to $\mathtt{x}$. In other words, $\mathtt{x} = \mathtt{c}_{\mathtt{h}} \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{x}))$ is equivalent to

$$\lim_{\gamma\to 0} \frac{1}{\gamma}(W_s(\mathtt{x}) + \gamma\eta) - W_s(x)) = 0, \tag{2.5}$$

where η is a difference of two probability measures (see [63] for the proof of this statement). The BP threshold $\mathtt{h}_{\mathrm{BP}}$ and the MAP threshold $\mathtt{h}_{\mathrm{MAP}}$ can be obtained from the analysis of the stationary points of the potential function. See [63, 82] for more details and a rigorous discussion of this issue.

Remark about notations. In the remainder of the chapter, we omit the subscript $\mathtt{h}$ from $\mathtt{c}_{\mathtt{h}}$ and the argument h from $\mathtt{x}(h)$ most of the time. This is because we will need a subscript (resp. an argument) z that represents the position along the chain in the discrete (resp. continuous) case.

2.2.2 Spatially Coupled System

The definitions of the BP and MAP thresholds above extend to the spatially coupled setting. For standard (uncoupled) LDPC codes, the BP threshold $\mathtt{h}_{\mathrm{BP}}$ is, in general, lower than the MAP threshold $\mathtt{h}_{\mathrm{MAP}}$. Spatial coupling exhibits two attractive properties. First, *the MAP threshold is conserved* under coupling in the limit of $L_c \to +\infty$ and for all w. The proof of this statement is found in [82]. Second, *the BP threshold of the coupled system saturates to the MAP threshold*, as proved in [43, 63]. This phenomenon is called *threshold saturation*, the main consequence of which is that one can decode perfectly up to the $\mathtt{h}_{\mathrm{MAP}}$.

We now describe the density evolution and potential functional formalism for the spatially coupled code ensemble. Consider $L_c + w$ "copies" of the single system described in Section 2.2.1, placed on the spatial coordinates $z = -w+1, \dots, L_c+w-1$. The system at position z is coupled to its w neighboring systems by means of a uniform coupling window. For simplicity, we take a uniform coupling window. We denote by $\tilde{\mathtt{x}}_z^{(t)}$ the variable node output distribution at position $z = -w+1, \dots, L_c$ on the spatial axis and at time $t \in \mathbb{N}$. The DE equation of the coupled system takes the form

$$\tilde{\mathtt{x}}_z^{(t+1)} = \mathtt{c}_z \circledast \lambda^{\circledast}\Bigg(\frac{1}{w}\sum_{i=0}^{w-1} \rho^{\boxast}\Big(\frac{1}{w}\sum_{j=0}^{w-1} \tilde{\mathtt{x}}_{z+i-j}^{(t)}\Big)\Bigg). \tag{2.6}$$

In this equation, $\mathtt{c}_z = \mathtt{c}$, for $z = 1, \dots, L_c + w - 1$ and $\mathtt{c}_z = \Delta_\infty$ for $z = -w+1, \dots, 0$. Furthermore, we fix the left boundary to $\mathtt{x}_z^{(t)} = \Delta_\infty$ for $z = -w+1, \dots, 0$, for all $t \in \mathbb{N}$. These conditions express perfect information at the left boundary which is what allows seeding and decoding wave propagation along the chain of coupled codes. The initial condition for the recursion in (2.6) is $\mathtt{x}_z^{(0)} = \Delta_0$ for $z = 1, \dots, L_c + w - 1$.

It is more convenient for our analysis to work with the check node input distribution than with the variable node output distribution. The check node input distribution at position z on the coupling axis and time $t \in \mathbb{N}$ is denoted by $\mathtt{x}_z^{(t)}$ and is equal to the normalized sum of the averaged variable node output distributions at positions $z - w + 1, \dots, z$. This is due to the forward averaging technique in which coupling is made (in our work). More formally, we write the check node input distribution as

$$\mathtt{x}_z^{(t)} = \frac{1}{w}\sum_{i=0}^{w-1} \tilde{\mathtt{x}}_{z-i}^{(t)}.$$

Then, using this change of variables, we rewrite (2.6) as

$$\mathtt{x}_z^{(t+1)} = \frac{1}{w}\sum_{i=0}^{w-1} \mathtt{c}_{z-i} \circledast \lambda^{\circledast}\Bigg(\frac{1}{w}\sum_{j=0}^{w-1} \rho^{\boxast}\Big(\mathtt{x}_{z-i+j}^{(t)}\Big)\Bigg). \tag{2.7}$$

Just as in the single system case, this DE equation can be expressed as the stationarity condition of a potential functional (see [63])

$$\begin{aligned} W(\underline{\mathtt{x}}) = \sum_{z=-w+1}^{L_c} \Big\{ & \frac{1}{R'(1)} H(R^{\boxast}(\mathtt{x}_z)) + H(\rho^{\boxast}(\mathtt{x}_z)) - H(\mathtt{x}_z \boxast \rho^{\boxast}(\mathtt{x}_z)) \\ & - \frac{1}{L'(1)} H\Big(\mathtt{c}_z \circledast L^{\circledast}\big(\frac{1}{w}\sum_{i=0}^{w-1} \rho^{\boxast}(\mathtt{x}_{z+i})\big)\Big)\Big\}, \end{aligned} \tag{2.8}$$

where $\underline{\mathtt{x}} = (\mathtt{x}_{-w+1}, \dots, \mathtt{x}_{L_c+w-1})$. The fixed point form of (2.7) is equivalent to $\lim_{\gamma \to 0} \gamma^{-1}(W(\underline{\mathtt{x}}+\gamma\underline{\eta}) - W(\underline{\mathtt{x}}) = 0$ for $\underline{\eta} = (\eta_{-w+1}, \dots, \eta_{L_c+w-1})$ where η_i are differences of probability measures.

2.2.3 Phenomenological Observations

Our derivation is far from rigorous and is based on an assumption derived from a phenomenological picture observed from simulations. We summarize the main observations in this paragraph for the case of transmission over the BEC channel. This channel also gives us the opportunity to illustrate the formalism outlined in Sections 2.2.1 and 2.2.2 in a concrete case.

The channel distribution of the BEC can be expressed as $\mathtt{c}_\epsilon(h) = \epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$, where ϵ is the erasure probability, and $H(\mathtt{c}_\epsilon) = \epsilon$ (hence $\mathtt{h} = \epsilon$). The density of the BP estimates of log-likelihood variables can be parametrized as $\mathtt{x}^{(t)}(h) = x^{(t)}\Delta_0(h) + (1 - x^{(t)})\Delta_\infty(h)$, where $x^{(t)} \in [0,1]$ is interpreted as an erasure probability at iteration $t \in \mathbb{N}$. The DE equation becomes a one dimensional iterative map, expressed as

$$x^{(t+1)} = \epsilon\lambda(1 - \rho(1 - x^{(t)})), \tag{2.9}$$

over scalars in $[0,1]$. These iterations are always initialized with $x^{(0)} = 1$, or equivalently with $x^{(0)} = \epsilon$. The corresponding fixed point equation is the stationarity condition for a (single) potential function

$$W_{\mathrm{BEC}}(x) = \frac{1}{R'(1)}(1 - R(1-x)) - x\rho(1-x) - \frac{\epsilon}{L'(1)}L(1 - \rho(1-x)). \tag{2.10}$$

Note that the potential function is defined up to a constant that is set, in our work, such that $W_{\mathrm{BEC}}(0) = 0$. Figure 1.6 illustrates the (single) potential function for a $(3,6)$ regular Gallager ensemble, for several values of ϵ. For $\epsilon < 0.43$, (2.10) is strictly increasing; equivalently, the DE iterations are driven to the unique minimum at $x = 0$. At $\epsilon = \epsilon_{\mathrm{BP}} = 0.43$, a horizontal inflexion point appears, at which we obtain a non-trivial local minimum x_{BP}; this corresponds to the non-trivial fixed point reached by DE iterations. It is known that the MAP threshold is equal to the erasure probability at which the non-trivial minimum is at the same height as the trivial one, and that decoding becomes impossible once the non-trivial minimum becomes the global minimum. In the example, this occurs when $\epsilon = \epsilon_{\mathrm{MAP}} = 0.4881$. In Figure 1.6, we also show the *energy gap* defined for $\epsilon_{\mathrm{BP}} < \epsilon < \epsilon_{\mathrm{MAP}}$, that is defined by $\Delta E = W_{\mathrm{BEC}}(x_{\mathrm{BP}}) - W_{\mathrm{BEC}}(0)$. At the MAP threshold, we have $\Delta E = 0$.

We now describe the phenomenology of the solitonic wave for spatially coupled codes. Our discussion is limited to the case where the underlying code ensemble has a single non-trivial DE fixed point (equivalently, the single potential function has a single non-trivial local minimum). One can show that this is always the case for regular code ensembles. For irregular degree distributions, the situation may be more complicated as several non-trivial fixed points might appear. For the case of transmission over the BEC, Equ. (2.7) reads

$$x_z^{(t+1)} = \frac{1}{w}\sum_{i=0}^{w-1} \epsilon_{z-i}\lambda\Big(\frac{1}{w}\sum_{j=0}^{w-1}\big(1 - \rho(1 - x_{z-i+j}^{(t)})\big)\Big). \tag{2.11}$$

Here, $\epsilon_z = \epsilon$, for $z = 1, \dots, L_c + w - 1$ and $\epsilon_z = \Delta_\infty$ for $z = -w+1, \dots, 0$. Furthermore, we fix the left boundary to $x_z^{(t)} = 0$ for $z = -w+1, \dots, 0$, for all $t \in \mathbb{N}$. This is the "seed" that initiates decoding at the boundary of the coupled chain. The initial condition for the iterations is $x_z^{(0)} = 1$ (or ϵ) for $z = 1, \dots, L_c + w - 1$.

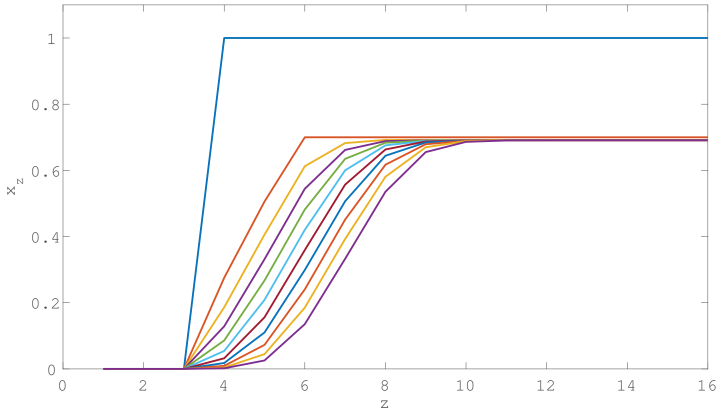

Figure 2.1: We consider the spatially coupled LDPC($\lambda(x), \rho(x)$) ensemble, with $\lambda(x) = 0.3x^3 + 0.4x^5 + 0.3x^6$ and $\rho(x) = x^5$, with transmission over the BEC(0.7). The parameters of coupling are $L_c + w = 16$ and $w = 3$. We plot the decoding profile during the first 10 iterations, where the profile is initialized to 0 for $z \leq 3$ and to 1 elsewhere. We observe that the segment initialized to 1 decreases quickly and converges to the BP threshold $x_{\text{BP}} = 0.6907$. We observe that the transient phase is only about 5 iterations long for this example, before the decoding profile converges to a fixed shape.

The evolution of the decoding wave can be decomposed into two phases: a *transient* and a *stationary* one. In the transient phase, we observe a profile of erasure probabilities $\underline{x} = (x_{-w+1}, \dots, x_{L_c})$ changing shape. The segment initialized to $x_z^{(0)} = 1$ quickly drops to $x_z \approx x_{\text{BP}}$ where it remains stuck on the far right for large values of z. The seeding region, however, starts progressing towards the right-hand side and, after a few iterations, a fixed profile shape develops. This transient phase is illustrated in Figure 2.1 for an irregular code. Overall, it only lasts for a few iterations (around 5 iterations for this example). After this transient phase is over, one observes a stationary phase with a *solitonic behavior*, as depicted in Figure 2.2. The profile of erasure probabilities has a stationary shape with a front centered at some position z_{front} that moves towards the right at a constant speed. The position z_{front} is relatively well-localized within approximately $2w$ positions around the front of the profile. The front quickly approaches $x_z \to 0$ for $z < z_{\text{front}}$ and $x_z \to x_{\text{BP}}$ for $z > z_{\text{front}}$. The stationary phase and its soliton are depicted in Figure 2.2 for a finite spatially coupled $(3,6)$-regular ensemble with chain length $L_c = 50$ and $w = 3$ for $\epsilon = 0.46$. In this figure, we plot the decoding profile every 30 iterations starting from the 30th iteration (the leftmost curve) and up till the 150th iteration (the rightmost curve). The front (or kink) increases sharply from $x_z = 0$ to $x_z = x_{\text{BP}} = 0.3789$ over a width of the order of $2w = 6$.

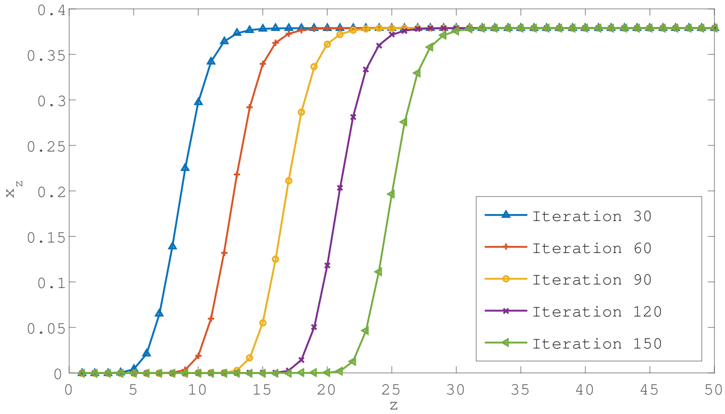

Figure 2.2: We consider the $(3, 6)$-regular LDPC spatially coupled code with $L_c = 50$, $w = 3$ on the BEC(0.46). We plot the error probability along the spatial dimension z and observe the "decoding wave". This "soliton" is plotted every 30 iterations up till iteration 150 and is seen to make a quick transition from zero error probability to the BP-value $x_{\mathsf{BP}} = 0.3798$ of the error probability. The optimal (MAP) noise threshold is $\epsilon_{\mathsf{MAP}} = 0.4881$.

2.3 Continuum Limit and Main Result

2.3.1 Continuum Limit

We now consider the coupled system in the *continuum limit*, in which the length of the coupling chain L_c is first taken very large $L_c \to +\infty$, and then the window size is taken very large $w \to +\infty$. The continuum limit has already been considered for the special case of the BEC in [69, 70, 72, 73]. To avoid superfluous notation, we shall keep the same symbols for the profile, the spatial position, and the channel distribution in the continuum limit. We thus denote by $\mathtt{x}$ the *continuous* profile of distributions and set $\mathtt{x}(\frac{z}{w}, t) \equiv \mathtt{x}_z^{(t)}$. We then replace $\frac{z}{w} \to z$ so that the new z is the *continuous* variable on the spatial axis, $z \in \mathbb{R}$.

In view of the discussion of the phenomenology in Section 2.2.3, we consider the class of profiles satisfying the "natural boundary conditions" $\mathtt{x}(z, t) \to \Delta_\infty$ when $z \to -\infty$ for all $t \in \mathbb{R}$, $\mathtt{x}(z, t) \to \mathtt{x}_{\mathrm{BP}}$ when $z \to +\infty$ for all $t \in \mathbb{N}$, where $\mathtt{x}_{\mathrm{BP}}$ is the unique non-trivial stable fixed point of the DE recursion for the single system Equ. (2.3).

The BMS channel distribution is now also continuous, and we denote by $\mathtt{c}(z)$ the channel distribution at the continuous spatial position $z \in \mathbb{R}$. The DE recursion (2.7) then takes the form

$$\mathtt{x}(z, t+1) = \int_0^1 \mathrm{d}u\, \mathtt{c}(z-u) \circledast \lambda^{\circledast}\Big(\int_0^1 \mathrm{d}s\, \rho^{\boxast}\big(\mathtt{x}(z-u+s, t)\big)\Big). \tag{2.12}$$

The initial condition at $t = 0$ is given by a Heaviside-like profile $\mathtt{x}(z, 0)$ that interpolates between the two limiting values of the boundary condition, namely $\mathtt{x}(z, 0) = \Delta_\infty$ when $z < 0$ and $\mathtt{x}(z, 0) = \mathtt{x}_{\mathrm{BP}}$ when $z \geq 0$.

2.3.2 Statement of Main Result

We consider the case when the channel entropy $\mathtt{h}$ is in the range $[\mathtt{h}_{\mathrm{BP}}, \mathtt{h}_{\mathrm{MAP}}]$. The phenomenology tells us that: (i) after a transient phase, the profile develops a fixed shape $\mathtt{X}$; (ii) the shape is independent of the initial condition; (iii) the shape travels at constant speed v; (iv) the shape satisfies the boundary conditions $\mathtt{X}(z) \to \Delta_\infty$ for $z \to -\infty$ and $\mathtt{X}(z) \to \mathtt{x}_{\mathrm{BP}}$ for $z \to +\infty$. We formalize these observations by making the ansatz:

Ansatz. For each $\mathtt{h} \in [\mathtt{h}_{\mathrm{BP}}, \mathtt{h}_{\mathrm{MAP}}]$, there exists a velocity $v \geq 0$ and a family of probability measures $\mathtt{X}(z)$ (indexed by $z \in \mathbb{R}$) satisfying the boundary conditions $\mathtt{X}(z) \to \Delta_\infty$ for $z \to -\infty$ and $\mathtt{X}(z) \to \mathtt{x}_{\mathrm{BP}}$ for $z \to +\infty$, such that for $t \to +\infty$ and $|z - vt| = O(1)$, the solution of DE (2.12) is independent of the initial condition and satisfies $\mathtt{x}(z,t) \to \mathtt{X}(z - vt)$.

Implicit in this ansatz is the fact that we restrict ourselves to underlying code ensembles that have only one non-trivial stable BP fixed point. This is always true for regular codes, for example, but is not limited to this case. Ensembles with several non-trivial fixed points could lead to more complicated phenomenologies, as emphasized in [98]; this would require a different ansatz.

Velocity of the soliton for general BMS channels. Under the assumptions above, the velocity of the soliton is given by

$$v = \frac{\Delta E}{\int_{\mathbb{R}} \mathrm{d}z\, H\Big(\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)^{\boxast 2}\Big)}, \tag{2.13}$$

where ΔE is the *energy gap* defined as

$$\Delta E = W_s(\mathtt{x}_{\mathrm{BP}}) - W_s(\Delta_\infty), \tag{2.14}$$

where we recall that W_s is the potential of the *uncoupled system* shown in (2.4), $\mathtt{x}_{\mathrm{BP}}$ is the non-trivial BP fixed point to which the uncoupled system converges, and Δ_∞ is the trivial fixed point (the Dirac mass at infinity).

We make a few remarks. In this formula, the prime denotes the derivative $\mathtt{X}'(z) = \lim_{\delta \to 0} \delta^{-1}(\mathtt{X}(z+\delta) - \mathtt{X}(z))$ that is to be interpreted as a difference between two measures. The energy gap is only defined for $\mathtt{h}_{\mathrm{BP}} \leq \mathtt{h} \leq \mathtt{h}_{\mathrm{MAP}}$; that is, when the single potential W_s has a non-trivial non-negative local minimum (see, for example, Figure 1.6). The energy gap is equal to zero only when $\mathtt{h} = \mathtt{h}_{\mathrm{MAP}}$, which confirms the fact that the velocity of decoding is zero (no decoding occurs) in this case. We note also that, with our normalizations, $W_s(\Delta_\infty) = 0$.

Formula (2.13) involves the shape $\mathtt{X}$. Using the DE equation, the ansatz $\mathtt{x}(z,t) \to \mathtt{X}(z - vt)$, and the approximation $\mathtt{x}(z, t+1) - \mathtt{x}(z,t) \approx -v\mathtt{X}'(z - vt)$, which is valid for small values of the velocity v, we find (after a change of variables) that $\mathtt{X}(z)$ is the solution of

$$\mathtt{X}(z) - v\mathtt{X}'(z) = \int_0^1 \mathrm{d}u\, \mathtt{c}(z-u) \circledast \lambda^{\circledast}\Big(\int_0^1 \mathrm{d}s\, \rho^{\boxast}\Big(\mathtt{X}(z-u+s)\Big)\Big). \tag{2.15}$$

To obtain the shape $\mathtt{X}(z)$ and the velocity v, one must iteratively solve the closed system of equations formed by (2.13) and (2.15). Note that the assumption of

v being small is strictly valid for values of $\mathtt{h}$ close to $\mathtt{h}_{\mathrm{MAP}}$. However, numerical simulations confirm that in practice, the resulting formula for the velocity is precise over the whole range $[\mathtt{h}_{\mathrm{BP}}, \mathtt{h}_{\mathrm{MAP}}]$.

2.4 Derivation of Main Result

Let us briefly outline the main steps of derivation. We first write down a potential functional that gives, in the continuous setting, the DE fixed point equation corresponding to (2.12). This enables us to formulate the DE iterations as a sort of gradient descent equation (Section 2.4.1). From there on, we use the ansatz in Section 2.3.2 to derive the formula for the velocity (2.13).

2.4.1 Density Evolution as Gradient Descent

We denote by $\Delta\mathcal{W}(\mathtt{x})$ the potential functional of the coupled system in the continuum limit obtained from (2.8) (see Section 1.6). This limit involves an integral over the spatial direction $z \in \mathbb{R}$, and in order to obtain a convergent result, we must subtract a "reference energy". Essentially, any static reference profile, here called $\mathtt{x}_0(z)$, that satisfies the boundary conditions $\mathtt{x}_0(z) \to \Delta_\infty$, $z \to -\infty$ and $\mathtt{x}_0(z) \to x_{\mathrm{BP}}$, $z \to +\infty$, will suffice. For concreteness, one can take a Heaviside-like profile $\mathtt{x}_0(z) = \Delta_\infty$, $z < 0$, $\mathtt{x}_0(z) = x_{\mathrm{BP}}$, $z \geq 0$. The potential functional is then defined as

$$\Delta\mathcal{W}(\mathtt{x}) = \int_{\mathbb{R}} \mathrm{d}z\, \{P(z,\mathtt{x}) - P(z,\mathtt{x}_0)\}, \tag{2.16}$$

where $P(z,\mathtt{x})$ is a z-dependent functional of $\mathtt{x}$ equal to

$$\begin{aligned} P(z,\mathtt{x}) = \frac{1}{R'(1)} H(R^{\boxast}(\mathtt{x}(z,t))) + H(\rho^{\boxast}(\mathtt{x}(z,t))) - H(\mathtt{x}(z,t) \boxast \rho^{\boxast}(\mathtt{x}(z,t))) \\ - \frac{1}{L'(1)} H\Big(\mathtt{c}(z) \circledast L^{\circledast}\big(\int_0^1 \mathrm{d}s\, \rho^{\boxast}(\mathtt{x}(z+s,t))\big)\Big). \end{aligned} \tag{2.17}$$

In Section 2.7, we calculate the functional derivative of $\Delta\mathcal{W}(\mathtt{x})$ in a direction $\eta(z,t)$; this derivative is defined as

$$\frac{\delta\Delta\mathcal{W}}{\delta\mathtt{x}}[\eta(z,t)] = \frac{\mathrm{d}}{\mathrm{d}\gamma} \Delta\mathcal{W}(\mathtt{x}+\gamma\eta)\Big|_{\gamma=0}, \tag{2.18}$$

and find

$$\frac{\delta\Delta\mathcal{W}}{\delta\mathtt{x}}[\eta(z,t)] \tag{2.19}$$

$$= \int_{\mathbb{R}} \mathrm{d}z\, H\Bigg(\Big(\int_0^1 \mathrm{d}u\, \mathtt{c}(z-u) \circledast \lambda^{\circledast}\big(\int_0^1 \mathrm{d}s\, \rho^{\boxast}(\mathtt{x}(z-u+s,t))\big) \tag{2.20}$$

$$- \mathtt{x}(z,t)\Big) \boxast \rho'^{\boxast}(\mathtt{x}(z,t)) \boxast \eta(z,t)\Bigg). \tag{2.21}$$

From (2.12) and (2.21), we deduce that

$$\int_{\mathbb{R}} \mathrm{d}z\, H\Big((\mathtt{x}(z,t+1) - \mathtt{x}(z,t)) \boxast \rho'^{\boxast}(\mathtt{x}(z,t)) \boxast \eta(z,t)\Big) = \frac{\delta\Delta\mathcal{W}}{\delta\mathtt{x}}[\eta(z,t)]. \tag{2.22}$$

Using the duality rule (2.2) with $\mathtt{a} = \mathtt{x}(z, t+1) - \mathtt{x}(z,t)$ and $b = \rho'^{\boxast}(\mathtt{x}(z,t)) \boxast \eta(z,t)$ (where η must be a difference of two measures so that b is also such a difference) and the associativity of $\boxast$, we note that the equation above can also be formulated as

$$\int_{\mathbb{R}} \mathrm{d}z \, H\Big((\mathtt{x}(z,t+1) - \mathtt{x}(z,t)) \circledast (\rho'^{\boxast}(\mathtt{x}(z,t)) \boxast \eta(z,t)) \Big) = -\frac{\delta \Delta \mathcal{W}}{\delta \mathtt{x}}[\eta(z,t)]. \quad (2.23)$$

In this form, we recognize a sort of infinite-dimensional gradient descent equation in a space of measures. This reformulation of DE forms the basis of the derivation of the formula for the velocity.

2.4.2 Final Steps of the Derivation

The potential functional can be decomposed into two parts: a "single system" part $\mathcal{W}_s(\mathtt{x})$ and an "interaction" part $\mathcal{W}_i(\mathtt{x})$ that results from coupling. We have

$$\Delta \mathcal{W}(\mathtt{x}) = \mathcal{W}_s(\mathtt{x}) + \mathcal{W}_i(\mathtt{x}), \quad (2.24)$$

where the single part is defined as

$$\mathcal{W}_s(\mathtt{x}) = \int_{\mathbb{R}} \mathrm{d}z \, \{P_s(z, \mathtt{x}) - P_s(z, \mathtt{x}_0)\}, \quad (2.25)$$

where

$$\begin{aligned} P_s(z,\mathtt{x}) = {} & \frac{1}{R'(1)} H(R^{\boxast}(\mathtt{x}(z,t))) + H(\rho^{\boxast}(\mathtt{x}(z,t))) - H(\mathtt{x}(z,t) \boxast \rho^{\boxast}(\mathtt{x}(z,t))) \\ & - \frac{1}{L'(1)} H\Big(\mathtt{c}(z) \circledast L^{\circledast}\big(\rho^{\boxast}(\mathtt{x}(z,t))\big)\Big), \end{aligned} \quad (2.26)$$

and the interaction part is defined as

$$\mathcal{W}_i(\mathtt{x}) = \frac{1}{L'(1)} \int_{\mathbb{R}} \mathrm{d}z \, \{P_i(z,\mathtt{x}) - P_i(z, \mathtt{x}_0)\}, \quad (2.27)$$

where

$$P_i(z,\mathtt{x}) = H\Big(\mathtt{c}(z) \circledast L^{\circledast}\big(\rho^{\boxast}(\mathtt{x}(z,t))\big)\Big) - H\Big(\mathtt{c}(z) \circledast L^{\circledast}\Big(\int_0^1 \mathrm{d}u \, \rho^{\boxast}(\mathtt{x}(z+u,t))\Big)\Big). \quad (2.28)$$

For future use, we note that $P_s(z,\mathtt{x}) = W_s(\mathtt{x}(z,t))$ is the single system potential (2.4) "at position z". With these definitions, the gradient descent equation (2.23) can be written as

$$\begin{aligned} & \int_{\mathbb{R}} \mathrm{d}z \, H\Big((\mathtt{x}(z,t+1) - \mathtt{x}(z,t)) \boxast \rho'^{\boxast}(\mathtt{x}(z,t)) \boxast \eta(z,t) \Big) \\ & \qquad = \frac{\delta \mathcal{W}_s}{\delta \mathtt{x}}[\eta(z,t)] + \frac{\delta \mathcal{W}_i}{\delta \mathtt{x}}[\eta(z,t)] \end{aligned} \quad (2.29)$$

We use the ansatz to compute the three terms in this equation in the regime $t \to +\infty$, $z \to +\infty$ such that $|z - vt| = O(1)$. We find it convenient to choose the direction $\eta(z,t) = \mathtt{X}'(z - vt)$.

We start with the left-hand side of (2.29). We use the ansatz $\mathtt{x}(z,t) \to \mathtt{X}(z-vt)$, the approximation $\mathtt{x}(z,t+1)-\mathtt{x}(z,t) \approx -v\mathtt{X}'(z-vt)$, and the special choice of $\eta(z,t) = \mathtt{X}'(z-vt)$ to rewrite the left-hand side of (2.29) as

$$v \int_{\mathbb{R}} \mathrm{d}z \, H\big(\mathtt{X}'(z-vt) \boxast \rho'^{\boxast}(\mathtt{X}(z-vt)) \boxast \mathtt{X}'(z-vt,t)\big). \tag{2.30}$$

Using the commutativity of the operator $\boxast$, we can see that this is equal to

$$v \int_{\mathbb{R}} \mathrm{d}z \, H\big(\rho'^{\boxast}(\mathtt{X}(z-vt)) \boxast \mathtt{X}'(z-vt)^{\boxast 2}\big). \tag{2.31}$$

Note that we can shift the argument in the integrals $z-vt \to z$ to make this term independent of time.

We now consider the first functional derivative on the right-hand side of (2.29), again when $\eta(z,t) = \mathtt{X}'(z-vt)$. It should be clear that we can immediately make the change of variables in the integrals $z-vt \to z$, which simplifies the formulas. Using the calculations in Section 2.7, we find

$$\begin{aligned}\frac{\delta \mathcal{W}_s}{\delta \mathtt{X}}[\mathtt{X}'(z)] = \int_{\mathbb{R}} \mathrm{d}z \Big\{ & H\big(\mathtt{X}(z) \circledast [\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)]\big) \\ & - H\big(\mathsf{c} \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{X}(z))) \circledast [\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)]\big)\Big\}.\end{aligned} \tag{2.32}$$

In order to simplify the above, we remark the following.

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}z}\Big\{\frac{1}{R'(1)} H(R^{\boxast}(\mathtt{X}(z))) - H(\mathtt{X}(z) \boxast \rho^{\boxast}(\mathtt{X}(z))) \\ &\qquad + H(\rho^{\boxast}(\mathtt{X}(z))) - \frac{1}{L'(1)} H\big(\mathsf{c}(z) \circledast L^{\circledast}(\rho^{\boxast}(\mathtt{X}(z)))\big)\Big\} \\ &= H(\rho^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)) - H(\mathtt{X}'(z) \boxast \rho^{\boxast}(\mathtt{X}(z))) \\ &\qquad - H(\mathtt{X}(z) \boxast \rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)) + H(\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)) \\ &\qquad - H(\mathsf{c}(z) \circledast \lambda^{\circledast}(\rho^{\boxast}(\mathtt{X}(z))) \circledast [\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)]).\end{aligned}$$

The first two terms on the right-hand side cancel out. Also, using the duality rule (2.2) for the third term, we obtain the integrand in (2.32). In other words, we can rewrite (2.32) as

$$\begin{aligned}\frac{\delta \mathcal{W}_s}{\delta \mathtt{X}}[\mathtt{X}'(z)] = \int_{\mathbb{R}} \mathrm{d}z \, \frac{\mathrm{d}}{\mathrm{d}z} P_s(z, \mathtt{X}) &= \int_{\mathbb{R}} \mathrm{d}z \, \frac{\mathrm{d}}{\mathrm{d}z} W_s(\mathtt{X}(z)) \\ &= W_s(\mathtt{x}_{\mathrm{BP}}) - W_s(\Delta_\infty) \\ &= \Delta E.\end{aligned} \tag{2.33}$$

We now show that the functional derivative of the interaction part in (2.29) does not contribute to the sum of derivatives when $\eta(z,t)$ is chosen to be equal to $\mathtt{X}'(z-vt)$. By directly applying the definition of the functional derivative, we find

$$\begin{aligned}\frac{\delta \mathcal{W}_i}{\delta \mathtt{X}}[\mathtt{X}'(z)] = \int_{\mathbb{R}} \mathrm{d}z \Big\{ & H\big(\mathsf{c} \circledast \big[\lambda^{\circledast}(\rho^{\boxast}(\mathtt{X}(z))) \circledast (\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z))\big]\big) \\ & - H\big(\mathsf{c} \circledast \big[\lambda^{\circledast}\big(\int_0^1 \mathrm{d}u \, \rho^{\boxast}(\mathtt{X}(z+u))\big) \\ & \qquad \circledast \big(\int_0^1 \mathrm{d}s \, \rho'^{\boxast}(\mathtt{X}(z+s)) \boxast \mathtt{X}'(z+s)\big)\big]\big)\Big\}.\end{aligned} \tag{2.34}$$

We notice that the integrand is a total derivative. More specifically, it is equal to

$$\frac{1}{L'(1)}\frac{\mathrm{d}}{\mathrm{d}z}\Big\{H\big(\mathsf{c}\circledast L^{\circledast}(\rho^{\boxast}(\mathtt{X}(z)))\big)-H\big(\mathsf{c}\circledast L^{\circledast}(\int_0^1 \mathrm{d}u\,\rho^{\boxast}(\mathtt{X}(z+u)))\big)\Big\}.$$

Due to the boundary conditions, we have $\lim_{z\to-\infty}\mathtt{X}(z)=\lim_{z\to-\infty}\mathtt{X}(z+u)=\Delta_\infty$ and $\lim_{z\to+\infty}\mathtt{X}(z)=\lim_{z\to-\infty}\mathtt{X}(z+u)=\mathtt{x}_{\mathrm{BP}}$. We thus conclude that the total derivative integrates to zero and

$$\frac{\delta\mathcal{W}_i}{\delta\mathtt{X}}[\mathtt{X}'(z)]=0. \tag{2.35}$$

Finally, replacing (2.31), (2.33), (2.35) in (2.29) we obtain the simple relationship

$$v\int_{\mathbb{R}}\mathrm{d}z\,H\big(\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)^{\boxast 2}\big)=\Delta E, \tag{2.36}$$

which yields the formula for the velocity (2.13).

2.5 Applications and Numerical Experiments

2.5.1 Binary Erasure Channel (BEC)

When transmission takes place over the BEC, the formula for the velocity can be obtained by directly simplifying the general formula in (2.13).[5] We will suppose that the underlying LDPC(λ,ρ) code is such that the DE equation has a single non-trivial fixed point that we denote by $x_{\mathrm{BP}}\neq 0$. Furthermore, we fix ϵ in the range $[\epsilon_{\mathrm{BP}},\epsilon_{\mathrm{MAP}}]$ (recall that the channel entropy reduces to $H(\mathsf{c})=\mathtt{h}=\epsilon$ when transmission takes place over the BEC).

The channel distribution can be written as $\mathsf{c}=\epsilon\Delta_0+(1-\epsilon)\Delta_\infty$, and the profile is of the form $\mathtt{x}(z,t)=x(z,t)\Delta_0+(1-x(z,t))\Delta_\infty$, where $x(z,t)\in[0,1]$ is the scalar erasure probability at position z and time t. This tends to a fixed shape defined by

$$\mathtt{X}(z)=X(z)\Delta_0+(1-X(z))\Delta_\infty, \tag{2.37}$$

where $X(z)\in[0,1]$ satisfies $\lim_{z\to-\infty}X(z)=0$, $\lim_{z\to+\infty}X(z)=x_{\mathrm{BP}}$. We also characterize the derivative of the shape with

$$\mathtt{X}'(z)=X'(z)\Delta_0-X'(z)\Delta_\infty. \tag{2.38}$$

We note the following identities, that are valid for scalar maps $f,g:\mathbb{R}\to[0,1]$ (such as λ, ρ, L, R and their derivatives)

$$\begin{cases} f^{\circledast}(\mathtt{X}(z))=f(X(z))\Delta_0+(1-f(X(z)))\Delta_\infty,\\ g^{\boxast}(\mathtt{X}(z))=(1-g(1-X(z))\Delta_0+g(1-X(z))\Delta_\infty. \end{cases} \tag{2.39}$$

---

[5] We can also use the formula for general scalar systems in Chapter 3, that covers cases beyond coding theory, because the BEC yields a scalar system.

To compute the denominator of (2.13), we first use (1.41), (2.38), and (2.39) to write

$$
\begin{aligned}
&\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)^{\boxast 2} \\
&\quad = \{(1-\rho'(1-X(z)))\Delta_0 + \rho'(1-X(z))\Delta_\infty\} \boxast \{X'(z)\Delta_0 - X'(z)\Delta_\infty\}^{\boxast 2} \\
&\quad = \{(1-\rho'(1-X(z)))\Delta_0 + \rho'(1-X(z))\Delta_\infty\} \boxast \{X'(z)^2\Delta_\infty - X'(z)^2\Delta_0\} \\
&\quad = (1-\rho'(1-X(z)))X'(z)^2\Delta_0 - (1-\rho'(1-X(z)))X'(z)^2\Delta_0 \\
&\qquad\quad + \rho'(1-X(z))X'(z)^2\Delta_\infty - \rho'(1-X(z))X'(z)^2\Delta_0\} \\
&\quad = \rho'(1-X(z))X'(z)^2\Delta_\infty - \rho'(1-X(z))X'(z)^2\Delta_0.
\end{aligned}
$$

By using $H(\Delta_0) = 1$, $H(\Delta_\infty) = 0$ and properties of the entropy functional, we write the denominator of (2.13) as

$$H\big(\rho'^{\boxast}(\mathtt{X}(z)) \boxast \mathtt{X}'(z)^{\boxast 2}\big) = -\rho'(1-X(z))X'(z)^2.$$

For the numerator of (2.13), we have $\Delta E = W_{\mathrm{BEC}}(x_{\mathrm{BP}}) - W_{\mathrm{BEC}}(0)$, where the single system potential on the BEC is obtained from (2.4) by using, again, the properties in (1.41) and (2.39). The exercise yields

$$W_{\mathrm{BEC}}(x) = \frac{1}{R'(1)}(1-R(1-x)) - x\rho(1-x) - \frac{\epsilon}{L'(1)}L(1-\rho(1-x)).$$

Putting together these results, the formula in (2.13) becomes

$$v_{\mathrm{BEC}} = -\frac{W_{\mathrm{BEC}}(x_{\mathrm{BP}}) - W_{\mathrm{BEC}}(0)}{\int_{\mathbb{R}} \mathrm{d}z\, \rho'(1-X(z))X'(z)^2}. \tag{2.40}$$

(Note that with our normalizations $W_{\mathrm{BEC}}(0) = 0$ for all ϵ.) The erasure profile $X(z)$ has to be computed from the one-dimensional integral equation

$$X(z) - v_{\mathrm{BEC}}X'(z) = \epsilon \int_0^1 \mathrm{d}u\, \lambda\Big(1 - \int_0^1 \mathrm{d}s\, \rho(1 - X(z-u+s))\Big). \tag{2.41}$$

The velocity vanishes when $\epsilon \to \epsilon_{\mathrm{MAP}}$ because $W_{\mathrm{BEC}}(x_{\mathrm{BP}}) \to W_{\mathrm{BEC}}(0) = 0$ in that case.

An important quantity is the slope of the velocity at ϵ_{MAP}, that can serve as a linear approximation for the velocity close to this value of the channel parameter. To compute it, we remark that W_{BEC} has an explicit dependence on ϵ, as well as an implicit one through $x_{\mathrm{BP}}(\epsilon)$. Thus,

$$
\begin{aligned}
\frac{\mathrm{d}W_{\mathrm{BEC}}}{\mathrm{d}\epsilon} &= \frac{\partial W_{\mathrm{BEC}}}{\partial \epsilon} + \frac{\partial W_{\mathrm{BEC}}}{\partial x_{\mathrm{BP}}}\frac{\mathrm{d}x_{\mathrm{BP}}}{\mathrm{d}\epsilon} \\
&= \frac{\partial W_{\mathrm{BEC}}}{\partial \epsilon} \\
&= -\frac{1}{L'(1)}L(1-\rho(1-x_{\mathrm{BP}})),
\end{aligned}
$$

so that for $\epsilon \to \epsilon_{\mathrm{MAP}}$, the first-order Taylor expansion of the single potential yields

$$W_{\mathrm{BEC}}(x_{\mathrm{BP}}) \approx -(\epsilon - \epsilon_{\mathrm{MAP}})\frac{1}{L'(1)}L(1-\rho(1-x_{\mathrm{MAP}})).$$

(Note that we used $x_{\mathrm{BP}}(\epsilon) \to x_{\mathrm{BP}}(\epsilon_{\mathrm{MAP}}) = x_{\mathrm{MAP}}$ when $\epsilon \to \epsilon_{\mathrm{MAP}}$, where x_{MAP} is defined as the non-zero point at which the potential is stationary and vanishes.) This yields the linear approximation for the velocity

$$v_l = \frac{(\epsilon - \epsilon_{\mathrm{MAP}})}{L'(1)} \frac{L(1 - \rho(1 - x_{\mathrm{MAP}}))}{\int_{\mathbb{R}} \mathrm{d}z\, \rho'(1 - X_{\mathrm{MAP}}(z))(X'_{\mathrm{MAP}}(z))^2}, \tag{2.42}$$

where X_{MAP} is the erasure probability profile obtained when $\epsilon = \epsilon_{\mathrm{MAP}}$.

It is interesting to compare (2.40) with the upper bound of Theorem 1 in [98] for a *discrete* system

$$v_{\mathrm{B}} = \alpha \frac{W_{\mathrm{BEC}}(x_{\mathrm{BP}}) - W_{\mathrm{BEC}}(0)}{\sum\limits_{z \in \mathbb{Z}} \rho'(1 - x_z)(x_z - x_{z-1})^2}, \qquad \alpha \le 2. \tag{2.43}$$

In [98], the derivation of the bound yields $\alpha \le 2$ (for L_c and w large enough), but it is conjectured, based on numerical simulations, that $\alpha = 1$ would be a tight bound. It is clear that (2.43) and (2.40) are consistent. We note, for the purpose of reference, that another upper bound is derived in [98], namely

$$v_{B_2} = \frac{\alpha(W_{\mathrm{BEC}}(x_{\mathrm{BP}}; \epsilon) - W_{\mathrm{BEC}}(0; \epsilon))}{2W_{\mathrm{BP}}(x_u; \epsilon) - W_{\mathrm{BP}}(x_{\mathrm{BP}}; \epsilon)},$$

where x_u and x_{BP} are, respectively, the non-trivial unstable and stable fixed points of the potential of the uncoupled system W_{BEC}. We do not discuss this bound in detail because it turns out to be a very loose bound in practice.

We now compare the results of the *analytical* formula for the velocity (2.40) with the *empirical* velocity (called v_e below) that is obtained by simulations (running the discrete DE recursion). We show that the formula provides a very good approximation for the (real) empirical value of the velocity, even for relatively small values of the window size w.

For the simulations, we consider the spatially coupled $(3, 6)$- and $(4, 6)$-regular code ensembles, as well as two irregular LDPC codes (that we describe later). We run the simulations for several values of the chain length $L_c = 256, 1024$ and the window size $w = 3, 5, 8, 16$. The empirical velocity is the velocity calculated from erasure probability profiles of the discrete DE recursion (2.11). Consider two (discrete) profiles $\underline{x}^{(t_1)}$ and $\underline{x}^{(t_2)}$ at any two iterations t_1 and t_2, respectively, with $t_1 < t_2$. After the transient phase is over the profiles are identical up to translation. We call a "kink" the part of the profile where there is a fast increase from 0 to x_{BP} in the erasure probability. The kink "position" is the coordinate such that the height is equal to $x_{\mathrm{BP}}/2$, and Δz is the difference of two such positions (at times t_1 and t_2). Then, the empirical velocity v_e is defined as

$$v_e = \frac{\Delta z}{w(t_2 - t_1)}. \tag{2.44}$$

In practice, we obtain reliable results by taking pairs of profiles separated by 20 iterations and averaging the ratio above over every consecutive pair of profiles. Note that we normalize the velocity by w to be able to compare systems with different window widths.

