

Bootstrapping the Gross-Neveu-Yukawa Archipelago and Skydiving Algorithm

Thesis by
Aike Liu

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended on May 13, 2024

© 2024

Aike Liu

ORCID: 0009-0009-8509-4635

All rights reserved

Acknowledgements

I am profoundly grateful to a number of people whose support has been indispensable in my journey of completing my PhD.

Firstly, I owe my deepest gratitude to my advisor, Dr. David Simmons-Duffin, whose guidance, patience, profound expertise, and broad knowledge base laid the cornerstone of my graduate research, as well as my future career. There are tons of moments throughout the years when I was overwhelmed by doubts of my projects, research interest, and future career, and it was only because of his encouragement that I was able to continue my PhD studies. His insights and innovative ideas greatly enriched my experience and his support has been a vital component in my growth as a researcher. Furthermore, his mentorship extended beyond academic guidance, providing me with invaluable advice on career choices and research opportunities. He is always open to questions and discussions, and I am constantly amazed by how considerate and understanding a person could be. I am immensely thankful for his generosity in sharing his knowledge and experience and his genuine care.

I would also like to thank my amazing collaborators: Dr. Rajeev Erramilli, Dr. Luca Iliesiu, Dr. Petr Kravchuk, Dr. David Poland, Dr. Ning Su, and Dr. Balt C. van Rees. Their perspectives, expertise and innovations were invaluable, and our discussions always led to better clarity and improved results. The exchange of ideas and the collective tackling of complex problems were profoundly enriching experiences that I will carry forward into my future career.

My thanks extends to the Simons Collaboration on the Nonperturbative Bootstrap which David introduced me to. Through the summer schools and the winter annual conferences of our collaboration, I have built connections and friendships with many brilliant researchers working at the frontier of my field. Our meetings are always fruitful while most enjoyable.

I want to express my gratitude to all the members of the theoretical physics group with special thanks to Dr. Hiroshi Ooguri, Dr. Anton Kapustin, and Dr. David Hsieh for serving on my candidacy and thesis committees. And a special note of thanks goes to Carol Silberstein, our administrative assistant, whose kindness and considerations have always created a welcoming and supportive environment.

I am deeply thankful to all my peer students and postdocs in the theory group, especially my officemate, Jaeha Lee, and my friend since undergraduate, Yufeng Du, for their unwavering support throughout the years. PhD is a long and most of the time, stressful period, and I could not have completed my PhD without my friends. Their wholehearted care and generosity strengthened my resolve and capacity to persevere through challenges.

Lastly, but most importantly, I would like to thank my parents. It has been almost nine years since I left home and started my studies in a foreign land, but the physical distance has never been an obstacle in our family. We are always close and there are thousands of evenings when we chatted for hours till midnight. Their unconditional love and encouragement, have not only shaped my career but also sustained me through the ups and downs of my life. I am eternally grateful for their belief in me and dedicated support.

Completing this journey would not have been possible without the contributions and support of each one of these individuals. I am deeply thankful to everyone who played a part in making this achievement possible.

Abstract

The goal of the conformal bootstrap is to solve conformal field theories (CFTs) by imposing physical constraints including symmetries and unitarity. It has been a powerful tool to rigorously constrain CFT data, especially for strongly-coupled theories where traditional perturbative methods fail. Based solely on unitarity, symmetry, and assumptions about the spectrum of scaling dimensions, the bootstrap method has produced stringent bounds on critical exponents of several universality classes describing real-world statistical and quantum phase transitions.

The numerical bootstrap method combines the physical constraints with convex optimization. Specifically, the physics problems are converted into semidefinite programs and solved numerically. Such methods have led to precise and rigorous predictions on critical exponents of condensed-matter systems, such as 3d Ising models and the $O(N)$ models. In my first research project, we perform a bootstrap analysis of a mixed system of four-point functions of bosonic and fermionic operators in parity-preserving 3d CFTs with $O(N)$ global symmetry. Our results provide rigorous bounds on scaling dimensions and OPE coefficients of the $O(N)$ symmetric Gross-Neveu-Yukawa (GNY) fixed-points, constraining these theories to live in isolated islands in the space of CFT data. We delivered the bounds on the critical points with $N = 1, 2, 4,$ and 8 which have applications to phase transitions in condensed matter systems. We were also able to demonstrate the existence of the supercurrent when supersymmetry emerges at $N = 1$ without prior assumptions of the symmetry.