In Table 2.1, we give the empirical values v_e of the normalized velocities for the spatially coupled $(4, 6)$-regular code ensemble with transmission over the BEC(0.6), when the spatial length is $L_c = 1024$ and the channel parameter is fixed to $\epsilon = 0.6$ (between the BP and MAP thresholds), for different values of the window size w. We observe that the result of our formula v_{BEC} serves as a good estimate of the empirical velocity v_e for all the demonstrated values of the window size. We also observe that the linear approximation serves as a good estimate when the channel parameter is not too far from the MAP threshold ϵ_{MAP}. The upper bound v_B [98], however, serves as a better estimate as the window size grows larger.

Table 2.1: Normalized velocities for LDPC(x^4, x^6) on the BEC with a spatial length $L_c = 1024$, for several w sizes, and $\epsilon = 0.6$. The values in the table can be compared to $v_{\text{BEC}} = 0.0333$ and $v_l = 0.0293$.

	$w = 3$	$w = 5$	$w = 8$	$w = 16$
v_e	0.0325	0.0335	0.0337	0.0339
v_B/α	0.0473	0.0410	0.0380	0.0356

In Table 2.2, we give the empirical values v_e of the normalized velocities for the spatially coupled $(3,6)$-regular code ensemble with transmission over the BEC(ϵ), when the spatial length is $L_c = 1024$ and the window size is $w = 8$, for different values of the channel parameter ϵ. One can compare these values with those in [98] (up to a factor equal to w due to the normalization). We find that the result of the formula for v_{BEC} gives the closest estimate to the empirical velocity v_e for all values of ϵ.

Table 2.2: Normalized velocities for the LDPC(x^3, x^6) on the BEC, spatial length $L_c = 1024$, $w = 8$, and several ϵ values.

	$\epsilon = 0.45$	$\epsilon = 0.46$	$\epsilon = 0.47$	$\epsilon = 0.48$
v_e	0.0667	0.0458	0.0267	0.0117
v_{BEC}	0.0660	0.0449	0.0272	0.0115
v_l	0.0506	0.0373	0.0240	0.0108
v_B/α	0.0781	0.0541	0.0332	0.0142
v_{B_2}/α	0.6970	0.5008	0.3068	0.1291

Figures 2.3 and 2.4 show the empirical velocity v_e, the analytical velocity v_{BEC}, and the upper bound v_B for the spatially coupled $(3,6)$-regular code ensemble, with chain length $L_c = 256$ and window size $w = 3$. We remark that our formula approximates the empirical velocity very well, for all values of the channel parameter $\epsilon \in [\epsilon_{\text{BP}}, \epsilon_{\text{MAP}}] = [0.43, 0.488]$. As or the $(4,6)$ code, we find good agreement between the two values for more than half of the interval $[\epsilon_{\text{BP}}, \epsilon_{\text{MAP}}] \approx [0.515, 0.719]$.

We also illustrate the results for two irregular code ensembles in Figures 2.5 and 2.6. The first one has node degree distributions $L(x) = 0.3x^2 + 0.6x^3 + 0.1x^5$ and $R(x) = x^4$, spatial length $L_c = 1024$, and window size $w = 4$. The agreement between v_{BEC} and v_e is excellent for the whole range $\epsilon \in [\epsilon_{\text{BP}}, \epsilon_{\text{MAP}}] = [0.657, 0.719]$. The second one has $L(x) = 0.4x^3 + 0.3x^4 + 0.3x^5$ and $R(x) = 0.5x^8 + 0.5x^{12}$, spatial length $L_c = 256$, and window size $w = 3$. The agreement between the velocities is also very good for most of the range $\epsilon \in [\epsilon_{\text{BP}}, \epsilon_{\text{MAP}}] = [0.311, 0.385]$.

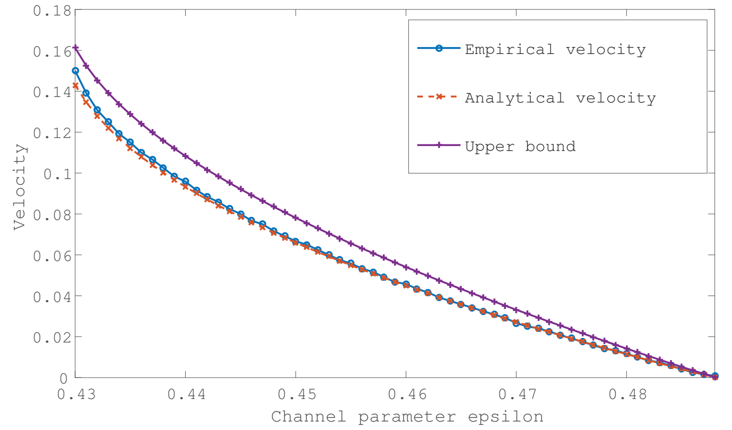

Figure 2.3: Normalized velocities v_e, v_{BEC}, and v_{B}/α (in the order of the legend) for the $(3,6)$-regular ensemble with spatial length $L_c = 256$, window size $w = 3$, and $\epsilon_{\mathsf{BP}} = 0.43 < \epsilon < \epsilon_{\mathsf{MAP}} = 0.4881$.

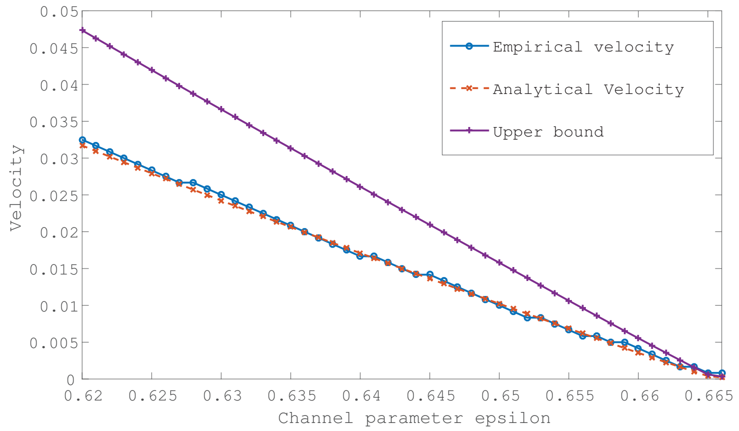

Figure 2.4: Normalized velocities v_e, v_{BEC}, and v_{B}/α (in the order of the legend) of the decoding profile for the $(4,6)$-regular ensemble of spatial length $L_c = 256$, window size $w = 3$, and $\epsilon_{\mathsf{BP}} = 0.515 < \epsilon < \epsilon_{\mathsf{MAP}} = 0.666$.

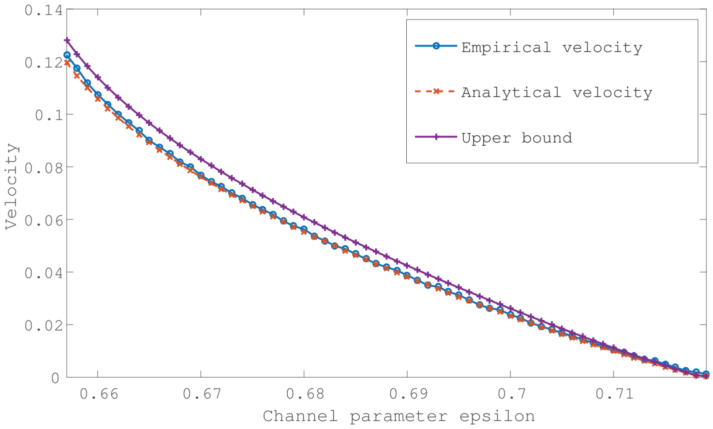

Figure 2.5: Normalized velocities v_{BEC}, v_e, and v_B/α for an ensemble with $L(x) = 0.3x^2 + 0.6x^3 + 0.1x^5$, $R(x) = x^4$ of spatial length $L_c = 1024$ and window size $w = 4$. Here, $\epsilon \in [\epsilon_{\mathsf{BP}}, \epsilon_{\mathsf{MAP}}] = [0.657, 0.719]$.

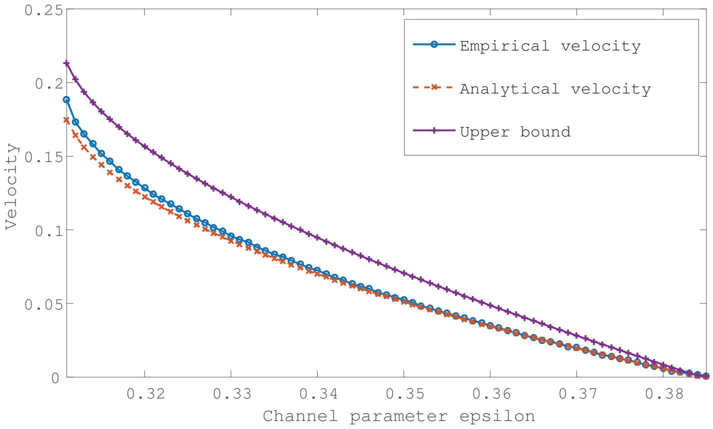

Figure 2.6: Normalized velocities v_{BEC}, v_e, v_B/α for an ensemble with $L(x) = 0.4x^3 + 0.3x^4 + 0.3x^5$, $R(x) = 0.5x^8 + 0.5x^{12}$ of spatial length $L_c = 256$ and window size $w = 3$. Here, $\epsilon \in [\epsilon_{\mathsf{BP}}, \epsilon_{\mathsf{MAP}}] = [0.311, 0.385]$.

2.5.2 Gaussian Approximation (GA)

The DE equations relate probability densities to each other, and as such, we need to track an infinite set of parameters (except for the BEC case where the space of densities can be parametrized by a single real number). In many situations, such as large degrees for example, the densities are well approximated by Gaussians. This allows us to project the DE equations down to a low-dimensional space. There are several variants of the Gaussian approximation (see for example [37, 102, 103]), and here we use it in a form called the "reciprocal channel approximation" proposed in [102, 103].

The idea is to assume that the densities of the LLR messages appearing in the DE equations are symmetric Gaussian densities. Such densities take the form

$$\mathrm{d}\mathtt{x}(h) = \frac{\mathrm{d}h}{\sqrt{2\pi\sigma^2}} \exp\Big(- \frac{(h-m)^2}{2\sigma^2} \Big), \tag{2.45}$$

where the mean m and variance σ^2 satisfy $\sigma^2 = 2m$. Furthermore, the channel density $\mathtt{c}$ is replaced by that corresponding to a BIAWGNC(σ_n^2) with the same entropy $H(\mathtt{c})$. Density evolution can then conveniently be expressed in terms of the entropies $p_z^{(t)} = H(\mathtt{x}_z^{(t)})$. This is done as follows. Let $\psi(m)$ denote the entropy of a symmetric Gaussian density of mean m given by[6]

$$\psi(m) = \frac{1}{\sqrt{4\pi m}} \int_{\mathbb{R}} \mathrm{d}z \, e^{-\frac{(z-m)^2}{4m}} \log_2(1 + e^{-z}). \tag{2.46}$$

Thus, $\psi^{-1}(p)$ denotes the mean of a symmetric Gaussian density $\mathtt{x}$ of entropy $p = H(\mathtt{x})$. Consider two symmetric Gaussian densities $\mathtt{x}_1$ and $\mathtt{x}_2$ with means m_1 and m_2 and entropies $p_1 = \psi(m_1)$ and $p_2 = \psi(m_2)$, respectively. Then, we have

$$H(\mathtt{x}_1 \circledast \mathtt{x}_2) = \psi(\psi^{-1}(p_1) + \psi^{-1}(p_2)), \tag{2.47}$$

which just expresses the fact that a usual convolution of two Gaussian densities of means m_1 and m_2 is a Gaussian density of mean $m_1 + m_2$.

On the contrary, $\mathtt{x}_1 \boxast \mathtt{x}_2$ is not exactly Gaussian so there is no exact formula to describe its entropy, but the idea here is to preserve the duality rule $H(\mathtt{x}_1 \boxast \mathtt{x}_2) + H(\mathtt{x}_1 \otimes \mathtt{x}_2) = H(\mathtt{x}_1) + H(\mathtt{x}_2)$. Writing this relation as

$$1 - H(\mathtt{x}_1 \boxast \mathtt{x}_2) = H((\Delta_0 - \mathtt{x}_1) \circledast (\Delta_0 - \mathtt{x}_2)),$$

and noting that $H(\Delta_0 - \mathtt{x}_1) = 1 - p_1$, $H(\Delta_0 - \mathtt{x}_2) = 1 - p_2$, we obtain the approximation

$$H(\mathtt{x}_1 \boxast \mathtt{x}_2) = 1 - \psi(\psi^{-1}(1-p_1) + \psi^{-1}(1-p_2)). \tag{2.48}$$

We limit ourselves to regular codes for simplicity. Using the entropy relations in (2.47) and (2.48), we obtain

$$\begin{cases} H(\mathtt{x}^{(t)}) = H(\mathtt{c} \circledast \mathtt{y}^{(t)\circledast(\ell-1)}), \\ H(\mathtt{y}^{(t+1)}) = H(\mathtt{x}^{(t)\boxast(r-1)}). \end{cases} \tag{2.49}$$

---

[6] For indications on the numerical implementation of this function, see [61], pp.194 and 237.

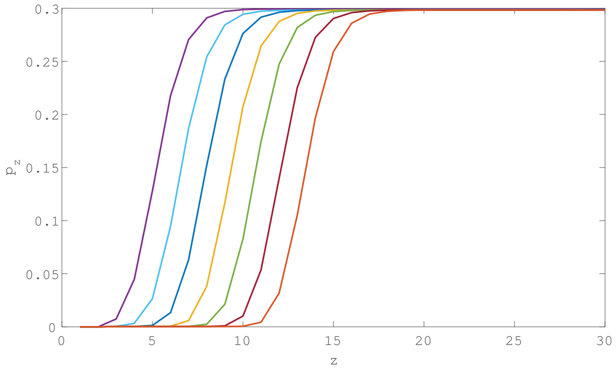

Figure 2.7: The profile of entropies $p(z,t)$ plotted every 10 iterations starting from iteration 20. We take the $(3,6)$-regular LDPC code with spatial length $L_c = 30$ and window size $w = 3$, and we consider the BIAWGN channel with mean $\psi^{-1}(H(\mathtt{c})) = 2.4$.

Setting $p^{(t)} = H(\mathtt{x}^{(t)})$ and $q^{(t)} = H(\mathtt{y}^{(t)})$, we find

$$\begin{cases} p^{(t)} = \psi\big(\psi^{-1}(H(\mathtt{c})) + (\ell-1)\psi^{-1}(q^{(t)})\big), \\ q^{(t+1)} = 1 - \psi\big((r-1)\psi^{-1}(1-p^{(t)})\big). \end{cases} \tag{2.50}$$

These equations can be combined into

$$p^{(t+1)} = \psi\Big((\ell-1)\psi^{-1}\big(1-\psi((r-1)\psi^{-1}(1-p^{(t)}))\big) + \psi^{-1}(H(\mathtt{c}))\Big). \tag{2.51}$$

The corresponding potential function is then easily obtained from (2.4)

$$\begin{aligned} W_{\mathrm{GA}}(p) = {} & \frac{1}{r}\big(1-\psi\big(r\psi^{-1}(1-p)\big)\big) \\ & + \psi\big(r\psi^{-1}(1-p)\big) - \psi\big((r-1)\psi^{-1}(1-p)\big) \\ & - \frac{1}{\ell}\psi\big(\psi^{-1}(H(\mathtt{c})) + \ell\psi^{-1}\big(1-\psi((r-1)\psi^{-1}(1-p)).\big)\big) \end{aligned} \tag{2.52}$$

For the coupled system, we denote by p_z the *average* of the entropy of symmetric Gaussian densities emanating from the variable nodes *over the positions* $z, \dots, z+w$. The coupled DE recursion then takes the form

$$\begin{aligned} p_z^{(t+1)} = \frac{1}{w}\sum_{i=0}^{w-1}\psi\Big((\ell-1)\psi^{-1}\Big(\frac{1}{w}\sum_{j=0}^{w-1}\big(1-\psi((r-1)\psi^{-1}(1-p_{z-i+j}^{(t)}))\big)\Big) \\ + \psi^{-1}(H(\mathtt{c}))\Big). \end{aligned} \tag{2.53}$$

This coupled recursion can be solved with appropriate boundary conditions. As shown in Figure 2.7, it demonstrates the wave propagation phenomenon.

We are now ready to discuss the application of the approximation to the formula for the velocity. The continuum limit is obtained exactly as in Section 2.3.1. The assumption that the density $\mathtt{x}(z,t)$ tends to a fixed shape $\mathtt{X}(z-v_{\mathrm{GA}}t)$ after the transient phase implies that its entropy $p(z,t)$ tends to $P(z-v_{\mathrm{GA}}t) \equiv H(\mathtt{X}(z-v_{\mathrm{GA}}t))$

where $P(z)$ is a scalar function (independent of initial conditions) satisfying the integral equation

$$\begin{aligned} &P(z) - v_{\mathrm{GA}} P'(z) && (2.54) \\ &= \int_0^1 \mathrm{d}u\, \psi\Big(\psi^{-1}(H(\mathtt{c})) + (\ell-1)\psi^{-1}\big(1 - \int_0^1 \mathrm{d}s\, \psi\big((r-1)\psi^{-1}(1 - P(z-u+s))\big)\big)\Big) && (2.55) \end{aligned}$$

and the boundary conditions $\lim_{z\to-\infty} P(z) = 0$, $\lim_{z\to+\infty} = p_{\mathrm{BP}}$, where p_{BP} is the non-trivial fixed point of (2.51). We will now show that the formula for the velocity reduces to

$$v_{\mathrm{GA}} = -\frac{W_{\mathrm{GA}}(p_{\mathrm{BP}}) - W_{\mathrm{GA}}(0)}{(r-1)\int_{\mathbb{R}} \mathrm{d}z\, P'(z)^2 \frac{\psi''((r-2)\psi^{-1}(1-P(z)))}{\psi'(\psi^{-1}(1-P(z)))^2}}. \tag{2.56}$$

To derive (2.56) we consider the denominator in (2.13) and write it as

$$\begin{aligned} &(r-1)H\big(\mathtt{x}(z)^{\boxast(r-2)} \boxast \mathtt{x}'(z)^{\boxast 2}\big) \\ &\quad = (r{-}1)\lim_{\delta\to 0}\frac{1}{\delta^2}\Big\{H\big(\mathtt{x}(z)^{\boxast(r-2)} \boxast \mathtt{x}(z+\delta)^{\boxast 2}\big) \\ &\qquad - 2H\big(\mathtt{x}(z)^{\boxast(r-2)} \boxast \mathtt{x}(z+\delta) \boxast \mathtt{x}(z)\big) + H\big(\mathtt{x}(z)^{\boxast(r-2)} \boxast \mathtt{x}(z)^{\boxast 2}\big)\Big\} \end{aligned} \tag{2.57}$$

By computing each entropy in the Gaussian approximation, we find for the bracket on the right-hand side

$$\begin{aligned} \frac{1}{\delta^2}\Big\{&\big[1 - \psi((r-2)\psi^{-1}(1-p(z)) + 2\psi^{-1}(1-p(z+\delta)))\big] \\ &- 2\big[1 - \psi((r-1)\psi^{-1}(1-p(z)) + \psi^{-1}(1-p(z+\delta)))\big] \\ &+ \big[1 - \psi(r\psi^{-1}(1-p(z)))\big]\Big\} \end{aligned} \tag{2.58}$$

In Appendix 2.7.3, we compute the limit of this term when $\delta \to 0$ by appropriate Taylor expansions and find

$$-\left(\frac{\mathrm{d}P(z)}{\mathrm{d}z}\right)^2 \frac{\psi''\big((r-2)\psi^{-1}(1-P(z))\big)}{\big(\psi'(\psi^{-1}(1-P(z)))\big)^2}. \tag{2.59}$$

This concludes the derivation of the formula in (2.56).

Table 2.3 gives a comparison of the analytical and empirical velocities v_{GA} and $v_{e,\mathrm{GA}}$, respectively, that are obtained for the $(3,6)$- and the $(4,8)$-regular ensembles, for a spatial length of $L_c = 100$ and $w = 3$ for different values of $\psi^{-1}(H(c)) = \sigma_n^2/2$ (twice the signal to noise ratio). We also plot both velocities for the $(3,6)$-regular ensemble for the same parameters in Figure 2.8. We conjecture that the errors incurred are due to numerical errors involved in computing the functions ψ and its inverse.

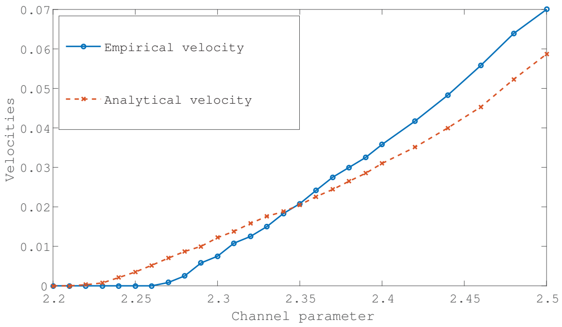

Figure 2.8: Normalized velocities v_{GA} and $v_{e,\text{GA}}$ for the spatially coupled $(3,6)$-regular ensemble within the Gaussian approximation framework, for a spatial length of $L_c = 100$ and window size $w = 3$, as a function of $\psi^{-1}(H(\mathtt{c})) = 2/\sigma_n^2$.

Table 2.3: Normalized velocities of the profiles within the Gaussian approximation on the $(3,6)$- and $(4,8)$-regular code ensembles with $L_c = 100$, $w = 3$.

$2/\sigma_n^2$	**2.33**	**2.35**	**2.38**	**2.40**
v_{GA} $(3,6)$	0.0176	0.0205	0.0265	0.0310
v_e $(3,6)$	0.0150	0.0208	0.0300	0.0358
v_{GA} $(4,8)$	0.0237	0.0258	0.0312	0.0381
v_e $(4,8)$	0.0217	0.0250	0.0308	0.0342

2.6 Application to Scaling Laws for Finite-Length Coupled Codes

The authors in [96] propose a scaling law to predict the error probability of a finite-length spatially coupled (ℓ, r, L_c) code when transmission takes place over the BEC. The derived scaling law depends on *scaling parameters*, one of which we will relate to the velocity of the decoding wave. The (ℓ, r, L_c) ensemble considered in [96] differs slightly from the purely random ensemble we consider in this thesis. However, as we will see, our formula for the velocity yields results that are reasonably good for this application. We briefly describe this ensemble and the scaling law.

The (ℓ, r, L_c) ensemble combines the benefits of purely random codes (that we consider in this thesis) and protograph-based codes [85]. The randomness involved in the construction makes the ensemble relatively easy to analyze, and the structure added to the construction due to its similarity to protograph-based code improves the performance of the code. The ensemble is constructed as follows: Make $L_c + w$ copies of an uncoupled code at positions $z = -w + 1, \ldots, L_c$. All edges are erased then reconnected such that a variable node at position z_0 has exactly one edge with each set of check nodes at positions $z_0 + i$, where $i = 0, \ldots, \ell - 1$. The check nodes are chosen such that the regularity of their degree is maintained. Therefore, every variable node has ℓ emanating edges and every check node has r such edges.

We consider transmission over the BEC. In this case, the BP decoder can be seen as a peeling decoder [104]. Whenever a variable node is decoded, it is removed from the graph along with its edges. One way to track this peeling process is to analyze the evolution of the degree distribution of the residual graph across iterations; this serves as a sufficient statistic. This statistic can be described by a system of differential equations, whose solution determines the mean and variance of the *fraction of degree-one check nodes* and the variance around this mean at any time during the decoding process. We denote by $\hat{r}_1$ the mean.

It has been shown in [96] that there exists a *steady state phase* where the mean and the variance are constant. It is exactly during this phase that one can observe the solitonic behavior.[7]

Let $\epsilon_{(\ell,r,L_c)}$ denote the BP threshold of the finite-size (ℓ, r, L_c) ensemble (for large values of L_c, this is close to ϵ_{MAP} due to threshold saturation). We can write the first-order Taylor expansion of $\hat{r}_1\big|_{\epsilon}$ around $\epsilon < \epsilon_{(\ell,r,L_c)}$ as

$$\hat{r}_1\big|_{\epsilon} \approx \hat{r}_1\big|_{\epsilon_{(\ell,r,L_c)}} + \gamma\Delta\epsilon + \mathcal{O}(\Delta\epsilon^2),$$

where $\Delta\epsilon = \epsilon_{(\ell,r,L_c)} - \epsilon$. Thus, for a given $\epsilon < \epsilon_{(\ell,r,L_c)}$ and by using $\hat{r}_1\big|_{\epsilon_{(\ell,r,L_c)}} = 0$ (by definition), we obtain $\gamma \approx \hat{r}_1\big|_{\epsilon}/\Delta\epsilon$. The parameter γ enters in the scaling law and is therefore of importance. In previous work, γ was determined experimentally. It would clearly be desirable to have a theoretical handle on γ. It is argued in [96] that $\gamma \approx \bar{\gamma}$, where $\bar{\gamma} = x_{\mathrm{BP}}/c$ and c is a real positive constant that behaves like $\Delta\epsilon/v_{\mathrm{BEC}}$, i.e.,

$$\bar{\gamma} \approx \frac{x_{\mathrm{BP}}\, v_{\mathrm{BEC}}}{\Delta\epsilon}. \tag{2.60}$$

It is expected that this formula becomes exact in an asymptotic limit $\epsilon_{(\ell,r,L_c)} \to \epsilon_{\mathrm{MAP}}$, where threshold saturation takes place. Using the linearization (2.42), we obtain

$$\bar{\gamma} \to \frac{x_{\mathrm{MAP}}}{L'(1)} \frac{L(1-\rho(1-x_{\mathrm{MAP}}))}{\int_{\mathbb{R}} \mathrm{d}z\, \rho'(1-X_{\mathrm{MAP}}(z))(X'_{\mathrm{MAP}}(z))^2}. \tag{2.61}$$

The parameter $\bar{\gamma}$ is simply equal to the erasure probability multiplied by the slope of the velocity at ϵ_{MAP}.

We compare the values of γ and $\bar{\gamma}$ for different values of ℓ and r, at a channel parameter $\epsilon = \epsilon_{(\ell,r,L_c)} - 0.04$, in Table 2.4. The experimental values of γ are taken from [96], and to obtain the values of $\bar{\gamma}$, we use the analytical velocity (2.40). We observe that the numbers roughly agree. There are two reasons that can explain the discrepancies. First, we use the velocity for the purely random spatially coupled graph ensemble whereas the ensemble considered in [96] is more structured. Note also that as ℓ, r increase, the window size of the structured ensemble increases so the finite size effects at fixed spatial length $L_c = 100$ may be more marked. Second, the expression (2.60) is valid when $\epsilon \to \epsilon_{(\ell,r,L_c)}$, whereas in Table 2.4, $\Delta\epsilon = 0.04$ which is relatively large (this choice in [96] is due to stability issues in numerical integration techniques when $\epsilon \to \epsilon_{(\ell,r,L_c)}$). We conjecture that the second issue is the dominant reason for the difference between the values of γ and $\bar{\gamma}$, and that in fact the velocity for the structured ensemble is not very different from the one predicted by the analytical formula (2.40).

---

[7] In this thesis, we consider one-sided termination instead of two-sided termination (as considered in [96]), so the fraction $\hat{r}_1$ here is equal to half the fraction $\hat{r}_1(*)$ in [96].

Table 2.4: Values of γ and $\bar{\gamma}$ for $\epsilon_{(\ell,r,L_c)} - \epsilon = 0.04$, $L_c = 100$, and several values of ℓ and r

ℓ	r	ϵ_{MAP}	γ	$\bar{\gamma}$
3	6	0.4881	2.155	1.960
4	8	0.4977	2.120	1.779
5	10	0.4994	2.095	1.733
6	12	0.4999	2.075	1.722
4	12	0.3302	2.140	1.778
5	15	0.3325	2.115	1.746
4	6	0.6656	2.100	1.735

2.7 Appendix

In this section, we show how the equations (2.21), (2.32), and (2.34) are derived using functional derivatives.

2.7.1 Derivation of Equation (2.21)

We calculate the functional derivative of $\Delta\mathcal{W}(\mathtt{x})$ in a direction $\eta(z,t)$ as follows,

$$\frac{\delta\Delta\mathcal{W}}{\delta\mathtt{x}}[\eta(z,t)] = \frac{\partial}{\partial\gamma}\Delta\mathcal{W}(\mathtt{x}+\gamma\eta)\Big|_{\gamma=0} = \frac{\partial}{\partial\gamma}\int_{\mathbb{R}}\mathrm{d}z\,P(z,\mathtt{x}+\gamma\eta)\Big|_{\gamma=0},$$

where the function P is defined in (2.17). Then, taking the derivative with respect to γ yields

$$\begin{aligned}\int_{\mathbb{R}}\mathrm{d}z\Big\{&H(\rho^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)) + H(\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t))\\ &- H(\rho^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)) - H(\mathtt{x}(z,t)\boxast\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t))\\ &- H\Big(\mathsf{c}(z)\circledast\lambda^{\circledast}\Big(\int_0^1\mathrm{d}s\,\rho^{\boxast}(\mathtt{x}(z+s,t))\Big)\\ &\qquad\circledast\Big[\int_0^1\mathrm{d}u\,\rho'^{\boxast}(\mathtt{x}(z+u,t))\boxast\eta(z+u,t)\Big]\Big)\Big\}.\end{aligned}$$

We notice that the first and third terms in the integral cancel out due to the commutativity of the operator $\boxast$. By rearranging the averaging functions in the last term, we obtain

$$\begin{aligned}\int_{\mathbb{R}}\mathrm{d}z\Big\{&H(\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)) - H(\mathtt{x}(z,t)\boxast\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t))\\ &- H\Big(\int_0^1\mathrm{d}u\,\mathsf{c}(z-u)\circledast\lambda^{\circledast}\Big(\int_0^1\mathrm{d}s\,\rho^{\boxast}(\mathtt{x}(z-u+s,t))\Big)\\ &\qquad\circledast\Big[\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)\Big]\Big)\Big\}.\end{aligned}$$

By noticing that $\mathtt{y} = \int_0^1\mathrm{d}u\,\mathsf{c}(z-u)\circledast\lambda^{\circledast}(\int_0^1\mathrm{d}s\,\rho^{\boxast}(\mathtt{x}(z-u+s,t)))$ is a probability measure and $\mathtt{a} = \rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)$ is a difference of probability measures, we can use the second duality rule in (2.2) (that is $H(\mathtt{y}\boxast\mathtt{a})+H(\mathtt{y}\circledast\mathtt{a})=H(\mathtt{a})$) in order to rewrite the above as (freely using the commutativity of $\boxast$)

$$\begin{aligned}
&\int_{\mathbb{R}} \mathrm{d}z \Big\{ -H(\mathtt{x}(z,t)\boxast\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)) \\
&\qquad\qquad + H\Big(\Big[\int_0^1 \mathrm{d}u\, \mathtt{c}(z-u)\circledast\lambda^{\circledast}\Big(\int_0^1 \mathrm{d}s\,\rho^{\boxast}(\mathtt{x}(z-u+s,t))\Big)\Big] \\
&\qquad\qquad\qquad \boxast\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t)\Big)\Big\} \\
&= \int_{\mathbb{R}} \mathrm{d}z\,\rho'^{\boxast}(\mathtt{x}(z,t))\boxast\eta(z,t \\
&\qquad \boxast\Big(\int_0^1 \mathrm{d}u\,\mathtt{c}(z-u)\circledast\lambda^{\circledast}\Big(\int_0^1 \mathrm{d}s\,\rho^{\boxast}(\mathtt{x}(z-u+s,t))\Big)-\mathtt{x}(z,t)\Big).
\end{aligned}$$

2.7.2 Derivation of Equation (2.32)

We calculate the functional derivative of $\mathcal{W}_s(\mathtt{X})$ in a direction $\mathtt{X}'(z)$ as follows,

$$\begin{aligned}
&\frac{\delta\mathcal{W}_s}{\delta\mathtt{X}}[\mathtt{X}'(z)] = \frac{\partial}{\partial\gamma}\mathcal{W}_s(\mathtt{X}+\gamma\mathtt{X}')\Big|_{\gamma=0} \\
&= \frac{\partial}{\partial\gamma}\int_{\mathbb{R}} \mathrm{d}z\,P_s(z,\mathtt{X}+\gamma\mathtt{X}')\Big|_{\gamma=0} \\
&= \int_{\mathbb{R}} \mathrm{d}z\,\Big\{H(\rho^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)) + H(\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)) \\
&\qquad - H(\rho^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)) - H(\mathtt{X}(z)\boxast\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)) \\
&\qquad - H(\mathtt{c}(z)\circledast\lambda^{\circledast}(\rho^{\boxast}(\mathtt{X}(z)))\circledast[\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)])\Big\}.
\end{aligned}$$

We notice here that on the right-hand side of last equality, the first and third terms under the integral cancel out. Using the second duality rule in (2.2), and noticing that $\mathtt{X}(z)$ is a probability measure and $\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)$ is a difference of probability measures, we can rewrite the functional derivative as

$$\begin{aligned}
&\int_{\mathbb{R}} \mathrm{d}z\,\Big\{H(\mathtt{X}(z)\circledast[\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)]) \\
&\qquad - H(\mathtt{c}(z)\circledast\lambda^{\circledast}(\rho^{\boxast}(\mathtt{X}(z)))\circledast[\rho'^{\boxast}(\mathtt{X}(z))\boxast\mathtt{X}'(z)])\Big\}.
\end{aligned}$$

2.7.3 Derivation of Expression (2.59)

Our goal in this section is to show that (2.58) reduces to (2.59) when $\delta\to 0$. We first reorganize (2.58) as follows (up to multiplication by $1/\delta^2$)

$$\begin{aligned}
&\psi\Big((r-2)\psi^{-1}(1-p(z))+2\psi^{-1}(1-p(z+\delta))\Big) \\
&\quad - 2\psi\Big((r-2)\psi^{-1}(1-p(z))+\psi^{-1}(1-p(z+\delta)) \\
&\quad + \psi^{-1}(1-p(z))\Big) + \psi\Big((r-2)\psi^{-1}(1-p(z))+2\psi^{-1}(1-p(z))\Big).
\end{aligned}$$

When we Taylor expand each entropy $\psi(\cdots)$ around $(r-2)\psi^{-1}(1-p(z))$ to the second order, we observe that the first order terms cancel and what remains is

$$-\frac{1}{2}\psi''\big((r-2)\psi^{-1}(1-p(z))\big)\Big\{4\psi^{-1}(1-p(z+\delta))^2 \\ -4\psi^{-1}(1-p(z))^2-2\big(\psi^{-1}(1-p(z+\delta))-\psi^{-1}(1-p(z))\big)^2\Big\}.$$

This is equal to

$$-\psi''\big((r-2)\psi^{-1}(1-p(z))\big)\Big(\psi^{-1}(1-p(z+\delta))-\psi^{-1}(1-p(z))\Big)^2.$$

Next, we write $\psi^{-1}(1-p(z+\delta))$ as

$$\psi^{-1}(1-p(z)-(p(z+\delta)-p(z))),$$

and Taylor expand $\psi^{-1}(\cdots)$ around $1-p(z)$ to obtain

$$-(p(z+\delta)-p(z))^2((\psi^{-1})'(1-p(z)))^2\psi''\big((r-2)\psi^{-1}(1-p(z))\big).$$

Multiplying by $1/\delta^2$, taking the limit $\delta\to 0$, and using the relation $(\psi^{-1})'(\cdots)=1/(\psi'(\psi^{-1}(\cdots)))$ we finally obtain

$$-\left(\frac{\mathrm{d}p(z)}{\mathrm{d}z}\right)^2\frac{\psi''\big((r-2)\psi^{-1}(1-p(z))\big)}{\Big(\psi'(\psi^{-1}(1-p(z)))\Big)^2}.$$

Propagating Wave Velocity for General Scalar Systems

3

3.1 Introduction

Although spatial coupling was first introduced and used in the context of coding [44,45,77], it was later applied to several other problems, such as compressive sensing [52–54], random constraint satisfaction problems [55,56], and a coupled Curie-Weiss (toy) model [59,60]. For all these problems, the spatially coupled systems exhibit the "threshold saturation" phenomenon: The dynamic threshold of the coupled system saturates to the static threshold of the underlying uncoupled system [71]. In the context of coding, the dynamic (or algorithmic) and static thresholds are the belief propagation (BP) and maximum a-posteriori (MAP) thresholds, respectively. That is, the dynamic threshold is the noise level up to which the BP decoder can fully decode, and the static threshold is that up to which the optimal (MAP) decoder can decode. In the present chapter, we consider spatially coupled *general scalar systems* governed by a set of iterative message-passing equations.[1] The precise definitions of the dynamic and static thresholds, then, depend on the application and are related, respectively, to the performance of the message-passing algorithm and the "optimal solver". We formalize the setting further in Section 3.2.

As usual, the spatially coupled version of such general scalar systems is obtained by taking some, say L_c+w, copies of the underlying (uncoupled) system on the spatial axis $z = -w+1, \dots, L_c$, and coupling every w adjacent such systems by means of a coupling-window function. The performance of the message-passing algorithm on the coupled system can be expressed by means of a DE-like recursion that updates a vector $\underline{x} = (x_{-w+1}, \dots, x_{L_c})$; we call this vector the "profile". In the context of coding over the binary erasure channel (BEC), for instance, the z^{th} component of the profile is the average erasure probability of the bits of the copy on the z^{th}

---

[1] The content of this chapter is based on previous work [74,97].

[74] R. El-Khatib, N. Macris, *"The velocity of the propagating wave for general coupled scalar systems,"* Information Theory Workshop (ITW) 2016, Cambridge, UK, IEEE pp. 246–250.

[97] R. El-Khatib, N. Macris, *"The velocity of the propagating wave for spatially coupled systems with applications to LDPC codes,"* in preparation (2016) to be submitted to the IEEE Transactions on Information Theory.

position of the coupling chain. For general scalar systems, the interpretation of the z^{th} component of the vector depends on the application. We call the *desirable* or *trivial* fixed point the vector profile obtained by running the DE-like recursion when the initial condition on the profile is "perfect" information. A *non-trivial* fixed point can be obtained by running the recursion with different initial conditions or "bad" information; we say that it is *undesirable*. In the present chapter, we assume that we have a unique non-trivial stable fixed point.

Threshold saturation can be viewed in terms of the trivial (desirable) and non-trivial (undesirable) fixed points: The profile associated with an uncoupled system converges to the trivial fixed point for system parameters that are up to the dynamic/algorithmic threshold, and to the non-trivial fixed point otherwise; whereas the profile associated with a coupled system converges to the trivial fixed point for system parameters up to the static threshold that is *higher* than the dynamic threshold. For the general scalar systems that we consider, we can think of the system parameter as a level of noise or distortion; the real interpretation depends on the application. For coding on the BEC(ϵ), for example, this is the channel parameter ϵ. In the case of compressive sensing, the parameter is the inverse measurement ratio (the length of the signal divided by the number of measurements made to reconstruct it).

As with coding, we seed a few copies at the boundary of the coupled chain with perfect information, and the latter is propagated into the chain upon iterations of the message-passing algorithm. When the system parameter is between the algorithmic and static thresholds, and after a few "transient" iterations of the recursion, the profile exhibits a *solitonic behavior*; that is, it develops a *fixed shape* and propagates with a *constant velocity*. In Figure 3.1, we plot the profile for the compressive sensing problem every 20 iterations (up till iteration 160) and observe the "wave propagation" phenomenon.[2] In this chapter, we find an analytical formula for the velocity of the propagating wave, assuming that the solitonic behavior is indeed exhibited by the coupled system profile.

The solitonic behavior exhibited by the propagating wave has already been studied in the context of coding when transmission takes place over the BEC. In [69,70], the authors prove the existence of a solitonic wave solution for coding on the BEC and derive bounds on the velocity. In [98], the authors provide bounds on the velocity for coding schemes that might have more than two stable fixed points (in contrast with the systems we consider). In addition, a formula for the velocity in the context of the coupled Curie-Weiss toy model was derived in [53], along with some approximations for it.

To derive the formula for the velocity of the wave, we follow the steps taken in Chapter 2. We consider the coupled system in the continuum limit $L_c \gg w \gg 1$ in order to make the derivations more tractable. We also make use of the (continuous) coupled potential functional that is simply the variational formulation of the message-passing recursion; that is, its stationary points are the fixed points of the recursion.

This chapter is organized as follows. In Section 3.2, we introduce the notation and preliminaries that we need and briefly describe the wave phenomenon exhibited

---

[2] We can see, in this example, that the solitonic behavior is already exhibited starting at the 20^{th} iteration of the recursion.

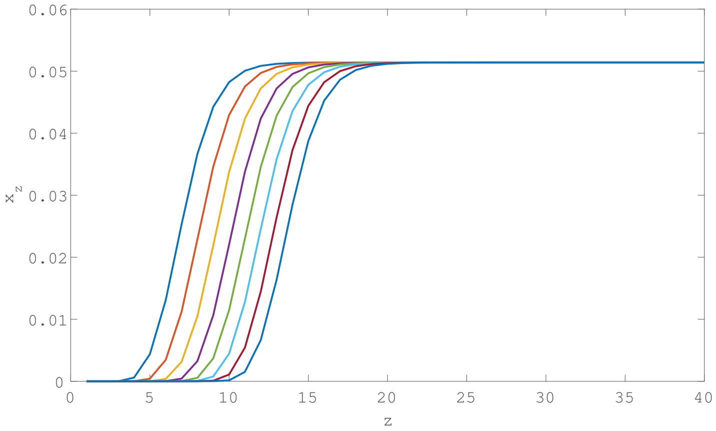

Figure 3.1: We consider the compressive sensing problem with $\mathtt{snr} = 10^5$, $\epsilon = 0.1$, and $\delta = 0.17$. The coupling parameters are the chain length $L_c = 40$ and the window size $w = 3$. We plot here the profile $x(z,t)$ as a function of the spatial position z at iterations t, every 20 iterations, starting from $t = 20$ till $t = 160$. We notice that the profile is flat almost everywhere and rises abruptly from $x_{\text{good}} = 0$ to $x_{\text{bad}} \simeq 0.05$.

by the soliton. In Section 3.3, we further describe the system in the continuum limit and state the main result. In Section 3.4, we show the derivation of the main result. Finally, in Section 3.5, we provide numerical simulations for GLDPC codes on the BEC and BSC, as well as for compressive sensing.

3.2 Preliminaries

We adopt the framework and notation in [71]. We denote by $\epsilon \in [0, \epsilon_{\max}]$, where $\epsilon_{\max} \in (0, \infty)$, the interval of values for the control parameter ϵ. Consider bounded, smooth functions that are increasing in both their arguments $g : [0, x_{\max}(\epsilon)] \times [0, \epsilon_{\max}] \to [0, y_{\max}(\epsilon)]$ and $f : [0, y_{\max}(\epsilon)] \times [0, \epsilon_{\max}] \to [0, x_{\max}(\epsilon)]$, where $x_{\max}(\epsilon)$, $y_{\max}(\epsilon) \in (0, \infty)$ and $y_{\max}(\epsilon) = g(x_{\max}(\epsilon); \epsilon)$. The scalar recursions that interest us are of the form

$$x^{(t+1)} = f(g(x^{(t)}; \epsilon); \epsilon), \tag{3.1}$$

where $t \in \mathbb{N}$ is the iteration number. The recursion is initialized with $x^{(0)} = x_{\max}$. As $f(g([0, x_{\max}(\epsilon)])) \subset [0, x_{\max}(\epsilon)]$, the initialization of (3.1) implies that $x^{(1)} \leq x^{(0)} = x_{\max}$ and more generally $x^{(t+1)} \leq x^{(t)}$. Thus, $x^{(t)}$ will converge to a limiting value $x^{(\infty)}$, and this limit is a fixed point because f and g are continuous. The fixed points of the recursion (3.1) can be described as stationary points of a *single system* potential function U_s defined as

$$U_s(x) = xg(x; \epsilon) - G(x; \epsilon) - F(g(x; \epsilon); \epsilon), \tag{3.2}$$

where $F(x; \epsilon) = \int_0^x \mathrm{d}s\, f(s; \epsilon)$ and $G(x; \epsilon) = \int_0^x \mathrm{d}s\, g(s; \epsilon)$. Without loss of generality, this function is normalized so that $U_s(x) = 0$.

We define $x_{\text{good}}(\epsilon)$ as the fixed point of (3.1) that is reached when x is initialized so that $x^{(0)} = 0$ ("perfect" information). Furthermore, the *algorithmic threshold*[3] is defined as

$$\epsilon_a = \sup\{\epsilon \mid x^{(\infty)} = x_{\text{good}}\}. \tag{3.3}$$

The monotonicity of f and g implies that, for $\epsilon < \epsilon_a$, the basin of attraction of x_{good} is the whole interval $[0, x_{\max}(\epsilon)]$. Moreover, x_{good} is the unique stationary point of the potential function, and it is a minimum because it is an attractive fixed point. For $\epsilon > \epsilon_a$, we have $x^{(\infty)} \neq x_{\text{good}}$ and we denote this fixed point by $x^{(\infty)} = x_{\text{bad}}$. Note that this is an attractive fixed point and is thus a (local) minimum of $U_s(x)$. The two attractive fixed points are separated by at least one unstable fixed point x_{unst} that is a local maximum of $U_s(x)$. We henceforth assume that there does not appear any other fixed points besides x_{good}, x_{unst}, x_{bad}. With this assumption in mind, we define an *energy gap* as

$$\Delta E = U_s(x_{\text{bad}}) - U_s(x_{\text{good}}), \tag{3.4}$$

and the *potential threshold* as the unique value ϵ_{pot} such that $\Delta E = 0$ (it can be shown that ΔE in non-increasing in ϵ).

The corresponding spatially coupled recursions are obtained by placing $L_c + w$ replicas of the single system on the spatial positions $z = -w + 1, \dots, L_c$ and coupling them with a uniform coupling window of size w. The coupled recursion takes the form

$$x_z^{(t+1)} = \frac{1}{w} \sum_{j=0}^{w-1} f\Big(\frac{1}{w} \sum_{k=0}^{w-1} g(x_{z-j+k}^{(t)}; \epsilon); \epsilon\Big). \tag{3.5}$$

Motivated by the phenomenology observed in many examples (e.g. for the BEC or for compressive sensing), we fix the boundary conditions as $x_z^{(t)} = x_{\text{good}}$, for $z = \{-w+1, \dots, 0\}$ and all $t \in \mathbb{N}$ in order to study the stationary phase, during which a soliton appears. The initialization of the recursion is $x_z^{(0)} = x_{\text{bad}}$ for $z = \{0, \dots, L_c + w - 1\}$. The corresponding *potential functional* is given by

$$U_c(\underline{x}) = \sum_{z=-w+1}^{L_c} \big(x_z g(x_z; \epsilon) - G(x_z; \epsilon)\big) - \sum_{z=-w+1}^{L_c} F\Big(\frac{1}{w} \sum_{i=0}^{w-1} g(x_{z+i}; \epsilon); \epsilon\Big), \tag{3.6}$$

where $\underline{x} = (x_{-w+1}, \dots, x_{L_c})$. The fixed point equation (3.5) can be obtained by setting to zero the derivative of the potential $U_c(\underline{x})$ with respect to $\underline{x}$.

The spatially coupled recursions (3.5) display the *threshold saturation* property. Namely, for all $\epsilon < \epsilon_{\text{pot}}$ the fixed point $x_z^{(\infty)}$, $z = -w+1, \dots, L_c$, of the recursion (3.5) is equal to a constant profile $\underline{x}_{\text{good}} = (x_{\text{good}}, \dots, x_{\text{good}})$.

In the remainder of the chapter, we consider the range $\epsilon \in [\epsilon_a, \epsilon_{\text{pot}}]$. It is for these values of the parameter ϵ that a soliton propagating at finite speed is observed, after a transient phase that last for only a few iterations.

---

[3] The algorithmic threshold depends on the governing message-passing algorithm.

3.2.1 Phenomenological Observations

We consider the case when the control parameter satisfies $\epsilon \in [\epsilon_a, \epsilon_{\text{pot}})$. We consider coupled systems whose single potentials have exactly two minima and one maximum for this range of values of ϵ. Equivalently, we consider systems whose underlying system recursions have exactly two stable fixed points and one unstable fixed point.

We recall that the coupled DE equation governing the spatially coupled system is given by (3.5). The left boundary is fixed to perfect information; that is, $x_z^{(t)} = x_{\text{good}}$ for $z = -w+1, \dots, 0$ and all $t \in \mathbb{N}$. This is the "seed" that causes the propagation of perfect information into the coupled chain. The rest of the coupled chain is initialized with $x_z^{(0)} = x_{\text{bad}}$ for $z = 1, \dots, L_c + w - 1$.

Just as in coding (Chapter 2), we observe a solitonic behavior once we apply the scalar recursion (3.5). That is, with iterations of the recursion, we notice a wave-like phenomenon: the profile $\underline{x}$ appears to have a fixed shape that is moving with a constant velocity. We call this profile a wave or a soliton. The soliton is a kink with a front at position z_{front}, making a quick transition of width $O(2w)$, between the two values $x_z^{(t)} \approx x_{\text{good}}$ for $z << z_{\text{front}}$ and $x_z^{(t)} \approx x_{\text{bad}}$ for $z >> z_{\text{front}}$.

The simplest example to keep in mind for all of the above setting, as well as the next paragraph, is the case of LDPC(λ, ρ) codes with transmission over the BEC(ϵ) where $f(x;\epsilon) = \epsilon\lambda(x)$ and $g(x;\epsilon) = 1-\rho(1-x)$ and $U_s(x)$ is equal to (2.10). For this example, $\epsilon_a = \epsilon_{\text{BP}}$, $\epsilon_{\text{pot}} = \epsilon_{\text{MAP}}$, $x_{\text{good}} = 0$ and $x_{\text{bad}} = x_{\text{BP}}$ is the non-trivial stable BP fixed point.

3.3 Continuum Limit and Main Result

3.3.1 Continuum Limit

We consider the system in the limit $L_c \gg w \gg 1$ and formulate a continuum approximation. The coupled recursion (3.5) becomes

$$x(z, t+1) = \int_0^1 \mathrm{d}u\, f\Big(\int_0^1 \mathrm{d}s\, g(x(z-u+s, t); \epsilon); \epsilon\Big). \tag{3.7}$$

We take the boundary condition $x(z,t) \to x_{\text{good}}$ when $z \to -\infty$ and $x(z,t) \to x_{\text{bad}}$ when $z \to +\infty$. This boundary condition captures the profiles obtained after the transient phase has passed, and is well adapted to the study of the soliton propagation.

3.3.2 Statement of Main Result

As before, we assume that there exists a constant velocity $v > 0$ such that, for $t \to +\infty$ and $|z - vt| = O(1)$, the profile $x(z,t) \to X(z - vt)$, where $X(z)$ is independent of the initial condition $x(z, 0)$ and satisfies $\lim_{z \to -\infty} X(z) = x_{\text{good}}$, $\lim_{z \to +\infty} X(z) = x_{\text{bad}}$. Under this assumption, the velocity of the soliton is

$$v = \frac{\Delta E}{\int_{\mathbb{R}} \mathrm{d}z\, g'(X(z);\epsilon) X'(z)^2}, \tag{3.8}$$

where the shape $X(z)$ satisfies

$$X(z) - vX'(z) = \int_0^1 \mathrm{d}u\, f\Big(\int_0^1 \mathrm{d}s\, g(X(z-u+s); \epsilon); \epsilon\Big). \tag{3.9}$$

3.4 Derivation of Main Result

The derivation of the formula in (3.8) closely follows the one in Section 2.4.2, so we will be quite brief. The first step is to introduce a continuum version of $U_c(\underline{x})$, that we call $\Delta\mathcal{U}_c(x)$.[4] We define $x_0(z)$ as a static (time-independent) profile that satisfies the boundary conditions $x_0(z) \to x_{\text{good}}$ when $z \to -\infty$ and $x_0(z) \to x_{\text{bad}}$ when $z \to +\infty$ (for example, one may take a Heaviside-like step function). This is a reference profile that we need so that we obtain well-defined integrals in the following expression

$$\Delta\mathcal{U}_c(x) = \int_{\mathbb{R}} \mathrm{d}z \Big\{ P_c(z,x) - P_c(z,x_0) \Big\},$$

where $P_c(z,x)$ is a z-dependent functional of x equal to

$$P_c(z,x) = x(z,t)g(x(z,t);\epsilon) - G(x(z,t);\epsilon) - F\Big(\int_0^1 \mathrm{d}u\, g(x(z+u,t);\epsilon);\epsilon\Big).$$

As long as $x(z,t)$ and $x_0(z)$ converge to their limiting values fast enough, the integrals over the spatial axis are well-defined. Evaluating the functional derivative of $\Delta\mathcal{U}_c(x;\epsilon)$ in an arbitrary direction η, defined as $\lim_{\gamma\to 0}\gamma^{-1}(\Delta\mathcal{U}_c(x+\gamma\eta)-\Delta\mathcal{U}_c(x))$, we find that (3.7) is equivalent to a gradient descent equation

$$\int_{\mathbb{R}} \mathrm{d}z\, g'(x(z,t);\epsilon)\big(x(z,t+1) - x(z,t)\big)\eta(z,t) = -\frac{\delta\Delta\mathcal{U}_c}{\delta X}[\eta(z,t)]. \tag{3.10}$$

We first consider the left-hand side of (3.10). We use the ansatz $x(z,t) \to X(z-vt)$ and consider the special direction $\eta(z,t) = X'(z-vt)$ to analyze (3.10). Using the approximation $X(z-vt) \approx X(z) - vX'(z)$ for small v, we obtain (after a change of variables $z \to z+vt$)

$$v\int_{\mathbb{R}} \mathrm{d}z\, X'(z)^2 g'(X(z);\epsilon) = \frac{\delta\Delta\mathcal{U}_c}{\delta x}[X'(z)]. \tag{3.11}$$

We proceed to compute the right-hand side of (3.10). The potential functional can be divided into two parts: the "single system potential" $\mathcal{U}_s(x)$ that remains if we ignore the coupling effect, and the interaction potential $\mathcal{U}_i(x)$ that captures the effect of coupling. That is, $\Delta\mathcal{U}_c = \mathcal{U}_s + \mathcal{U}_i$, with

$$\begin{aligned}\mathcal{U}_s(x) = \int_{\mathbb{R}} \mathrm{d}z \Bigg\{ &\Big[x(z,t)g(x(z,t);\epsilon) - G(x(z,t);\epsilon) - F(g(x(z,t);\epsilon)\Big] \\ &- \Big[x_0(z)g(x_0(z);\epsilon) - G(x_0(z);\epsilon) - F(g(x_0(z);\epsilon);\epsilon)\Big]\Bigg\},\end{aligned}$$

$$\begin{aligned}\mathcal{U}_i(x) = \int_{\mathbb{R}} \mathrm{d}z \Bigg\{ &\Big[F(g(x(z,t);\epsilon);\epsilon) - F(\int_0^1 \mathrm{d}u\, g(x(z-u,t);\epsilon)\Big] \\ &- \Big[F(g(x_0(z);\epsilon);\epsilon) - F\Big(\int_0^1 \mathrm{d}u\, g(x_0(z-u);\epsilon);\epsilon\Big)\Big]\Bigg\}.\end{aligned}$$

---

[4] We have already defined this potential functional in (1.74) but will repeat it here for convenience.

The computation of each functional derivative at $x(z,t) \to X(z-vt)$ in the direction $X'(z-vt)$ yields

$$\begin{aligned}\frac{\delta \mathcal{U}_s}{\delta X}[X'(z)] &= \int_{\mathbb{R}} \mathrm{d}z\, X'(z)\Big(X(z)g'(X(z);\epsilon) - g'(X(z);\epsilon)f(g(X(z);\epsilon);\epsilon)\Big) \\ &= \int_{\mathbb{R}} \mathrm{d}z\, \frac{\mathrm{d}}{\mathrm{d}z}\Big\{X(z)g(X(z);\epsilon) - G(X(z);\epsilon) - F(g(X(z);\epsilon);\epsilon)\Big\} \\ &= \Big[U_s(X(z))\Big]_{-\infty}^{+\infty} \\ &= U_s(x_{\mathrm{bad}};\epsilon) - U_s(x_{\mathrm{good}};\epsilon), \end{aligned} \tag{3.12}$$

and

$$\begin{aligned}\frac{\delta \mathcal{U}_i}{\delta X}[X'(z)] &= \int_{\mathbb{R}} \mathrm{d}z\, X'(z)\Big\{f(g(X(z);\epsilon);\epsilon)g'(X(z)) \\ &\qquad - f\Big(\int_0^1 \mathrm{d}u\, g(X(z-u);\epsilon);\epsilon\Big)\int_0^1 \mathrm{d}u\, g'(X(z-u);\epsilon)\Big\} \\ &= \int_{\mathbb{R}} \mathrm{d}z \frac{\mathrm{d}}{\mathrm{d}z}\Big\{F(g(X(z);\epsilon);\epsilon) - F\Big(\int_0^1 \mathrm{d}u\, g(X(z-u);\epsilon);\epsilon\Big)\Big\} \\ &= 0. \end{aligned} \tag{3.13}$$

Replacing (3.12) and (3.13) in (3.10) we obtain the formula for the velocity (3.8).

3.5 Applications and Numerical Experiments

The general formula for the velocity of the soliton for scalar systems (3.8) can be applied on several examples. In particular, we recover the results of Chapter 2 for standard LDPC codes over the BEC, as well as on general BMS channels within the scalar Gaussian approximation. In this section, we provide two more scalar applications, namely GLDPC codes and compressive sensing.

The predictions of our formula are compared with the observed, empirical velocity v_e that is obtained by running the scalar recursions. To obtain v_e, we plot the discrete profile $\underline{x}$ at different iterations of the scalar recursions and find the average of $\Delta z/(w\Delta I)$, where Δz is the spatial difference between the kinks of the profiles, ΔI is the difference in the number of iterations, and w is the size of the coupling window. We note that we normalize all velocities by the window size w so that we can compare the velocities of systems that have different such sizes.

3.5.1 Generalized LDPC (GLDPC) Codes

A GLDPC code is a code represented by a bipartite graph, where the rules of the check nodes do not depend on parity (as do usual LDPC codes) but on a primitive BCH code. An attractive property of BCH codes is that they can be designed to correct a chosen number of errors. For instance, one can design a BCH code so that it corrects all patterns of at most e erasures on the BEC, and all error patterns of weight at most e on the binary symmetric channel (BSC). We consider a GLDPC code with degree-2 variable nodes and degree-n check nodes, where the rules of the latter are given by a primitive BCH code of blocklength n.

We give a short description of a BCH code of blocklength n and minimum distance $d = 2e + 1$ (for more details, see [105]). A BCH code is a cyclic code over a finite field GF(b^β), where b is a prime power and β is an integer. Let a be a primitive element of GF(b^β). Then, each element of GF(b^β) can be written in the form a^i, $i \in \mathbb{N}$. For each element a^i, we can define a minimal polynomial m_i that is the monic polynomial over GF(b) with smallest degree. The generator polynomial θ over GF(b) of the BCH code is defined as the least common multiple of $m_1, \dots, m_d$.

Consider transmission on the BEC or BSC and denote by ϵ the channel parameter. The density evolution recursions have been derived in [106] for both channels, based on a bounded distance decoder for the BCH code. For n and e fixed, we can write the update equations of the message-passing algorithm as (3.1) with

$$\begin{cases} f(x;\epsilon) = \epsilon x, \\ g(x) = \sum_{i=e}^{n-1} \binom{n-1}{i} x^i (1-x)^{n-i-1}. \end{cases}$$

Here, we have $\epsilon_{\max} = x_{\max} = y_{\max} = 1$. Moreover, we can easily check, by differentiation with respect to x, that the potential function $U_{\mathrm{GLDPC}}(x)$ of the system is given by

$$U_{\mathrm{GLDPC}}(x) = \frac{e}{n} g(x;\epsilon) - \frac{g'(x;\epsilon)}{n\, x(1-x)} - \frac{\epsilon}{2} g^2(x;\epsilon).$$

For numerical implementation purposes, it is useful to note that

$$g'(x) = \frac{x^{e-1}(1-x)^{n-e-1}}{B(e, n-e)},$$

where $B(a,b) = \frac{(a-1)!(b-1)!}{(a+b-1)!}$ denotes the Beta function, and that $g(x)$ is equal to the regularized incomplete Euler Beta function so that

$$g(x) = \frac{1}{B(e,n-e)} \int_0^x \mathrm{d}s\, s^{e-1}(1-s)^{n-e-1}.$$

This potential has $x = 0$ as a trivial stationary point[5] and develops a non-trivial minimum at $x_{\mathrm{BP}} \neq 0$ when $\epsilon > \epsilon_{\mathrm{BP}}$. Note that, as usual, the channel parameter ϵ_{BP} is the value for which the first horizontal inflexion point appears. The MAP threshold is given by the channel paramter ϵ_{MAP} for which $U_{\mathrm{GLDPC}}(x_{\mathrm{BP}}) = U_{\mathrm{GLDPC}}(0) = 0$.

The formula for the velocity of the soliton in the case of coupled GLDPC codes is obtained from (3.8). The energy gap for $\epsilon_{\mathrm{BP}} < \epsilon < \epsilon_{\mathrm{MAP}}$ is now defined as $\Delta E = U_{\mathrm{GLDPC}}(x_{\mathrm{BP}}) - U_{\mathrm{GLDPC}}(0)$. Figure 3.2 shows the velocities (normalized by w) for the spatially coupled GLPDC code with $n = 15$ and $e = 3$, when the coupling parameters satisfy $L_c + w = 500$ and $w = 3$. We plot the velocities for $\epsilon \in [\epsilon_{\mathrm{BP}}, \epsilon_{\mathrm{MAP}}] = [0.348, 0.394]$. We observe that the formula for the velocity provides a very good estimation of the empirical velocity v_e.

3.5.2 Compressive Sensing

Let $\mathbf{s}$ be a length-n signal vector where the components are i.i.d. copies of a random variable S. We take m linear measurements of the signal and assume that the

---

[5] Equivalently, $x = 0$ is a trivial fixed point of DE as can be seen from the expressions of f and g.

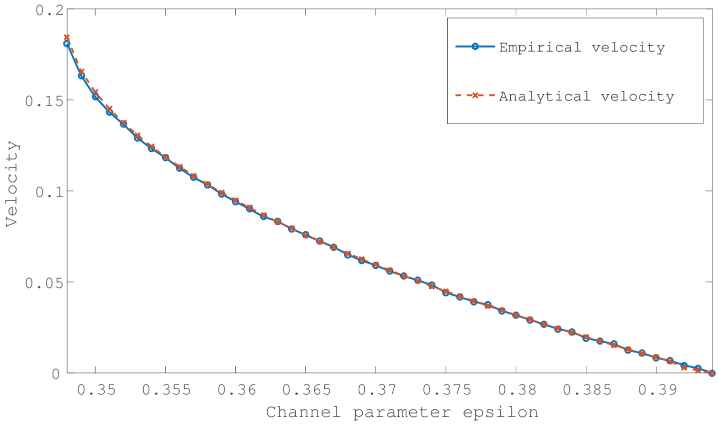

Figure 3.2: We consider a GLDPC code with $n = 15$ and $e = 3$, with spatial length $L_c + w = 500$ and uniform coupling window with $w = 3$. We plot the normalized velocities v_{GLDPC} and v_e as a function of the channel parameter ϵ when ϵ is between the BP threshold $\epsilon_s = \epsilon_{\text{BP}} = 0.348$ and the potential threshold $\epsilon_c = \epsilon_{\text{MAP}} \approx 0.394$.

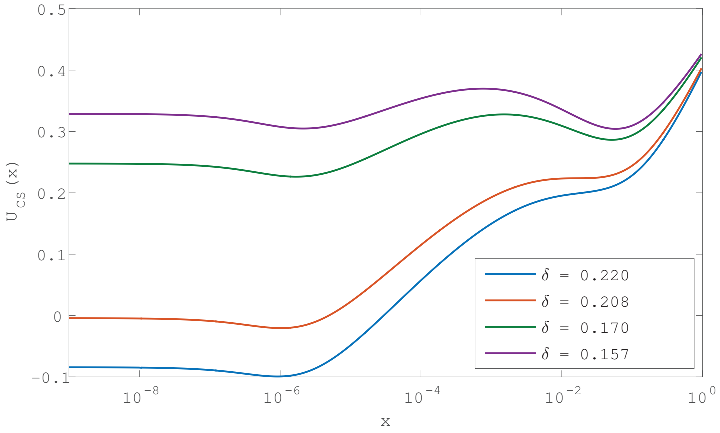

Figure 3.3: Potential function for compressive sensing with a Gaussian-Bernoulli prior. The sparsity parameter is $\rho = 0.1$ and the signal to noise ratio $\texttt{snr} = 10^5$. We show the potential for several values of the measurement fraction δ. For $\delta > \delta_{\text{AMP}}$ we have a single minimum x_{good}. At $\delta_{\text{AMP}} = 0.205$ there is a horizontal inflexion point and for smaller measurement fractions a second minimum x_{bad} appears. At $\delta_{\text{opt}} = 0.157$ the gap ΔE vanishes.

measurement matrix has i.i.d Gaussian elements distributed like $\mathcal{N}(0, 1/\sqrt{n})$. We define $\delta = m/n$ as the measurement ratio and fix it to a constant value as $n \to \infty$. The relation between δ and the parameter ϵ defined in Section 3.2 is $\epsilon = \delta^{-1}$. We assume that the power of the variable S is normalized to 1; that is, $\mathbb{E}[S^2] = 1$. We also assume that each component of the signal $\mathbf{s}$ is corrupted by independent Gaussian noise of variance $\sigma^2 = 1/\texttt{snr}$. To recover $\mathbf{s}$, one implements the so-called approximate message-passing (AMP) algorithm.

It is well known that the analysis of the AMP algorithm is given by state evolution [54]. Let $Y = \sqrt{\texttt{snr}}\, S + Z$, where $Z \sim \mathcal{N}(0,1)$, and let $\hat{S}(Y) = \mathbb{E}_{S|Y}[S|Y]$ be the minimum mean square estimator. We set

$$\texttt{mmse}(\texttt{snr}) = \mathbb{E}_{S,Y}[(S - \hat{S}(Y, \texttt{snr}))^2],$$

for the `mmse` function. The state evolution equations (that track the mean squared error of the AMP estimate) then correspond to the recursion (3.1) with

$$\begin{cases} f(x,\delta) = \texttt{mmse}(\texttt{snr} - x), \\ g(x,\delta) = \texttt{snr} - \frac{1}{\frac{1}{\texttt{snr}} + \frac{x}{\delta}}. \end{cases}$$

Here, x is interpreted as the mean square error predicted by the AMP estimate of the signal. State evolution is initialized with $x = 1$, which corresponds to no knowledge about the signal. We will take δ as the control parameter. Note that we have $\delta_{\max} = 1$, $x_{\max} = \texttt{mmse}(0)$, $y_{\max} = g(x_{\max})$. The potential function is equal to

$$\begin{aligned} U_{\text{CS}}(x) = &-\frac{x}{\frac{1}{\texttt{snr}} + \frac{x}{\delta}} + \delta \ln\Big(1 + \frac{x\,\texttt{snr}}{\delta}\Big) \\ &- 2I\Big(S; \sqrt{\texttt{snr}}S + Z\Big) + 2I\Big(S; \sqrt{\frac{1}{\frac{1}{\texttt{snr}} + \frac{x}{\delta}}}S + Z\Big). \end{aligned}$$

where $I(A;B)$ denotes the mutual information between two random variables A and B. To check that this potential gives back the correct state evolution equation as a stationarity condition, we simply differentiate it with respect to x using to the well-known relation $\frac{\mathrm{d}}{\mathrm{d}\texttt{snr}} I(S; \sqrt{\texttt{snr}}S + Z) = \frac{1}{2}\texttt{mmse}(\texttt{snr})$ [107].

To illustrate the potential function in a concrete case, we take the Bernoulli-Gaussian distribution as the prior distribution over the signal components

$$q_0(s) = (1-\rho)\delta(s) + \rho \frac{e^{-s^2/2}}{\sqrt{2\pi}}.$$

Figure 3.3 shows the potential function for $\rho = 0.1$, $\texttt{snr} = 10^5$, and several values of δ (the measurement fraction). We observe that for $\delta > \delta_{\text{AMP}} = 0.205$, there is a unique minimum x_{good} that is a fixed point of state evolution once the recursion is initialized with $x = 1$. During this phase, there are enough measurements so that the reconstruction of the signal is good and the mean square error is small. At $\delta_{\text{AMP}} = 0.205$, a horizontal inflexion point develops in the potential function. For $\delta < \delta_{\text{AMP}}$, a second minimum appears at a higher mean square error x_{bad} and the reconstruction of the AMP algorithm is bad. The optimal threshold δ_{opt} corresponding to the minimum mean square error estimator is found when δ is such that the two minima of the potential are at the same height. More specifically, δ_{opt} is given by the solution of the equation $U_{\text{CS}}(x_{\text{bad}}) = U_{\text{CS}}(x_{\text{good}})$. For our example, one finds $\delta_{\text{opt}} = 0.157$. This threshold is reached by the AMP algorithm on the spatially coupled system.

We fix the value of δ in the range $[\delta_{\text{opt}}, \delta_{\text{AMP}}] = [0.157, 0.205]$. In this regime, the solution of the spatially coupled state evolution equations develops a solitonic behavior; this represents the profile of mean square errors along the spatial direction. The formula for the velocity v_{CS} of this soliton is obtained from (3.8), where the energy gap is now defined as $\Delta E = U_{\text{CS}}(x_{\text{bad}}) - U_{\text{CS}}(x_{\text{good}})$. Figure 3.4 shows the

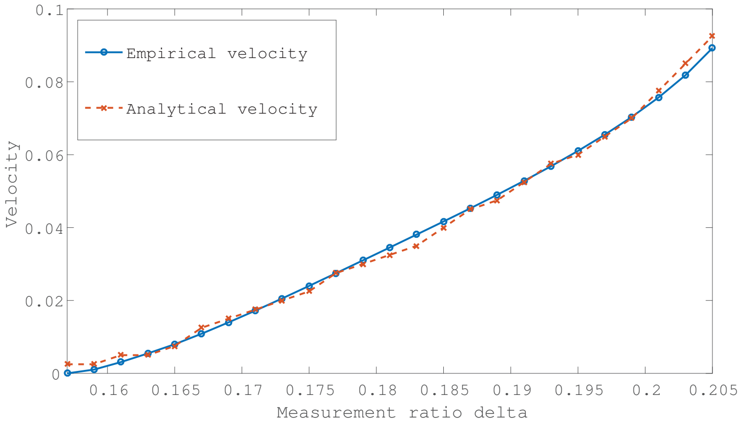

Figure 3.4: We consider the compressive sensing problem with $\texttt{snr} = 10^5$ and Gaussian-Bernoulli prior for the signal components with sparsity parameter $\rho = 0.1$. We have $L_c + w = 250$ and uniform coupling window with $w = 4$. We plot the normalized velocities v_{CS} and v_e as a function of the measurement fraction δ when δ is between the potential threshold $\delta_{opt} = 0.157$ and $\delta_{\mathsf{AMP}} = 0.205$.

velocities (normalized by w) for the spatially coupled compressive sensing system with $\texttt{snr} = 10^5$, $\rho = 0.1$ when the coupling parameters satisfy $L_c + w = 250$ and $w = 4$. In Figure 3.4, we plot the velocities for $\delta \in [\delta_{\text{opt}}, \delta_{\text{AMP}}] = [0.157, 0.205]$. It is clear that the formula for the velocity provides a good estimation of the empirical velocity v_e.

Displacement Convexity for (ℓ, r)-regular Codes on the BEC

4

4.1 Introduction

Spatially coupled codes have been proved to be universally capacity-achieving under the low-complexity belief propagation (BP) decoding algorithm. This property is attributed to the “threshold saturation” phenomenon that describes the fact that the spatially coupled ensembles generally have a BP threshold that is as high as the maximum-a-posteriori (MAP) threshold of the ensemble, as well as that of its underlying ensemble; i.e., their threshold is *saturated* to the highest possible value. In the context of coding, this result was proved for transmission over the binary erasure channel (BEC) [41, 69, 70, 100] and for transmission over general binary-input memoryless symmetric-output (BMS) channels [43, 63].

In this chapter, we introduce one further tool for the analysis of such systems: the concept of *displacement convexity* [92, 93].[1] Displacement convexity plays a crucial role in the theory of optimal transport and, as we show in the present chapter and in Chapter 5, it also serves as a useful tool for the analysis of spatially coupled graphical models. We use the simple case of regular LDPC Gallager ensembles and transmission over the BEC to explain how the concept of displacement convexity can help us simplify existing proofs and derive new results. In Chapter 5, we provide a more general analysis by using displacement convexity for general spatially coupled systems (not restriced to coding) that are governed by a set of scalar density-evolution (DE) equations.

We analyze the performance of the BP algorithm on the spatially coupled codes by using a DE recursion that tracks the bit erasure probabilities along the coupling chain. This recursion can be expressed in variational form by using the potential functional, where the fixed points of the DE recursion are the stationary points of

---

[1] The content of this chapter is based on previous work [72, 108].
[72] R. El-Khatib, N. Macris, R. Urbanke, *“Displacement convexity, a useful framework for the study of spatially coupled codes,”* Information Theory Workshop (ITW) 2013, Sevilla, Spain, IEEE pp. 1–5.
[108] R. El-Khatib, N. Macris, R. Urbanke, *“Displacement convexity, a useful framework for the study of spatially coupled codes,”* 2013, http://arxiv.org/abs/1304.6026.

the potential. Therefore, finding the stable fixed points of the DE recursion, which we call the "profiles", becomes equivalent to finding the minima of the potential. Furthermore, we consider the coupled system in the continuum limit, in which we first take the coupling length L_c, and then the window size w, to infinity. This enables us to express the DE recursion and the potential functional in the continuous space, the natural setting for the study of displacement convexity. For the (ℓ, r)-regular Gallager ensembles that we consider in this chapter, the potential functional is not convex *in the traditional sense*. However, we show that it is *displacement convex*; that is, it is convex with respect to an alternative structure in the space of probability measures. Roughly speaking, these are the probability measures associated with erasure probability profiles viewed as cumulative distribution functions (cdf's).

We consider the static case: when the decoder phase-transition threshold is equal to the MAP threshold and the BP decoder gets "blocked".[2] The displacement convexity property plays a fundamental role in characterizing the set of minimizing profiles in a suitable space of *increasing profiles*. For our case, we obtain strict displacement convexity once a global translational degree of freedom of the profiles is removed. These results enable us to conclude that, in a suitable space of increasing profiles, the minimizer of the potential functional is unique up to translation, therefore, so is the solution of the DE equation.

Our goal is to find the stable fixed point(s) of the DE equation or, equivalently, the minimizer(s) of the potential. In this chapter, we use displacement convexity to do the latter. We start by looking at the minimization problem in a more general space of profiles $\mathcal{S}$ that contains general continuous profiles (that are not necessarily non-decreasing) with an additional constraint: their limits at $-\infty$ and $+\infty$ on the continuous spatial axis are equal to the trivial and non-trivial BP fixed point values, respectively.[3] We further define the subset $\mathcal{S}'$ of $\mathcal{S}$ that contains non-decreasing such profiles. We prove that we can restrict our search for the minimizer(s) to the space $\mathcal{S}'$. The first step in doing so is to prove that truncating the profile at the value of the non-trivial BP fixed point would yield a smaller potential value. We then use rearrangement inequalities to show that rearranging the mass of a profile, so that the resultant profile is non-decreasing, would also decrease the potential value. This result is crucial for our derivations, as it enables us to view the profiles at hand as cumulative distribution functions (cdf's) – up to scaling – and this is the natural setting for the analysis of displacement convexity.[4] We then define the subset $\mathcal{S}''$ of $\mathcal{S}'$ that contains strictly increasing profiles. We furthermore restrict the search of minimizers to the space $\mathcal{S}''$ by showing that if the minimizer is non-decreasing, then it is necessarily strictly increasing.

An important question to ask is whether a minimizer for the potential functional *exists* or, in other words, whether the potential attains its minimum. By using the

---

[2] We know that for regular LDPC codes with transmission over the BEC, the DE recursion of the underlying uncoupled code has exactly two stable fixed points: we called them the trivial (desirable) and the non-trivial (undesirable) fixed points. At these values, the potential function of the uncoupled code is zero because we consider the static phase.

[3] We assume in this chapter that the profiles reach the limits "fast enough", so that the integrals in which we use them are well-defined. In Chapter 5, however, we work around this assumption.

[4] It is not clear if and when the rearrangement inequalities are strict. As a result, the analysis falls short of establishing that there cannot exist profiles in $\mathcal{S}\backslash\mathcal{S}'$ that are minimizers and solutions of the DE equation.

direct method in the calculus of variations [109, 110], we prove that the potential functional does indeed achieve its minimum. Furthermore, we show that over the space $\mathcal{S}$, the potential attains its minimum in the subspace $\mathcal{S}''$.

Once we restrict our search of minimizing profiles to the space $\mathcal{S}''$ (in which we prove that the potential functional indeed attains its minimum), we prove that the potential is *strictly displacement convex*. To do this, we consider any two profiles y_0 and y_1 (that can be viewed as cdf's) and define their displacement interpolant y_λ that is obtained from the two original profiles by taking the linear interpolation of their inverses and then taking the inverse back into the space of profiles. The displacement interpolant is formally defined in (4.24). We then make use of a result by Villani [93] to prove that the potential functional is indeed displacement convex. As the potential is invariant upon translations on the profile it takes as an argument, we prove that the potential is strictly convex, once we remove the profile's translational degree of freedom (by centering it at the origin).

The three main results of this chapter are: (i) Theorem 4.15 that states that the potential functional attains its minimum in the space of strictly increasing profiles, and that it does not have a minimum in the space of profiles that are non-decreasing (but not strictly increasing); (ii) Theorem 4.2 that establishes the displacement convexity of the potential in the space of non-decreasing profiles, and its strict displacement convexity in the space of centered strictly increasing ones; and finally (iii) Corollary 4.3 that concludes that any minimizer of the potential in the space of non-decreasing profiles are translates of the unique minimizer that exists in the space of *centered* strictly increasing profiles. This corollary establishes that when the BEC parameter is equal to the MAP threshold, the DE equation governing the spatially coupled (ℓ, r)-regular Gallager ensemble has a unique solution in the space of centered strictly increasing profiles, and all its other minimizers in the space of non-decreasing profiles are translates of it. To establish these results, we prove intermediate ones such as the restriction of the space of minimizers to "truncated" profiles in Lemma 4.5 and furthermore to non-decreasing profiles in Lemma 4.6 and strictly increasing ones in Lemma 4.10.

The chapter is organized as follows. In Section 4.2, we introduce the framework for our analysis and our main results. In Section 4.5.1, we give a quick introduction of the notion of displacement convexity. Finally, in Section 4.4, we present a proof of existence of the profile that minimizes the potential, and in Section 4.5, we prove that the functional is displacement convex.

4.2 Preliminaries

In this section, we introduce the model and the associated variational problem to which we apply the displacement convexity proof technique, and we state our main result.

4.2.1 Single System

We consider the (ℓ, r)-regular Gallager code ensemble, where every variable node has ℓ outgoing edges, and every check node has r such edges. For the channel, we take a BEC with parameter ϵ. We analyze the asymptotic performance of the system, under BP iterations, by tracking the evolution of the output erasure probability y of

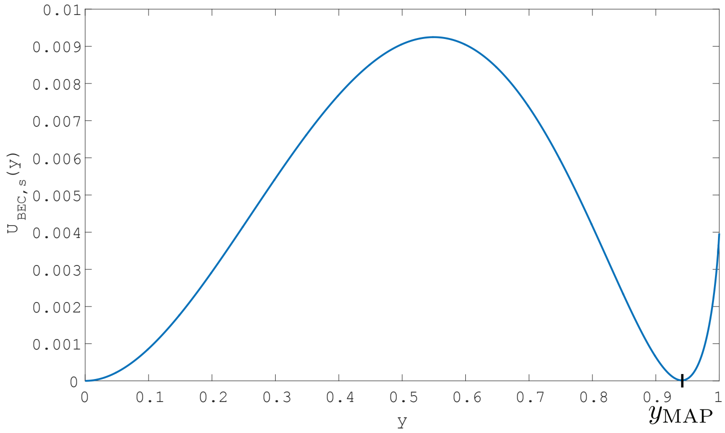

Figure 4.1: The plot of the single system potential $W_s(y)$ as a function of the check-erasure probability y, for a $(3, 6)$-regular uncoupled ensemble and $\epsilon = \epsilon_{\mathsf{MAP}} = 0.4881$. There are two minima at $y = 0$ and $y = y_{\mathrm{MAP}}$.

check nodes. We can thus write the DE recursion in terms of y (explained in more detail in Section 1.4.3) as

$$y^{(t+1)} = 1 - (1 - \epsilon(y^{(t)})^{\ell-1})^{r-1}, \tag{4.1}$$

with initial condition $y^{(0)} = 1$.

To this DE recursion, we associate a potential function, whose stationarity condition yields the fixed points of the recursion. The "single" (uncoupled) potential function is expressed as

$$U_{\mathrm{BEC,s}}(y) = y + \left(1 - \frac{1}{r}\right)\left(1 - (1-y)^{r/(r-1)} - 1\right) - \frac{\epsilon}{\ell} y^{\ell}, \tag{4.2}$$

and can be recovered from the expression in (1.48).

4.2.2 Spatially Coupled System

Consider the spatially coupled (ℓ, r, L_c, w)-regular ensemble, described in detail in [43], where the parameters represent the left degree, right degree, system length, and coupling window size (or smoothing parameter), respectively. More specifically, the ensemble is constructed as follows: consider $L_c + w$ replicas of an (uncoupled) (ℓ, r)-regular ensemble. We couple these components by connecting every variable node to ℓ check nodes, and every check node to r variable nodes. The connections are chosen randomly: for a variable node at position z, each of its ℓ connections is chosen uniformly and independently at random in the range $z, \dots, z + w - 1$, and for a check node at position z, each of its r connections is chosen uniformly and independently in the range $z - w + 1, \dots, z$.

Let $\check{y}_z$ for $z = -w + 1, \dots, L_c$, denote the output erasure probability of the check node at position z. We find it most convenient to formulate the problem with the average over a window in terms of $y_z \equiv \frac{1}{w}\sum_{i=0}^{w-1} \check{y}_{z+k}$; this describes the input erasure probability to a variable node at position z. Then the fixed-point (FP) condition implied by DE is

$$y_z = 1 - \Big(1 - \frac{\epsilon}{w} \sum_{i=0}^{w-1} \big(\frac{1}{w} \sum_{j=0}^{w-1} y_{z-i+j}\big)^{\ell-1}\Big)^{r-1}.$$

This can be seen as a special case of the more general DE recursion expressed in terms of the variable node input erasure probability in (1.61) for the case of the (ℓ, r)-regular Gallager ensemble. The corresponding potential function (adapted from (1.66)) is

$$U_{\mathrm{BEC}}(\underline{y}) = \frac{1}{w} \sum_{z=-w+1}^{L_c} \Big\{ y_z + \Big(1 - \frac{1}{r}\Big)\big((1-y_z)^{\frac{r}{r-1}} - 1\big) - \frac{\epsilon}{\ell}\Big(\frac{1}{w}\sum_{i=0}^{w-1} y_{z+i}\Big)^{\ell} \Big\}, \tag{4.3}$$

where $\underline{y} = (y_{-w+1}, \dots, y_{L_c})$. The normalization $1/w$ is a convenience for which the reason will immediately appear.

4.3 Continuum Limit and Main Results

4.3.1 Continuum Limit

The natural setting for displacement convexity is the continuum case. Therefore, we consider the continuum limit of the potential in (4.3). To obtain the expressions of the DE recursion and the potential in the continuum limit, we define the rescaled variables $\tilde{z} = \frac{z}{w}$, $u = \frac{i}{w}$ and the rescaled function $\tilde{y}(\frac{z}{w}) \equiv y_z$, as we have done in Chapter 1. It is easy to see that with this rescaling, (4.3) becomes a Riemann sum. When we take the limits $L_c \to +\infty$ first and then $w \to +\infty$, we find the following expression for the potential functional in the continuum limit

$$\begin{aligned} \mathcal{W}_{\mathrm{BEC}}(y) = \int_{\mathbb{R}} \mathrm{d}z \Big\{ \Big(1 - \frac{1}{r}\Big)\Big(1 - y(z)\Big)^{\frac{r}{r-1}} - (1 - y(z)) \\ + \frac{1}{r} - \frac{\epsilon}{\ell}\Big(\int_0^1 \mathrm{d}u\, y(z+u)\Big)^{\ell} \Big\}. \end{aligned} \tag{4.4}$$

We note that the potential above is related to that in (1.80) by

$$\mathcal{W}_{\mathrm{BEC}}(y) = \int_{\mathbb{R}} \mathrm{d}z\, P_y(z, y),$$

and we slightly abuse notation by keeping the same label. In this chapter, we do not have to subtract a reference energy from the continuous version of (4.3) because we consider the system at the decoder phase transition threshold $\epsilon = \epsilon_{\mathrm{MAP}}$ (also called the static phase). During the static phase, the profile $\underline{y}$ converges to limits at $z \to \pm\infty$ at which the potential vanishes. Therefore, the potential above is well-defined. At the end of this paragraph, we give the conditions on the erasure probability profile that are required in order to have a well-defined problem. From now on, the reader should think of the noise level as fixed to the value $\epsilon = \epsilon_{\mathrm{MAP}}$; we will make it explicit by always writing the parameter as ϵ_{MAP} in the formulas that follow.

The continuum limit of the DE recursion expressed in terms of $y(z)$ reads

$$y(z) = 1 - \Big(1 - \epsilon_{\mathrm{MAP}} \int_0^1 \mathrm{d}u \Big(\int_0^1 \mathrm{d}s\, y(z-u+s)\Big)^{\ell-1}\Big)^{r-1} \tag{4.5}$$

One can check that (4.5) gives the stationary points of (4.4).