On the other hand, as we progress towards larger systems to study and to obtain more precise bounds on various CFTs, the limits on computational resources cannot be overlooked. To tackle the numerical challenges and improve efficiency, my second research project studies families of semidefinite programs (SDPs) that depend nonlinearly on a small number of “external” parameters.

Such families appear universally in numerical bootstrap computations. The traditional method for finding an optimal point in parameter space works by first solving an SDP with fixed external parameters, then moving to a new point in parameter space and repeating the process. Instead, we unify solving the SDP and moving in parameter space in a single algorithm that we call “skydiving”. We test skydiving on some representative problems in the conformal bootstrap, finding significant speedups compared to traditional methods.

Published Content and Contributions

- [1] A. Liu, D. Simmons-Duffin, N. Su and B. C. van Rees,
“Skydiving to Bootstrap Islands,” [arXiv:2307.13046 [hep-th]].
<https://arxiv.org/abs/2307.13046>

Contributions:

- Identified the theoretical physical constraints on the Gross-Neveu-Yukawa (GNY) theory at criticality.
- Programmed these constraints in Haskell and ran the semidefinite program (SDP) solver on high performance computing clusters to locate the theory in a multi-dimensional parameter space.

- [2] R. S. Erramilli, L. V. Iliesiu, P. Kravchuk, A. Liu, D. Poland and D. Simmons-Duffin,
“The Gross-Neveu-Yukawa archipelago,”
Journal of High Energy Physics, Volume 2023, article number 36, (2023),
JHEP **02**, 036 (2023),
doi:10.1007/JHEP02(2023)036, [arXiv:2210.02492 [hep-th]].
[https://link.springer.com/article/10.1007/JHEP02\(2023\)036](https://link.springer.com/article/10.1007/JHEP02(2023)036)

Contributions:

- Studied families of semidefinite programs (SDPs) that depend nonlinearly on a small number of “external” parameters, a typical problem in numerical bootstrap computations.
- Designed an algorithm named “skydiving” which unifies solving the SDP and optimizing in parameter space.
- Implemented the new algorithm in C++ and wrote a Haskell interface to test the solver on real physical problems.

Table of Contents

Acknowledgements	iii
Abstract	v
Published Content and Contributions	vii
Table of Contents	viii
Chapter I: Introduction	1
Chapter II: The Gross-Neveu-Yukawa Archipelago	7
2.1 Introduction	7
2.2 Theoretical background and spectrum assumptions	12
2.3 Numerical setup	24
2.4 Results	42
2.5 Discussion	47
2.6 Appendix	49
Chapter III: Skydiving to Bootstrap Islands	60
3.1 Introduction	60
3.2 Non-linear semidefinite programming problems	63
3.3 Skydiving algorithm	73
3.4 Implementation	87
3.5 Example runs	90
3.6 Conclusions and future directions	104
Bibliography	107

Chapter 1

Introduction

Among quantum field theories (QFTs), conformal field theories (CFTs) are the theories that are invariant under the conformal group. While the familiar concept of Poincaré group, the symmetry group of relativistic field theory in flat space, leaves the flat space metric $\eta_{\mu\nu} = \text{diag}(-, +, +, +)$ invariant, the conformal group consists of spacetime transformations that preserve angles, but not necessarily distances. In terms of the spacetime metric, we are requiring that the transformation preserves the metric up to a scaling factor,

$$g'_{\mu\nu}(x') \rightarrow \Lambda(x)g_{\mu\nu}.$$

The motivations of studying conformal symmetry come from various fields of physics. First of all, CFTs are the end points of renormalization flows of QFTs and hence, studying CFTs amounts to studying the definition of QFTs. They appear in real-world condensed matter systems as CFTs describe critical points where a continuous phase transition occurs. This gives us the power to show the power law decay of correlation functions and predict the correlation length that can be confirmed by experiments. Finally, they draw attention in the realm of holography and quantum gravity thanks to the AdS/CFT correspondence.

Thanks to the fact that the conformal field theories have more symmetry than standard QFTs, a new set of methods, the conformal bootstrap, was developed to solve the space of CFTs relying purely on mathematical consistency conditions. One of the primary merits of CFT bootstrap is that the method is non-perturbative compared to traditional QFT calculations. Hence, we are

able to give rigorous bounds on strongly coupled CFTs where the perturbation theory loses its accuracy.