Equation (4.4) can be expressed as a sum of two contributions $\mathcal{W}_{\mathrm{single}}(y)$ and $\mathcal{W}_{\mathrm{int}}(y)$ that are defined as

$$\begin{aligned}\mathcal{W}_{\mathrm{single}}(y) &= \int_{\mathbb{R}} \mathrm{d}z \left\{ \Big(1 - \frac{1}{r}\Big)(1-y(z))^{\frac{r}{r-1}} - (1-y(z)) + \frac{1}{r} - \frac{\epsilon_{\mathrm{MAP}}}{\ell} y(z)^{\ell} \right\} \\ &\equiv \int_{\mathbb{R}} \mathrm{d}z\, W_s(y(z)), \qquad (4.6)\\ \mathcal{W}_{\mathrm{int}}(y) &= \frac{\epsilon_{\mathrm{MAP}}}{\ell} \int_{\mathbb{R}} \mathrm{d}z \left\{ y(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, y(z+u)\Big)^{\ell} \right\}. \qquad (4.7)\end{aligned}$$

We call (4.6) the "single-system potential functional" and (4.7) the "interaction functional". The following remarks explain the interpretation suggested by these names. The term (4.7) vanishes when evaluated for a constant $y(z) = y$. Moreover, the integrand of (4.6), namely $W_s(y(z)) = W_s(y)$ is just the potential of the underlying uncoupled code ensemble. This is easily seen by recognizing that the usual DE equation for the variable input erasure probability is recovered by setting the derivative of $W_s(y)$ to zero. We call $W_s(y)$ the "single system potential". A plot of $W_s(y)$ for the $(3,6)$-ensemble is shown in Figure 4.1 when $\epsilon = \epsilon_{\mathrm{MAP}}$. The figure shows that the single potential vanishes at $y = 0$ and $y = y_{\mathrm{MAP}}$, some positive value. This is a generic feature of all (ℓ, r)-regular code ensembles as long as $\ell \geq 3$ (for cycle codes $\ell = 2$, we have $y_{\mathrm{MAP}} = 0$). In particular, this shows that in order for the integrals in (4.4) to be well-defined, we have to consider profiles $y(z)$ that tend to their limits, as $z \to \pm\infty$, quickly enough so that the potential vanishes. Also, for simplicity, we will restrict our analysis to continuous profiles.

The above remarks motivate us to define the following spaces of profiles. Let

$$\begin{aligned}\mathcal{S} = \{ y : \mathbb{R} \to \mathbb{R}^{+} &\text{ continuous and s.t.} \\ &\lim_{z\to-\infty} z y(z) = 0,\ \lim_{z\to+\infty} z(y(z) - y_{\mathrm{MAP}}) = 0\}. \qquad (4.8)\end{aligned}$$

Note that the left limit is 0 and the right limit is y_{MAP}. We also define a space of non-decreasing profiles (that can be flat over some intervals)

$$\mathcal{S}' = \{ y \in \mathcal{S} :\ y \text{ is non} - \text{decreasing}\},$$

and a space of strictly increasing[5] profiles,

$$\mathcal{S}'' = \{ y \in \mathcal{S} :\ y \text{ is strictly increasing}\}.$$

As will become apparent, it is useful to think of profiles in $\mathcal{S}'$ and $\mathcal{S}''$ as cdf's of measures over $\mathbb{R}$. Here, these measures are normalized so that the measure of $\mathbb{R}$ is

---

[5] To make a clear distinction between non-decreasing and increasing functions, we furthermore describe the latter as *strictly increasing*.

y_{MAP}.[6] Note that, for $y \in \mathcal{S}'$, the support of the associated measure is not necessarily the whole real line, while it is the whole real line for $y \in \mathcal{S}''$. Finally, we introduce the space $\mathcal{S}_0''$ of strictly increasing profiles that are pinned at the origin; more precisely, $y \in \mathcal{S}_0''$ if and only if $y \in \mathcal{S}''$ and $y(0) = y_{\text{MAP}}/2$.

4.3.2 Main Results

It is easy to see that $\mathcal{W}_{\text{BEC}}(y)$ is bounded from below; more precisely,

$$\inf_{y \in \mathcal{S}} \mathcal{W}_{\text{BEC}}(y) \geq -\epsilon_{\text{MAP}} y_{\text{MAP}}^{\ell}/(2\ell).$$

Indeed, $W_s(y) \geq 0$, as seen in Figure 4.1 and, using Jensen's inequality, we show that $\mathcal{W}_{\text{int}}(y) \geq -\epsilon_{\text{MAP}} y_{\text{MAP}}^{\ell}/(2\ell)$ in Lemma 4.4. The first non-trivial question one may ask is whether the minimum is attained in $\mathcal{S}$. Using a rearrangement inequality of Brascamp-Lieb-Luttinger [111], we reduce this question to the same one in $\mathcal{S}' \subset \mathcal{S}$ in Lemma 4.6. We then use the "direct method", a standard technique in the calculus of variations [109, 110] in Section 4.4.2 to show the existence of a minimizer in $\mathcal{S}'$. Once we know that a minimizer exists in $\mathcal{S}'$ we prove that it cannot be in $\mathcal{S}' \setminus \mathcal{S}''$. This existence theorem is formally stated and proven in Section 4.4.2; we write it below for clarity.

Theorem 4.1. *Let $\epsilon = \epsilon_{MAP}$. The functional $\mathcal{W}_{BEC}(y)$ achieves its minimum over $\mathcal{S}$ in the subspace $\mathcal{S}''$. There does not exist a minimum in $\mathcal{S}' \setminus \mathcal{S}''$.*

The existence of a minimum in $\mathcal{S}' \setminus \mathcal{S}''$ is excluded; i.e., a minimizer that is non-decreasing has to be strictly increasing. However, we are not able to exclude the existence of a minimizer in $\mathcal{S} \setminus \mathcal{S}'$ that would have "oscillations". In order to exclude such minimizers, we would have to study the conditions under which the Brascamp-Lieb-Luttinger inequality is strict in our context. We do not address this issue in the present work. In general, this can be a difficult problem (see [112, 113]).

When functionals are convex, one obtains important information on the set of minimizers. For example, strict convexity implies that the minimizer is unique. Thus, the next natural question is whether or not the functional $\mathcal{W}_{\text{BEC}}(y)$ is (strictly) convex. This is in fact not true, but we will show that it is *displacement convex* in $\mathcal{S}'$ (hence also in $\mathcal{S}''$). Here, displacement convexity refers to convexity under an interpolation path that is different from the usual linear combination and is explained in detail in Section 1.7. The functional is invariant under a global translation; i.e., we have $\mathcal{W}_{\text{BEC}}(y(\cdot + \tau)) = \mathcal{W}_{\text{BEC}}(y(\cdot))$. Thus, the system certainly has at least one translational degree of freedom and so the displacement convexity in $\mathcal{S}'$ (and $\mathcal{S}''$) cannot be strict. However, once we remove this degree of freedom by pinning the profiles, say at the origin, we prove strict displacement convexity of the functional in the space $\mathcal{S}_0''$ of pinned and strictly increasing profiles. This is done in Section 4.6.

Theorem 4.2. *Let $\epsilon = \epsilon_{MAP}$. The functional $\mathcal{W}_{BEC}(y)$ is displacement convex on $\mathcal{S}'$, and strictly displacement convex on $\mathcal{S}_0''$.*

---

[6] In Chapter 5 we normalize such profiles such that the measure of $\mathbb{R}$ is 1. In this respect the two chapters are "inconsistent" in order to demonstrate the effect of the "real" limit of the profiles (y_{MAP} in this chapter) on the expressions we use throughout the analysis.

Proof of Theorem 4.2. The displacement convexity of $\mathcal{W}_{\mathrm{BEC}}(y)$ in $\mathcal{S}'$ follows from that of the single potential in Proposition 4.16 and that of the interaction potential in Proposition 4.17. In Section 4.6, we show that $\mathcal{W}_{\mathrm{int}}(y)$ is strictly displacement convex in $\mathcal{S}_0''$. Combining this with Proposition 4.16 immediately yields the strict displacement convexity of $\mathcal{W}_{\mathrm{BEC}}(y)$ in $\mathcal{S}_0''$. $\square$

This implies that there is a unique minimizer in $\mathcal{S}_0''$. But, as the existence of a minimizer is excluded in $\mathcal{S}' \smallsetminus \mathcal{S}''$ (by Theorem 4.15), we can conclude that the only minimizers in $\mathcal{S}'$ are translates of the unique one in $\mathcal{S}_0''$. These consequences also translate into properties of solutions of the DE equation (4.5).

Corollary 4.3. *Let $\epsilon = \epsilon_{MAP}$. In the space $\mathcal{S}_0''$, the functional $\mathcal{W}_{BEC}(y)$ has a unique minimizer. In $\mathcal{S}'$, all minimizers are translates of it. Similarly, in $\mathcal{S}_0''$, the DE equation* (4.5) *has a unique solution, and in $\mathcal{S}'$, all solutions are translates of it.*

Proof of Corollary 4.3. The uniqueness of the minimizer of $\mathcal{W}_{\mathrm{BEC}}(y)$ in $\mathcal{S}_0''$ follows from strict displacement convexity. Indeed, suppose there are two distinct minimizers y_1 and y_2 with $\mathcal{W}_{\mathrm{BEC}}(y_1) = \mathcal{W}_{\mathrm{BEC}}(y_2)$, and let y_λ be their displacement interpolant. Then, $\mathcal{W}_{\mathrm{BEC}}(y_\lambda) < (1-\lambda)\mathcal{W}_{\mathrm{BEC}}(y_1) + \lambda\mathcal{W}_{\mathrm{BEC}}(y_2)$ for $\lambda \neq 0, 1$, which implies that $\mathcal{W}_{\mathrm{BEC}}(y_\lambda) < \mathcal{W}_{\mathrm{BEC}}(y_1)$. We must also have $\mathcal{W}_{\mathrm{BEC}}(y_1) \leq \mathcal{W}_{\mathrm{BEC}}(y_\lambda)$, hence $\mathcal{W}_{\mathrm{BEC}}(y_1) < \mathcal{W}_{\mathrm{BEC}}(y_\lambda)$, which is a contradiction.

Let us now show that all minimizers $y_1 \in \mathcal{S}'$ are translates of the unique minimizer $y_0 \in \mathcal{S}_0''$. Due to Lemma 4.10, we know that $y_1 \in \mathcal{S}''$; i.e., it has to be strictly increasing. Thus, there is a unique position, say z_1, such that $y_1(z_1) = y_{\mathrm{MAP}}/2$. Consider the set of profiles $\mathcal{S}_{z_1}''$ obtained by translating the set $\mathcal{S}_0''$ by the vector z_1. Clearly, y_1 is the unique minimizer in $\mathcal{S}_{z_1}''$. But it is also clear that $\mathcal{W}_{\mathrm{BEC}}(y_0(\cdot - z_1)) = \mathcal{W}_{\mathrm{BEC}}(y_1(\cdot))$. Thus, as $y_0(\cdot - z_1) \in \mathcal{S}_{z_1}''$, we must have $y_0(\cdot - z_1) = y_1(\cdot)$ as announced.

Finally, let us discuss the consequences for the solutions of the DE equation (4.5). We show that in the space $\mathcal{S}'$, a solution of the DE equation is necessarily a minimum of $\mathcal{W}_{\mathrm{BEC}}(y)$. This implies the statement of the theorem. Let $y_0 \in \mathcal{S}'$ denote a solution of the DE equation. Consider any other profile $y_1 \in \mathcal{S}'$, and consider their resulting displacement interpolant y_λ. A computation of the derivative shows that $\frac{\mathrm{d}}{\mathrm{d}\lambda}\mathcal{W}_{\mathrm{BEC}}(y_\lambda)|_{\lambda=0} = 0$ because y_0 is a solution of the DE equation. As the map $\lambda \to \mathcal{W}_{\mathrm{BEC}}(y_\lambda)$ is displacement convex, $\lambda = 0$ must be a minimum of this map. Thus $\mathcal{W}_{\mathrm{BEC}}(y_\lambda) \geq \mathcal{W}_{\mathrm{BEC}}(y_0)$, and in particular with $\lambda = 1$, we obtain $\mathcal{W}_{\mathrm{BEC}}(y_1) \geq \mathcal{W}_{\mathrm{BEC}}(y_0)$. Thus y_0 is a minimum of the functional in $\mathcal{S}'$. $\square$

We would like to point out that, while displacement convexity itself is quite general and can presumably be generalized to the general potential functionals of [100], the issue of strict displacement convexity is more subtle. In fact, T. Richardson [114] pointed out examples of systems where "internal" translation degrees of freedom may exist (besides the global one) which would spoil the unicity up to global translations.

4.4 Existence of Minimizing Profile

In this section, we prove that the functional $\mathcal{W}_{\mathrm{BEC}}$ attains its minimum using the "direct method" in the calculus of variations. Before stating and proving the theorem on the existence of the minimizer in Section 4.4.2, we establish some useful preliminaries in Section 4.4.1.

4.4.1 Preliminaries

The results in this section will allow us to restrict the search of minimizing profiles to those in $\mathcal{S}$ that are monotone non-decreasing.

The first lemma states that the interaction potential is bounded from below.

Lemma 4.4. *For any y in $\mathcal{S}$,*

$$\int_{\mathbb{R}} \mathrm{d}z \left\{ y(z)^{\ell} - \left(\int_0^1 \mathrm{d}u \, y(z+u) \right)^{\ell} \right\} \geq -\frac{1}{2} y_{MAP}^{\ell}.$$

Proof. As $h(u) = u^{\ell}$, $u \geq 0$, is a convex function, we use Jensen's inequality to write

$$\int_0^1 \mathrm{d}u \, y(z+u)^{\ell} \geq \Big(\int_0^1 \mathrm{d}u \, y(z+u) \Big)^{\ell}. \tag{4.9}$$

Further,

$$\begin{aligned} \int_{-M}^{M} \mathrm{d}z \int_0^1 \mathrm{d}u \, y(z+u)^{\ell} &\overset{(a)}{=} \int_0^1 \mathrm{d}u \int_{-M+u}^{M+u} \mathrm{d}\tilde{z} \, y(\tilde{z})^{\ell} \\ &= \int_0^1 \mathrm{d}u \Big(\int_{-M+u}^{-M} \mathrm{d}\tilde{z} \, y(\tilde{z})^{\ell} + \int_{-M}^{M} \mathrm{d}\tilde{z} \, y(\tilde{z})^{\ell} + \int_{M}^{M+u} \mathrm{d}\tilde{z} \, y(\tilde{z})^{\ell} \Big), \end{aligned}$$

where (a) is obtained by first changing the order of integration (which is admissible because the integral converges) and then making the change of variable $\tilde{z} = z + u$. And so, by combining this identity with (4.9), we obtain

$$\begin{aligned} &\int_{-M}^{M} \mathrm{d}z \left\{ y(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u \, y(z+u) \Big)^{\ell} \right\} \\ &\qquad + \int_0^1 \mathrm{d}u \int_{-M+u}^{-M} \mathrm{d}\tilde{z} \, y(\tilde{z})^{\ell} + \int_0^1 \mathrm{d}u \int_{M}^{M+u} \mathrm{d}\tilde{z} \, y(\tilde{z})^{\ell} \geq 0. \end{aligned}$$

Now, we take the limit $M \to +\infty$ for each term of this inequality. By an application of Lebesgue's dominated convergence theorem, the last two terms tend to zero and $\frac{1}{2} y_{\mathrm{MAP}}^{\ell}$, respectively. Therefore, the limit of the first term is bounded from below by $-\frac{1}{2} y_{\mathrm{MAP}}^{\ell}$, which concludes the proof. □

We remark that any constant lower bound is sufficient for our purposes: finding profiles that minimize a potential is equivalent to finding those that minimize a potential added to a constant.

The following lemma states that a truncation of the profile at the value y_{MAP} decreases the potential functional, so we may restrict our search of minimizing profiles to those with range $y(z) \in [0, y_{\mathrm{MAP}}]$.

Lemma 4.5. *Define $\bar{y}(z) = \min\{y(z), y_{MAP}\}$. For all $y \in \mathcal{S}$ we have*

$$\mathcal{W}_{BEC}(y) \geq \mathcal{W}_{BEC}(\bar{y}),$$

and the inequality is strict if $y \neq \bar{y}$.

Proof. It is easy to prove that a truncation of $y(z)$ at y_{MAP} yields a smaller value for the single system potential $\mathcal{W}_{\mathrm{single}}(y)$ (see e.g. the Figure 4.1 for intuition). Therefore we have $\mathcal{W}_{\mathrm{single}}(y) \geq \mathcal{W}_{\mathrm{single}}(\bar{y})$.

We now treat the functional corresponding to the interaction term. We define the function δ_y as $\delta_y(z) = y(z) - \bar{y}(z)$ and notice that

$$\begin{cases} y(z) \leq y_{\mathrm{MAP}} \Rightarrow \delta_y(z) = 0 \text{ and } \bar{y}(z) = y(z), \\ y(z) > y_{\mathrm{MAP}} \Rightarrow \delta_y(z) > 0 \text{ and } \bar{y}(z) = y_{\mathrm{MAP}}. \end{cases} \tag{4.10}$$

We need to show that $\mathcal{W}_{\mathrm{int}}(\bar{y}) \leq \mathcal{W}_{\mathrm{int}}(y)$ or, equivalently, that

$$\begin{aligned} &\int_{\mathbb{R}} \mathrm{d}z \left\{ \bar{y}(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u \, \bar{y}(z+u) \Big)^{\ell} \right\} \\ &\qquad \leq \int_{\mathbb{R}} \mathrm{d}z \left\{ (\bar{y}(z) + \delta_y(z))^{\ell} - \Big(\int_0^1 \mathrm{d}u \, (\bar{y}(z+u) + \delta_y(z+u)) \Big)^{\ell} \right\}. \end{aligned}$$

We use the binomial expansion to write this as

$$\sum_{i=0}^{\ell-1} \binom{\ell}{i} \int_{\mathbb{R}} \mathrm{d}z \left\{ \bar{y}(z)^i \delta_y(z)^{\ell-i} - \Big(\int_0^1 \mathrm{d}u \, \bar{y}(z+u) \Big)^i \Big(\int_0^1 \mathrm{d}u \, \delta_y(z+u) \Big)^{\ell-i} \right\} \geq 0.$$

In the following steps, we show that the integral inside the summation above is positive for any fixed value of i; the inequality follows directly. We see that

$$\begin{aligned} \Big(\int_0^1 \mathrm{d}u \, \bar{y}(z+u) \Big)^i \Big(\int_0^1 \mathrm{d}u \, \delta_y(z+u) \Big)^{\ell-i} &\leq y_{\mathrm{MAP}}^i \Big(\int_0^1 \mathrm{d}u \, \delta_y(z+u) \Big)^{\ell-i} \\ &\leq y_{\mathrm{MAP}}^i \int_0^1 \mathrm{d}u \, \delta_y(z+u)^{\ell-i}, \end{aligned}$$

where the first inequality is due to the property $\bar{y}(z) \leq y_{\mathrm{MAP}}$ and the second is obtained using the convexity of the function $h(u) = u^{\ell}$; $u \geq 0$. We integrate over z, then make the change of variables $\tilde{z} = z + u$ on the right-hand side to obtain

$$\begin{aligned} &\int_{\mathbb{R}} \mathrm{d}z \Big(\int_0^1 \mathrm{d}u \, \bar{y}(z+u) \Big)^i \Big(\int_0^1 \mathrm{d}u \, \delta_y(z+u) \Big)^{\ell-i} \\ &\qquad \leq \int_{\mathbb{R}} \mathrm{d}z \, y_{\mathrm{MAP}}^i \int_0^1 \mathrm{d}u \, \delta_y(z+u)^{\ell-i} = \int_{\mathbb{R}} \mathrm{d}\tilde{z} \, y_{\mathrm{MAP}}^i \delta_y(\tilde{z})^{\ell-i}. \end{aligned} \tag{4.11}$$

Using the properties of δ_y in (4.10), we remark that

$$\int_{\mathbb{R}} \mathrm{d}z \, y_{\mathrm{MAP}}^i \delta_y(z)^{\ell-i} = \int_{\mathbb{R}} \mathrm{d}z \, \bar{y}(z)^i \, \delta_y(z)^{\ell-i},$$

and so the difference of quantities in the inequality (4.11) is integrable. We thus obtain the following inequality

$$\int_{\mathbb{R}} \mathrm{d}z \Big(\int_0^1 \mathrm{d}u \, \bar{y}(z+u) \Big)^i \Big(\int_0^1 \mathrm{d}u \, \delta_y(z+u) \Big)^{\ell-i} \leq \int_{\mathbb{R}} \mathrm{d}z \, \bar{y}(z)^i \delta_y(z)^{\ell-i},$$

for any i. This yields the desired result. □

We next restrict our search of minimizing profiles to non-decreasing ones. In order to achieve this, we will use rearrangement inequalities. We will use a notion of increasing rearrangement[7] introduced by [94]. In words, an increasing rearrangement associates to any function $y \in \mathcal{S}$ with range $[0, y_{\mathrm{MAP}}]$ a non-decreasing function $y^* \in \mathcal{S}'$ so that the total mass is preserved. More formally, any non-negative function y in $\mathcal{S}$ can be represented in layer cake form

$$y(z) = \int_0^{+\infty} \mathrm{d}t\, \mathbb{1}_{E_t}(z),$$

where $\mathbb{1}_{E_t}(z)$ is the indicator function of the level set $E_t = \{z | y(z) > t\}$. For each t, the level set E_t can be written as the union of a bounded set A_t and a half line $(a_t, +\infty)$. We define the associated rearranged set as $E_t^* = (a_t - |A_t|, +\infty)$. The increasing rearrangement of y is the new function y^* whose level sets are E_t^*. More explicitly,

$$y^*(z) = \int_0^{+\infty} \mathrm{d}t\, \mathbb{1}_{E_t^*}(z).$$

Lemma 4.6. *Take any $y \in \mathcal{S}$ and let $y^* \in \mathcal{S}'$ be its increasing rearrangement. Then,*

$$\mathcal{W}_{BEC}(y) \geq \mathcal{W}_{BEC}(y^*).$$

Before proceeding with the proof of Lemma 4.6, we state a definition and a general rearrangement inequality of Brascamp, Lieb and Luttinger [111].

Definition 4.7 (Definition 1.1 [111])**.** *Let f be a non-negative measurable function on $\mathbb{R}$, let $K_y^f = \{x | f(x) \geq y\}$ and let $M_y^f = \mu(K_y^f)$, where μ denotes the Lebesgue measure. Assume that $M_\alpha^f < \infty$ for some $\alpha < \infty$. If $\hat{f}$ is another function on $\mathbb{R}$ with the same properties as f and, additionally, $\hat{f}(x) = \hat{f}(-x)$ for all $x \in \mathbb{R}$, and for any $0 < x_1 < x_2$, we have $\hat{f}(x_2) \leq \hat{f}(x_1)$, and finally, $M_y^{\hat{f}} = M_y^f$ for all $y > 0$, then $\hat{f}$ is called a* symmetric decreasing rearrangement *of f.*

Theorem 4.8 (Theorem 1.2 [111])**.** *Let f_j, $1 \leq j \leq k$ be non-negative measurable functions on $\mathbb{R}$, and let a_{jm}, $1 \leq j \leq k$, $1 \leq m \leq n$, be real numbers. Then, if $\hat{f}_j$ is the symmetric decreasing rearrangement of f_j, we have*

$$\int_{\mathbb{R}^n} \mathrm{d}^n x \prod_{j=1}^{k} f_j\left(\sum_{m=1}^{n} a_{jm} x_m\right) \leq \int_{\mathbb{R}^n} \mathrm{d}^n x \prod_{j=1}^{k} \hat{f}_j\left(\sum_{m=1}^{n} a_{jm} x_m\right) \tag{4.12}$$

Remark 4.9. *Theorem 4.8 is non-trivial only if $k > n$. Otherwise, both integrals diverge and the inequality trivially holds. We will see in this section that $k > n$ in our case.*

Proof of Lemma 4.6. It is sufficient to prove that the increasing rearrangement of a profile decreases $\mathcal{W}_{\mathrm{int}}(y)$ because $\mathcal{W}_{\mathrm{single}}(y)$ is invariant under rearrangement.

Theorem 4.8 applies to symmetric decreasing rearrangements. Therefore, it is convenient to first "symmetrize" the profiles we consider and the potential functional. Consider a profile $y \in \mathcal{S}$ such that $y(z) \in [0, y_{\mathrm{MAP}}]$, $z \in \mathbb{R}$, (due to Lemma 4.5) and denote by y_s the function such that $y_s(z) = y(z)$, $z < R$ and $y_s(z) = y_s(2R -$

---

[7] Previously, the term "increasing" was used to denote functions that are currently referred to as "non-decreasing". So an increasingly rearranged function is allowed to have flat spots. In our notation, we always use "non-decreasing" to refer to such functions. However, we keep the name "increasing rearrangement" to be consistent with the literature.

$z)$, $z > R$. The value of R is chosen (large enough) so that $y(R)$ is arbitrarily close to y_{MAP}. Note that y_s is integrable over $\mathbb{R}$.

We recall the expression of $\mathcal{W}_{\text{int}}(y)$ in (4.7) and rewrite it as

$$\mathcal{W}_{\text{int}}(y) = \frac{\epsilon_{\text{MAP}}}{\ell} \lim_{R \to +\infty} \left\{ \int_{-\infty}^{R} \mathrm{d}z\, y(z)^{\ell} - \int_{-\infty}^{R} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, y(z+u) \Big)^{\ell} \right\}. \tag{4.13}$$

We now express both integrals between brackets in terms of the symmetrized profile. For the first one, this is immediate

$$\int_{-\infty}^{R} \mathrm{d}z\, y(z)^{\ell} = \frac{1}{2} \int_{\mathbb{R}} \mathrm{d}z\, y_s(z)^{\ell}. \tag{4.14}$$

For the second one, some care has to be taken with the averaging over u when z is near R. One has

$$\int_{-\infty}^{R} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, y(z+u) \Big)^{\ell} = \frac{1}{2} \int_{\mathbb{R}} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, y_s(z+u) \Big)^{\ell} + o\Big(\frac{1}{R^{\ell}}\Big). \tag{4.15}$$

Replacing these two equations in (4.13), we obtain

$$\mathcal{W}_{\text{int}}(y) = \frac{\epsilon_{\text{MAP}}}{2\ell} \int_{\mathbb{R}} \mathrm{d}z\, y_s(z)^{\ell} - \frac{\epsilon_{\text{MAP}}}{2\ell} \int_{\mathbb{R}} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, y_s(z+u) \Big)^{\ell}. \tag{4.16}$$

Now, we consider $\hat{y}_s$, the symmetric decreasing rearrangement of y_s. The first term in (4.16) is invariant under rearrangement. It remains to prove that the second term in (4.16) increases upon rearrangement. We express it as follows (dropping $\epsilon_{\text{MAP}}/2\ell$)

$$\begin{aligned}
\int_{\mathbb{R}} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, y_s(z+u) \Big)^{\ell} &= \int_{\mathbb{R}} \mathrm{d}z \int_{\mathbb{R}^{\ell}} \prod_{i=1}^{\ell} \mathrm{d}u_i\, y_s(z+u_i) \mathbb{1}_{[0,1]}(u_i) \\
&\overset{(b)}{=} \int_{\mathbb{R}} \mathrm{d}\tilde{z} \int_{\mathbb{R}^{\ell}} \prod_{i=1}^{\ell} \mathrm{d}\tilde{u}_i\, y_s(\tilde{z} + R + \tilde{u}_i + \frac{1}{2})\, \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}(\tilde{u}_i) \\
&\overset{(c)}{\le} \int_{\mathbb{R}} \mathrm{d}\tilde{z} \int_{\mathbb{R}^{\ell}} \prod_{i=1}^{\ell} \mathrm{d}\tilde{u}_i\, \hat{y}_s(\tilde{z} + R + \tilde{u}_i + \frac{1}{2})\, \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}(\tilde{u}_i) \\
&\overset{(d)}{=} \int_{\mathbb{R}} \mathrm{d}z \int_{\mathbb{R}^{\ell}} \prod_{i=1}^{\ell} \mathrm{d}u_i\, \hat{y}_s(z+u_i)\, \mathbb{1}_{[0,1]}(u_i) \\
&= \int_{\mathbb{R}} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, \hat{y}_s(z+u) \Big)^{\ell},
\end{aligned}$$

where the equality in (b) is due to the changes of variables $\tilde{z} = z - R$ and $\tilde{u}_i = u_i - \frac{1}{2}$; $i = 1 \dots \ell$, the inequality in (c) is due Theorem 4.8, and the equality in (d) is obtained by first remarking that the indicator function $\mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}(\tilde{u}_i)$ is unchanged upon rearrangement and then by making the reverse changes of variables $z = \tilde{z} + R$ and $u_i = \tilde{u}_i + \frac{1}{2}$; $i = 1 \dots \ell$. So far we have obtained

$$\mathcal{W}_{\text{int}}(y) \ge \frac{\epsilon_{\text{MAP}}}{2\ell} \int_{\mathbb{R}} \mathrm{d}z\, \hat{y}_s(z)^{\ell} - \frac{\epsilon_{\text{MAP}}}{2\ell} \int_{\mathbb{R}} \mathrm{d}z \Big(\int_{0}^{1} \mathrm{d}u\, \hat{y}_s(z+u) \Big)^{\ell}.$$

To prove that $\mathcal{W}_{\text{int}}(y) \ge \mathcal{W}_{\text{int}}(y^*)$, it remains to remark that y^* is the left-hand "unsymmetrized" version of $\hat{y}_s$ and to reverse the steps (4.13)-(4.16). □

We can thus restrict the search of minimizing profiles to the space of non-decreasing profiles (however, as already explained, we cannot exclude the possibility that there exists a profile $y \in \mathcal{S}$ such that $\mathcal{W}_{\text{BEC}}(y) = \mathcal{W}_{\text{BEC}}(y^*)$). In fact, the following lemma allows us to further restrict the search of minimizing profiles to strictly increasing ones.

Lemma 4.10. *Let $y \in \mathcal{S}'$ be a minimizer of the potential functional $\mathcal{W}_{BEC}(y)$. Then it must be strictly increasing; i.e., $y \in \mathcal{S}''$.*

We establish Lemma 4.10 as a corollary of the two following lemmas.

Lemma 4.11. *If y minimizes $\mathcal{W}_{BEC}(y)$, then it satisfies the DE equation.*

Proof. See Section 4.7.1. □

Lemma 4.12. *If y is non-decreasing and satisfies the DE equation, then it cannot be strictly flat on an interval $(a, b) \subset \mathbb{R}$ with positive measure.*

Proof. See Section 4.7.2. □

Proof of Lemma 4.10. The proposition follows directly from Lemmas 4.11 and 4.12. □

Below, we mention a lemma that is interesting in itself but not necessary for our results.

Lemma 4.13. *If y minimizes $\mathcal{W}_{BEC}(y)$ then it cannot have a flat spot (i.e., $y(z) = y_{\text{flat}}$ with $0 < y_{\text{flat}} < y_{MAP}$) in a bounded interval $z \in [a, b]$ such that $b - a > 1$.*

Proof. See Section 4.7.3. □

The final step of these preliminaries concerns a necessary condition that any *minimizing sequence* in $\mathcal{S}_0''$ must satisfy. It is useful to think of such profiles as cdf's. A minimizing sequence in $\mathcal{S}_0''$ is, by definition, any sequence $y_n \in \mathcal{S}_0''$ such that

$$\lim_{n\to\infty} \mathcal{W}_{\text{BEC}}(y_n) = \inf_{y \in \mathcal{S}_0''} \mathcal{W}_{\text{BEC}}(y). \tag{4.17}$$

Such a sequence exists as long as the functional is bounded from below. This is true because $W_s(y) \geq 0$ and due to Lemma 4.4. Consider the sequence of probability measures associated to the sequence of cdf's y_n. The following lemma states that this sequence of measures is *tight*.

Lemma 4.14. *Let $y_n \in \mathcal{S}_0''$ be a minimizing sequence of cdf's. For any $\delta > 0$, we can find $M_\delta > 0$ (independent of n) such that*

$$y_n(M_\delta) - y_n(-M_\delta) > (1 - \delta) y_{MAP},$$

for all n.

Proof. Consider a minimizing sequence of cdf's y_n; i.e., satisfying (4.17). Fix any $\delta > 0$ and suppose that

$$y_n(M) - y_n(-M) < (1-\delta) y_{\text{MAP}}. \tag{4.18}$$

We will show that (4.18) implies that necessarily $M \le c/\delta^2$ for a fixed constant $c > 0$. Taking the contrapositive we find that: choosing $M_\delta = c'/\delta^2$ with $c' > c$ implies that any minimizing sequence satisfies $y_n(M_\delta) - y_n(-M_\delta) > (1-\delta) y_{\text{MAP}}$.

From Lemma 4.4 we have

$$\mathcal{W}_{\text{BEC}}(y_n) \ge \mathcal{W}_{\text{single}}(y_n) - \frac{\epsilon_{\text{MAP}} y_{\text{MAP}}^{\ell}}{2\ell} \ge \int_{-M}^{M} \mathrm{d}z \, W_s(y_n(z)) - \frac{\epsilon_{\text{MAP}} y_{\text{MAP}}^{\ell}}{2\ell}. \tag{4.19}$$

Now, assuming (4.18) there must be a mass at least δy_{MAP} outside of the interval $[-M, M]$. Thus, we have $y_n(-M) \ge \delta y_{\text{MAP}}/2$ or $y_{\text{MAP}} - y_n(M) \ge \delta y_{\text{MAP}}/2$. Recall that $y_n \in \mathcal{S}_0''$ so $y_n(0) = y_{\text{MAP}}/2$. Therefore, in $[-M, 0]$ or in $[0, M]$ the profile $y_n(z)$ must be $\delta y_{\text{MAP}}/2$ away from the minima 0 and y_{MAP} of W_s. Moreover, one can check that $W_s(y)$ has a parabolic shape near the minima at 0 and y_{MAP} so that away from these minima $W_s(y) \ge C\delta^2 y_{\text{MAP}}^2/4$ for a constant $C > 0$ depending only on ℓ. These remarks imply that

$$\int_{-M}^{M} \mathrm{d}z \, W_s(y_n(z)) \ge \frac{1}{2} M C \delta^2 y_{\text{MAP}}^2. \tag{4.20}$$

As y_n is a minimizing sequence, for n large enough its cost must be smaller than the cost of a fixed reference profile, say $\rho(z) = 0, z \le 0$, $\rho(z) = y_{\text{MAP}}$, $z > 0$. More formally,

$$\mathcal{W}_{\text{BEC}}(y_n) < \mathcal{W}_{\text{BEC}}(\rho) = -\frac{\epsilon_{\text{MAP}}}{\ell(\ell+1)} y_{\text{MAP}}^{\ell}. \tag{4.21}$$

Finally, combining (4.19), (4.20) and (4.21), we find that

$$M \le 2 \frac{\epsilon_{\text{MAP}}(\ell-1)}{\ell(\ell+1)} \frac{y_{\text{MAP}}^{\ell-2}}{C\delta^2}. \tag{4.22}$$

□

4.4.2 The Direct Method

The direct method in the calculus of variations [109, 110] is a standard scheme that is used to prove that minimizers exist. We use this method to obtain the following theorem.

Theorem 4.15. *Let $\epsilon = \epsilon_{MAP}$. The functional $\mathcal{W}_{BEC}(y)$ achieves its minimum over $\mathcal{S}$ in the subspace $\mathcal{S}''$. There does not exist a minimum in $\mathcal{S}' \setminus \mathcal{S}''$.*

Proof. Let us take any minimizing sequence y_n of cdf's; i.e., a sequence that satisfies (4.17). By Lemma 4.14, the corresponding sequence of measures is tight. Thus, by a simple version of Prokhorov's theorem for measures on the real line, we can extract a (pointwise) convergent subsequence of cdf's $y_{n_k} \to y_l$ as $k \to +\infty$ with $y_l \in \mathcal{S}_0''$. We can thus directly apply Fatou's lemma to deduce that the potential functional is lower-semi-continuous, which means that

$$\mathcal{W}_{\text{BEC}}(y_l) \le \liminf_{k \to +\infty} \mathcal{W}_{\text{BEC}}(y_{n_k}). \tag{4.23}$$

Putting (4.17) and (4.23) together, we obtain

$$\mathcal{W}_{\mathrm{BEC}}(y_l) \le \liminf_{k\to+\infty} \mathcal{W}_{\mathrm{BEC}}(y_{n_k}) = \lim_{n\to+\infty} \mathcal{W}_{\mathrm{BEC}}(y_n) = \inf_{\mathcal{S}_0''} \mathcal{W}_{\mathrm{BEC}}(y).$$

On the other hand, $\inf_{\mathcal{S}_0''} \mathcal{W}_{\mathrm{BEC}}(y) \le \mathcal{W}_{\mathrm{BEC}}(y_l)$. Thus, we conclude that $\inf_{\mathcal{S}_0''} \mathcal{W}_{\mathrm{BEC}}(y) = \mathcal{W}_{\mathrm{BEC}}(y_l)$.

We have shown that the minimum is achieved in $\mathcal{S}_0''$, the space of strictly increasing profiles pinned at the origin. Hence, it is achieved in $\mathcal{S}'$ and $\mathcal{S}''$ (note that, by translation invariance, translations of y_l are minimizers in these spaces). Finally, Lemma 4.10 ensures that there is no minimum in $\mathcal{S}' \smallsetminus \mathcal{S}''$. □

4.5 Displacement Convexity

This section contains the main results of the chapter, namely that the potential functional $\mathcal{W}_{\mathrm{BEC}}(y)$ is displacement convex in $\mathcal{S}'$ and strictly displacement convex in $\mathcal{S}_0''$.

4.5.1 Discussion on Displacement Convexity

We provide a brief introduction to displacement convexity in Section 1.7. In this section, we remind the reader of some basic definitions and relate them to the content of the present chapter.

Lemma 4.6 in Section 4.4 shows that we can restrict the minimization problem to the space of non-decreasing profiles. Thus, the discussion below assumes that we consider such profiles. This is the correct setting for defining displacement convexity.

A non-decreasing profile with left limit 0 and right limit y_{MAP} can be thought of as a cdf (up to scaling because the right limit might not be equal to 1). Further, such non-decreasing functions have non-decreasing inverse functions (that can also be thought of as cdf's, up to scaling). More precisely, consider the following bijective maps that associate (with an abuse of notation) to a cdf y its inverse z.

$$\begin{aligned} z(y) &= \inf\{\, z : \ y(z) > y\}, \\ y(z) &= \inf\{\, y : \ z(y) > z\}. \end{aligned}$$

For any two non-decreasing profiles $y_0, y_1 \in \mathcal{S}'$, we consider z_0, z_1 their respective inverses under the maps defined above. Then, for any $\lambda \in [0,1]$, the interpolated profile y_λ is defined as follows.

$$\begin{aligned} z_\lambda(y) &= (1-\lambda) z_0(y) + \lambda z_1(y), \\ y_\lambda(z) &= \inf\{\, y : \ z_\lambda(y) > z\}. \end{aligned} \tag{4.24}$$

The graphical construction of the interpolant p_λ of two profiles p_0 and p_1 is shown in Figure 1.14. It is not difficult to see that if y_0 and y_1 are in $\mathcal{S}'$, $\mathcal{S}''$, or $\mathcal{S}_0''$, then so is the respective interpolating profile y_λ for all $\lambda \in [0,1]$.

Displacement convexity of the potential functional $\mathcal{W}_{\mathrm{BEC}}(y)$ on the space $\mathcal{S}'$ means that the following inequality holds

$$\mathcal{W}_{\mathrm{BEC}}(y_\lambda) \le (1-\lambda)\mathcal{W}_{\mathrm{BEC}}(y_0) + \lambda \mathcal{W}_{\mathrm{BEC}}(y_1), \tag{4.25}$$

for any y_0, $y_1 \in \mathcal{S}'$ and all $\lambda \in [0,1]$. Strict displacement convexity means that this inequality is strict as long as y_0 and y_1 are distinct and $\lambda \in (0,1)$. We will that $\mathcal{W}_{\text{BEC}}(y)$ is displacement convex by separately proving this property, in Sections 4.5.2 and 4.5.3, for the single potential (4.6) and the interaction potential (4.7), respectively. Moreover, we will see that (4.7) is strictly displacement convex in $\mathcal{S}_0''$.

4.5.2 Displacement Convexity of the Single-Potential

We first prove that the single-potential functional $\mathcal{W}_{\text{single}}(y)$ is displacement convex. Note that the single system potential function $W_s(y)$ is *not convex* in the usual sense (see Figure 4.1, for example).

Proposition 4.16. *Let y_0 and y_1 be in $\mathcal{S}'$ and let y_λ be their interpolating profile as defined in Section 4.24. Then*

$$\mathcal{W}_{\text{single}}(y_\lambda) = (1-\lambda)\mathcal{W}_{\text{single}}(y_0) + \lambda\mathcal{W}_{\text{single}}(y_1).$$

Proof. Recall that $\mathcal{W}_{\text{single}}(y) = \int_{\mathbb{R}} \mathrm{d}z\, W_s(y(z))$. Recall also that $y_\lambda(z)$ as defined in Section 4.24 is the inverse of $z_\lambda(y) = (1-\lambda)z_0(y) + \lambda z_1(y)$. Thus

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z\, W_s(y_\lambda(z)) &= \int_0^{y_{\text{MAP}}} \mathrm{d}z_\lambda(y)\, W_s(y)\\ &= (1-\lambda)\int_0^{y_{\text{MAP}}} \mathrm{d}z_0(y)\, W_s(y) + \lambda\int_0^{y_{\text{MAP}}} \mathrm{d}z_1(y)\, W_s(y)\\ &= (1-\lambda)\int_{\mathbb{R}} \mathrm{d}z\, W_s(y_0(z)) + \lambda\int_{\mathbb{R}} \mathrm{d}z\, W_s(y_1(z)).\end{aligned}$$

□

Thus, the function $\lambda \to \mathcal{W}_{\text{single}}(y_\lambda)$ is linear, hence convex.

4.5.3 Displacement Convexity of the Interaction-Potential

The proof of displacement convexity of the interaction potential term is more involved.

Proposition 4.17. *Let y_0 and y_1 be in $\mathcal{S}'$ and let y_λ be their interpolating profile as defined in Section 4.24. Then*

$$\mathcal{W}_{\text{int}}(y_\lambda) \le (1-\lambda)\mathcal{W}_{\text{int}}(y_0) + \lambda\mathcal{W}_{\text{int}}(y_1). \tag{4.26}$$

Proof. As y can be seen as a cdf (up to scaling), we associate with it a probability measure μ such that $y(z) = y_{\text{MAP}}\int_{-\infty}^{z} \mathrm{d}\mu(u)$. We then rewrite the interaction functional in the form

$$\mathcal{W}_{\text{int}}(y_\lambda) = \int_{\mathbb{R}} \mathrm{d}\mu_\lambda(z_1)\ldots\mathrm{d}\mu_\lambda(z_\ell)V(z_1,\ldots,z_\ell), \tag{4.27}$$

where $V(z_1,\ldots,z_\ell)$ is a totally symmetric "kernel function" that we will compute. There is an argument by Villani [93] that allows to conclude (4.26) whenever V is jointly convex (in the usual sense). Let us briefly explain this argument here. Consider the measures μ_0, μ_1 associated to the cdf's y_0, y_1. Then there exists a unique non-decreasing map $T : \mathbb{R} \to \mathbb{R}$ such that $\mu_1 = T\#\mu_0$. Here $T\#\mu_0$ is the push-forward[8] of μ_0 under T. Then, from $z_\lambda(y) = (1-\lambda)z_0(y) + \lambda z_1(y)$, we have that $\mu_\lambda = T_\lambda \# \mu_0$ where $T_\lambda(z) = (1-\lambda)z + \lambda T(z)$. Equation (4.27) can thus be written as

$$\mathcal{W}_{\text{int}}(y_\lambda) = \int_{\mathbb{R}} \mathrm{d}\mu_0(z_1)\ldots\mathrm{d}\mu_0(z_\ell) V(T_\lambda(z_1),\ldots,T_\lambda(z_\ell)) \tag{4.28}$$

$$= \ell! \int_{S_z} \mathrm{d}\mu_0(z_1)\ldots\mathrm{d}\mu_0(z_\ell) V(T_\lambda(z_1),\ldots,T_\lambda(z_\ell)). \tag{4.29}$$

In (4.29), we restrict the integrals over the sector $S_z = \{\mathbf{z} = (z_1,\cdots,z_\ell) : z_i \geq z_j \text{ if } i < j\}$. This restriction is allowed because V is totally symmetric, as we will see. It is important to notice that we have $T_\lambda(z_1) \geq \cdots \geq T_\lambda(z_\ell)$ for any $\lambda \in [0,1]$ because T is a non-decreasing map. Moreover, the λ dependence in the kernel function is linear. Thus, the proof of displacement convexity ultimately rests on checking that the kernel function is jointly convex (in the usual sense) in one sector, say S_z.[9] If we can establish the usual convexity of V on S_z, then the linearity in λ will pop out of the integrals over the measures, as we had for the single potential functional. In fact, the kernel function is translation-invariant and can be expressed as a function of the distances $d_{1i} \equiv z_1 - z_i$, $i = 1,\ldots,\ell$, with $d_{11} = 0$. We will prove that V is jointly convex as a function of these distances.

It remains to compute V and to investigate its joint convexity. With appropriate usage of Fubini's theorem and after some manipulations, we find

$$\mathcal{W}_{\text{int}}(y) = \frac{\epsilon_{\text{MAP}} y_{\text{MAP}}^{\ell}}{\ell} \int_{\mathbb{R}^\ell} \prod_{i=1}^{\ell} \mathrm{d}\mu_0(z_i) \times \left\{ \int_{[0,1]^\ell} \prod_{i=1}^{\ell} \mathrm{d}u_i \int_{\mathbb{R}} \mathrm{d}z \left(\prod_{i=1}^{\ell} \theta(z - z_i) - \prod_{i=1}^{\ell} \theta(z - (z_i - u_i)) \right) \right\}, \tag{4.30}$$

where $\theta(z)$ denotes the Heaviside step function. So the kernel $V(z_1,\ldots,z_\ell)$ in (4.27) is the integrand (between the brackets) of the first ℓ integrals in (4.30) (as well as the constant $\epsilon_{\text{MAP}} y_{\text{MAP}}^{\ell}/\ell$ before the integrals that we drop to alleviate notation). Our goal, henceforth, is to prove that V is convex in the usual sense. We will reduce the problem further by proving that, in fact, $V_{\mathbf{u}}$ is convex for all fixed $\mathbf{u}$, where $\mathbf{u} = (u_1,\ldots,u_\ell)$ and

$$V_{\mathbf{u}}(\mathbf{z}) = \int_{\mathbb{R}} \mathrm{d}z \left(\prod_{i=1}^{\ell} \theta(z - z_i) - \prod_{i=1}^{\ell} \theta(z - (z_i - u_i)) \right).$$

We recall here that we restrict our analysis to the sector of the space of ordered variables S_z. Also, we remark that $\prod_{i=1}^{\ell} \theta(a_i) = \theta(\max_{i=1\ldots\ell} a_i)$. We observe that $V_{\mathbf{u}}$ can be written in terms of the distances $d_{1i} = z_1 - z_i$, $i = 1,\ldots,\ell$ as

$$V_{\mathbf{u}}(\mathbf{z}) = \int_{\mathbb{R}} \mathrm{d}z \left\{ \theta(z - z_1) - \theta(\max_{i=1\ldots\ell}(z - (z_i - u_i))) \right\}$$
$$= -\min_{i=1\ldots\ell}(z_1 - z_i + u_i) = -\min_{i=1\ldots\ell}(d_{1i} + u_i)$$

---

[8] Given a measurable map $T : \mathbb{R} \to \mathbb{R}$, the push-forward of μ under T is the measure $T\#\mu_0$ such that, for any bounded continuous function ϕ, $\int_{\mathbb{R}} \phi(T(z))\mathrm{d}\mu(z) = \int_{\mathbb{R}} \phi(z)\mathrm{d}(T\#\mu)(z)$. This is explained in more detail in Section 1.7.

[9] By symmetry, convexity in one sector implies convexity in other sectors. However this does not mean that convexity holds if arguments are taken in different sectors. And, indeed, in the present problem, one can check that convexity only holds within each sector.

Lemma 4.18 below states that $V_{\mathbf{u}}$ is jointly convex in S_z for all fixed $\mathbf{u}$. This implies that $V(\mathbf{z})$ is jointly convex in S_z. This completes the proof. □

Lemma 4.18. *The function* $f_{\mathbf{u}}(\mathbf{d}) = \min_i (d_{1i} + u_i)$ *is concave in* $\mathbf{d}$*, where* $\mathbf{d} = (d_{12}, \ldots, d_{1l})$ *and* $d_{1i} \equiv 0$.

Proof. Let $\mathbf{d}$ and $\tilde{\mathbf{d}}$ be two instances of the argument of $f_{\mathbf{u}}$. Then, for $\lambda \in [0,1]$,

$$\begin{aligned} f_{\mathbf{u}}((1-\lambda)\mathbf{d} + \lambda\tilde{\mathbf{d}}) &= \min_i((1-\lambda)d_{1i} + \lambda\tilde{d}_{1i} + u_i) \\ &= \min_i((1-\lambda)(d_{1i} + u_i) + \lambda(\tilde{d}_{1i} + u_i)) \\ &\geq (1-\lambda)\min_i(d_{1i} + u_i) + \lambda\min_i(\tilde{d}_{1i} + u_i) \\ &= (1-\lambda)f_{\mathbf{u}}(\mathbf{d}) + \lambda f_{\mathbf{u}}(\tilde{\mathbf{d}}). \end{aligned}$$

This shows concavity.

□

4.6 Strict Displacement Convexity

We now prove that, for two distinct functions y_0 and y_1 in $\mathcal{S}_0''$, the inequality (4.26) is strict whenever $\lambda \neq 0, 1$.

As we already know that the kernel function $V(\mathbf{z})$ is convex in the sector S_z, it is sufficient to show it is *strictly convex as a function of the distances* $\mathbf{d} = (d_{12}, \ldots, d_{1\ell})$ on some *subset of positive measure* of the $\mathbf{d}$-space. This is sufficient because in (4.29) the kernel function is integrated against the measure μ_0 with full support $\mathbb{R}$. Recall that μ_0 has full support because the profile y_0 belongs to $\mathcal{S}''$.

Below, we give explicit formulas for V in terms of the distances d_{1i}. These formulas allow to prove that V is strictly convex in a subset of non-zero measure. Concretely, this subset is a small enough neighborhood of the origin $d_{1i} = 0$, $i = 2, \ldots, \ell$.

We remark that it is not easy to see that V is convex in the whole $\mathbf{d}$-space directly from these formulas.[10] In fact, the formulas show that it is certainly not strictly convex when some of the distances become greater than 1.

4.6.1 Explicit Expressions of Kernel Function

In this section, we compute the kernel function V of Section 4.5.3 and illustrate some of its properties. In particular, we show that it is strictly convex in a set of positive measure.

Recall that the function is totally symmetric under permutations. It is therefore enough to compute it in a fixed sector $S_z = \{\mathbf{z} = (z_1, \cdots, z_\ell) : z_i \geq z_j \text{ if } i < j\}$. We express V in terms of the distances d_{1i}, that are ordered such that $d_{1i} < d_{1j}$ if $i < j$.

We first discuss the explicit examples $\ell = 2$ and $\ell = 3$ and then find a general formula for the kernel. For $\ell = 2$ an explicit computation yields,

$$V_{(\ell=2)}(d_{12}) = \begin{cases} -\frac{1}{2} & \text{if} \quad d_{12} \geq 1, \\ -\frac{1}{2} + \frac{1}{6}(1 - d_{12})^3 & \text{if} \quad d_{12} < 1. \end{cases}$$

---

[10] However we already know that V is convex by the method of proof of the previous section.

By taking the second derivative, it is easy to see that $V_{(\ell=2)}$ is convex everywhere, and strictly convex for $d_{12} < 1$.

For $\ell = 3$, we have $d_{12} < d_{13}$ and the computation yields,

$$V_{(\ell=3)}(d_{12}, d_{13}) = \begin{cases} V_{(\ell=2)}(d_{12}) & \text{if } d_{13} \geq 1, \\ \\ -\frac{1}{2} + \frac{1}{6}(1-d_{12})^3 \\ +\frac{1}{12}(1-d_{13})^4 & \text{if } d_{13} < 1. \\ +\frac{1}{6}d_{12}(1-d_{13})^3 \end{cases}$$

For $d_{13} < 1$ the Hessian is

$$\begin{pmatrix} 1 - d_{12} & -\dfrac{1}{2}(d_{13}-1)^2 \\ -\dfrac{1}{2}(d_{13}-1)^2 & -(d_{13}-1)(1+d_{12}-d_{13}) \end{pmatrix},$$

and the corresponding eigenvalues are

$$\lambda_{1,2} = \frac{1}{2}\{2 - 2d_{13} - d_{12}d_{13} + d_{13}^2 \pm \sqrt{\Delta}\},$$

where

$$\begin{aligned} \Delta = 1 &+ 4d_{12}^2 - 4d_{13} - 8d_{12}d_{13} - 4d_{12}^2 d_{13} + 10d_{13}^2 \\ &+ 8d_{12}d_{13}^2 + d_{12}^2 d_{13}^2 - 8d_{13}^3 - 2d_{12}d_{13}^3 + 2d_{13}^4. \end{aligned}$$

A plot of the eigenvalues shows that they are non-negative in the region $0 \leq d_{12} \leq d_{13} \leq 1$. In fact, one eigenvalue is strictly positive everywhere in this region, and the other is strictly positive everywhere in this region except at the boundary $d_{13} = 1$, where it becomes equal to zero. This is consistent with the fact that $V_{(\ell=3)}(d_{12}, d_{13}) = V_{(\ell=2)}(d_{12})$ when $d_{13} \geq 1$. For $d_{13} \geq 1$, the Hessian always has a vanishing eigenvalue, and a strictly positive one when $d_{12} < 1$. For $d_{12} \geq 1$, the kernel $V_{(\ell=3)}(d_{12}, d_{13})$ is constant and both eigenvalues vanish. To summarize the kernel is always convex, and strictly convex for $0 \leq d_{13} < 1$.

These results can be generalized for all ℓ. We find the general expression of the kernel

$$\begin{aligned} V_\ell(d_{12}, \cdots, d_{1\ell}) = \sum_{k=2}^{\ell} \sum_{m=k}^{\ell} & \frac{(1-d_{1m})^{m-k+3}}{(m-k+3)(m-k+2)} \\ & \times \Big(\sum_{\mathcal{S} \subseteq \{2,\ldots,m-1\}} \prod_{n \in \mathcal{S}} d_{1n} \Big). \end{aligned}$$

The corresponding Hessian (H_{ij}) is a symmetric matrix of dimension $(\ell-1) \times (\ell-1)$ with matrix elements that are polynomials in d_{1i}, $i = 1, \ldots, \ell$. In particular, at $d_{1i} = 0$ for all i we have $H_{ii} = 1$ and $H_{ij} = -\frac{1}{j+1} + \sum_{m=j+1}^{\ell} \frac{1}{(m-1)(m-2)} = -\frac{1}{\ell-1}$; $j > i$. Defining $\mathbf{v}$ as the $(\ell-1)$-dimensional vector of 1's and denoting by $\mathbb{1}$ the $(\ell-1)$-dimensional identity matrix, we remark that H at the origin can be expressed as

$$H = (1 + \frac{1}{\ell-1})\mathbb{1} - \frac{1}{\ell-1}\mathbf{v}\mathbf{v}^{\mathrm{T}}.$$

The eigenvalues of this matrix are $1+\frac{1}{\ell-1}$ and 1 (with $1+\frac{1}{\ell-1}$ having degeneracy $\ell-2$). As these eigenvalues are strictly positive, the Hessian is strictly positive definite at the origin, and thus (by continuity) also in a small neighborhood of the origin. Thus V is a strictly convex function of d_{1i}, $i = 2, \dots, \ell$ in a small neighborhood of the origin.

4.7 Appendix

4.7.1 Proof of Lemma 4.11

Proof of Lemma 4.11. Consider a profile y and a function η such that $\lim_{z\to\pm\infty} \eta(z) = 0$. We compute the directional derivative (also called the functional derivative) of the potential $\mathcal{W}_{\mathrm{BEC}}(y)$ in the direction of η,

$$\frac{\delta \mathcal{W}_{\mathrm{BEC}}}{\delta y}[\eta] = \lim_{\gamma\to 0} \frac{\mathcal{W}_{\mathrm{BEC}}(y+\gamma\eta) - \mathcal{W}_{\mathrm{BEC}}(y)}{\gamma}.$$

A calculation gives

$$\frac{\delta \mathcal{W}_{\mathrm{BEC}}}{\delta y}[\eta] = \int_{\mathbb{R}} \mathrm{d}z\, \eta(z) \Big\{ 1 - (1-y(z))^{\frac{1}{r-1}} - \epsilon_{\mathrm{MAP}} \int_0^1 \mathrm{d}s \Big(\int_0^1 \mathrm{d}u\, y(z+u-s) \Big)^{\ell-1} \Big\}.$$

Now consider the function

$$\eta_y(z) = -\Big\{ 1 - (1-y(z))^{\frac{1}{r-1}} - \epsilon_{\mathrm{MAP}} \int_0^1 \mathrm{d}s \Big(\int_0^1 \mathrm{d}u\, y(z+u-s) \Big)^{\ell-1} \Big\}.$$

The directional derivative of $\mathcal{W}_{\mathrm{BEC}}(y)$ in the direction of η_y satisfies

$$\frac{\delta \mathcal{W}_{\mathrm{BEC}}}{\delta y}[\eta_y] \le 0 \tag{4.31}$$

because the integrand is a square. Now assume that y is a minimizing profile. In the case where equality is met in (4.31), y satisfies the DE equation. Consider the case where the inequality is strict. Then,

$$\frac{\delta \mathcal{W}_{\mathrm{BEC}}}{\delta y}[\eta_y] = \lim_{\gamma\to 0} \frac{\mathcal{W}_{\mathrm{BEC}}(y+\gamma\eta_y) - \mathcal{W}_{\mathrm{BEC}}(y)}{\gamma} < 0,$$

and we can find γ_0 small enough such that

$$\mathcal{W}_{\mathrm{BEC}}(y+\gamma_0\eta_y) < \mathcal{W}_{\mathrm{BEC}}(y).$$

So y cannot be a minimizing profile, and this concludes the proof by contradiction. □

4.7.2 Proof of Lemma 4.12

Proof of Lemma 4.12. If y satisfies the DE equation, then

$$1 - (1-y(z))^{\frac{1}{r-1}} = \epsilon_{\mathrm{MAP}} \int_0^1 \mathrm{d}s \Big(\int_0^1 \mathrm{d}u\, y(z+u-s) \Big)^{\ell-1}$$

By taking the derivative with respect to z on each side, we find that (with y' denoting the derivative of y)

$$\begin{aligned}(1-y(z))^{\frac{2-r}{r-1}}\frac{1}{r-1}y'(z) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad & \\ = \epsilon_{\mathrm{MAP}}\int_0^1 \mathrm{d}s\,(l-1)\Big(\int_0^1 \mathrm{d}u\, y(z+u-s)\Big)^{\ell-2}\int_0^1 \mathrm{d}w\, y'(z+w-s). &\end{aligned} \tag{4.32}$$

Notice that $\int_0^1 \mathrm{d}v\, y'(z+v-s) = \int_0^1 \mathrm{d}v\, \frac{\mathrm{d}}{\mathrm{d}v}y(z+v-s) = y(z+1-s)-y(z-s)$.

Now assume that there exists a flat spot of y for $z \in (a,b)$ where it takes some value y_{flat}. We consider "maximal" intervals (a,b) such that y takes values different than y_{flat} for all $z \notin (a,b)$. On this flat spot, (4.32) becomes

$$0 = \int_0^1 \mathrm{d}s\Big(\int_0^1 \mathrm{d}u\, y(z+u-s)\Big)^{\ell-2}\big(y(z+1-s)-y(z-s)\big). \tag{4.33}$$

We will now show that this equality cannot be satisfied.

Let us first consider the case when a and b are finite. As (a,b) is maximal and y is non-decreasing, we know that $0 < y_{\text{flat}} < y_{\mathrm{MAP}}$, $y(z) < y_{\text{flat}}$ for all $z < a$ and $y(z) > y_{\text{flat}}$ for all $z > b$. Now let us fix $z \in [b-1+\delta_0, b]$, where $0 < \delta_0 < 1$ is chosen so that the equality (4.33) holds (the derivative is zero). But for such z and for all $0 \le s < 1$, $\int_0^1 \mathrm{d}u\, y(z+u-s) \ge \int_{u>s} \mathrm{d}u\, y(z+u-s) \ge (1-s)y_{\text{flat}} > 0$. Thus we should have $y(z+1-s) = y(z-s)$ almost everywhere for $s \in [0,1]$ for (4.33) to hold. This is not possible. Indeed, take $s \in [0,\delta_1]$ with $0 < \delta_1 < \delta_0$ is small enough so that $z-s < b < z+1-s$ and thus $y(z+1-s)-y(z-s) > y(b+\delta_0-\delta_1)-y(b) > 0$. These arguments prove that a non-decreasing solution of DE cannot be flat for $z \in [b-1+\delta_0, b]$. We repeat the argument on $[b-k+\delta_0, b-(k-1)+\delta_0]$ for all $1 < k < K$ such that $b-K+\delta_0 < a$ and find that a non-decreasing solution of DE cannot be flat on each of those intervals, and thus on $[b-(K-1)+\delta_0, b]$. Finally, we repeat the argument on the last interval $[a, b-(K-1)+\delta_0]$ and deduce that a non-decreasing solution of DE cannot be flat on $[a,b]$.

Next, we consider the case when $a = -\infty$. In this case, we have that $y(z) = 0$ for all $z \le b$ and $y(z) > 0$ for all $z > b$. The analysis is similar to the preceding one. First fix z in the interval $\in [b-1+\delta_0, b]$, $0 < \delta_0 < 1$. The derivative $y^{(}z)$ is zero for such a value of z and so the equality (4.33) is satisfied. Now take $s \in [0,\delta_1]$ with $0 < 2\delta_1 < \delta_0$. Then $y(z+1-s)-y(z-s) = y(z+1-s) > y(b+\delta_0-\delta_1) > 0$, and for $1-\delta_0+2\delta_1 < u < 1$ we have $y(z+u-s) > y(b+\delta_1)$ so $\int_0^1 \mathrm{d}u\, y(z+u-s) > (1+\delta_0-2\delta_1)y(b+\delta_1) > 0$. Thus the right hand side of (4.33) does not vanish which is a contradiction. We carry out the same analysis as above for $z \in [b-k+\delta_0, b-(k-1)+\delta_0]$ for $k \in \mathbb{N}$, and thus deduce that y cannot be flat on $[-\infty, b]$.

Finally, consider the case when $b = +\infty$. The analysis is essentially symmetric to the preceding one. In this case we have $y(z) = y_{\mathrm{MAP}}$ for $z > a$ and $y(z) < y_{\mathrm{MAP}}$ for $z < a$. First fix z in the interval $[a, a+1-\delta_0]$. For such z (4.33) holds. For $s \in [1-\delta_1, 1]$ with $\delta_1 < \delta_0$ we have $y(z+1-s)-y(z-s) > y_{\mathrm{MAP}} - y(a-\delta_0+\delta_1) > 0$. Moreover it is clear that $\int_0^1 \mathrm{d}u\, y(z+u-s) > 0$. So the right hand side of (4.33) cannot vanish, and we arrive at a contradiction. We repeat the argument for $z \in [a+k-\delta_0, a+(k+1)-\delta_0]$ for $k \in \mathbb{N}$, and conclude that y cannot be flat on $[a, +\infty]$.

□

4.7.3 Proof of Lemma 4.13

Proof of Lemma 4.13. Suppose that y is a non-decreasing minimizing profile and that it has a constant value $0 < y_{\text{flat}} < y_{\text{MAP}}$ on a bounded interval of length greater than 1. We will construct another profile that has strictly less energy.

We start by expressing the single potential as follows

$$\begin{aligned}\mathcal{W}_{\text{single}}(y) &= \int_{-\infty}^{a} \mathrm{d}z\, W_{\text{s}}(y(z)) + \int_{a}^{b-1} \mathrm{d}z\, W_{\text{s}}(y(z)) + \int_{b-1}^{+\infty} \mathrm{d}z\, W_{\text{s}}(y(z)) \\ &= \int_{-\infty}^{a} \mathrm{d}z\, W_{\text{s}}(y(z)) + (b-1-a)W_{\text{s}}(y(a)) + \int_{b-1}^{+\infty} \mathrm{d}z\, W_{\text{s}}(y(z)).\end{aligned}$$

By applying the change of variables $\tilde{z} = z - (b-1) + a$ on the rightmost integral, we express it as

$$\int_{b-1}^{+\infty} \mathrm{d}z\, W_{\text{s}}(y(z)) = \int_{a}^{+\infty} \mathrm{d}z\, W_{\text{s}}(y(z+b-1-a)).$$

We define the profile $\tilde{y}$ by

$$\tilde{y}(z) = \begin{cases} y(z) & \text{if} \quad z \le a, \\ y(z+b-1-a) & \text{if} \quad z > a, \end{cases}$$

and remark that it is also non-decreasing. We thus obtain

$$\mathcal{W}_{\text{single}}(y) = \mathcal{W}_{\text{single}}(\tilde{y}) + (b-1-a)W_{\text{s}}(y_{\text{flat}}).$$

Note that as $0 < y_{\text{flat}} < y_{\text{MAP}}$ we have $W_{\text{s}}(y_{\text{flat}}) > 0$. Thus $\mathcal{W}_{\text{single}}(y) > \mathcal{W}_{\text{single}}(\tilde{y})$.

For the interaction potential, we prove that $\mathcal{W}_{\text{int}}(y) = \mathcal{W}_{\text{int}}(\tilde{y})$. Indeed,

$$\begin{aligned}\mathcal{W}_{\text{int}}(y) = \frac{\epsilon_{\text{MAP}}}{\ell} \int_{\mathbb{R}} \mathrm{d}z \Big\{ y(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, y(z+u) \Big)^{\ell} \Big\} \\ = \frac{\epsilon_{\text{MAP}}}{\ell} \int_{-\infty}^{a} \mathrm{d}z \Big\{ y(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, y(z+u) \Big)^{\ell} \Big\} \quad (4.34) \\ + \frac{\epsilon_{\text{MAP}}}{\ell} \int_{a}^{b-1} \mathrm{d}z \Big\{ y(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, y(z+u) \Big)^{\ell} \Big\} \quad (4.35) \\ + \frac{\epsilon_{\text{MAP}}}{\ell} \int_{b-1}^{+\infty} \mathrm{d}z \Big\{ y(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, y(z+u) \Big)^{\ell} \Big\}. \quad (4.36)\end{aligned}$$

We use the same definition of $\tilde{y}$ as above, and denote the functionals in (4.34), (4.35), and (4.36) by $\mathcal{T}_1(y)$, $\mathcal{T}_2(y)$, and $\mathcal{T}_3(y)$ respectively. Observe that

- $\mathcal{T}_1(y) = \mathcal{T}_1(\tilde{y})$ because

$$\begin{aligned} y(z) &= \tilde{y}(z) \quad \text{if} \quad z < a \\ y(z+u) &= \tilde{y}(z+u) \quad \text{if} \quad z < a, \text{ and } 0 < u < 1. \end{aligned}$$

- $\mathcal{T}_2(y) = 0$ because $y(z) = y(z+u) = y_{\text{flat}}$ when $a < z < b-1$.

- $\mathcal{T}_3(y) = \mathcal{T}_2(\tilde{y}) + \mathcal{T}_3(\tilde{y})$ because, by the change of variables $\tilde{z} = z - (b-1) + a$, we have

$$\begin{aligned}\mathcal{T}_3(y) &= \frac{\epsilon_{\text{MAP}}}{\ell} \int_{a}^{+\infty} \mathrm{d}z \Big\{ y(z+b-1-a)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, y(z+b-1-a+u) \Big)^{\ell} \Big\} \\ &= \frac{\epsilon_{\text{MAP}}}{\ell} \int_{a}^{+\infty} \mathrm{d}z \Big\{ \tilde{y}(z)^{\ell} - \Big(\int_0^1 \mathrm{d}u\, \tilde{y}(z+u) \Big)^{\ell} \Big\}.\end{aligned}$$

Thus $\mathcal{T}_1(y) + \mathcal{T}_2(y) + \mathcal{T}_3(y) = \mathcal{T}_1(\tilde{y}) + \mathcal{T}_2(\tilde{y}) + \mathcal{T}_3(\tilde{y})$

Combining these results we obtain

$$\mathcal{W}_{\mathrm{BEC}}(y) = \mathcal{W}_{\mathrm{BEC}}(\tilde{y}) + (b - 1 - a)\, W_{\mathrm{s}}(y_{\mathrm{flat}}) > \mathcal{W}_{\mathrm{BEC}}(\tilde{y})$$

□

Displacement Convexity for General Scalar Systems

5

5.1 Introduction

Considering the excellent performance that spatially coupled systems have exhibited in various frameworks, such as coding, compressive sensing, statistical physics, and random constraint satisfaction problems, it should hardly come as a surprise that there are fundamental mathematical structures behind spatially coupled systems. This chapter is concerned with the somewhat hidden convexity structure, called *displacement convexity*, in coupled systems that are governed by a set of scalar density-evolution (DE) equations. This structure was introduced by McCann [92] and is well known in the theory of optimal transport [93].

In this chapter, we consider general coupled systems governed by scalar DE (or DE-like) equations.[1] We look at the system during the *static phase*, when the system parameter is equal to the static threshold and the DE equations can no longer modify the state of the system. In the context of coding, for instance, this corresponds to the case when the channel parameter is equal to the maximum a-posteriori (MAP) threshold. The definition is different for more general systems (see Section 5.2). We define an associated potential functional whose stationary point equations are the fixed points of the DE equations. Then, we use displacement convexity to establish the strict convexity of this potential. This proves that the potential admits a unique minimizer. By finding the space of profiles to which this minimizer belongs, we characterize the unique fixed point of the coupled system's DE equations.

Displacement convexity is a mathematical tool that was first introduced by McCann to describe interacting gases and equilibrium crystals [92] and later applied to various problems, including optimal transport [93, 115, 116] and probability the-

---

[1] The content of this chapter is based on previous work [73, 75].

[73] R. El-Khatib, N. Macris, T. Richardson, R. Urbanke, *"Analysis of coupled scalar systems by displacement convexity,"* International Symposium on Information Theory (ISIT) 2014, Honolulu, Hawaii, IEEE pp. 2321 – 2325.

[75] R. El-Khatib, N. Macris, T. Richardson, R. Urbanke, *"Displacement convexity in spatially coupled scalar recursions,"* in preparation (2016) to be submitted to the IEEE Transactions on Information Theory.

ory [117, 118]. A functional that is displacement convex is essentially convex with respect to an alternative structure in the space of probability measures. In order to apply this tool on the potential functional that we consider in this chapter, we will first show that the profiles that solve the DE equations can be seen as probability measures, and then give mild conditions under which the potential is (strictly) convex, with respect to the alternative structure.

As usual, we construct a spatially coupled system from an uncoupled one by placing $L_c + W$ copies of the latter on a spatial axis and coupling them locally within some window of size W. We run an iterative message-passing algorithm on the spatially coupled system and assess its large-system asymptotic performance by using the solutions of the associated coupled DE-like equations. The fixed points of these equations can be viewed as the stationary point equations of the potential functional. In the continuum limit $L_c \gg W \gg 1$, the DE equations and the potential become continuous space objects.

In Chapter 4, we considered the spatially coupled (ℓ, r)-regular Gallager ensemble with transmission over the binary erasure channel (BEC). Here, we instead consider general scalar coupled systems (with general coupling window functions), but also restrict our analysis to the *static phase*. Although the formalism in Chapter 4 can be extended to a few general scalar recursions, such as those pertaining to irregular LDPC codes, it does not appear to extend to a very wide class of general scalar recursions. The main purpose of the present chapter is to prove that a rather general class of scalar systems also exhibits the property of displacement convexity, and even strict displacement convexity, under rather mild assumptions. Although the analysis in the present chapter is similar in spirit to that in Chapter 4, it is also significantly different and more far-reaching in its range of applications.

We mention the main differences between the approaches taken in Chapter 4 and the present chapter. There are various possible formulations of the potential functional. Here, we use the representation from [69,70] for scalar systems, expressed in terms of both profiles x and y, whereas in the previous chapter, we consider a potential functional expressed solely in terms of the profile y. The functional we use in the present chapter enables us to obtain much more general proofs that hold under quite mild conditions. Moreover, here we allow the update functions f and g to have discontinuities, a property that the update functions in Chapter 4 inherently lack (in the case of the specific code ensemble we consider). Although we assume, in Chapter 4, that the profiles we consider have high (enough) rates of convergence towards their limits, in this chapter we circumvent this restriction by using analysis via "K-saturated profiles" that are cut off to the function's left limit when the spatial coordinate is smaller than $-K$, and to the right limit when the spatial coordinate is larger than K (see Definition 5.9).

In the present chapter, we prove that, under mild conditions, the potential functional governing the general coupled system is strictly displacement convex in the space of non-decreasing functions with finite left and right limits (that we define later), and that it attains its minimum in this space. These conditions can be summarized as the strictly positive gap condition that was defined in [69, 70], as well as some boundedness conditions on the coupling window.

At first, we begin our analysis with general profiles that have normalized images $[0, 1]$ and normalized limits 0 and 1 at $-\infty$ and $+\infty$, respectively, on the spatial axis of coupling. Then, using rearrangement inequalities, we prove that the coupled potential functional yields smaller values with the *rearranged non-decreasing* counterparts of these functions than with the original ones. This enables us to restrict our search for minimizers of the potential to the space of non-decreasing functions with the above limits. At this point, we can view the profiles as cumulative distribution functions (cdf's), which is the right setting for displacement convexity. Using the direct method in the calculus of variations [109], we prove that the potential *attains* its minimum in this space of profiles, as we have done in Chapter 4. We note that this existence result was proved using a different technique in [69, 70].

To prove that the potential functional is displacement convex, we first split it into two contributions that we call the single and interaction potentials. We then prove that the single potential is affine in the interpolation parameter λ and the interaction potential is convex in it. To do this, we first prove these properties for "K-saturated" non-decreasing profiles, then we map the result back to more general non-decreasing ones that have limits 0 and 1 and $-\infty$ and $+\infty$, respectively. Finally, we use displacement interpolation and again some saturation properties to prove that the minimizing pair of profiles is unique up to translation.

The main propositions of this chapter are: (i) Proposition 5.17 that states that the potential functional has the displacement convexity property; (ii) Proposition 5.21 that asserts that monotonic minimizers of the potential functional are fixed-point solutions of the spatially coupled DE equations (in a generalized sense); (iii) Proposition 5.27 that gives the condition for unicity of the minimizers up to translations along the spatial axis. It is also of interest that the potential functional satisfies a rearrangement inequality, namely (iv) Proposition 5.15 that ensures that one can find minimizers among monotonic spatial fixed points. The conditions for our results to hold are rather mild and essentially match those in [69, 70] for the existence of spatial fixed points.

This chapter is organized as follows. In Section 5.2, we introduce spatially coupled recursions and the variational formulation. In Section 5.3, we prove rearrangement inequalities that allow us to reduce the search for minima of the potential to a space of monotonic functions, and in Section 5.4, we discuss the existence question using the direct method from functional analysis. In Section 5.5, we prove that the potential is displacement convex. In Section 5.6, we generalize the notion of fixed-point solutions to the DE equations and show that such generalized solutions are minimizers of the potential. In Section 5.7, we address the unicity of the minimizer, and finally, in Section 5.8, we illustrate displacement convexity with applications to coding and compressive sensing.

5.2 Preliminaries

5.2.1 Single System

In this section, we explain the set-up for general spatially coupled scalar recursions and give a variational formulation of these recursions. The fixed-point equations of the scalar recursions will be generically called "density evolution" (DE) equations. The case of the (ℓ, r)-regular LDPC code ensemble with transmission over the BEC(ϵ) will serve as a concrete running example for the setting. We note that, although this specific application has been analyzed in detail in Chapter 4, the approach we take in the present one is different, as it considers a system of DE equations, instead of only one DE recursion.

Consider the pair of DE fixed point equations

$$\begin{cases} u = g(v), \\ v = f(u), \end{cases} \tag{5.1}$$

where $u, v \in [0,1]$. The *update functions* f, g are assumed to be non-decreasing from $[0,1]$ to $[0,1]$, and normalized such that $f(0) = g(0) = 0$ and $f(1) = g(1) = 1$. We will think of them as EXIT-like curves of DE $(u, f(u))$ and $(g(v), v)$ for $u, v \in [0,1]$ (see Figure 5.1). It is always possible to adopt this normalization in specific applications.

Example 5.1. *Take an (ℓ, r)-regular Gallager ensemble, with transmission over the BEC(ϵ). Let $\mathtt{y}$ (resp. $\mathtt{x}$) be the erasure probabilities emitted by the check (resp. variable) nodes. The DE fixed-point equations are $\mathtt{y} = 1 - (1 - \mathtt{x})^{r-1}$ and $\mathtt{x} = \epsilon \mathtt{y}^{\ell-1}$. In this chapter, we are interested in the specific value of the channel parameter $\epsilon = \epsilon_{MAP}$. We denote by $\mathtt{x}_{MAP}$, $\mathtt{y}_{MAP}$ the non-trivial stable fixed point when $\epsilon = \epsilon_{MAP}$. To achieve the normalization of (5.1), we make the change of variables $\mathtt{y} = \mathtt{y}_{MAP} u$ and $\mathtt{x} = \mathtt{x}_{MAP} v$, so that the DE equations become $u = \mathtt{y}_{MAP}^{-1}(1 - (1 - \mathtt{x}_{MAP} v)^{r-1})$ and $v = \epsilon_{MAP} \mathtt{x}_{MAP}^{-1} \mathtt{y}_{MAP}^{\ell-1} u^{\ell-1}$. Note that we must have $1 = \mathtt{y}_{MAP}^{-1}(1 - (1 - \mathtt{x}_{MAP})^{r-1})$ and $1 = \epsilon_{MAP} \mathtt{x}_{MAP}^{-1} \mathtt{y}_{MAP}^{\ell-1}$. We then set*

$$\begin{cases} g(v) = \mathtt{y}_{MAP}^{-1}(1 - (1 - \mathtt{x}_{MAP} v)^{r-1}), \\ f(u) = u^{\ell-1}, \end{cases} \tag{5.2}$$

that satisfy the required normalizations $f(0) = g(0) = 0$ and $f(1) = g(1) = 1$. The corresponding EXIT curves have three intersections. The one at $(0,0)$ corresponds to the trivial fixed point of DE, the one at $(1,1)$ corresponds to the stable non-trivial fixed point of DE, and the third one at some middle point corresponds to the unstable fixed point.

We will use the "single" potential function associated to the DE equations (5.1); it was introduced in [69, 70] and is defined as

$$\phi(f, g; u, v) = \int_0^u \mathrm{d}\tilde{u}\, g^{-1}(\tilde{u}) + \int_0^v \mathrm{d}\tilde{v}\, f^{-1}(\tilde{v}) - uv. \tag{5.3}$$

When the update functions f and g are clear from context or irrelevant, we often drop them as arguments from the notation and denote this potential function by $\phi(u, v)$. As g^{-1} and f^{-1} are non-decreasing, $\phi(u, v)$ is convex in u for fixed v, and convex in v for fixed u. It is minimized over v by setting $v = f(u)$, and over u by setting $u = g(v)$.

By substituting $v = f(u)$ in (5.3), we obtain the integral of the *signed area* between the two EXIT curves (see fig. 5.1) as

$$\begin{aligned} A(f, g; u) &= \phi(f, g; u, f(u)) \\ &= \int_0^u \mathrm{d}\tilde{u}\, (g^{-1}(\tilde{u}) - f(\tilde{u})). \end{aligned} \tag{5.4}$$

The signed area A and the potential function ϕ prove to be crucial quantities in our analysis and setting. For instance, the following result was shown in [69, 70] that relates them to the DE system of equations (5.6) that corresponds to the spatially coupled system (that we describe in Section 5.2.2).

Lemma 5.2. *If there exists an interpolating fixed point solution to* (5.6)*, then*

$$\phi(f,g;u,v) \geq 0,$$

for all $u, v \in [0,1]$ *and* $A(f,g;1) = \phi(f,g;1,1) = 0$.

The result applies not only to interpolating fixed points, but also to a relaxed definition of interpolating "consistent" fixed points that we define in Section 5.6. In [69,70], when the assumption $\phi(f,g;1,1) = 0$ is made, the condition $\phi(f,g;u,v) \geq 0$ for all $u, v \in [0,1]$ is termed the *positive gap condition* (PGC). In this chapter, however, we will assume that $\phi(f,g;1,1) = 0$ throughout, so the term "positive gap condition" will be used to imply both this equality and the inequality in Lemma 5.2.

When the inequality in Lemma 5.2 is strict, i.e., $\phi(f,g;u,v) > 0$ for $(u,v) \notin \{(0,0),(1,1)\}$, the condition is termed the *strictly positive gap condition* (SPGC) in [69,70]. In this case it was shown that an interpolating fixed-point profile exists, provided that the coupling-window function w is strictly positive on the interior of some interval $[-W, W]$ and zero off of the interval. This support condition on the function w can be relaxed under various other conditions, see [69,70].

Definition 5.3. *We say that the positive gap condition (PGC) is satisfied when* $\phi(f,g;1,1) = 0$ *and* $\phi(f,g;u,v) \geq 0$ *for all* $u, v \in [0,1]$. *The strictly positive gap condition is satisfied when* $\phi(f,g;1,1) = 0$ *and* $\phi(f,g;u,v) > 0$ *for* $(u,v) \notin \{(0,0),(1,1)\}$.

Example 5.4. *For the* (ℓ, r)*-regular Gallager ensemble with transmission over the BEC*(ϵ)*, when* $\epsilon = \epsilon_{MAP}$*, we have the potential function*

$$\phi(u,v) = \frac{1}{\mathtt{x}_{MAP}} \Big\{ u - \frac{r-1}{\mathtt{y}_{MAP} r} \big(1 - (1 - \mathtt{y}_{MAP} u)^{\frac{r}{r-1}}\big) \Big\} + \frac{\ell - 1}{\ell} v^{\frac{\ell}{\ell-1}} - uv,$$

and the signed area

$$A(f,g;u) = \frac{1}{\mathtt{x}_{MAP}} \Big\{ u + \frac{r-1}{\mathtt{y}_{MAP} r} \big((1 - \mathtt{y}_{MAP} u)^{\frac{r}{r-1}} - 1\big) \Big\} + \frac{u^{\ell}}{\ell}.$$

Moreover, we have $A(f,g;1) = 0$. *In fact, this last constraint together with the two fixed point equations* $\mathtt{y}_{MAP} = 1 - (1 - \mathtt{x}_{MAP})^{r-1}$ *and* $\mathtt{x}_{MAP} = \epsilon_{MAP} \mathtt{y}_{MAP}^{\ell-1}$ *completely determine* ϵ_{MAP}, $\mathtt{x}_{MAP}$ *and* $\mathtt{y}_{MAP}$. *The SPGC holds for this example (see Section 5.8 for further illustration).*

5.2.2 Spatially Coupled System

To obtain the spatially coupled version of the "single" system above, we place $L_c + W$ replicas of the latter on the positions $z = -W+1, \ldots, L_c$ of the spatial axis, and reconnect their edges locally within a coupling window, so that the original degree distributions of the nodes are preserved.[2]

The natural setting for displacement convexity of spatial coupling is the continuum setting that can be thought of as an approximation of the corresponding discrete system in the regime of large spatial length $L_c \to +\infty$ and coupling window

---

[2] We loosely define the construction of the spatially coupled ensemble in this chapter because it has been described in all the previous chapters. One can consult with Section 1.5 for more details on this construction.

size $W \to +\infty$. The continuum limit has already been introduced in the literature as a convenient means to analyze the behavior of an originally discrete model [54, 55, 69, 70].

The potential functional for the spatially coupled system is described in discrete form in (1.67) and in the continuum limit in (1.83). We write the coupled system potential here directly in the continuum limit. Consider a spatially coupled system with an averaging window function $w : \mathbb{R} \to \mathbb{R}$ that is always assumed to be bounded, non-negative, even, integrable, and normalized such that $\int_{\mathbb{R}} \mathrm{d}x\, w(x) = 1$. The averaging window is the means for "coupling" in "spatial coupling". So far in the thesis, we have only considered the uniform averaging window, but in the present chapter we consider the more general window function that we have just described. Let us define the constant

$$C_w := \int_{\mathbb{R}} \mathrm{d}z\, |z| w(z). \tag{5.5}$$

We assume throughout the chapter that C_w is finite. As we shall see, this is directly related to the finiteness of the potential. Let $x, y : \mathbb{R} \to [0,1]$ be two functions and denote by $x^w = x \otimes w$ and $y^w = y \otimes w$ their usual convolution with w, i.e., $x^w(z) = \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, x(\tilde{z}) w(z - \tilde{z})$ and $y^w(z) = \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, y(\tilde{z}) w(z - \tilde{z})$. The pair of fixed-point DE equations of a spatially coupled scalar continuous system are

$$\begin{cases} y(z) = g(x^w(z)), \\ x(z) = f(y^w(z)), \end{cases} \tag{5.6}$$

where $z \in \mathbb{R}$ is the spatial position. We refer to the functions x, y as *profiles* and to f, g as *update functions.* A pair of profiles $x, y : \mathbb{R} \to [0,1]$ that solves the above equations almost everywhere is called a *fixed point.* Note that (5.6) are *non-local* equations.

In this chapter, we are interested in profiles $p : \mathbb{R} \to [0,1]$ (p denotes a generic profile like x and y) that satisfy the limit conditions

$$\lim_{z \to -\infty} p(z) = 0, \qquad \lim_{z \to +\infty} p(z) = 1. \tag{5.7}$$

We note that these two limit values are the extreme fixed points of (5.1). We refer to such profiles as *interpolating profiles.* A pair x, y of interpolating profiles that solves (5.6) is called an *interpolating fixed point.*

Definition 5.5. *A function $p : \mathbb{R} \to [0,1]$ that satisfies* (5.7) *is called an interpolating profile. A pair x, y of interpolating profiles that solves* (5.6) *almost everywhere, i.e., up to a set of measure zero, is called an interpolating fixed point.*

In Section 5.3 we show that, when minimizing the potential functional over the space of interpolating profiles, we can focus on monotonic (non-decreasing) profiles.

The solutions of spatially coupled DE equations (5.6) are given by the stationary points of a *potential functional* $\mathcal{W}_2$ of x and y that we define in (1.83). We repeat it below for convenience. This can be checked by setting the functional derivatives of this potential functional with respect to each of x and y to zero. We set

$$\mathcal{W}_2(x, y) = \int_{\mathbb{R}} \mathrm{d}z\, I_{x,y,w}(z), \tag{5.8}$$

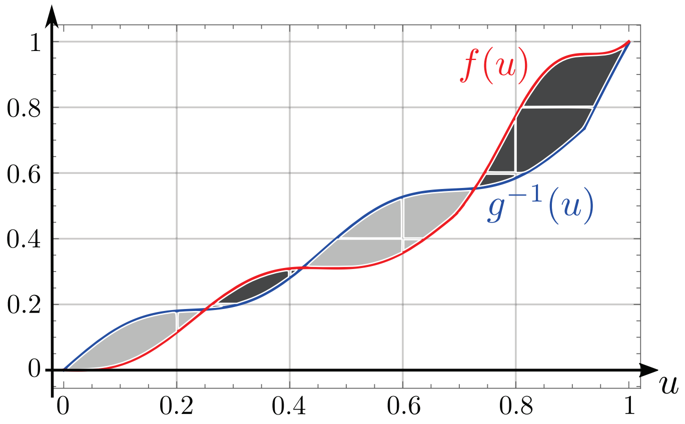

Figure 5.1: A generic example of the systems we consider. The EXIT-like curves are f (in red) and g^{-1} (in blue). The signed area $A(f, g; 1)$ from (5.4) is the sum of the light gray areas (positively signed) and the dark gray areas (negatively signed), and it is equal to 0.

where we have introduced the notation

$$I_{x,y,w}(z) = \int_0^{y(z)} \mathrm{d}u\, g^{-1}(u) + \int_0^{x(z)} \mathrm{d}v\, f^{-1}(v) - x^w(z)y(z). \tag{5.9}$$

We note that the potential above is related to that in (1.83) by

$$\begin{aligned} \mathcal{W}_2(x, y) &= \Delta\mathcal{W}_2(x, y) + \int_{\mathbb{R}} \mathrm{d}z\, P_2(z, x_0, y_0) \\ &= \int_{\mathbb{R}} \mathrm{d}z\, P_2(z, x, y), \end{aligned}$$

Here, we do not have to subtract a reference energy as done in (1.83). As explained in the introduction, we consider the static phase in this chapter. Due to the paragraph below, as well as some "saturation" properties that we prove, we can see that this integral is well-defined.