Let us now briefly describe what a CFT consists of and how to characterize it. The operators in a CFT can be classified into two categories, the primary operators and their descendants. The primary operators are irreducible representations of the conformal group $SO(d+1, 1)$. They are labeled by their characters, the scaling dimension Δ , and the spin ℓ . The descendants can be obtained by acting derivatives on the primary operators.

$$\text{primaries: } \mathcal{O}_{\Delta, \ell}, \quad \text{descendants: } (\partial^{\mu_1} \dots \partial^{\mu_k})(\mathcal{O}_{\Delta, \ell})$$

Given this relation, we can limit our attention to the properties of the primary operators in a certain theory. Furthermore, if we study the two-point and three-point correlation functions of the primary operators, we observe the following forms:

$$\langle \mathcal{O}_i(x_1) \mathcal{O}_j(x_2) \rangle \sim \frac{1}{|x_1 - x_2|^{2\Delta_{\mathcal{O}}}},$$

$$\langle \mathcal{O}_i(x_1) \mathcal{O}_j(x_2) \mathcal{O}_k(x_3) \rangle = \sum_{a \in \text{spin comb.}} \lambda_{ijk}^a T_{a,ijk}(x_1, x_2, x_3),$$

where the T functions are known mathematical function. The variables in these expressions are the scaling dimensions $\Delta_{\mathcal{O}}$ and the coefficients, which we call the operator product expansion (OPE) coefficients, λ_{ijk}^a . As any higher-point correlation functions can be expanded as sums of products of the two-point and three-point correlation functions, once the set of $\{\Delta_{\mathcal{O}}, \lambda_{ijk}\}$ are fixed, all the correlation functions in this CFT is known. Therefore, the aim of identifying CFT is nailed down to identify this set of parameters $\{\Delta_{\mathcal{O}}, \lambda_{ijk}\}$ of the primary operators, which we call the CFT data.

So how can we solve, or at least bound, these parameters for a CFT? To begin with, the most basic requirement we can impose is unitarity, requiring that all the states in a physical CFT have to have positive norm. From these constraints on two types of descendants, $|P|\mathcal{O}\rangle|^2, |P^2|\mathcal{O}\rangle|^2 \geq 0$, we have

$$\begin{aligned} \Delta &\geq d - 2 + \ell && \text{for } \ell \geq 1, \\ \Delta &\geq \frac{d-1}{2} && \text{for } \ell = 1/2, \\ \Delta &\geq \frac{d-2}{2} && \text{for } \ell = 0. \end{aligned}$$

But we can do much better. The major source of power of the conformal bootstrap comes from the four-point function and crossing symmetry,

$$\langle \mathcal{O}_i(x_1)\mathcal{O}_j(x_2)|\mathcal{O}_k(x_3)\mathcal{O}_l(x_4)\rangle = \pm \langle \mathcal{O}_k(x_3)\mathcal{O}_j(x_2)|\mathcal{O}_i(x_1)\mathcal{O}_l(x_4)\rangle, \quad (1.1)$$

which is the associative properties of the operator product expansion,

$$\sum \begin{array}{c} 1 \\ \diagdown \\ \text{O} \\ \diagup \\ 2 \end{array} \begin{array}{c} 4 \\ \diagdown \\ \text{O} \\ \diagup \\ 3 \end{array} = \sum \begin{array}{c} 1 \\ \diagdown \\ \text{O} \\ \diagup \\ 2 \end{array} \begin{array}{c} 4 \\ \diagdown \\ \text{O} \\ \diagup \\ 3 \end{array}$$

Just as the two-point and three-point functions are constrained by the conformal symmetry to have almost fixed forms, so are the four-point functions,

$$\langle \mathcal{O}_i(x_1)\mathcal{O}_j(x_2)\mathcal{O}_k(x_3)\mathcal{O}_l(x_4)\rangle = \sum_I g_{ijkl}^I(u, v) T_{I,ijkl}(x_1, x_2, x_3, x_4),$$

except that now the coefficients in front of the T functions are no longer a number, but a function $g_{ijkl}^I(u, v)$ of the cross-ratios

$$u = \frac{|x_1 - x_2||x_3 - x_4|}{|x_1 - x_3||x_2 - x_4|}, \quad v = \frac{|x_1 - x_2||x_3 - x_4|}{|x_2 - x_3||x_1 - x_4|}.$$

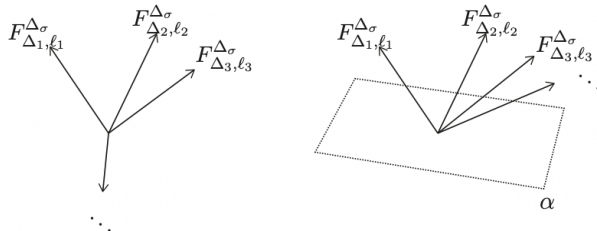
These functions g 's are also known functions of the cross-ratios and the CFT data $\{\Delta_{\mathcal{O}}, \lambda_{ijk}\}$. Hence, the crossing equation 1.1 puts constraints on our CFT data. The rest of conformal bootstrap is to employ various methods to find out which sets of $\{\Delta_{\mathcal{O}}, \lambda_{ijk}\}$ obey these constraints and which ones are excluded.