Example 5.6. *For the* (ℓ, r)*-regular LDPC code and transmission over the BEC(*ϵ*) the potential* (5.8) *is*

$$\begin{aligned} \mathcal{W}_2(x, y) = \int_{\mathbb{R}} \mathrm{d}z \Big\{ \frac{1}{\mathtt{x}_{MAP}} \{y(z) - \frac{r-1}{\mathtt{y}_{MAP} r}(1 - (1 - \mathtt{y}_{MAP} y(z))^{\frac{r}{r-1}})\} \\ + \frac{\ell - 1}{\ell} x(z)^{\frac{\ell}{\ell-1}} - x^w(z)y(z) \Big\}. \end{aligned}$$

Note that the limit of the integrand in (5.8) (and the example) vanishes when $z \to -\infty$ because of the condition (5.7) on the profiles. It also vanishes when $z \to +\infty$ because of (5.7) and $A(f, g; 1) = 0$. However, this does not suffice for the existence of the integral, essentially due to the fact that $x^w - x$ may not be Lebesgue integrable (for monotonic profiles this difficulty does not arise). So, it is possible that $\mathcal{W}_2(x, y)$ fails to be well-defined as a Lebesgue integral for some choices of the interpolating profiles.

Once we consider interpolating profiles and assume the PGC and that $C_w < \infty$, we can circumvent this technical issue by defining the potential functional as

$$\mathcal{W}_2(x,y) = \lim_{A,B\to\infty} \int_{-A}^{B} \mathrm{d}z I_{x,y,w}(z). \tag{5.10}$$

We show below that this limit always exists (it is possibly $+\infty$).

Lemma 5.7. *Assuming the PGC, for any interpolating profile pair x, y, we have that*

$$\mathcal{W}_2(x,y) \geq \int_{\mathbb{R}} \mathrm{d}z\, \phi(x^w(z), y(z)), \tag{5.11}$$

and, given a sequence of interpolating pairs x_i, y_i converging pointwise almost everywhere to the interpolating pair x, y, we have

$$\liminf_{i\to\infty} \mathcal{W}_2(x_i, y_i) \geq \mathcal{W}_2(x,y). \tag{5.12}$$

Proof. Define $H_f(x) = \int_0^x \mathrm{d}v\, f^{-1}(v)$ and $H_g(y) = \int_0^y \mathrm{d}u\, g^{-1}(u)$. We use the notation $(H_f \circ x)(z) = H_f(x(z))$ and $(H_g \circ y)(z) = H_g(y(z))$. Note that

$$I_{x,y,w} = (H_g \circ y) + (H_f \circ x) - x^w y.$$

Now, if we define

$$\tilde{I}_{x,y,w} = (H_g \circ y) + (H_f \circ x)^w - x^w y,$$

then

$$\begin{aligned}
\int_{-A}^{B} \mathrm{d}z\, I_{x,y,w}(z) &= \int_{-A}^{B} \mathrm{d}z\, \tilde{I}_{x,y,w}(z) + \int_{-A}^{B} \mathrm{d}z\, (I_{x,y,w}(z) - \tilde{I}_{x,y,w}(z)) \\
&= \int_{-A}^{B} \mathrm{d}z\, \tilde{I}_{x,y,w}(z) + \int_{-A}^{B} \mathrm{d}z\, ((H_f \circ x) - (H_f \circ x)^w(z)).
\end{aligned}$$

Taking the limits $A, B \to +\infty$ by definition (5.10) and Lemma 5.29 we obtain

$$\mathcal{W}_2(x,y) = \lim_{A,B\to+\infty} \int_{-A}^{B} \mathrm{d}z\, \tilde{I}_{x,y,w}(z).$$

We will shortly see that the PGC implies that $\tilde{I}_{x,y,w}(z)$ is non-negative so that $\mathcal{W}_2(x,y)$ is well-defined (it is possibly $+\infty$). This also means that it is possible to adopt

$$\mathcal{W}_2(x,y) = \int_{\mathbb{R}} \mathrm{d}z\, \tilde{I}_{x,y,w}(z), \tag{5.13}$$

as an alternative expression for $\mathcal{W}_2(x,y)$.

Now, note that $H_f(x)$ and $H_g(y)$ are convex functions because f^{-1} and g^{-1} are non-decreasing. Indeed,

$$\begin{aligned}
H_f(x+a) - H_f(x) &= \int_x^{x+a} \mathrm{d}v\, f^{-1}(v) \\
&\geq a f^{-1}(x) \\
&= a H_f'(x).
\end{aligned}$$

By Jensen's inequality we have

$$(H_f \circ x)^w \geq (H_f \circ x^w),$$

and we therefore obtain

$$\tilde{I}_{x,y,w}(z) \geq \phi(x^w(z), y(z)), \tag{5.14}$$

that proves the non-negativity of $\tilde{I}_{x,y,w}(z)$ because $\phi(x^w(z), y(z))$ is non-negative by the PGC.

Integrating (5.14) and using (5.13), we obtain the first claim (5.11) of the lemma. Furthermore, we obtain the second claim (5.12) directly by applying Fatou's lemma to (5.13) (we can apply Fatou's lemma because by (5.14) $\tilde{I}_{x_i,y_i,w}$ is a non-negative sequence, and it converges to $\tilde{I}_{x,y,w}$). ☐

Let us remark that, in the process of proving this lemma, we have seen that $\mathcal{W}_2(x,y)$ can be defined as (5.10) or, equivalently, as (5.13), as long as we assume the PGC, interpolating profiles, and $C_w < +\infty$.

5.2.3 Discussion

In Section 5.6, we show that among all interpolating profiles, monotonic interpolating consistent fixed points are minimizers of $\mathcal{W}_2$. To do this, we use rearrangement properties that are summarized in Section 5.3. For a fixed x, we always have $\mathcal{W}_2(x,y) \geq \mathcal{W}_2(x, g \circ x^w)$. This is because $I_{x,y,\omega}(z)$ is convex in $y(z)$, for fixed $x(z)$, and setting $y(z) = g(x^w(z))$ minimizes $I_{x,y,\omega}(z)$ over $y(z)$, for fixed $x(z)$.

One of the main results of this chapter is to show the *displacement convexity* of $\mathcal{W}_2$ in its two arguments. More precisely, we can think of interpolating between two pairs (x_0, y_0) and (x_1, y_1) of *monotonic* profiles by interpolating their inverse functions. Hence, we consider

$$\begin{cases} x_\lambda^{-1} & = (1-\lambda)x_0^{-1} + \lambda x_1^{-1}, \\ y_\lambda^{-1} & = (1-\lambda)y_0^{-1} + \lambda y_1^{-1}, \end{cases}$$

and show that $\mathcal{W}_2(x_\lambda, y_\lambda)$ is a convex function of λ. Note that, for a monotonic interpolating profile p, the inverse function $p^{-1}(u)$ is uniquely defined for almost all $u \in (0,1)$; also, the right and left limits, $p^{-1}(u+)$ and $p^{-1}(u-)$, respectively, are uniquely determined. Displacement convexity is explained in more detail in Sections 1.7 and 5.5.

Displacement convexity applies only to monotonic profiles. In the next section, we address the conditions under which one can conclude that minimizers of $\mathcal{W}_2$ satisfying (5.7) can be taken to be monotonic.

The following quantities will play a crucial role in the remainder of the chapter,

$$\Omega(z) = \int_{-\infty}^{z} \mathrm{d}\tilde{z}\, w(\tilde{z}), \qquad V(z) = \int_{-\infty}^{z} \mathrm{d}\tilde{z}\, \Omega(\tilde{z}). \tag{5.15}$$

Here V is called *the kernel* for reasons that will become clear. As will be seen, displacement convexity arises from the convexity of V.

Lemma 5.8. *Assume that $C_w < \infty$. Then V is well-defined and convex.*

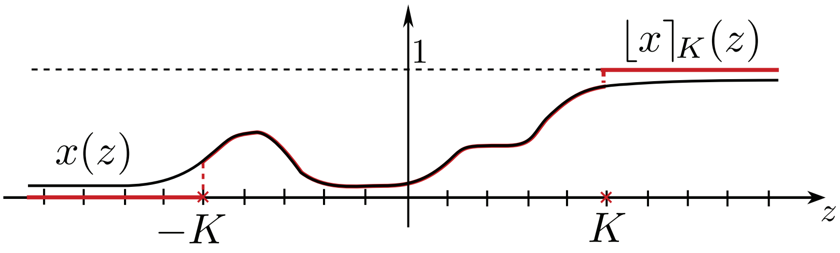

Figure 5.2: A profile x and its saturated version $\lfloor x \rceil_K$.

Proof. Using integration by parts, we can write

$$V(z) = \int_{-\infty}^{z} \mathrm{d}\tilde{z}\, \Omega(\tilde{z}) = \tilde{z}\,\Omega(\tilde{z})\Big|_{-\infty}^{z} - \int_{-\infty}^{z} \mathrm{d}\tilde{z}\, \tilde{z}\, w(\tilde{z}). \tag{5.16}$$

For $z \le 0$, we have

$$\int_{-\infty}^{z} \mathrm{d}\tilde{z}\, |\tilde{z}| w(\tilde{z}) \ge \int_{-\infty}^{z} \mathrm{d}\tilde{z}\, |z| w(\tilde{z}) = |z|\Omega(z) \ge 0,$$

so taking $\tilde{z} \to -\infty$ shows $\lim_{\tilde{z}\to-\infty} \tilde{z}\,\Omega(\tilde{z}) = 0$. Due to (5.16), we conclude

$$V(z) = z\,\Omega(z) - \int_{-\infty}^{z} \mathrm{d}\tilde{z}\, \tilde{z}\, w(\tilde{z}). \tag{5.17}$$

Thus, V is finite and well-defined. Convexity follows because $V''(z) = w(z) \ge 0$. ☐

Much of the analysis in this chapter proceeds relatively simply under the simple assumption that

$$\int_{\mathbb{R}} \mathrm{d}z\, (1 - x^w(z))y(z) < \infty,$$

where the integral on the left-hand side can be seen as the *interaction potential.* Most of our results will first be established under this assumption. In general, however, this assumption is not needed and it is sufficient that $C_w < \infty$. We typically generalize our results to this case by taking limits. Let us discuss this issue.

Definition 5.9. *We say that a function x is* saturated *off of the finite interval $[-K, K]$ if $x(z) = 0$ for $z \in (-\infty, -K)$ and $x(z) = 1$ for $z \in (K, \infty)$.*

Given a profile x, let us define $\lfloor x \rceil_K$ by

$$\lfloor x \rceil_K(z) = \mathbb{1}_{\{|z|<K\}} x(z) + \mathbb{1}_{\{z \ge K\}}\,.$$

By definition, $\lfloor x \rceil_K$ is saturated off of $[-K, K]$ (see Figure 5.2).

Lemma 5.10. *Let x, y be interpolating profiles and assume the PGC and that $C_w < \infty$, then*

$$\lim_{K\to\infty} \mathcal{W}_2(\lfloor x \rceil_K, \lfloor y \rceil_K) = \mathcal{W}_2(x, y).$$

Proof. See Section 5.9.2. ☐

We end this section with another useful definition.

Definition 5.11. *Assuming it exists, we define the "single potential"*

$$\begin{aligned}\mathcal{W}_{2,s}(x,y) &= \mathcal{W}_2(x,y) - \int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) \\ &= \int_{\mathbb{R}} \mathrm{d}z \left(\int_0^{x(z)} \mathrm{d}v\, f^{-1}(v) - \int_0^{y(z)} \mathrm{d}u\,(1 - g^{-1}(u)) \right). \end{aligned} \tag{5.18}$$

As we will see, the functional $\mathcal{W}_{2,s}(x,y)$ captures the "single" (uncoupled) part of $\mathcal{W}_2$: It is invariant under increasing rearrangements and linear under displacement interpolation.

5.3 Rearrangements

Displacement convexity is usually defined on a space of probability measures. For measures on the real line, it is most convenient to view displacement convexity on a space of cumulative distribution functions (cdf's). It is therefore fortunate that the search for the global minimum of the potential functional (5.8) can be reduced to the space of profiles x and y that are *non-decreasing*. In this section, we use the tool of *increasing rearrangements* to show that such rearrangements of x and y can only decrease the potential.

The notions of rearrangements that we use were also mentioned in Chapter 4; we repeat them here for completeness. Symmetric decreasing rearrangements are a classical tool in analysis (see [119]). Here, we use their closely related cousin, namely, increasing rearrangements (see [94]). Our presentation is self-contained and no previous exposure to rearrangements is needed. Consider a profile $p : \mathbb{R} \to [0,1]$ that satisfies (5.7). The *increasing rearrangement*[3] of p is the increasing function $\bar{p}$ that has the same limits and that somewhat preserves the mass of each level set (here, the mass of a level set could be infinite). More formally, let us represent p in layer cake form as

$$p(z) = \int_0^{p(z)} \mathrm{d}t = \int_0^1 \mathrm{d}t\, \mathbb{1}_{E_t}(z), \tag{5.19}$$

where $\mathbb{1}_{E_t}$ is the indicator function of the level set $E_t = \{z \,|\, p(z) > t\}$. For each value $t \in [0,1)$, the level set E_t can be written as the disjoint union of a bounded set A_t and a half line $(a_t, +\infty)$. We define the rearranged set $\bar{E}_t = (a_t - |A_t|, +\infty)$, and then

$$\bar{p}(z) = \int_0^1 \mathrm{d}t\, \mathbb{1}_{\bar{E}_t}(z). \tag{5.20}$$

A simple example capturing the notion of increasing rearrangement is shown in Figure 5.3.

Lemma 5.12. *Let p and q be two profiles satisfying* (5.7), *and let $\bar{p}$ and $\bar{q}$ denote their respective increasing rearrangements. Then, assuming the left integral exists, we have*

$$\int_{\mathbb{R}} \mathrm{d}z\,(p(z) - q(z)) = \int_{\mathbb{R}} \mathrm{d}z\,(\bar{p}(z) - \bar{q}(z)).$$

---

[3] Note that an increasing rearrangement is not necessarily strictly increasing.

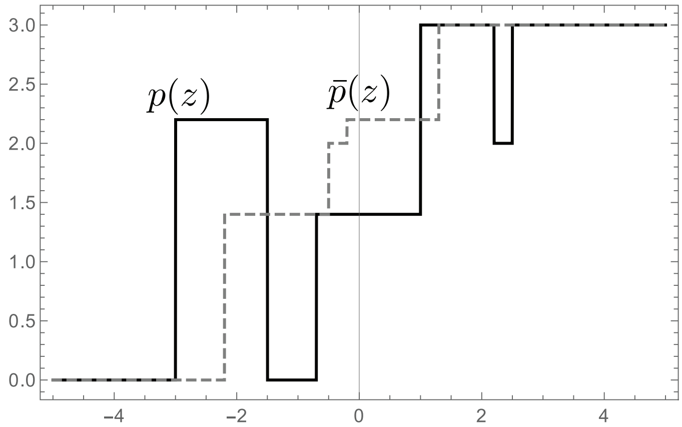

Figure 5.3: Simple example of an increasing rearrangement for step functions.

Proof. For each $t \in (0,1)$, there exists a minimal a_t such that $(a_t,\infty) \subset \{z : p(z) > t\} \cap \{z : q(z) > t\}$. Define $B_{p,t} = \{z : p(z) > t\} \setminus (a_t,\infty)$ and $B_{q,t} = \{z : q(z) > t\} \setminus (a_t,\infty)$. We also define the same quantities for the rearranged profiles $\bar{p}$ and $\bar{q}$, namely $\bar{a}_t$, $B_{\bar{p},t}$ and $B_{\bar{q},t}$. We show below that

$$|B_{p,t}| - |B_{q,t}| = |B_{\bar{p},t}| - |B_{\bar{q},t}|, \tag{5.21}$$

Equation (5.21) gives the result because, by using the layer cake representation, it follows that

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z\,(p(z) - q(z)) &= \int_0^1 \mathrm{d}t\,(|B_{p,t}| - |B_{q,t}|)\\ &= \int_0^1 \mathrm{d}t\,(|B_{\bar{p},t}| - |B_{\bar{q},t}|)\\ &= \int_{\mathbb{R}} \mathrm{d}z\,(\bar{p}(z) - \bar{q}(z)).\end{aligned}$$

Let us give an explicit argument for (5.21). We note that the infinite part of a level set can only increase under an increasing rearrangement, thus $(a_t,\infty) \subset (\bar{a}_t,\infty)$. So, (a_t,∞) is common to $\{z : \bar{p}(z) > t\}$ and $\{z : \bar{q}(z) > t\}$, and subtracting it leaves two finite sets with the same finite measure because rearrangements are measure-preserving; i.e.,

$$|\{z : \bar{p}(z) > t\} \setminus (a_t,\infty)| = |\{z : p(z) > t\} \setminus (a_t,\infty)|,$$

(with the *same* a_t on both sides). Thus,

$$\begin{aligned}|B_{\bar{p},t}| &= |\{z : \bar{p}(z) > t\} \setminus (\bar{a}_t,\infty)|\\ &= |\{z : \bar{p}(z) > t\} \setminus (a_t,\infty)| - |(\bar{a}_t, a_t)|\\ &= |B_{p,t}| - |(\bar{a}_t, a_t)|.\end{aligned}$$

Similarly, $|B_{\bar{q},t}| = |B_{q,t}| - |(\bar{a}_t, a_t)|$, and (5.21) follows from these two identities. □

Lemma 5.13. *For any interpolating x and y, we have*

$$\int_{\mathbb{R}} \mathrm{d}z\,(1 - x(z))y(z) \ge \int_{\mathbb{R}} \mathrm{d}z\,(1 - \bar{x}(z))\bar{y}(z).$$

Proof. If the left hand side is infinite then the result is immediate, so we assume that it is finite.

This result is very similar to the Hardy-Littlewood inequality for symmetric rearrangements. We will, however, give a self-contained elementary proof. The key inequality is the following, that holds for all $t, s \in (0,1)$,

$$|\{z : 1 - x(z) > t\} \cap \{z : y(z) > s\}| \geq |\{z : 1 - \bar{x}(z) > t\} \cap \{z : \bar{y}(z) > s\}|. \tag{5.22}$$

This gives the result because

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z\,(1 - x(z))y(z) &= \int_{\mathbb{R}} \mathrm{d}z \int_0^1 \mathrm{d}t\, \mathbb{1}_{\{1-x(z)>t\}}(z) \int_0^1 \mathrm{d}s\, \mathbb{1}_{\{y(z)>s\}}(z)\\ &= \int_0^1 \int_0^1 \mathrm{d}t\,\mathrm{d}s\, |\{z : 1 - x(z) > t\} \cap \{z : y(z) > s\}|.\end{aligned}$$

To see (5.22) for $s, t \in (0,1)$, note that $\{z : 1 - x(z) > t\} = (-\infty, a_t) \cup A_t$ and $\{z : y(z) > s\} = (b_s, +\infty) \cup B_s$ where the unions are disjoint and $|A_t|, |B_s| < \infty$. If $a_t + |A_t| < b_s - |B_s|$ (see case a) in Figure 5.4), then the right hand side of (5.22) is 0 and (5.22) is immediate so we assume otherwise. If $a_t \geq b_s$ (see case b) in Figure 5.4), then we trivially have equality in (5.22) so we also assume $a_t < b_s$. This is case c) in Figure 5.4. We now have

$$\begin{aligned}|\{z : 1 - \bar{x}(z) > t\} \cap \{z : \bar{y}(z) > s\}| =& |\,(-\infty, a_t + |A_t|)\ \cap\ (b_s - |B_s|, +\infty)\,|\\ =& |A_t| + |B_s| - (b_s - a_t).\end{aligned}$$

Remark that the last line is non-negative because we are *not* in the case $a_t + |A_t| < b_s - |B_s|$. Now note that A_t and B_s can intersect only in the interval $[a_t, b_s]$, so we have

$$\begin{aligned}|\{z : 1 - x(z) > t\} \cap \{z : y(z) > s\}| =& |A_t| + |B_s| - |(A_t \cup B_s) \cap [a_t, b_s]|\\ \geq& |A_t| + |B_s| - (b_s - a_t),\end{aligned}$$

and the lemma follows. □

Lemma 5.14. *For any interpolating x and y, we have*

$$\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) \geq \int_{\mathbb{R}} \mathrm{d}z\,(1 - \bar{x}^w(z))\bar{y}(z). \tag{5.23}$$

Proof. If the left hand side is infinite, the inequality holds. Hence, we suppose it is finite. We have

$$\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) = \int_{\mathbb{R}} \mathrm{d}z \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, w(\tilde{z})(1 - x(z - \tilde{z}))y(z).$$

As the integrand is non-negative and the integral is finite, we can apply the Fubini theorem to rewrite

$$\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) = \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, w(\tilde{z}) \int_{\mathbb{R}} \mathrm{d}z\,(1 - x(z))y(z + \tilde{z}). \tag{5.24}$$

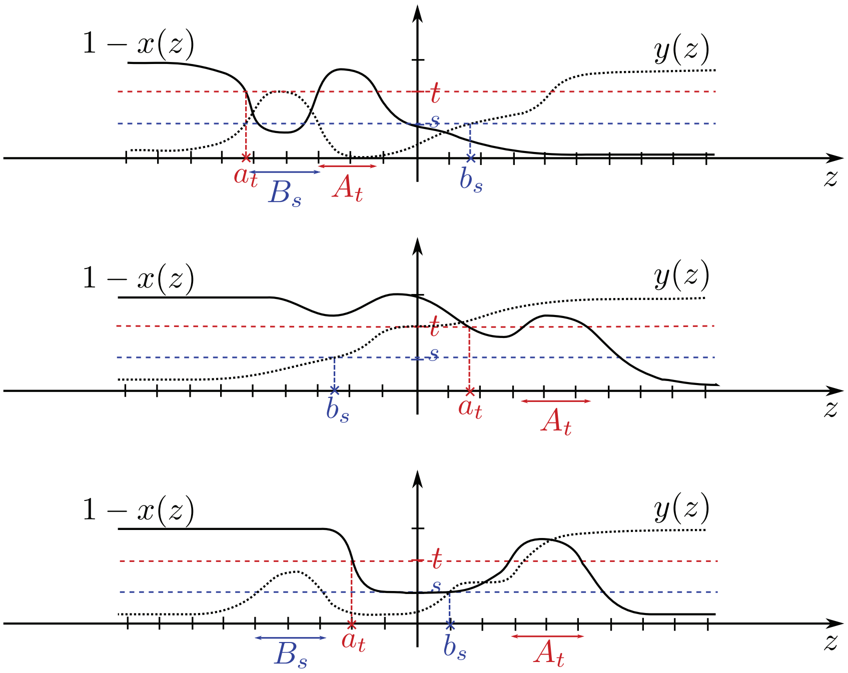

Figure 5.4: Illustration of level sets used in the proof of Lemma 5.13. a) For $a_t + |A_t| < b_s - |B_s|$. The intersection in the right hand side of (5.22) is empty and the inequality is trivial. b) For $a_t + |A_t| > b_s - |B_s|$ and $a_t \geq b_s$ (5.22) is an equality. c) For the last case $a_t + |A_t| > b_s - |B_s|$ and $a_t < b_t$ the inequality (5.22) is non trivial.

Now, we apply Lemma 5.13 to the functions x and $y_{\tilde{z}}$, where $y_{\tilde{z}}(z) = y(z+\tilde{z})$. Note that $y_{\tilde{z}}$ is simply a translated version of y so its rearrangement is just obtained by the same translation of $\bar{y}$, i.e., $\bar{y}_{\tilde{z}}(z) = \bar{y}(z+\tilde{z})$. Thus

$$\int_{\mathbb{R}} \mathrm{d}z\,(1-x(z))y(z+\tilde{z}) \geq \int_{\mathbb{R}} \mathrm{d}z\,(1-\bar{x}(z))\bar{y}(z+\tilde{z}).$$

Multiplying by $w(\tilde{z})$, integrating over $\tilde{z}$ and using (5.24) we obtain (5.23).

☐

We are now ready to prove a rearrangement inequality for $\mathcal{W}_2$.

Proposition 5.15 (Monotonicity of Minimizers)**.** *Let x and y be profiles that satisfy* (5.7)*, and let $\bar{x}$ and $\bar{y}$ be their respective increasing rearrangements. Assume the PGC and that $C_w < \infty$, then we have*

$$\mathcal{W}_2(x,y) \geq \mathcal{W}_2(\bar{x},\bar{y}). \tag{5.25}$$

Proof. If the left hand side of (5.25) is infinite, then the result is immediate, so we assume that $\mathcal{W}_2(x,y)$ is finite. Let us first assume that $\int_{\mathbb{R}} \mathrm{d}z\,(1-x^w(z))y(z) < \infty$ (in fact, we can assume the saturated case). It then follows that $\mathcal{W}_{2,s}(x,y)$ in Equ. (5.18) is finite. Note that, if $F : \mathbb{R} \to [0,1]$ is monotone, then the increasing rearrangement of $F \circ p$ is equal to $F \circ \bar{p}$. Thus, the increasing rearrangement of $\int_0^{x(z)} \mathrm{d}v\, f^{-1}(v)$ is equal to $\int_0^{\bar{x}(z)} \mathrm{d}v\, f^{-1}(v)$, and similarly for the term $\int_0^{y(z)} \mathrm{d}u\,(1-g^{-1}(u))$. We can now apply Lemma 5.12 (with suitable scaling) to conclude that $\mathcal{W}_{2,s}(x,y) = \mathcal{W}_{2,s}(\bar{x},\bar{y})$. For this case, the proposition now follows from Lemma 5.14.

Now, we consider the general case where, possibly, $\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) = \infty$. Due to Lemma 5.10, we have

$$\mathcal{W}_2(x,y) = \lim_{K\to\infty} \mathcal{W}_2(\lfloor x\rceil_K, \lfloor y\rceil_K). \tag{5.26}$$

We remark that $\int_{\mathbb{R}} \mathrm{d}z\,(1 - (\lfloor x\rceil_K^w(z))\lfloor y\rceil_K(z) < \infty$ due to Lemma 5.30, Equ. (5.48). Therefore, using the saturated case we have already established above, we have

$$\mathcal{W}_2(\lfloor x\rceil_K, \lfloor y\rceil_K) \geq \mathcal{W}_2(\overline{\lfloor x\rceil_K}, \overline{\lfloor y\rceil_K}). \tag{5.27}$$

Finally, it is easy to see that, for any interpolating profile x, we have $\overline{\lfloor x\rceil_K} \to \overline{x}$ pointwise. By Lemma 5.7, Equ. (5.12), we obtain

$$\liminf_{K\to\infty} \mathcal{W}_2(\overline{\lfloor x\rceil_K}, \overline{\lfloor y\rceil_K}) \geq \mathcal{W}_2(\bar{x},\bar{y}). \tag{5.28}$$

Combining (5.26), (5.27), (5.28) concludes the proof. □

Proposition 5.15 shows that minimizers x, y of the functional $\mathcal{W}_2(x,y)$ can be found in the spaces of *non-decreasing profiles*. From now on, we therefore restrict the functional to those spaces.

5.4 Existence of Minimizers

The existence of a monotonic fixed point, that we will show is a minimizer of $\mathcal{W}_2$, is proved in [69, 70]. In this section, we give an alternative proof, under similar conditions, using the direct method of the calculus of variations [109], as was done in Chapter 4.

In the direct method of the calculus of variations, one constructs a minimizer as a limit point of a minimizing sequence. As $\mathcal{W}_2(x,y)$ is invariant under a common translation of x and y, it is necessary to center the sequence in order to carry out the method. We can do this by translating x and y so that $\frac{1}{2} \in [x(0-), x(0+)]$. We call such a profile pair *centered*.

Proposition 5.16. *Assume $C_w < \infty$ and assume the SPGC is satisfied. Then, there exists a monotonic non-decreasing profile pair $(x(z), y(z))$ that minimizes $\mathcal{W}_2$ under the condition that (x,y) has limit $(1,1)$ at $z = \infty$ and limit $(0,0)$ at $z = -\infty$.*

Proof. We already remarked that we can adopt the alternative expression (5.13) for the potential functional, namely

$$\mathcal{W}_2(x,y) = \int_{\mathbb{R}} \mathrm{d}z\, \tilde{I}_{x,y,w}(z),$$

where $\tilde{I}_{x,y,w}(z) \geq 0$. Therefore, $\mathcal{W}_2(x,y)$ is bounded from below, thus, by Proposition 5.15, there exists a minimizing sequence (x_i,y_i) of monotonic profiles satisfying the limit condition; i.e.,

$$\lim_{i\to+\infty} \mathcal{W}_2(x_i,y_i) = \inf \mathcal{W}_2(x,y).$$

Let us center the sequence so that $\frac{1}{2} \in [x_i(0-), x_i(0+)]$ for each i. Interpreting x_i and y_i as cumulative probability distributions, our aim is to show the tightness of the sequence; i.e., that the transition of x_i and y_i from ϵ to $1-\epsilon$ must occur in a bounded region for all i and for any $\epsilon \in (0,1)$.

Let C be an arbitrary finite constant. Then, we claim that, for any $\epsilon > 0$, there exists $Z < \infty$ such that $\mathcal{W}_2(x,y) < C$ implies that $x(z), y(z) > 1-\epsilon$ for $z > Z$ and $x(z), y(z) < \epsilon$ for $z < -Z$, (assuming x, y is a centered monotonic profile pair satisfying the limit conditions).

This claim completes the proof. Indeed, we can then extract from (x_i, y_i) a subsequence (x_{i_k}, y_{i_k}) that converges to a limit point (x_*, y_*) that necessarily satisfies the limit conditions, and by Fatou's lemma

$$\int_{\mathbb{R}} \mathrm{d}z\, \tilde{I}_{x_*,y_*,w}(z) \le \liminf_{k\to+\infty} \int_{\mathbb{R}} \mathrm{d}z\, \tilde{I}_{x_{i_k},y_{i_k},w}(z),$$

so $\mathcal{W}_2(x_*,y_*) \le \inf \mathcal{W}_2(x,y)$, and x_*, y_* is a monotone minimizing pair for the potential functional.

Now, we prove the claim. Due to Lemma 5.31, we have

$$\int_{\mathbb{R}} \mathrm{d}z\, |x^w(z) - x(z)| \le C_w.$$

We see that $\mathcal{W}_2(x,y) < C$ implies that

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z\, \phi(f,g;y(z),x(z)) &= \mathcal{W}_2(x,y) - \int_{\mathbb{R}} \mathrm{d}z\, (x(z) - x^w(z))y(z) \\ &\le C + C_w.\end{aligned}$$

By the strictly positive gap condition, there exists $\eta > 0$ such that $\phi(f,g;y(z),x(z)) > \eta$ unless we have either $x(z), y(z) < \epsilon$ or $x(z), y(z) > 1-\epsilon$. Let z_+ be the smallest value of z for which $x(z), y(z) > 1-\epsilon$. Then,

$$C + C_w \ge \int_0^{z_+} \mathrm{d}z\, \phi(f,g;y(z),x(z)) > \eta z_+\,,$$

and $z_+ < (C+C_w)/\eta$. Thus, for each i we have $x_i(z), y_i(z) > 1-\epsilon$ for $z > (C+C_w)/\eta$. Similarly, for each i, we have $x_i(z), y_i(z) < \epsilon$ for $z < -(C+C_w)/\eta$. ☐

5.5 Displacement Convexity

A generic functional $\mathcal{F}(p)$ on a space $\mathcal{X}$ (of profiles say) is said to be convex in the usual sense if, for any pair $p_0, p_1 \in \mathcal{X}$, and for all $\lambda \in [0,1]$, and for the *linear interpolation* $(1-\lambda)p_0 + \lambda p_1$ of the profiles, the inequality $\mathcal{F}((1-\lambda)p_0 + \lambda p_1) \le (1-\lambda)\mathcal{F}(p_0) + \lambda\mathcal{F}(p_1)$ holds. Displacement convexity, on the other hand, is defined as convexity under an alternative interpolation called the *displacement interpolation*. The usual setting for displacement convexity is a space of probability measures. For measures over the real line, one can conveniently define the displacement interpolation in terms of the cdf's associated to the measures. This is the simplest setting and the one that we adopt here. We explain this in detail in Section 1.7 and summarize the main definitions that are relevant to the present chapter here.

We think of the increasing profiles p as right-continuous cdf's of some underlying measures $\mathrm{d}p$ over the real line. As already stated, the inverse $p^{-1}(u)$ is defined almost everywhere and the left and right limits, $p^{-1}(u_-)$ and $p^{-1}(u_+)$, respectively, are uniquely defined. However, at this point, it is useful to settle on the right continuous inverse that is defined for all $u \in (0,1)$, namely $p^{-1}(u) = \inf\{z \mid p(z) > u\}$.

Consider two profiles p_0 and p_1. If p_0 is continuous, the pushforward map $T_p : \mathbb{R} \to \mathbb{R}$ from $\mathrm{d}p_0$ to $\mathrm{d}p_1$ is given by

$$T_p(z) = p_1^{-1}(p_0(z)), \tag{5.29}$$

and the interpolant $p_\lambda(\cdot)$ is the cdf of the measure defined by

$$\mathrm{d}p_\lambda = ((1-\lambda)id + \lambda T_p)\#\mathrm{d}p_0,$$

where id denotes the identity map. Therefore, we have

$$\int \mathrm{d}p_1(z)\, h(z) = \int \mathrm{d}p_0(z)\, h(T_p(z)),$$
$$\int \mathrm{d}p_\lambda(z)\, h(z) = \int \mathrm{d}p_0(z)\, h((1-\lambda)z + \lambda T_p(z)),$$

whenever the integrals are well-defined.

The graphical construction of the interpolant p_λ is illustrated in Figure 1.14. Graphically, T_p finds the position $\tilde{z} = T_p(z)$ on the z-axis so that $p_1(\tilde{z}) = p_0(z)$ for some given z. Consider the linear interpolation between points on $\mathbb{R}$,

$$z_{p,\lambda} = (1-\lambda)z + \lambda T_p(z).$$

The displacement interpolant p_λ is defined so that the following equality holds for all $\lambda \in [0,1]$

$$p_\lambda(z_{p,\lambda}) = p_0(z).$$

In the case where p_0 is discontinuous, we have to be more careful in the definition. At points of discontinuity of p_0, the map $T_p(z)$ should not be singled valued. As we work in one dimension, this issue is easily circumvented and we can, in general, define p_λ via its inverse as

$$p_\lambda^{-1}(u) = (1-\lambda)p_0^{-1}(u) + \lambda p_1^{-1}(u), \tag{5.30}$$

and $p_\lambda(z) = \inf\{u \mid p_\lambda^{-1}(u) > z\}$ (which is right continuous). Correspondingly, if p is an interpolating non-decreasing profile then, under appropriate regularity of h, we can write

$$\int_{\mathbb{R}} \mathrm{d}p(z)\, h(z) = \int_0^1 \mathrm{d}u\, h(p^{-1}(u)),$$

and we have

$$\int \mathrm{d}p_\lambda(z)\, h(z) = \int_0^1 \mathrm{d}u\, h((1-\lambda)p_0^{-1}(u) + \lambda p_1^{-1}(u)).$$

With this in mind, we will continue to use the notation $T_p(z)$ when the above interpretation should be understood.

In the remainder of the chapter, we consider two pairs of interpolating profiles (x_0, x_1) and (y_0, y_1) and consider the corresponding interpolants x_λ and y_λ.

We now state one of the main results of this chapter.

Proposition 5.17. *Assume the PGC and $C_w < \infty$. Then, the potential $\mathcal{W}_2(x,y)$ is displacement convex; that is, for all $\lambda \in [0,1]$*

$$\mathcal{W}_2(x_\lambda, y_\lambda) \leq (1-\lambda)\mathcal{W}_2(x_0,y_0) + \lambda \mathcal{W}_2(x_1,y_1). \tag{5.31}$$

We first show that it is sufficient to prove the proposition under the assumption that (x_0,y_0) and (x_1,y_1) are saturated. We recall that, by Lemma 5.10, for any monotonic interpolating pair x,y we have

$$\lim_{K\to\infty} \mathcal{W}_2(\lfloor x \rceil_K, \lfloor y \rceil_K) = \mathcal{W}_2(x,y). \tag{5.32}$$

Given any monotonic interpolating pairs $(x_0,y_0),(x_1,y_1)$, let $x_{K,\lambda}$ denote the displacement interpolant of $\lfloor x_0 \rceil_K$ and $\lfloor x_1 \rceil_K$. It is easy to see that $x_{K,\lambda}$ converges pointwise to x_λ when $K \to +\infty$. By Lemma 5.7, Equ. (5.12), we therefore have

$$\liminf_{K\to\infty} \mathcal{W}_2(x_{K,\lambda}, y_{K,\lambda}) \geq \mathcal{W}_2(x_\lambda, y_\lambda). \tag{5.33}$$

In view of (5.32) and (5.33), we see that (5.31) follows from

$$\mathcal{W}_2(x_{K,\lambda}, y_{K,\lambda}) \leq (1-\lambda)\mathcal{W}_2(\lfloor x_0 \rceil_K, \lfloor y_0 \rceil_K) + \lambda \mathcal{W}_2(\lfloor x_1 \rceil_K, \lfloor y_1 \rceil_K), \tag{5.34}$$

which is the saturated case of (5.31). For the remainder of the section, we therefore assume the saturated case, and prove (5.34).

If x and y are saturated then we have

$$\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) < \infty\,.$$

Indeed,

$$(1 - x^w(z))y(z) = (1 - x(z))y(z) + (x(z) - x^w(z))y(z),$$

and the first term is integrable for saturated profiles x,y. The second term is also integrable for such profiles because of Lemma 5.31 (note that x^w is not necessarily saturated). This is the critical requirement because, by integrating by parts, we obtain

$$\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) = \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\,V(z-\tilde{z}). \tag{5.35}$$

The full derivation of this identity reads

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w(z))y(z) &= \iint\limits_{\mathbb{R}^2} \mathrm{d}z\,\mathrm{d}\tilde{z}\,(1 - x(z))w(z-\tilde{z})y(\tilde{z}) \\ &= \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}\tilde{z}\,y(\tilde{z})\Omega(z-\tilde{z}) \\ &= \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\,V(z-\tilde{z}),\end{aligned}$$

where we have used the fact the $V(z)$ is well-defined.

The identity (5.35) leads to the following key result:

Lemma 5.18. *Let* (x_0, y_0) *and* (x_1, y_1) *be saturated, then* $\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w_\lambda(z))y_\lambda(z)$ *is a convex function of* λ.

Proof. As x_λ and y_λ are saturated, we have, by (5.35), that

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w_\lambda(z))y_\lambda(z) &= \iint\limits_{\mathbb{R}^2} \mathrm{d}x_\lambda(z)\mathrm{d}y_\lambda(\tilde z)\,V(z - \tilde z)\\ &= \iint\limits_{\mathbb{R}^2} \mathrm{d}x_0(z)\mathrm{d}y_0(\tilde z)\,V((1-\lambda)(z - \tilde z) + \lambda(T_x(z) - T_y(\tilde z))).\end{aligned}$$

This is convex in λ because the kernel V is convex (see Lemma 5.8). □

Lemma 5.19. *For any saturated pairs* $(x_0, y_0), (x_1, y_1)$, *the functional* $\mathcal{W}_{2,s}(x_\lambda, y_\lambda)$ *is affine in* λ.

Proof. We will show that $\mathcal{W}_{2,s}(x_\lambda, y_\lambda) - \mathcal{W}_{2,s}(x_0, y_0)$ is linear in λ. We start by the considering the first term of this difference. Using the layer cake representation and the monotonicity of the functions we have

$$\begin{aligned}\int_{\mathbb{R}} \mathrm{d}z \Big(\int_0^{x_\lambda(z)} \mathrm{d}u\, f^{-1}(u) - \int_0^{x_0(z)} \mathrm{d}u\, f^{-1}(u) \Big) &= \int_{\mathbb{R}} \mathrm{d}z \int_{x_0(z)}^{x_\lambda(z)} \mathrm{d}u\, f^{-1}(u)\\ &= \int_0^1 \mathrm{d}u\, f^{-1}(u)(x_0^{-1}(u) - x_\lambda^{-1}(u)).\end{aligned}$$

Because of (5.30) this last term equals

$$\lambda \int_0^1 \mathrm{d}u\, f^{-1}(u)(x_0^{-1}(u) - x_1^{-1}(u)),$$

that is linear in λ. Similarly for the second term in the difference $\mathcal{W}_{2,s}(x_\lambda, y_\lambda) - \mathcal{W}_{2,s}(x_0, y_0)$, we obtain

$$\begin{aligned}&\int_{\mathbb{R}} \mathrm{d}z \Big(\int_0^{y_0(z)} \mathrm{d}u\,(1 - g^{-1}(u)) - \int_0^{y_\lambda(z)} \mathrm{d}u\,(1 - g^{-1}(u)) \Big)\\ &\quad = \lambda \int_0^1 \mathrm{d}u\,(1 - g^{-1}(u))(y_1^{-1}(u) - y_0^{-1}(u)).\end{aligned}$$

□

We are now ready to prove the main result of this section.

Proof of Proposition 5.17. If $\mathcal{W}_2(x_0, y_0) = +\infty$ or $\mathcal{W}_2(x_1, y_1) = +\infty$ then the result is immediate, so we assume both are finite. As argued above, we can assume that all functions are saturated. We rewrite the potential in (5.8) as follows.

$$\mathcal{W}_2(x_\lambda, y_\lambda) = \mathcal{W}_{2,s}(x_\lambda, y_\lambda) + \int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w_\lambda(z))y_\lambda(z). \tag{5.36}$$

By Lemma 5.19 the functional $\mathcal{W}_{2,s}(x_\lambda, y_\lambda)$ is affine and hence convex in λ. The second term was shown to be convex in Lemma 5.18. □

5.6 Fixed Points and Minimizers

The main goal of this section is to prove Proposition 5.21 that states that a pair of monotonic profiles minimizes $\mathcal{W}_2$ if and only if it is a "consistent" fixed point. It will be helpful to start with a preliminary discussion that explains the motivation behind defining consistent fixed points.

We already remarked that $\phi(f,g;u,v)$ is convex in v for fixed u and minimized (over v) by setting $v = f(u)$, and similarly for u and v interchanged. From the expression (5.9), a similar argument shows that $I_{x,y,w}(z) \geq I_{x,g\circ x^w,w}(z)$ and $I_{x,y,w}(z) \geq I_{f\circ y^w,y,w}(z)$, so

$$\mathcal{W}_2(x,y) \geq \mathcal{W}_2(x, g\circ x^w) \text{ and } \mathcal{W}_2(x,y) \geq \mathcal{W}_2(f\circ y^w, y).$$

Under some conditions, we can have $\mathcal{W}_2(x,y) = \mathcal{W}_2(x, g\circ x^w)$ even though it is not the case that $y = g\circ x^w$ almost everywhere. This can happen, in particular, if g is discontinuous and the pair f,g does not satisfy the strictly positive gap condition.

One of the main analytical tools used in [69, 70] was the construction of f and g given x, y, and w so that x, y form a "consistent" interpolating fixed point. Note that, from an interpolating fixed point, we can recover the graph of f as the parametric curve $(x(z), y^w(z))$ as $z \in (-\infty, +\infty)$. Given interpolating x and y we denote the f so obtained as $f_{[x,y^w]}$ (see [69, 70] for more details). The function $f_{[x,y^w]}$ is uniquely determined at points of continuity but may not be uniquely determined at points of discontinuity. In particular, if y^w is constant over some open interval I where x is increasing, then $f_{[x,y^w]}$ has a discontinuity at that value of $y^w(I)$, and we see that we cannot have $x = f_{[x,y^w]}\circ y^w$ almost everywhere. Nevertheless, it is the case that $x(z) \in [f_{[x,y^w]}(y^w(z)-), f_{[x,y^w]}(y^w(z)+)]$ for all z, and, in this sense, it satisfies the DE equation. In [69, 70], the notation

$$x \doteqdot f\circ y^w$$

was used to capture this case.[4] This motivates us to make the following definition:

Definition 5.20. *We say that an interpolating pair x, y of profiles is a consistent fixed point if $x \doteqdot f\circ y^w$ and $y \doteqdot g\circ x^w$. Recall that x, y is a fixed point if $x = f\circ y^w$ and $y = g\circ x^w$ almost everywhere, i.e., up to a set of measure zero.*

Proposition 5.21. *Let x, y be monotonic and interpolating. Then $\mathcal{W}_2(x,y)$ is minimal - in the sense $\mathcal{W}_2(x,y) \leq \mathcal{W}_2(\tilde{x},\tilde{y})$ for any monotonic interpolating $\tilde{x},\tilde{y}$ - if and only if x, y is a consistent fixed point.*

Proof. If x, y is not a consistent fixed point, then, either $\mathcal{W}_2(x,y) = \infty$ in which case the pair cannot be minimal, or we have either $\mathcal{W}_2(x,y) > \mathcal{W}_2(x, g\circ x^w)$ or $\mathcal{W}_2(x,y) > \mathcal{W}_2(f\circ y^w, y)$, which shows that $\mathcal{W}_2(x,y)$ is not minimal.

To prove the converse, assume x_0, y_0 is a consistent fixed point. The proof proceeds by contradiction. Hence, we suppose that there exists interpolating x_1, y_1 with $\mathcal{W}_2(x_0,y_0) > \mathcal{W}_2(x_1,y_1)$ and we shall deduce a contradiction. By Lemma 5.10, we can assume that x_1 and y_1 are saturated.

---

[4] More precisely if h plays the role of f, g we denote $v \doteqdot h(u)$ when $v = h(u)$ at points of continuity of h and $v \in [h(u-), h(u+)]$ at points of discontinuity of h.

We will show that we may also take x_0, y_0 to be saturated. Define

$$\begin{aligned} f_0^K &= f_{[\lfloor x_0 \rceil_K, \lfloor y_0 \rceil_K^w]}, \\ g_0^K &= g_{[\lfloor y_0 \rceil_K, \lfloor x_0 \rceil_K^w]}, \end{aligned}$$

so $\lfloor x_0 \rceil_K, \lfloor y_0 \rceil_K$ is a consistent fixed point for f_0^K, g_0^K. As x_1, y_1 are saturated, it follows easily that

$$\lim_{K \to \infty} \mathcal{W}_2(f_0^K, g_0^K; x_1, y_1) = \mathcal{W}_2(f_0, g_0; x_1, y_1) = \mathcal{W}_2(x_1, y_1),$$

and, by Lemma 5.32, we have

$$\lim_{K \to \infty} \mathcal{W}_2(f_0^K, g_0^K; \lfloor x_0 \rceil_K, \lfloor y_0 \rceil_K) = \mathcal{W}_2(x_0, y_0),$$

and we see that we can assume that x_0, y_0 are saturated.

As x_0, y_0 is a consistent fixed point, it follows that $\mathcal{W}_2(x_0, y_0) \le \mathcal{W}_2(\tilde{x}, y_0)$ and $\mathcal{W}_2(x_0, y_0) \le \mathcal{W}_2(x_0, \tilde{y})$ for all interpolating $\tilde{x}$ and $\tilde{y}$. Hence, we now have

$$\begin{aligned} \mathcal{W}_2(x_\lambda, y_\lambda) - \mathcal{W}_2(x_0, y_0) &\ge \mathcal{W}_2(x_\lambda, y_\lambda) - \mathcal{W}_2(x_\lambda, y_0) - \mathcal{W}_2(x_0, y_\lambda) + \mathcal{W}_2(x_0, y_0) \\ &= - \int_{\mathbb{R}} \mathrm{d}z \, (x_\lambda^w(z) - x_0^w(z))(y_\lambda(z) - y_0(z)) \\ &\ge -C\lambda^2, \end{aligned} \tag{5.37}$$

where C is some positive constant. The last step follows from $|x_\lambda^w(z) - x_0^w(z)| \le C_1 \lambda$ and $\int_{\mathbb{R}} \mathrm{d}z \, |y_\lambda(z) - y_0(z)| \le C_2 \lambda$ for some positive constants C_1 and C_2, which follows from the saturation of x_0, y_0 and x_1, y_1. By Proposition 5.17, we have

$$\mathcal{W}_2(x_\lambda, y_\lambda) - \mathcal{W}_2(x_0, y_0) \le \lambda(\mathcal{W}_2(x_1, y_1) - \mathcal{W}_2(x_0, y_0)). \tag{5.38}$$

Because of the assumption on x_1, y_1, the right hand side of (5.38) is strictly negative. Thus, (5.37) and (5.38) contradict themselves for λ sufficiently small. We conclude that no such x_1, y_1 can exist. □

We conclude this section with a pleasing expression in Lemma 5.23 for $\mathcal{W}_2(x, y)$ when x, y is a monotonic minimizer or, equivalently, a consistent fixed point.

We need the following functional from [69, 70],

$$\xi_\phi(w; x, y; z_1, z_2) = \iint \mathrm{d}x(z) \mathrm{d}y(\tilde{z}) \, (\mathbb{1}_{C_1} \Omega(z - \tilde{z}) + \mathbb{1}_{C_2} \Omega(\tilde{z} - z)), \tag{5.39}$$

with

$$C_1 = \{(z, \tilde{z}) : z \le z_2, \tilde{z} > z_1\},$$

and

$$C_2 = \{(z, \tilde{z}) : z > z_2, \tilde{z} \le z_1\}.$$

Note that ξ_ϕ is non-negative; this is closely related to the positive gap condition. One of the main results in [69] (Lemma 9) is the following (this result is used in Section 5.7).

Lemma 5.22. *Let* x, y *be a consistent fixed point for* (5.6)*, then*

$$\xi_\phi(w; x, y; z_1, z_2) = \phi(f, g; y(z_1+), x(z_2+)). \tag{5.40}$$

It turns out that, for our application, we only require the case $z_1 = z_2$ and in this case the right hand side of (5.39) simplifies to

$$\iint \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, \mathbb{1}_{\{(z-z_1)(\tilde{z}-z_1)\le 0\}}\Omega(-|z-\tilde{z}|). \tag{5.41}$$

Lemma 5.23. *If* x, y *is a consistent fixed point then*

$$\mathcal{W}_2(x, y) = \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, \kappa(z-\tilde{z}),$$

where

$$\kappa(z) := V(z) - z\Omega(z) = -\int_{-\infty}^{z} \mathrm{d}\tilde{z}\, \tilde{z}w(\tilde{z}).$$

Note that κ *is a non-negative even function that tends to* 0 *at* $\pm\infty$ *(recall that* w *is an odd function).*

Proof. By Lemma 5.32, it is enough to prove this for the saturated case. For the saturated case, we can integrate by parts to obtain

$$\int_{\mathbb{R}} \mathrm{d}z\, (1 - x(z))y(z) = \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, \mathbb{1}_{\{z-\tilde{z}\ge 0\}}(z-\tilde{z}).$$

From (5.22) and (5.41), we have

$$\int_{\mathbb{R}} \mathrm{d}z\, \phi(f, g; y(z), x(z)) = \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, |z-\tilde{z}|\Omega(-|z-\tilde{z}|). \tag{5.42}$$

Combining these two equations we obtain

$$\begin{aligned}
\mathcal{W}_{2,s}(x, y) &= \iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, (|z-\tilde{z}|\Omega(-|z-\tilde{z}|) - \mathbb{1}_{\{z-\tilde{z}\ge 0\}}(z-\tilde{z})) \\
&= -\iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, (z-\tilde{z})(\mathbb{1}_{\{z-\tilde{z}\ge 0\}} - \Omega(-|z-\tilde{z}|)) \\
&= -\iint\limits_{\mathbb{R}^2} \mathrm{d}x(z)\mathrm{d}y(\tilde{z})\, (z-\tilde{z})\Omega(z-\tilde{z}).
\end{aligned}$$

By (5.36), adding this to (5.35) gives the result. ☐

5.7 Strict Displacement Convexity

The existence of increasing interpolation solutions to (5.6) was established in [69, 70] under the assumption of the strictly positive gap condition and assuming that w is strictly positive on an interval $(-W, W)$, $W \le +\infty$ and 0 off of $[-W, W]$. We shall refer to this as the *interval support condition.* It was also shown in [69, 70] that the existence of such a fixed point implies the positive gap condition, and, by example, it was shown that if $A(f, g; u) = 0$ for some $u \in (0, 1)$, then there may be an infinite family of fixed point solutions that are not equivalent under translation. In this section, we use displacement convexity to show that the solution whose existence was proved in Section 5.4 and in [69, 70] under the strictly positive gap condition is unique up to displacement.

It follows from Proposition 5.21 that all interpolating minimizers have the same potential and that they are all consistent fixed points. By Proposition 5.17, we see that, if x_0, y_0 and x_1, y_1 are both monotonic interpolating consistent fixed points, then x_λ, y_λ is a consistent fixed point for all $\lambda \in [0,1]$. Displacement convexity can therefore not be strict in this case. The aim of the proof is to show that the strictly positive gap condition then leads to the conclusion that all consistent fixed points are equal up to translation.

Given x_0, y_0 and x_1, y_1 we define

$$D(u,v) = (x_1^{-1}(v) - y_1^{-1}(u)) - (x_0^{-1}(v) - y_0^{-1}(u)).$$

Lemma 5.24. *Let x_0, y_0 and x_1, y_1 be consistent fixed points and assume interval support condition. Then for all $\lambda \in [0,1]$ we have*

$$\mu\{(u,v) : |x_\lambda^{-1}(v) - y_\lambda^{-1}(u)| < W, D(u,v) \neq 0, \phi(u,v) \neq 0\} = 0,$$

where μ denotes the 2-d Lebesgue measure.

Proof. We assume throughout that x_0, y_0 and x_1, y_1 are consistent fixed points. Formally, we have

$$\frac{\mathrm{d}^2}{\mathrm{d}\lambda^2}\mathcal{W}_2(x_\lambda, y_\lambda) = \iint_{[0,1]^2} \mathrm{d}u\mathrm{d}v\, D(u,v)^2 w(x_\lambda^{-1}(v) - y_\lambda^{-1}(u)).$$

The formula is derived in Section 5.9.5 for saturated profiles. Note that the integrand is always non-negative so the integral is well-defined although it may take the value $+\infty$. We claim that

$$\int_0^1 \mathrm{d}\lambda \iint_{[0,1]^2} \mathrm{d}u\mathrm{d}v\, D(u,v)^2 w(x_\lambda^{-1}(v) - y_\lambda^{-1}(u)) = 0. \tag{5.43}$$

Assume the claim is false. Then, there exists a set $A \subset [0,1]^2$ on which $x_0^{-1}, y_0^{-1}, x_1^{-1}, y_1^{-1}$ are all bounded, such that

$$\int_0^1 \mathrm{d}\lambda \iint_A \mathrm{d}u\mathrm{d}v\, D(u,v)^2 w(x_\lambda^{-1}(v) - y_\lambda^{-1}(u)) = \eta > 0.$$

In the saturated case, it is easy to see that $\mathcal{W}_2(x_\lambda, y_\lambda)$ is absolutely continuous and so is $\frac{\mathrm{d}}{\mathrm{d}\lambda}\mathcal{W}_2(x_\lambda, y_\lambda)$. It now follows that for all K large enough we have

$$\int_{[0,1]} \mathrm{d}\lambda \frac{\mathrm{d}^2}{\mathrm{d}\lambda^2}\mathcal{W}_2(x_{K,\lambda}, y_{K,\lambda}) \geq \eta,$$

and therefore, using the convexity of $\mathcal{W}_2(x_{K,\lambda}, y_{K,\lambda})$ with respect to λ, we deduce that there is a positive constant γ such that for all K large enough we have

$$\mathcal{W}_2(\lfloor x_0 \rfloor_K, \lfloor y_0 \rfloor_K) + \mathcal{W}_2(\lfloor x_1 \rfloor_K, \lfloor y_1 \rfloor_K) - 2\mathcal{W}_2(x_{K,\frac{1}{2}}, y_{K,\frac{1}{2}}) > \gamma\,.$$

Applying Lemma 5.10 and Lemma 5.7(5.12), and noting that $x_{K,\lambda}, y_{K,\lambda}$ converge pointwise to x_λ, y_λ, yields

$$\mathcal{W}_2(x_0, y_0) + \mathcal{W}_2(x_1, y_1) - 2\mathcal{W}_2(x_{\frac{1}{2}}, y_{\frac{1}{2}}) \geq \gamma,$$

which contradicts Proposition 5.21, thereby establishing the claim.

Now, assume that for some $\lambda \in [0,1]$, we have

$$\mu\{(u,v) : |x_\lambda^{-1}(v) - y_\lambda^{-1}(u)| < W, D(u,v) \neq 0, \phi(u,v) > 0\} > 0.$$

By the continuity and inner regularity of the Lebesgue measure, there exists a constant $\eta > 0$ and a closed set $A \in (0,1)^2$ of positive measure such that, for all $(u,v) \in A$, we have

$$\begin{aligned} |x_{\tilde\lambda}^{-1}(v) - y_{\tilde\lambda}^{-1}(u)| &< W - \eta, \\ |D(u,v)| &> \eta, \\ \phi(u,v) &> \eta \end{aligned}$$

for all $\tilde\lambda \in [0,1]$ satisfying $|\tilde\lambda - \lambda| < \eta$.

For all $\delta \in [0, 2W]$, we define

$$\theta(\delta) = \min_{z \in [\delta, 2W]} (\Omega(z) - \Omega(z - \delta)).$$

Note that $\theta(\delta) > 0$ for $\delta > 0$ and that θ is non-decreasing.

For any $(u,v) \in A$ we have for $a < b$,

$$\begin{aligned} &\int_a^b \mathrm{d}\lambda\, D(u,v)^2 w(x_0^{-1}(v) - y_0^{-1}(u) + \lambda D(u,v)) \\ &\quad = D(u,v)\big[\Omega(x_0^{-1}(v) - y_0^{-1}(u) + bD(u,v)) - \Omega(x_0^{-1}(v) - y_0^{-1}(u) + aD(u,v))\big] \\ &\quad \geq |D(u,v)|\theta((b-a)|D(u,v)|) \\ &\quad \geq \eta\,\theta((b-a)\eta). \end{aligned}$$

By the Fubini theorem, this contradicts our above established claim (5.43). □

Let us define $D_W = \{(u,v) : |x^{-1}(v) - y^{-1}(u)| < W\}$. For $k \in \mathbb{R}$ let us define

$$\begin{aligned} T_1(k) &= \{(z, \tilde z) : z \geq k, \tilde z \leq k, z - \tilde z < W\}, \\ T_2(k) &= \{(z, \tilde z) : z \leq k, \tilde z \geq k, \tilde z - z < W\}. \end{aligned}$$

Let $B_\epsilon(u)$ denote the open interval centered at u of length 2ϵ.

Lemma 5.25. *Let x, y be a consistent fixed point and assume the SPGC and the interval support condition. For all $v \in (0,1)$, there exists $u \in (0,1)$ and $\epsilon > 0$ such that $B_\epsilon(u) \times B_\epsilon(v) \subset D_W$.*

Proof. Let $v \in (0,1)$ and define $z_m = \frac{1}{2}(x^{-1}(v+) + x^{-1}(v-))$. We must have $x^{-1}(v+) - x^{-1}(v-) < 2W$ because otherwise we obtain $\xi_\phi(w; x, y; z_m, z_m) = 0$, which, by Lemma 5.22, contradicts the SPGC.

More precisely, by (5.41), the SPGC implies that the $\mathrm{d}x\mathrm{d}y$ measure of at least one of $T_1(z_m)$ and $T_2(z_m)$ is strictly positive. We shall assume that the measure of

$T_1(z_m)$ is positive, and the other case can be handled similarly. It follows from the monotonicity of x and y that there exists $u \in (0,1)$ and ϵ sufficiently small such that $y^{-1}(B_\epsilon(u)) \times x^{-1}((v, v+\epsilon)) \subset T_1(z_m)$. It follows that $y^{-1}(B_\epsilon(u)) \subset (z_m - W, z_m]$ and because $x^{-1}((v-\epsilon, v)) \subset (z_m - W, z_m]$, we have $B_\epsilon(u) \times B_\epsilon(v) \subset D_W$. □

By Lemma 5.24, we have that, if x_0, y_0 and x_1, y_1 are consistent fixed points and the SPGC and interval support condition hold, then

$$\mu\{(u,v) : |x_\lambda^{-1}(v) - y_\lambda^{-1}(u)| < W, D(u,v) \neq 0\} = 0. \tag{5.44}$$

We claim that this implies that $x_1^{-1}(v) - x_0^{-1}(v)$ is essentially constant. Similarly, we have $y_1^{-1}(u) - y_0^{-1}(u)$ is essentially constant. Moreover, these two constants are equal.

Lemma 5.26. *Assume the SPGC and the interval support condition and that* x_1, y_1 *and* x_0, y_0 *are consistent fixed points. Then,* $x_1^{-1} - x_0^{-1}$ *is essentially constant on* $[0,1]$.

Proof. Let us assume that $x_1^{-1} - x_0^{-1}$ is not essentially constant; i.e., there exists a real value s so that $|\{v : D_x(v) > s\}| \in (0,1)$ and $|\{v : D_x(v) \leq s\}| \in (0,1)$. Then, there exists $v^* \in (0,1)$ that is in the support of both sets; i.e., for any $\epsilon > 0$, we have $|\{v : D_x(v) > s\} \cap B_\epsilon(v^*)| > 0$ and $|\{v : D_x(v) \leq s\} \cap B_\epsilon(v^*)| > 0$.

By Lemma 5.25, there exists $u \in (0,1)$ and $\epsilon > 0$ such that $B_\epsilon(u) \times B_\epsilon(v^*) \subset D_W$. Using the definition of v^*, we can see that there is a positive constant $\eta > 0$ such that $\int_{B_\epsilon(v^*)} \mathrm{d}v\, |D(u,v)| > \eta$ for all $u \in B_\epsilon(u)$, which now contradicts (5.44). This completes the proof. □

Proposition 5.27. *Assume the SPGC and the interval support condition and that* x_1, y_1 *and* x_0, y_0 *are interpolating monotonic consistent fixed points. Then, there exists* m *such that for almost all* z *we have* $x_1(z) = x_0(z+m)$ *and* $y_1(z) = y_0(z+m)$.

Proof. By Lemma 5.26, there exists m such that $x_1^{-1}(v) - x_0^{-1}(v) = m$ for almost all $v \in [0,1]$. Similarly, there exists $\tilde{m}$ such that $y_1^{-1}(u) - y_0^{-1}(u) = \tilde{m}$ for almost all $u \in [0,1]$. It follows that $D(u,v) = m - \tilde{m}$ for almost all $(u,v) \in [0,1]^2$. It now follows from Lemma 5.25 and (5.44) that $m = \tilde{m}$. □

Even though we have stated and proved the results for consistent fixed points, under the assumptions of this section, consistent fixed points are actually fixed points.

Lemma 5.28. *If* x, y *is a consistent fixed point and* f, g *satisfies the strictly positive gap condition and the interval support condition holds then* x, y *is a fixed point.*

Proof. If the SPGC and the interval support condition hold, then x^w and y^w are strictly increasing wherever they take values in $(0,1)$. This implies that x, y must be a fixed point (see [69, 70] for further detail.) □

5.8 Illustrations

In this work, we have shown (using Propositions 5.16, 5.17, 5.21, and 5.27) that under some conditions, the potential functional $\mathcal{W}$ is displacement convex and that its minimizer exists and is unique up to translation. These conditions are the strictly positive gap condition, $C_w < \infty$, and the interval support condition. In this section, we apply these results on different scalar systems when these conditions hold. In particular, for the applications we consider, we use the even uniform window with $W = 1/2$ which implies the two latter conditions. We illustrate for each application that the strictly positive gap condition holds.

To check the SPGC, one can directly look at $\phi(f, g; u, v)$, but there is a simpler way to check the condition. Indeed, we already remarked that, for fixed u, the potential is minimized by setting $v = f(u)$. Therefore,

$$\phi(f, g; u, v) \geq \phi(f, g; u, f(u)) = A(f, g; u).$$

So the SPGC is valid as long as the signed area $A(f, g; u) > 0$ for $u \in (0, 1)$. Similarly, for fixed v the potential is minimized by setting $u = g(v)$. Thus,

$$\phi(f, g; u, v) \geq \phi(f, g; g(v), v) = \tilde{A}(f, g; v),$$

where

$$\tilde{A}(f, g; v) = \int_0^v \mathrm{d}v'(f^{-1}(v') - g(v')),$$

is the alternative signed area bounded between the two EXIT curves and the horizontal axis at the origin and at height v. The SPGC is valid as long as $\tilde{A}(f, g; v) > 0$ for $v \in (0, 1)$.

Clearly, when $\phi(f, g; 1, 1) = 0$ as assumed in this paper, we also have $A(f, g; 1) = \tilde{A}(f, g; 1)$.

5.8.1 LDPC Code Ensembles on the BEC

We demonstrate our results on the $(3, 6)$-regular spatially coupled LDPC code ensemble when transmission takes place over the BEC(ϵ). For this ensemble, we have the (unscaled) uncoupled DE equations $\mathrm{x} = \epsilon \mathrm{y}^2$ and $\mathrm{y} = 1 - (1 - \mathrm{x})^5$. We already showed how to perform the right scaling $\mathrm{x} = \mathrm{x}_{\mathrm{MAP}} v$ and $\mathrm{y} = \mathrm{y}_{\mathrm{MAP}} u$; asking that $(u, v) = (1, 1)$ is a fixed point and $A(f, g; 1) = 0$, we find $\mathrm{y}_{\mathrm{MAP}} = 0.941$, $\mathrm{x}_{\mathrm{MAP}} = 0.432$ and $\epsilon = \epsilon_{\mathrm{MAP}} = 0.4881$. Replacing these numbers in the expression of the potential function, we find $\phi(u, v)$. Figures 5.5 and 5.6 illustrate the corresponding EXIT curves and the potential that is seen to satisfy the SPGC.

5.8.2 Generalized LDPC Codes

We consider a generalized LDPC (GLDPC) code where the check node constraints are given by a primitive BCH code with minimum distance $d = 2e + 1$ (see [106] for more information). We consider the code with degree-2 variable nodes and degree-n check nodes, with transmission over the BEC(ϵ). The (unscaled) uncoupled DE

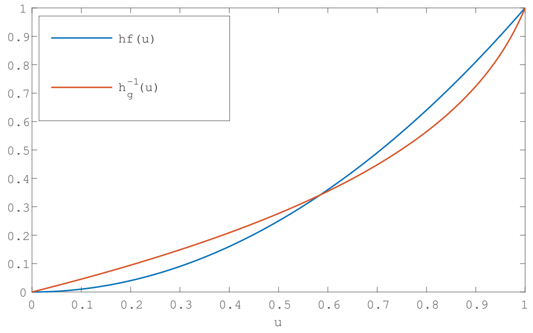

Figure 5.5: We plot the EXIT curves $f(u)$ and $g^{-1}(u)$ for $u \in [0,1]$ for the $(3,6)$-regular LDPC ensemble with transmission over the BEC(0.4881). We note that the signed area between the curves is equal to zero.

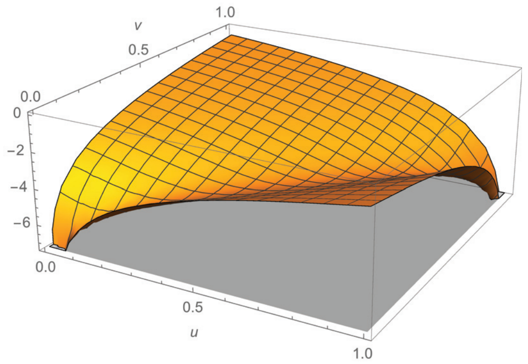

Figure 5.6: We consider the $(3,6)$-regular LDPC ensemble with transmission over the BEC(ϵ). We plot $\phi(f,g;u,v)$ for $(u,v) \in [0,1]^2$ in log scale when $\epsilon = \epsilon_{\text{MAP}} = 0.4881$. We can see that $\phi(f,g;u,v) > 0$ for $(u,v) \in (0,1)^2$ and $\phi(f,g;0,0) = \phi(f,g;1,1) = 0$.

equations are [71]

$$\begin{cases} \mathrm{x} = \epsilon \mathrm{y}, \\ \mathrm{y} = \sum_{i=e}^{n-1} \binom{n-1}{i} \mathrm{x}^i (1-\mathrm{x})^{n-i-1}. \end{cases}$$

Set $\epsilon = \epsilon_{\text{MAP}}$ and $\mathrm{y} = \mathrm{y}_{\text{MAP}} u$, $\mathrm{x} = \mathrm{x}_{\text{MAP}} v$. We then obtain the scaled equations (5.1); namely $v = f(u)$, $u = g(v)$

$$\begin{cases} f(u) = \epsilon_{\text{MAP}} \mathrm{x}_{\text{MAP}}^{-1} \mathrm{y}_{\text{MAP}} u, \\ g(v) = \mathrm{y}_{\text{MAP}}^{-1} \sum_{i=e}^{n-1} \binom{n-1}{i} \mathrm{x}_{\text{MAP}}^i \mathrm{v}^i (1 - \mathrm{x}_{\text{MAP}} v)^{n-i-1}. \end{cases}$$

The normalization condition $f(1) = g(1) = 1$ and the condition $\tilde{A}(f,g;1) = 0$ completely determine ϵ_{MAP}, x_{MAP}, y_{MAP}. The potential function and (alternative) signed area are given by

$$\phi(u,v) = \frac{\mathrm{x}_{\text{MAP}} u^2}{2 \epsilon_{\text{MAP}} \mathrm{y}_{\text{MAP}}} + \int_0^v \mathrm{d}v' \, g^{-1}(v') - uv.$$

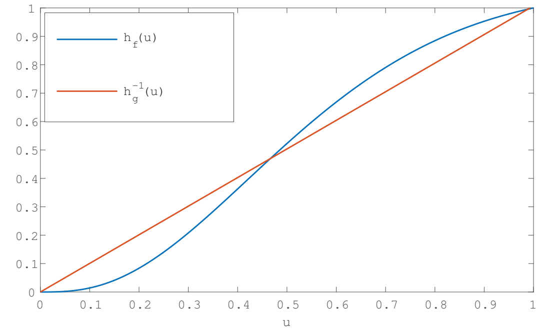

Figure 5.7: We plot the EXIT curves $f(u)$ and $g^{-1}(u)$ for $u \in [0,1]$ for the GLDPC code with $n = 15$ and $e = 3$, when transmission takes place over the BEC(0.3901). We note that the signed area between the curves is equal to zero.

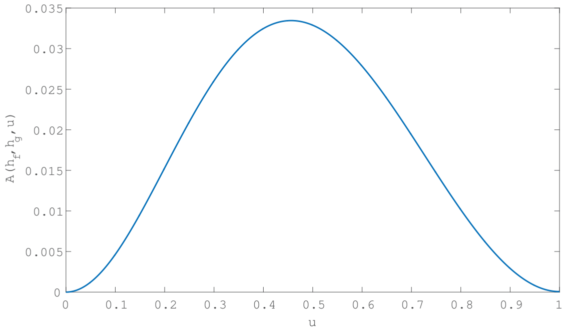

Figure 5.8: We consider the GLDPC code with $n = 15$ and $e = 3$, with transmission over the BEC(ϵ). We plot $\tilde{A}(f,g;v)$ for $v \in [0,1]$ when the channel parameter is $\epsilon = \epsilon_{\text{MAP}} = 0.3901$.

$$\tilde{A}(f,g;v) = \frac{\mathrm{x}_{\text{MAP}} v^2}{2\epsilon_{\text{MAP}} \mathrm{y}_{\text{MAP}}} - \frac{1}{\mathrm{y}_{\text{MAP}}} \sum_{i=e}^{n-1} \sum_{m=0}^{n-i-1} \binom{n-1}{i} \binom{n-i-1}{m} \frac{\mathrm{x}_{\text{MAP}}^{m+i} v^{m+i+1}}{m+i+1}.$$

The EXIT curves and signed area are illustrated in Figure 5.7. and Figure 5.8 for the GLDPC code with $n = 15$ and $e = 3$. This corresponds to $\mathrm{x}_{\text{MAP}} = 0.367$, $y_{\text{MAP}} = 0.9342$, $\epsilon_{\text{MAP}} = 0.3901$. Clearly the SPGC condition is satisfied.

5.8.3 The Gaussian Approximation

There are various forms of the Gaussian approximation [37,102,103] used to simplify the analysis of coding systems with transmission over binary memoryless symmetric (BMS) channels. Here, we consider a variant developed in [102, 103]. The method approximates the densities of the log-likelihood ratio (LLR) messages exchanged in the decoding graph with *symmetric* Gaussian densities; that is, densities of the form $\mathtt{x}(\alpha) = 1/\sqrt{2\pi\sigma^2}\exp(-\frac{(\alpha-m)^2}{2\sigma^2})$ with the property $\sigma^2 = 2m$. We also approximate the BMS channel $\mathtt{c}$ with a binary-input additive white Gaussian noise (BIAWGN) channel with parameter σ^2 and with the same entropy $H(\mathtt{c})$ as the original channel

$\mathtt{c}$. This makes the analysis one-dimensional and has been shown to serve as a good approximation.

The Gaussian approximation allows us to track the evolution of decoding by tracking the entropies of the LLR messages. Let $\psi(m)$ denote the entropy of a symmetric Gaussian density of mean m [61]. In particular, it can be expressed as

$$\psi(m) = \frac{1}{\sqrt{4\pi m}} \int_{\mathbb{R}} \mathrm{d}z\, e^{-\frac{(z-m)^2}{4m}} \log_2(1 + e^{-z}).$$

Note that $\psi(+\infty) = 0$ and $\psi(0) = 1$. We consider the $(3,6)$-regular LDPC code ensemble with transmission over the BMS.

The (unscaled) uncoupled DE equations are

$$\begin{cases} \mathrm{x} = \psi(\psi^{-1}(H(\mathtt{c})) + 2\psi^{-1}(\mathrm{y})), \\ \mathrm{y} = 1 - \psi(5\psi^{-1}(1 - \mathrm{x})). \end{cases}$$

We define m_{MAP} as the value of $\psi^{-1}(H(\mathtt{c}))$ at the MAP threshold and set $\psi^{-1}(H(\mathtt{c})) = m_{\mathrm{MAP}}$ and $\mathrm{x} = \mathrm{x}_{\mathrm{MAP}} v$, $\mathrm{y} = \mathrm{y}_{\mathrm{MAP}} u$. We then obtain the scaled equations (5.1), namely $v = f(u)$, $u = g(v)$,

$$\begin{cases} f(u) = \mathrm{x}_{\mathrm{MAP}}^{-1} \psi(m_{\mathrm{MAP}} + 2\psi^{-1}(\mathrm{y}_{\mathrm{MAP}} u)), \\ g(v) = \mathrm{y}_{\mathrm{MAP}}^{-1} - \mathrm{y}_{\mathrm{MAP}}^{-1} \psi(5\psi^{-1}(1 - \mathrm{x}_{\mathrm{MAP}} v)). \end{cases}$$

The normalization condition $f(1) = g(1) = 1$ and the condition $\tilde{A}(f, g; 1) = 0$ completely determine m_{MAP}, $\mathrm{x}_{\mathrm{MAP}}$, $\mathrm{y}_{\mathrm{MAP}}$. The potential function is given by

$$\begin{aligned} \phi(u,v) =& u(\mathrm{x}_{\mathrm{MAP}}^{-1} - v) + \mathrm{x}_{\mathrm{MAP}}^{-1} \int_0^u \mathrm{d}u'\, \psi\Big(\frac{1}{5}\psi^{-1}(1 - \mathrm{y}_{\mathrm{MAP}} u')\Big) \\ &+ \mathrm{y}_{\mathrm{MAP}}^{-1} \int_0^u \mathrm{d}v'\, \psi\Big(\frac{1}{2}\psi^{-1}(\mathrm{x}_{\mathrm{MAP}} v') - \frac{1}{2} m_{\mathrm{MAP}}\Big). \end{aligned}$$

A plot of the EXIT curves and potential function yields curves that are very similar to the case of the BEC (see e.g. Figs 5.5, 5.6).

5.8.4 Compressive Sensing

Consider a signal vector $\mathbf{s}$ of length n where the components are i.i.d. copies of a random variable S. We assume that $\mathbb{E}[S^2] = 1$ and that each component of $\mathbf{s}$ is corrupted with Gaussian noise $\mathcal{N}(0, \sigma^2 = 1/\mathtt{snr})$. We take m measurements of the signal and assume that the measurement matrix has i.i.d. Gaussian components $\mathcal{N}(0, 1/\sqrt{n})$. The measurement ratio is defined by $\delta = m/n$. Here we are interested in state evolution [54], which tracks the mean square error of the approximate message-passing (AMP) estimator (for the signal). We fix the $\mathtt{snr}$ to be large enough, and consider δ that is kept fixed as n becomes large.

The state evolution fixed point equations read

$$\begin{cases} \mathrm{y} = (\frac{1}{\mathtt{snr}} + \frac{\mathrm{x}}{\delta})^{-1}, \\ \mathrm{x} = \mathtt{mmse}(\mathrm{y}), \end{cases} \tag{5.45}$$

where the minimum mean square error function $\mathtt{mmse}$ is defined as follows. Let $Y = \sqrt{\mathtt{snr}} S + Z$ where Y is a scalar output and $Z \sim \mathcal{N}(0,1)$ and let $\hat{S}(Y, \mathtt{snr}) =$

$\mathbb{E}_{S|Y}[S|Y]$. Then, $\texttt{mmse}(\texttt{snr}) = \mathbb{E}_{S,Y}[(S - \hat{S}(Y, \texttt{snr}))^2]$. In the equations above, when we initialize with $x^{(0)} = 1$, $x^{(t)}$ is the average mean square error of the AMP estimator at iteration t.

We now put this system of equations in the form (5.1). Here, there is no trivial fixed point $\mathrm{x} = \mathrm{y} = 0$. However, the picture is very similar to LDPC coding like systems considered above. The role of the "trivial" fixed point is played by a fixed point $\mathrm{x}_*, \mathrm{y}_*$ that is obtained by initializing state evolution with $\mathrm{x} = 0$. Given $\texttt{snr}$, for δ below the algorithmic threshold this is the only fixed point, and for δ above this threshold one finds three solutions (besides $\mathrm{x}_*, \mathrm{y}_*$ which is stable, there are an unstable and a stable fixed point). Set $\mathrm{x}' = \mathrm{x} - \mathrm{x}_*$ and $\mathrm{y}' = \mathrm{y} - \mathrm{y}_*$. Equations (5.45) become

$$\begin{cases} \mathrm{y}' = -\mathrm{y}_* + (\frac{1}{\texttt{snr}} + \frac{\mathrm{x}_*+\mathrm{x}'}{\delta})^{-1}, \\ \mathrm{x}' = -\mathrm{x}_* + \texttt{mmse}(\mathrm{y}_* + \mathrm{y}'). \end{cases} \tag{5.46}$$

Note that $\mathrm{x}' = \mathrm{y}' = 0$ is a fixed point. We now scale $\mathrm{x}' = \mathrm{x}_{\mathrm{MAP}} v$, $\mathrm{y}' = \mathrm{y}_{\mathrm{MAP}} u$ where $\mathrm{x}_{\mathrm{MAP}}, \mathrm{y}_{\mathrm{MAP}}$ are chosen later on. Then (5.46) takes the form (5.1) with the EXIT curves defined as

$$\begin{cases} g(v) = -\mathrm{y}_* \mathrm{y}_{\mathrm{MAP}}^{-1} + (\frac{1}{\texttt{snr}} + \frac{\mathrm{x}_*+\mathrm{x}_{\mathrm{MAP}} v}{\delta})^{-1} \mathrm{y}_{\mathrm{MAP}}^{-1}, \\ f(u) = -\mathrm{x}_* \mathrm{x}_{\mathrm{MAP}}^{-1} + \texttt{mmse}(\mathrm{y}_* + \mathrm{y}_{\mathrm{MAP}} u) \mathrm{x}_{\mathrm{MAP}}^{-1}. \end{cases} \tag{5.47}$$

From these, one can compute the potential and the signed areas. Here, we illustrate the signed area. We have

$$g^{-1}(u) = -\mathrm{x}_* \mathrm{x}_{\mathrm{MAP}}^{-1} + \delta((\mathrm{y}_* + \mathrm{y}_{\mathrm{MAP}} u)^{-1} + \texttt{snr}^{-1})$$

from which it follows that

$$\begin{aligned} A(f, g; u) = & \frac{\delta}{\mathrm{x}_{\mathrm{MAP}} \mathrm{y}_{\mathrm{MAP}}} \ln(1 + \frac{\mathrm{y}_{\mathrm{MAP}}}{\mathrm{y}_*} u) + \frac{u\delta}{\mathrm{x}_{\mathrm{MAP}} \texttt{snr}} \\ & - \frac{1}{\mathrm{x}_{\mathrm{MAP}}} \int_0^u \mathrm{d}u' \, \texttt{mmse}(\mathrm{y}_* + \mathrm{y}_{\mathrm{MAP}} u') \end{aligned}$$

Finally, we set the signal to noise ratio to the value $\texttt{snr}_{\mathrm{MAP}}$ defined such that $f(1) = g(1) = 1$ and $A(f, g; 1)|_{\texttt{snr}_{\mathrm{MAP}}} = 0$. These conditions also determine $\mathrm{x}_{\mathrm{MAP}}, \mathrm{y}_{\mathrm{MAP}}$ (note also that these values are a "non-trivial" stable fixed point).

5.9 Appendix

This section contains the proof of the most general results that depend on taking limits. It contains proofs of the various limit results that allow generalization of arguments from the saturated case to the non-saturated case and some elementary technical results.

5.9.1 Integrability

Lemma 5.29. *Let p be an interpolating profile and assume $C_w < \infty$. Then,*

$$\lim_{A, B \to \infty} \int_{-A}^{B} \mathrm{d}z \, (p^w(z) - p(z)) = 0.$$

Proof. Assume that $C_w < \infty$. Note that by evenness of w we have

$$\begin{aligned} p^w(z) - p(z) &= \int_0^{+\infty} \mathrm{d}\tilde{z}\, w(\tilde{z})(p(z-\tilde{z}) - p(z)) + \int_{-\infty}^{0} \mathrm{d}\tilde{z}\, w(\tilde{z})(p(z-\tilde{z}) - p(z)) \\ &= \int_0^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z})\left(p(z-\tilde{z}) + p(z+\tilde{z}) - 2p(z)\right). \end{aligned}$$

Let us define $P(z) = \int_0^z \mathrm{d}\tilde{z}\, p(\tilde{z})$. Applying the Fubini theorem, we have

$$\int_{-A}^{B} \mathrm{d}z\,(p^w(z) - p(z)) = \int_0^{\infty} \mathrm{d}\tilde{z} w(\tilde{z})\left(D(B,\tilde{z}) - D(-A,\tilde{z})\right),$$

where

$$\begin{aligned} D(B,\tilde{z}) &= P(B-\tilde{z}) + P(B+\tilde{z}) - 2P(B) \\ &= \int_B^{B+\tilde{z}} \mathrm{d}\hat{z}\, p(\hat{z}) - \int_{B-\tilde{z}}^{B} \mathrm{d}\hat{z}\, p(\hat{z}) \\ &= \int_{B-\tilde{z}}^{B} \mathrm{d}\hat{z}\,(1 - p(\hat{z})) - \int_B^{B+\tilde{z}} \mathrm{d}\hat{z}\,(1 - p(\hat{z})). \end{aligned}$$

From these two expressions we obtain the two bounds

$$\begin{aligned} |D(B,\tilde{z})| &\le \tilde{z} \sup_{\hat{z}<B+\tilde{z}} p(\hat{z}), \\ |D(B,\tilde{z})| &\le \tilde{z} \sup_{\hat{z}>B-\tilde{z}} (1 - p(\hat{z})). \end{aligned}$$

Letting $K > 0$ be arbitrary, we have

$$\begin{aligned} \Big|\int_0^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z}) D(B,\tilde{z})\Big| &\le \int_0^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z}) |D(B,\tilde{z})| \\ &\le \int_K^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z})\tilde{z} + \sup_{z>B-K} (1 - p(z)) \int_0^K \mathrm{d}\tilde{z}\, w(\tilde{z})\tilde{z}. \end{aligned}$$

As $C_w < \infty$, we see, by choosing $K = B/2$, that we have

$$\lim_{B\to\infty} \int_0^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z}) D(B,\tilde{z}) = 0.$$

Similarly, we have

$$\Big|\int_0^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z}) D(-A,\tilde{z})\Big| \le \int_K^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z})\tilde{z} + \sup_{z<-A+K} p(z) \int_0^K \mathrm{d}\tilde{z}\, w(\tilde{z})\tilde{z},$$

which, by choosing $K = A/2$, gives

$$\lim_{A\to\infty} \int_0^{\infty} \mathrm{d}\tilde{z}\, w(\tilde{z}) D(-A,\tilde{z}) = 0.$$

□

5.9.2 Basic Bounds

We begin with some approximation bounds.

Lemma 5.30. *Let* p *be an interpolating profile (i.e., one satisfying* (5.7)*) and assume* $C_w < \infty$*. Then,*

$$\lim_{K\to\infty} \int_K^\infty \mathrm{d}z\,(1 - \lfloor p \rceil_K^w(z)) = 0, \tag{5.48}$$

$$\lim_{K\to\infty} \int_{-\infty}^{-K} \mathrm{d}z\,\lfloor p \rceil_K^w(z) = 0, \tag{5.49}$$

$$\lim_{K\to\infty} \int_{-K}^{K} \mathrm{d}z\,|p^w(z) - \lfloor p \rceil_K^w(z)| = 0. \tag{5.50}$$

Proof. Define

$$\xi(K) = \sup_{z \ge K}\{1 - p(z), p(-z)\},$$

and note that $\lim_{K\to\infty} \xi(K) = 0$. We have

$$\begin{aligned}
1 - \lfloor p \rceil_K(z) &\le \mathbb{1}_{\{z \le K/2\}} + \xi(K/2)\mathbb{1}_{\{K/2 < z \le K\}} \\
&= (1 - \xi(K/2))\mathbb{1}_{\{z \le K/2\}} + \xi(K/2)\mathbb{1}_{\{z \le K\}} \\
&\le \mathbb{1}_{\{z \le K/2\}} + \xi(K/2)\mathbb{1}_{\{z \le K\}},
\end{aligned}$$

from which we obtain (using changes of variables)

$$\begin{aligned}
\int_K^\infty \mathrm{d}z\,(1 - \lfloor p \rceil_K^w(z)) &= \int_K^\infty \mathrm{d}z \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, w(z - \tilde{z})(1 - \lfloor p \rceil_K(\tilde{z})) \\
&\le \int_K^\infty \mathrm{d}z\,\big(\Omega(K/2 - z) + \Omega(K - z)\xi(K/2)\big) \\
&= V(-K/2) + V(0)\xi(K/2),
\end{aligned}$$

and (5.48) now follows. The inequality (5.49) can be shown similarly by first noting that

$$\begin{aligned}
\lfloor p \rceil_K(-z) &\le (1 - \xi(K/2))\mathbb{1}_{\{z \le K/2\}} + \xi(K/2)\mathbb{1}_{\{z \le K\}} \\
&\le \mathbb{1}_{\{z \le K/2\}} + \xi(K/2)\mathbb{1}_{\{z \le K\}},
\end{aligned}$$

and writing

$$\begin{aligned}
\int_{-\infty}^{-K} \mathrm{d}z\,\lfloor p \rceil_K^w(z) &= \int_K^\infty \mathrm{d}z\,\lfloor p \rceil_K^w(-z) \\
&= \int_K^\infty \mathrm{d}z \int_{-\infty}^{K/2} \mathrm{d}\tilde{z}\, w(\tilde{z} - z)\lfloor p \rceil_K(-\tilde{z}).
\end{aligned}$$

Using again changes of variables and the upper bound on $\lfloor p \rceil_K(-\tilde{z})$, we find that $\int_{-\infty}^{-K} \mathrm{d}z\,\lfloor p \rceil_K^w(z) \le V(-K/2) + V(0)\xi(K/2)$, which proves (5.49).

Now, we show (5.50). We have

$$\begin{aligned}
|p^w(z) - \lfloor p \rceil_K^w(z)| &\le \int_{-\infty}^{-K} \mathrm{d}\tilde{z}\, w(z - \tilde{z})p(\tilde{z}) + \int_K^\infty \mathrm{d}\tilde{z}\, w(z - \tilde{z})(1 - p(\tilde{z})) \\
&\le \xi(K)\big(\Omega(-z - K) + \Omega(z - K)\big),
\end{aligned}$$

where we have used the definition of $\xi(K) = \sup_{z\geq K}\{1-p(z), p(-z)\}$. We thus obtain

$$\int_{-K}^{K} \mathrm{d}z\, |p^w(z) - \lfloor p\rceil_K^w(z)| \leq 2V(0)\xi(K)$$

and (5.50) follows. □

We can now prove Lemma 5.10. For convenience we restate the lemma.

(Lemma 5.10): Let x and y be interpolating profiles and assume the PGC and $C_w < \infty$. Then

$$\lim_{K\to\infty} \mathcal{W}_2(\lfloor x\rceil_K, \lfloor y\rceil_K) = \mathcal{W}_2(x,y).$$

Proof of Lemma 5.10. If $\mathcal{W}_2(x,y) = \infty$, then the result follows from Lemma 5.7 Equ. (5.12). We assume now that $\mathcal{W}_2(x,y) < \infty$, and it is sufficient to show that

$$\lim_{K\to\infty} \Big(\mathcal{W}_2(\lfloor x\rceil_K, \lfloor y\rceil_K) - \int_{-K}^{K} \mathrm{d}z\, I_{x,y,w}(z)\Big) = 0.$$

The expression in parentheses can be written as

$$\int_{-K}^{K} \mathrm{d}z\, (x^w(z) - \lfloor x\rceil_K^w(z))y(z) + \int_{K}^{+\infty} \mathrm{d}z \left(\int_0^1 \mathrm{d}u\, g^{-1}(u) + \int_0^1 \mathrm{d}v\, f^{-1}(v) - \lfloor x\rceil_K^w(z)\right)$$

$$= \int_{-K}^{K} \mathrm{d}z\, \big(x^w(z) - \lfloor x\rceil_K^w(z)\big)y(z) + \int_K^{\infty} \mathrm{d}z\, (1 - \lfloor x\rceil_K^w(z)),$$

where the last term follows from the fact that $A(f,g;1) = \phi(f,g;1,1) = 0$. The result now follows from Lemma (5.30). □

5.9.3 Rearrangement

Now, we focus on monotonic profiles. In particular, we prove the following lemma that is used throughout the chapter.

Lemma 5.31. *For any non-decreasing function h, we have*

$$\int_{\mathbb{R}} \mathrm{d}z\, |h^w(z) - h(z)| \leq C_w(h(+\infty) - h(-\infty)). \tag{5.51}$$

Proof. First we note that

$$h^w(z) - h(z) = \int_{\mathbb{R}} \mathrm{d}\tilde{z}(h(z-\tilde{z}) - h(z))w(\tilde{z}),$$

and we obtain

$$\begin{aligned}
&\int_{\mathbb{R}} \mathrm{d}z\, |h^w(z) - h(z)| \\
&\qquad \leq \int_{\mathbb{R}} \mathrm{d}z \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, |h(z-\tilde{z}) - h(z)| w(\tilde{z}) \\
&\qquad = \int_{\mathbb{R}} \mathrm{d}\tilde{z} \int_{\mathbb{R}} \mathrm{d}z\, |h(z-\tilde{z}) - h(z)| w(\tilde{z}) \\
&\qquad = \int_{\mathbb{R}} \mathrm{d}\tilde{z}\, (h(+\infty) - h(-\infty))|\tilde{z}| w(\tilde{z}) \\
&\qquad = C_w(h(+\infty) - h(-\infty)),
\end{aligned}$$

where the next to last step follows by the layer-cake representation and monotonicity of h. □

Let p be an interpolating profile. It it easy to see that, for any z, we have

$$\overline{\lfloor p \rceil_K}(z) = \bar{p}(z), \tag{5.52}$$

for all K large enough and that $\overline{\lfloor p \rceil_K}$ converges to $\bar{p}$ uniformly.

5.9.4 Minimizers

In this section we focus on limit results specific to consistent fixed points.

Lemma 5.32. *Assume $C_w < \infty$ and let x, y be an interpolating consistent fixed point. Let us define*

$$\begin{aligned} f^K &= f_{[\lfloor x \rceil_K, \lfloor y \rceil_K^w]}, \\ g^K &= g_{[\lfloor y \rceil_K, \lfloor x \rceil_K^w]}. \end{aligned}$$

Then,

$$\lim_{K \to \infty} \mathcal{W}_2(f^K, g^K; \lfloor x \rceil_K, \lfloor y \rceil_K) = \mathcal{W}_2(f, g; x, y).$$

Proof. From (5.40) and (5.41), for any consistent fixed point, we have

$$\int_{\mathbb{R}} \mathrm{d}z\, \phi(f, g; y(z), x(z)) = \iint_{\mathbb{R}^2} \mathrm{d}x(z) \mathrm{d}y(\tilde{z}) |z - \tilde{z}| \Omega(-|z - \tilde{z}|),$$

and, because $\lim_{z \to -\infty} z\Omega(z) = 0$, we clearly have

$$\lim_{K \to \infty} \iint_{\mathbb{R}^2} \mathrm{d}\lfloor x \rceil_K(z)\, \mathrm{d}\lfloor y \rceil_K(\tilde{z}) |z - \tilde{z}| \Omega(-|z - \tilde{z}|) = \iint_{\mathbb{R}^2} \mathrm{d}x(z)\, \mathrm{d}y(\tilde{z}) |z - \tilde{z}| \Omega(-|z - \tilde{z}|).$$

Thus, it remains only to show that

$$\lim_{K \to \infty} \int_{\mathbb{R}} \mathrm{d}z\, (\lfloor x \rceil_K(z) - \lfloor x \rceil_K^w(z)) \lfloor y \rceil_K(z) = \int_{\mathbb{R}} \mathrm{d}z\, (x(z) - x^w(z)) y(z).$$

By Lemma 5.31, we have

$$\lim_{K \to \infty} \int_{\mathbb{R} \setminus [-K, K]} \mathrm{d}z\, (x(z) - x^w(z)) y(z) = 0,$$

and by Lemma 5.30, Equ. (5.48), we have

$$\lim_{K \to \infty} \int_{\mathbb{R} \setminus [-K, K]} \mathrm{d}z\, (\lfloor x \rceil_K(z) - \lfloor x \rceil_K^w(z)) \lfloor y \rceil_K(z) = 0.$$

The result now follows from Lemma 5.30, Equ. (5.50). □

5.9.5 Second Derivative

We recall from the proof of Proposition 5.17 that, for saturated profiles,

$$\mathcal{W}_2(x_\lambda, y_\lambda) = \mathcal{W}_{2,s}(x_\lambda, y_\lambda) + \int_{\mathbb{R}} \mathrm{d}z (1 - x_\lambda^w(z)) y_\lambda(z).$$

The representation used in Lemma 5.18 for the second term is equivalent to

$$\begin{aligned}
&\int_{\mathbb{R}} \mathrm{d}z\,(1 - x^w_\lambda(z))y_\lambda(z) \\
&\qquad = \iint\limits_{[0,1]^2} \mathrm{d}u\mathrm{d}v\, V((1-\lambda)(x_0^{-1}(v) - y_0^{-1}(u)) + \lambda(x_1^{-1}(v) - y_1^{-1}(u))) \\
&\qquad = \iint\limits_{[0,1]^2} \mathrm{d}u\mathrm{d}v\, V(x_0^{-1}(v) - y_0^{-1}(u) + \lambda D(u,v)).
\end{aligned}$$

Moreover, we saw in Lemma 5.19 that $\mathcal{W}_{2,s}(x_\lambda, y_\lambda)$ is affine in λ. So, using $V''(z) = w(z)$, we immediately obtain

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}\lambda^2}\mathcal{W}_2(x_\lambda, y_\lambda) &= \iint\limits_{[0,1]^2} \mathrm{d}u\mathrm{d}v\, D(u,v)^2 w(x_0^{-1}(v) - y_0^{-1}(u) + \lambda D(u,v)) \\
&= \iint\limits_{[0,1]^2} \mathrm{d}u\mathrm{d}v\, D(u,v)^2 w(x_\lambda^{-1}(v) - y_\lambda^{-1}(u)).
\end{aligned}$$

Conclusions and Further Directions

6

6.1 Conclusions

Spatially coupled systems have recently received much attention due to their excellent performance across a variety of problems, including coding, in which they have been proved to be universally capacity-achieving [43, 63, 71], compressive sensing [52–54], and random constraint-satisfaction problems [55, 56], among others. The behavior exhibited by coupled systems has been coined “threshold saturation” and refers to the fact that the dynamic threshold of a spatially coupled system *saturates* to the static threshold of the underlying uncoupled one. In the context of coding, this means that the Belief Propagation (BP) threshold of the coupled code is equal to the maximum a-posteriori (MAP) threshold of the underlying uncoupled one.

In this thesis, we have analyzed the performance of spatially coupled systems under message-passing algorithms. We are interested in the behavior of the coupled system during the so-called “dynamic phase”, when the system parameter is between the dynamic and static thresholds, and at the static phase, when the parameter is equal to the static threshold. For this purpose, we first consider the system in the large-size limit, also called the *continuum limit*, obtained by first considering an infinite coupling chain length $L_c \to +\infty$ and next an infinite window size $w \to +\infty$. We thus approximate the original discrete system with a continuous one; this makes the analysis more tractable. Then, w express the performance of the coupled system in variational form, in terms of the continuous *potential functional* that describes its evolution along iterations of the algorithm, and we analyze this functional. The potential functional is inherently related to the “density evolution” (DE) equations that track the “error distributions” of the system; the stationary-point equations of the potential are the fixed-point equations of the DE equations.

In Chapter 2, we have considered irregular LDPC codes when transmission takes place over general binary-input memoryless symmetric-output (BMS) channels and decoding is done using the BP algorithm. We consider the system during the dynamic phase, when the channel parameter is between the BP and MAP thresholds and when the DE equations have a *unique non-trivial stable fixed point*. By plotting the vector of "error probabilities" (where the z^{th} vector component contains the error probability at the z^{th} position of the coupled chain) at different iterations of the BP algorithm, we observe that the vector (or profile) exhibits solitonic behavior. More specifically, it appears to have a fixed shape that moves at a constant speed. We conjecture that the profile indeed exhibits the solitonic behavior, and we find a formula for its velocity of propagation along the chain of coupling. This velocity is equal to the inverse of the number of iterations of BP required to decode an entire replica of the underlying system on the coupling axis. The main idea in the derivation of the formula is to rewrite the DE equations in terms of the potential functional. Once this is done, taking functional derivatives and making some algebraic manipulations yields the result. As we consider transmission over general BMS channels, the formula involves infinite-dimensional objects. We thus reduce its complexity by approximating all distributions involved in BP by symmetric Gaussian distributions [37]. This makes the distributions one-dimensional and gives relatively good estimates of the "real" (empirical) velocity. Moreover, we use our results, once specialized to the binary erasure channel (BEC), to estimate parameters that appear in the scaling law of finite-size spatially coupled (ℓ, r, L_c) ensembles [96].

The solitonic behavior is not specific to the coding problem, but is, in fact, also observed on other spatially coupled systems, such as the coupled compressive sensing and Curie-Weiss systems [53], among others. For this reason, we have considered general spatially coupled scalar systems governed by message-passing equations in Chapter 3. We also restrict ourselves in this setting to DE-like equations with only one non-trivial stable fixed point. Using again the variational tool of the potential functional, we derive the velocity of propagation of the soliton for such systems, and apply it to two examples: compressive sensing and generalized LDPC (GLDPC) codes when transmission takes place over the BEC or the binary symmetric channel (BSC).

So far, the shape of the soliton for spatially coupled systems, during the dynamic phase, has not been described analytically. Also, it is not known whether or not this shape is independent of the initial conditions on the profile. However, due to empirical observations, we conjecture that the profile attains a shape that is non-decreasing, such that its left and right limits are the trivial and non-trivial stable fixed-point values, respectively. These open questions bring into light the second part of the thesis, summarized below.

In Chapters 4 and 5, we have considered spatially coupled systems at the *static phase* and characterize the solution of the DE or DE-like equations. To do so, we introduce a new tool for the analysis of spatially coupled systems, namely the concept of *displacement convexity.* This tool makes use of an alternative structure of probability distributions; hence, it applies to an appropriate space of increasing profiles. We prove that the potential functional governing the systems we consider in these chapters are strictly convex under the alternative structure. This result implies that the potential functional admits a unique minimizing profile or, equivalently, that the DE equations governing the system admit a unique fixed point solution, in an appropriate space of profiles.

In Chapter 4, we have considered the spatially coupled (ℓ, r)-regular Gallager ensemble when transmission takes place over the BEC. We first prove that the potential functional attains a minimum; that is, that a minimizer for it exists, using the "direct method" in the calculus of variations [109, 110]. Displacement convexity can only be viewed over the space of probability distributions. Thus, we use truncation and increasing rearrangement techniques (see Section 4.4.1) to prove that the potential functional is minimized over the space of non-decreasing profiles that have finite limits. These can be seen as cumulative distribution functions (cdf's) (up to scaling), which confirms that displacement convexity is a tool appropriate for the analysis of spatially coupled systems. In fact, the left and right limits are the trivial and non-trivial stable fixed points of the DE equations, respectively; this matches perfectly with the conjecture resulting from our analysis in Chapters 2 and 3. Once we restrict our analysis to the space of cdf's (up to scaling), we prove that the potential functional is displacement convex over this space, and furthermore strictly displacement convex over the space of "centered" such profiles (where the jump from the left to the right limit is centered at the origin). We thus prove that the DE equations converge to a unique solution: a profile that is increasing from the trivial fixed-point value to the non-trivial fixed-point value, up to translation.

In Chapter 5, we have generalized our results from the previous one to general spatially coupled scalar systems, considering again the static phase. The approach in this chapter is different, as we consider the *system* of DE equations in terms of two profiles, instead of single DE recursion as considered in Chapter 5. Furthermore, we use "saturation" techniques (see Section 5.2.3) to avoid assuming that the minimizing cdf's converge to the limits "fast enough". Besides these (quite technical) aspects, the analysis follows the same steps (up to appropriate modifications). We establish the existence of a minimizer for the potential by using the direct method in the calculus of variations, although this result was already proved in [69]. We again use rearrangement inequalities to restrict our search of minimizing profiles to non-decreasing ones that can be viewed as cdf's. We show that, under mild conditions on the system, most of which are necessary for the existence of a fixed point, the potential is displacement convex. Under the conditions used in [69] to show the existence of spatial fixed points, we show that the displacement convexity is *strict*. This ensures that the pair of profiles minimizing the potential is unique up to translation. Therefore, for general spatially coupled scalar systems, we also characterize the solution to the DE equations; they are again non-decreasing profiles that increase from the trivial fixed-point value to the non-trivial one.

6.2 Further Directions

Continuum Limit:

Throughout the thesis, we have used a continuum approximation of the original discrete systems obtained by taking the spatial length L_c, and then the window size w, to be very large $L_c >> w >> 1$. This yields continuous versions of the DE equations and the potential functionals, that are originally discrete. In Chapters 2 and 3, we even confirm, using numerical simulations, that this continuum approximation is good (since the formulas for the velocities yield good estimates for the real, empirical velocities). It is an interesting open question to find the error in this approximation or at least to bound it in terms of the chain length and window size.

In Chapters 2 and 3, we have obtained an approximation of the velocity of the soliton based on the continuous formulation of the problem. We conjecture that this approximation becomes exact in an asymptotic limit of infinite spatial length

L_c and window size w (keeping the order $L_c >> w >> 1$). It is an interesting open problem to quantify the quality of this approximation already for L_c infinite and w finite but large. The numerical results tend to indicate that the approximation is already quite good for small values of w, when it is of the order of a few positions.

In the context of displacement convexity, however, we have used the continuum version of the potential functional because the continuous space is the natural setting in which displacement convexity has so far been developed [93]. An interesting question is whether it is possible to formulate the displacement convexity framework in the discrete setting, and to apply the analysis in this setting directly. More specifically, how can we define the displacement interpolant for two discrete vectors of scalar components? Once the interpolation is defined, how can we define the displacement convexity of a discrete potential? Otherwise, to account for the error made in approximating a discrete system with a continuous one, we can again try to estimate the error involved in this approximation.

The shape of the soliton:
In Chapters 2 and 3, we have considered the system in the dynamic setting, after the transient phase has passed and during the wave propagation phenomenon. We make an ansatz, namely that the profile has a unique shape, independent of the initial condition, that moves with a constant velocity. There are two interesting questions that directly arise from these assumptions. The first one concerns the minimum size of the seed required to initiate decoding or signal reconstruction at the boundaries, therefore wave propagation. The second concerns the (in)dependence between the size of the seed and the shape and speed of the soliton (as long as the seed is bigger than the minimum required size). The proof of the ansatz remains an important and interesting open question that could answer these questions. In particular, proving the ansatz may establish the independence of the size of the seed from the shape and speed of the soliton.

One way in which it is possible to characterize the *shape* of the propagating wave is to use displacement convexity for the analysis of the *dynamic state* of the spatially coupled system. In Chapters 4 and 5, we have characterized the fixed point of the DE equations using the tool of displacement convexity, during the *static phase* of the system. It would thus be interesting to reformulate the convexity problem in the context of the dynamic system. A first step in this direction is to consider the “dynamic potential” used in Chapters 2 and 3, along with the “dynamic DE equations” that still depend on time or the iteration number. Then, we can remove the “dynamic degree of freedom”, the velocity, from the profile by centering the moving profile at the origin. This is possible once we assume that the velocity of propagation is constant. Furthermore, using the tool of displacement convexity to analyze the dynamic system behavior can shed some light on the questions of the independence of the soliton shape from the initial conditions.

Approximations for the velocities:
The formulas we derive in Chapters 2 and 3 involve the entire shape of the soliton in the denominator, which could be expensive to compute. It would be desirable to find approximations (or bounds) for the velocities that are independent of the shape of the soliton, and that are good for the entire range of channel parameters between the BP and MAP thresholds, where the solitonic behavior is observed. The idea is to find

an approximation for the denominator that would involve only primitive quantities related to the underlying uncoupled ensemble (such as the degree distributions, the single system potential etc), as the numerator is already expressed in terms of these quantities. Such an approximation scheme has been proposed in [76] for the special case of coding with transmission over the BEC, where it works quite well close to the MAP threshold. It would be desirable to find an extension to more general settings.

Generalizations to analyses via displacement convexity:
By moving from Chapter 4 to Chapter 5, we have successfully generalized the analysis via displacement convexity from a particular code ensemble to general scalar systems. However, displacement convexity seems to be a generic tool that could also be applied to higher-dimensional problems, such as coding with transmission over general BMS channels or multiple-access channels. Furthermore, in Chapters 4 and 5, we have restricted our search of minimizing profiles to the space of increasing profiles. It is not clear in our settings when this inequality is strict, so we cannot exclude the existence of a minimizing pair outside the spaces of increasing profiles. It would be interesting to find the conditions under which the inequality is strict.

Bibliography

[1] M. Luby, M. Mitzenmacher, A. Shokrollah, and D. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” in *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing.* ACM, 1998, pp. 249–258.

[2] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[3] D. A. Huffman *et al.*, “A method for the construction of minimum-redundancy codes,” *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[4] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[5] ——, “Compression of individual sequences via variable-rate coding,” *IEEE transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[6] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[7] D. E. Muller, “Application of Boolean algebra to switching circuit design and to error detection,” *Transactions of the IRE Professional Group on Electronic Computers*, no. 3, pp. 6–12, 1954.

[8] I. S. Reed, “A class of multiple-error-correcting codes and the decoding schefe,” 1954.

[9] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[10] A. Hocquenghem, “Codes correcteurs d’erreurs,” *Chiffres*, vol. 2, no. 147-156, pp. 8–5, 1959.

[11] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[12] E. R. Berlekamp, *Algebraic coding theory.* McGraw-Hill New York, 1968, vol. 129.

[13] J. Massey, “Shift-register synthesis and BCH decoding,” *IEEE transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[14] M. Sudan, “Decoding of Reed-Solomon codes beyond the error-correction bound,” *Journal of complexity*, vol. 13, no. 1, pp. 180–193, 1997.

[15] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and algebraic-geometric codes,” in *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on.* IEEE, 1998, pp. 28–37.

[16] R. Koetter and A. Vardy, “Algebraic soft decoding of Reed-Solomon codes,” Oct. 14 2003, uS Patent 6,634,007.

[17] P. Elias, “Coding for noisy channels,” in *Proceedings of the Institute of Radio Engineers*, vol. 43, no. 3, 1955, pp. 356–356.

[18] J. M. Wozencraft, “Sequential decoding for reliable communication,” 1957.

[19] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[20] R. de Buda, “The upper error bound of a new near-optimal code,” *IEEE Transactions on Information Theory*, vol. 21, no. 4, pp. 441–445, 1975.

[21] ——, “Some optimal codes have structure,” *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 893–899, 1989.

[22] H.-A. Loeliger, “Averaging bounds for lattices and linear codes,” *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1767–1773, 1997.

[23] R. Urbanke and B. Rimoldi, “Lattice codes can achieve capacity on the AWGN channel,” *IEEE transactions on Information Theory*, vol. 44, no. 1, pp. 273–278, 1998.

[24] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes,” *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[25] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[26] D. J. MacKay and R. M. Neal, “Good codes based on very sparse matrices,” in *IMA International Conference on Cryptography and Coding.* Springer, 1995, pp. 100–111.

[27] ——, “Near Shannon limit performance of low density parity check codes,” *Electronics letters*, vol. 33, no. 6, pp. 457–458, 1997.

[28] D. J. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[29] D. A. Spielman, “Linear-time encodable and decodable error-correcting codes,” in *Proceedings of the Twenty-seventh Annual ACM Symposium on the Theory of Computing.* ACM, 1995, pp. 388–397.

[30] M. Sipser and D. A. Spielman, “Expander codes,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.

[31] N. Wiberg, *Codes and decoding on general graphs.* Citeseer, 1996.

[32] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[33] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Improved low-density parity-check codes using irregular graphs,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[34] ——, “Efficient erasure correcting codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[35] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *Proceedings of the Twenty-ninth Annual ACM Symposium on the Theory of Computing.* ACM, 1997, pp. 150–159.

[36] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[37] S.-Y. Chung, T. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, 2001.

[38] E. Arikan, “Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[39] E. Sasoglu, “Polar coding theorems for discrete systems,” 2011.

[40] M. Lentmaier, G. P. Fettweis, K. S. Zigangirov, and D. J. Costello Jr, “Approaching capacity with asymptotically regular ldpc codes,” in *Proc. Information Theory and Applications Workshop*, 2009, pp. 173–177.

[41] S. Kudekar, T. Richardson, and R. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 803–834, 2011.

[42] S. H. Hassani and R. Urbanke, “Universal polar codes,” in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2014, pp. 1451–1455.

[43] S. Kudekar, T. Richardson, and R. Urbanke, “Spatially coupled ensembles universally achieve capacity under belief propagation,” *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.

[44] M. Lentmaier, A. Sridharan, K. S. Zigangirov, and D. J. Costello Jr, "Terminated LDPC convolutional codes with thresholds close to capacity," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2005, pp. 1372–1376.

[45] M. Lentmaier, A. Sridharan, D. J. Costello Jr, and K. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5274–5289, 2010.

[46] S. Kudekar, C. Measson, T. Richardson, and R. Urbanke, "Threshold saturation on BMS channels via spatial coupling," in *6th International Symposium on Turbo Codes and Iterative Information Processing (ISTC).* IEEE, 2010, pp. 309–313.

[47] V. Rathi, R. Urbanke, M. Andersson, and M. Skoglund, "Rate-equivocation optimal spatially coupled LDPC codes for the BEC wiretap channel," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2011, pp. 2393–2397.

[48] A. Yedla, H. D. Pfister, and K. R. Narayanan, "Universality for the noisy Slepian-Wolf problem via spatial coupling," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2011, pp. 2567–2571.

[49] S. Kudekar and K. Kasai, "Threshold saturation on channels with memory via spatial coupling," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2011, pp. 2562–2566.

[50] ——, "Spatially coupled codes over the multiple access channel," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2011, pp. 2816–2820.

[51] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *2012 IEEE International Conference on Communications (ICC).* IEEE, 2012, pp. 2181–2186.

[52] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton).* IEEE, 2010, pp. 347–353.

[53] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 08, p. P08009, 2012.

[54] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2012, pp. 1231–1235.

[55] S. H. Hassani, N. Macris, and R. Urbanke, "Coupled graphical models and their thresholds," in *Proceedings of Information Theory Workshop IEEE (Dublin)*, 2010, pp. 1–5.

[56] ——, "Threshold saturation in spatially coupled constraint satisfaction problems," *Journal of Statistical Physics*, vol. 150, no. 5, pp. 807–850, 2013.

[57] D. Achlioptas, S. H. Hassani, N. Macris, and R. Urbanke, "Bounds for random constraint satisfaction problems via spatial coupling," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms.* SODA, 2016, pp. 469–479.

[58] S. H. Hassani, N. Macris, and R. Urbanke, "The space of solutions of coupled XORSAT formulae," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2013, pp. 2453–2457.

[59] ——, "Chains of mean field models," *Journal of Statistical Mechanics*, pp. 1–5, 2012.

[60] F. Caltagirone, S. Franz, R. G. Morris, and L. Zdeborová, "Dynamics and termination cost of spatially coupled mean-field models," *Physical Review E*, vol. 89, no. 1, p. 012102, 2014.

[61] T. Richardson and R. Urbanke, *Modern coding theory.* Cambridge University Press, 2008.

[62] N. Macris and R. Urbanke, "Statistical physics for communications, signal processing, and computer science," *http://ipg.epfl.ch/en/courses/2014-2015/statphys.*

[63] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7389–7415, 2014.

[64] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," *Advances in neural information processing systems*, vol. 13, 2001.

[65] M. Mézard, G. Parisi, and M. Virasoro, "Spin glasses and beyond," *World Scientific–Lecture Notes in Physics*, vol. 9, 1987.

[66] N. Macris, "Griffith-Kelly-Sherman correlation inequalities: A useful tool in the theory of error correcting codes," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 664–683, 2007.

[67] A. Montanari, "Tight bounds for LDPC and LDGM codes under MAP decoding," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3221–3246, 2005.

[68] S. Kudekar and N. Macris, "Sharp bounds for optimal decoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4635–4650, 2009.

[69] S. Kudekar, T. Richardson, and R. Urbanke, "Wave-like solutions of general one-dimensional spatially coupled systems," *preprint arXiv:1208.5273*, 2012.

[70] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Wave-like solutions of general 1-d spatially coupled systems," *IEEE Transactions on Information Theory*, vol. 61, no. 8, pp. 4117–4157, 2015.

[71] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of Maxwell saturation for coupled scalar recursions," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 6943–6965, 2014.

[72] R. El-Khatib, N. Macris, and R. Urbanke, "Displacement convexity, a useful framework for the study of spatially coupled codes," in *Information Theory Workshop (ITW), Sevilla, Spain.* IEEE, 2013, pp. 1–5.

[73] R. El-Khatib, N. Macris, T. Richardson, and R. Urbanke, "Analysis of coupled scalar systems by displacement convexity," in *International Symposium on Information Theory Proceedings (ISIT), Honolulu, Hawaii.* IEEE, 2014, pp. 2321–2325.

[74] R. El-Khatib and N. Macris, "The velocity of the propagating wave for general coupled scalar systems," in *Information Theory Workshop (ITW), Cambridge, UK.* IEEE, 2016.

[75] R. El-Khatib, N. Macris, T. Richardson, and R. Urbanke, "Displacement convexity in spatially coupled scalar recursions," in preparation to be submitted to the IEEE Transactions on Information Theory, 2016.

[76] R. El-Khatib and N. Macris, "The velocity of the decoding wave for spatially coupled codes on BMS channels," in *International Symposium on Information Theory Proceedings (ISIT), Barcelona, Spain.* IEEE, 2016, pp. 2119–2123.

[77] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Transactions on Information Theory*, pp. 2181–2190, 1999.

[78] R. M. Tanner, "Error-correcting coding system," Oct. 13 1981, uS Patent 4,295,218.

[79] ——, *Convolutional codes from quasi-cyclic codes: A link between the theories of block and convolutional codes.* University of California, Santa Cruz, Computer Research Laboratory, 1987.

[80] A. Sridharan, M. Lentmaier, D. J. Costello Jr, and K. Zigangirov, "Convergence analysis for a class of LDPC convolutional codes on the erasure channel," in *Annual Allerton Conference on Communication, Control and Computing (Allerton)*, 2004, pp. 953–962.

[81] M. Lentmaier and G. P. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs," in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2010, pp. 709–713.

[82] A. Giurgiu, N. Macris, and R. Urbanke, “Spatial coupling as a proof technique and three applications,” *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5281–5295, 2016.

[83] G. Richter, M. Kaupper, and K. S. Zigangirov, “Irregular low-density parity-check convolutional codes based on protographs,” in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2006, pp. 1633–1637.

[84] D. G. Mitchell, M. Lentmaier, and D. J. Costello, “New families of LDPC block codes formed by terminating irregular protograph-based LDPC convolutional codes,” in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2010, pp. 824–828.

[85] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *IPN progress report*, vol. 42, no. 154, pp. 42–154, 2003.

[86] D. Divsalar, S. Dolinar, and C. Jones, “Short protograph-based LDPC codes,” in *MILCOM 2007-IEEE Military Communications Conference.* IEEE, 2007, pp. 1–6.

[87] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, “Capacity-approaching protograph codes,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 876–888, 2009.

[88] J. Cho and L. Schmalen, “Construction of protographs for large-girth structured LDPC convolutional codes,” in *2015 IEEE International Conference on Communications (ICC).* IEEE, 2015, pp. 4412–4417.

[89] L. Schmalen, V. Aref, J. Cho, D. Suikat, D. Rösener, and A. Leven, “Spatially coupled soft-decision error correction for future lightwave systems,” *Journal of Lightwave Technology*, vol. 33, no. 5, pp. 1109–1116, 2015.

[90] D. Divsalar, S. Dolinar, and C. Jones, “Construction of protograph LDPC codes with linear minimum distance,” in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2006, pp. 664–668.

[91] C. Méasson, A. Montanari, T. Richardson, and R. Urbanke, “The generalized area theorem and some of its consequences,” *arXiv preprint cs/0511039*, 2005.

[92] R. McCann, “A convexity principle for interacting gases,” *Adv. Math.*, pp. 153–179, 1997.

[93] C. Villani, *Topics in Optimal Transportation.* AMS, 2003, vol. 58.

[94] G. Alberti and G. Bellettini, “A nonlocal anisotropic model for phase transitions Part I: The optimal profile problem,” *Math. Annalen. 310*, pp. 527–560, 1998.

[95] E. A. Carlen, M. C. Carvalho, R. Esposito, J. L. Lebowitz, and R. Marra, “Displacement convexity and minimal fronts at phase boundaries,” *Archive for rational mechanics and analysis*, vol. 194, no. 3, pp. 823–847, 2009.

[96] P. Olmos and R. Urbanke, “A scaling law to predict the finite-length performance of spatially-coupled LDPC codes,” *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3164–3184, 2014.

[97] R. El-Khatib and N. Macris, “The velocity of the propagating wave for spatially coupled systems with applications to LDPC codes,” in preparation to be submitted to the IEEE Transactions on Information Theory, 2016.

[98] V. Aref, L. Schmalen, and S. ten Brink, “On the convergence speed of spatially coupled LDPC ensembles,” in *51st Annual Allerton Conference on Comm., Control, and Comp.* IEEE, 2013, pp. 342–349.

[99] K. Takeuchi, T. Tanaka, and T. Kawabata, “A phenomenological study on threshold improvement via spatial coupling,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 5, pp. 974–977, 2012.

[100] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, “A simple proof of threshold saturation for coupled scalar recursions,” *7th Symp. on Turbo Codes and iterative Information Processing*, pp. 51–55, 2012.

[101] ——, “A simple proof of threshold saturation for coupled vector recursions,” pp. 25–29, 2012.

[102] S.-Y. Chung and G. D. J. Forney, “On the capacity of low-density parity-check codes,” in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2001, p. 320.

[103] S.-Y. Chung, “On the construction of some capacity-approaching coding schemes,” Ph.D. dissertation, Citeseer, 2000.

[104] M. Stinner, L. Barletta, and P. M. Olmos, “Finite-length scaling based on belief propagation for spatially coupled LDPC codes,” *arXiv preprint arXiv:1604.05111*, 2016.

[105] V. Pless, *Introduction to the theory of error-correcting codes.* John Wiley & Sons, 2011, vol. 48.

[106] Y.-Y. Jian, H. D. Pfister, and K. R. Narayanan, “Approaching capacity at high rates with iterative hard-decision decoding,” in *International Symposium on Information Theory Proceedings (ISIT).* IEEE, 2012, pp. 2696–2700.

[107] D. Guo, S. Shamai, and S. Verdú, “Mutual information and minimum mean-square error in Gaussian channels,” *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1261–1282, 2005.

[108] R. El-Khatib, N. Macris, and R. Urbanke, “Displacement convexity, a useful framework for the study of spatially coupled codes,” *http://arxiv.org/abs/1304.6026*, 2013.

[109] B. Dacorogna, *Direct Methods in the Calculus of Variations.* New York: Springer-Verlag, 1992.

[110] I. Fonseca and G. Leoni, *Modern Methods in the Calculus of Variations: Lp Spaces.* New York: Springer, 2007.

[111] H. J. Brascamp, E. H. Lieb, and J. M. Luttinger, "A general rearrangement inequality for multiple integrals," *Journal of Functional Analysis*, pp. 227–237, 1974.

[112] A. Burchard, "Cases of equality in the Riesz rearrangement inequality," *Annals of Mathematics, Second Series, Vol. 143, No. 3*, pp. 499–527, 1996.

[113] M. Christ and T. C. Flock, "Cases of equality in certain multilinear inequalities of Hardy-Riesz-Brascamp-Lieb-Luttinger type," *arXiv preprint arXiv:1307.8439*, 2013.

[114] T. Richardson, "Private communication," July 2013.

[115] C. Villani, *Optimal transport: old and new.* Springer Science & Business Media, 2008, vol. 338.

[116] A. Figalli, "Existence, uniqueness, and regularity of optimal transport maps," *SIAM Journal on Mathematical Analysis*, vol. 39, no. 1, pp. 126–137, 2007.

[117] N. Gozlan, C. Roberto, P.-M. Samson, and P. Tetali, "Displacement convexity of entropy and related inequalities on graphs," *Probability Theory and Related Fields*, vol. 160, no. 1-2, pp. 47–94, 2014.

[118] D. Cordero-Erausquin, W. Gangbo, and C. Houdré, "Inequalities for generalized entropy and optimal transportation," *Contemporary Mathematics*, vol. 353, pp. 73–94, 2004.

[119] G. E. Hardy, J. E. Littlewood, and G. Polya, *Inequalities.* London and New York: Cambridge University Press, 1952.

Curriculum Vitae

Personal Information

Nationality: Australian, Lebanese
Date of birth: 15.11.1990
Address: Chemin du Mottey 14, 1020 Renens, Switzerland
E-mail: rafah.elkhatib@gmail.com
Phone: (+41)78-884-3109

Education

École Polytechnique Fédérale de Lausanne (EPFL) *Sep 2012 - Dec 2016*

PhD in Communications and Computer Science
Laboratory of the Theory of Communications (**LTHC**)
Supervisors: Rüdiger Urbanke, Nicolas Macris

American University of Beirut (AUB) *Sep 2008 - Jul 2012*

Bachelor of Electrical and Computer Eng., Emphasis on Communications and Software

- Cumulative GPA: 91.82/100; Equivalence: 4/4
- Ranked 2^{nd} out of 150 in the department

Other sources and courses

- Attended Management of Innovation and Technology Transfer (MINTT) workshops at EPFL on intellectual property, start-ups, and the transfer of technology from research to industry – *Jun 2014*
- Taking Coursera course on Machine Learning, offered by Stanford University.

Experience

École Polytechnique Fédérale de Lausanne (EPFL) *Feb 2011 - Jul 2016*

Teaching assistant

- Courses: At AUB, Data Structures and Algorithms. At EPFL, Information Theory and Coding, Probabilities and Statistics, Information, Computation, and Communication.
- Express complicated concepts in simple terms to help students understand diverse topics.

École Polytechnique Fédérale de Lausanne (EPFL) *Jun 2011 - Sep 2011*

Intern in Electrical Engineering Dept., Group for Fiber Optics

- Integrated Optics: Built a set-up for the injection of light into polymer waveguides and measured subsequent power losses.
- Fiber Optics and Software: Programmed an optical microwave generator in C to enhance optical data collection of Brillouin scattering in fibers.

American University of Beirut *Apr 2010 - Jun 2010*

Computer Refurbisher (Center for Civic Engagement and Community Services)

- Collected and refurbished discarded computers from across the university campus. The computers were later distributed to municipalities that lacked funding.

Skills

Programming C++, Matlab, Mathematica, Java, LabVIEW, VHDL, ANTLR, LaTeX.
Languages English: native, Arabic: native, French: fluent.
Personal Leadership, communication, enthusiasm, team-worker, presentation skills.

AWARDS AND QUALIFICATIONS

- Received the Distinguished Service award for contributions in the graduate student association in the Computer Science Department **(EPFL)** – *Dec 2015*
- Received a Fellowship Scholarship for doctoral studies from the École Polytechnique Fédérale de Lausanne **(EPFL)** – *Sep 2012*
- Received the **Google** Anita Borg Memorial Scholarship Award – *Jun 2011*
- Granted a Full Merit Scholarship by the American University of Beirut (**AUB**) for Bachelor's degree – *Sep 2008 - Jun 2012*
- Ranked first in Beirut, seventh in Lebanon on Lebanese Baccalaureate Exams – *Jun 2008*
- Received Academic Distinction Award and Mathematics Award at Saint Mary's Orthodox College (**SMOC**) – *Jun 2008*
- Received Best Position Paper Award by the Model United Nations (**MUN**) for representing Eritrea in the Model Office of the United Nations High Commissioner for Refugees (Model **UNHCR**) – *Jun 2008*

JOURNAL PUBLICATIONS

- The velocity of the propagating wave for spatially coupled systems, *Rafah El-Khatib and Nicolas Macris – in preparation to be submitted to the IEEE Transaction on Information Theory*
- Spatially coupled scalar recursions and displacement convexity, *Rafah El-Khatib, Nicolas Macris, Thomas Richardson, Rüdiger Urbanke – in preparation to be submitted to the IEEE Transaction on Information Theory*

BOOK CHAPTERS

- Error correcting codes and spatial coupling, *Rafah E-Khatib, Jean Barbier, Ayaka Sakata, Rüdiger Urbanke*, book chapter in "Statistical Physics, Optimization, Inference, and Message-Passing Algorithms", Eds.: F. Krzakala et al, **2016**.

CONFERENCE PUBLICATIONS

- The velocity of the propagating wave for general scalar systems, *Rafah El-Khatib and Nicolas Macris*, Information Theory Workshop (**ITW 2016**), Cambridge, UK.
- The velocity of the decoding wave for spatially coupled codes on BMS channels, *Rafah El-Khatib and Nicolas Macris*, IEEE International Symposium on Information Theory (**ISIT 2016**), Barcelona, Spain.
- Analysis of coupled scalar systems by displacement convexity, *Rafah El-Khatib, Nicolas Macris, Thomas Richardson, Rüdiger Urbanke*, IEEE International Symposium on Information Theory (**ISIT 2014**) pp. 2321-2325, Honolulu, USA.
- Displacement convexity, a useful framework for the study of spatially coupled codes, *Rafah El-Khatib, Nicolas Macris, Rüdiger Urbanke*, Information Theory Workshop (**ITW 2013**), Sevilla, Spain.

SELECTED PROGRAMMING PROJECTS

- *Speed of Decoding* *EPFL, May 2015 - Jul 2016*
  As part of my PhD, I used Matlab and C to run numerical experiments of decoding on communication systems to analyze the performance of the "belief propagation" decoding algorithm on spatially coupled codes.

- *Satisfiability Modulo Theory Solvers* *AUB, Dec 2010 - Jun 2011*
  As part of an individual undergraduate research program, under the supervision of Dr. Fadi Zaraket, I built a theoretical level analysis technique that translates imperative software programs into high order logic and decides correctness using Satisfiability Modulo Theory (SMT) solvers. I also developed on existing C++ code.
- *Heuristic SAT Solver* *AUB, Sep 2009 - Feb 2010*
  In a group of two, I helped program a heuristic satisfiability (SAT) solver in C.

ORGANIZATIONAL EXPERIENCE

Conference of Western Switzerland's Universities *Aug 2014 - Dec 2016*

Representative of EPFL

- Contribute in organizing workshops and academic events in Western Switzerland, regarding logistics and finance.

Graduate Student Association of CS department *Jan 2014 - Dec 2015*

Co-President

- Received the "Distinguished Service" award by the Computer Science department.
- Organized talks, social events, and outings. This included leading teams of volunteers.
- Transformed the voluntary association to a formal association of EPFL.

Electrical Engineering Department Committee *Sep 2011 - Jul 2012*

Student body representative of department

- Contributed in voting and giving my opinion regarding any decision that concerns the student body of the department. This includes changes in the department's rules, the curriculum of the Bachelor studies in Electrical Engineering, and personal decisions concerning the enrollment and unenrollment of students on an individual level.

ICAMES Conference, Bogazici University, Istanbul *May 2011*

Team Leader

- Developed a programming language using ANTLR in a group of three.
- Organized the application procedure and transformed our project into conference material.

TALKS AND TRAINING

- Presented research results at the Information Theory Workshop (**ITW**) in Cambridge, UK – *Sep 2016*
- Invited to a Coding Theory seminar in Schloss Dagstuhl, Germany, entitled *Coding Theory in the Time of Big Data – Aug 2016*
- Presented research results at the International Symposium of Information Theory (**ISIT**) in Barcelona, Spain – *Jul 2016*
- Presented poster concerning latest research results at the European School of Information Theory (**ESIT**) in Zandvoort, the Netherlands – *Apr 2015*
- Invited to the Los Alamos National Laboratory (**LANL**) in New Mexico, USA, for a seminar in the Center of Non-linear Studies – *July 2014*
- Presented research results at **ISIT** in Honolulu, USA – *Jul 2014*
- Presented research results at **ITW** in Sevilla, Spain – *Sep 2013*
- Attended École de Physique des Houches, France, for an automn school on statistical physics, optimization, inference and message-passing algorithms, and presented poster – *Oct 2013*