It is not surprising that such system of variables and constraints can be gigantic and very difficult to solve by hand in general. There have been extensive analytic results in 2D and some in 4D thanks to the algebraic properties of the primary operators in the particular dimensions. But in odd dimensions, the most powerful tool comes from numerical methods and convex optimizations.

Numerical bootstrap[13, 5, 19, 20], is a specific application of the conformal bootstrap approach which formulates the physical constraint as a semidefinite program (SDP) and utilizes numerical convex optimization techniques to solve the SDP. The idea can be outlined pictorially. After applying OPE to the crossing equation, we obtain some constraints in the following form:

$$\sum_{\mathcal{O}_{\Delta, \ell}} \lambda_{\mathcal{O}}^2 F_{(\Delta, \ell)}(u, v) = 0$$

where F is some linear combinations of the $g(u, v)$ functions. If we think of the F 's as vectors and since the coefficients are all non-negative, the sum can only be zero if the vectors are distributed in certain ways. The two scenarios in the figure correspond to the cases when the constraint can be satisfied and when it cannot.



The task of the semidefinite program is to find a functional α , or a hyperplane in the figure, to prove that the constraint can never be satisfied no matter what values of λ 's take and hence the CFT which $\{\Delta_{\mathcal{O}}, \lambda_{ijk}\}$ encodes is ruled out.

The scale of a typical SDP in numerical bootstrap is large, especially when we want to use more and more crossing equations of various operators to study more complex theories and to obtain more precise results. Specifically, SDP solver SDPB [1, 2] was implemented to tackle the large-scale bootstrap problems specifically. Furthermore, accompanying softwares were developed to compute conformal blocks of both scalar and spinning operators efficiently [16, 17, 15, 18]. And advanced algorithms including the Delaunay search and “cutting surface” OPE search algorithms developed in [10]. allow us to explore a multi-dimensional subspace of CFTs $\{\Delta_{\mathcal{O}}, \lambda_{ijk}\}$.

Thanks for the advanced computational methods, the numerical conformal bootstrap has led to abundant new results in the last decade. These include precise determinations of critical exponents of the 3d critical Ising model [3, 4, 5, 1, 6, 7] which describes liquid-vapor transitions and uniaxial magnets, and the $O(N)$ models [8, 9, 6] — in particular the $O(2)$ model [10] which describes the superfluid transition in ^4He , and the $O(3)$ model [11] which describes classical Heisenberg ferromagnets.

In Chapter 2, we extended the methods listed above to study the 3d $O(N)$ -symmetric Gross-Neveu-Yukawa (GNY) model. This is the first bootstrap project exploring a scalar-fermion system. We used the formalisms developed

in [14, 15] and the conformal block computation algorithms developed and implemented in `blocks_3d` [16, 17, 15, 18]. We also implemented several Haskell libraries (described in appendix 2.6.1) to efficiently and robustly set up systems of mixed correlators and compute the resulting bounds. We explored the space of external scaling dimensions and external OPE coefficients using the Delaunay search and “cutting surface” OPE search algorithms developed in [10]. We delivered the rigorous bounds on the critical points with $N = 1, 2, 4$, and 8 with 3 to 5 significant digits. At the special case $N = 1$ where the model processes an emergent supersymmetry, we were also able to demonstrate the existence of the supercurrent without prior assumptions of the symmetry.

On the other hand, as the numerical toolbox enable us to study ever larger systems of crossing equations to obtain more precise bounds on various CFTs, they pose new numerical challenges. For example, the dimension of the matrix variables we have to solve in a SDP grows from $O(10^3)$ to $O(10^6)$ and the physical parameter (scaling dimensions or OPE coefficients) space we have to search over grows from $\text{dim}= 1$ to $\text{dim}= 8$. If we simply combine the exiting Delaunay search algorithm with SDPB, we encounter the common curse of dimensionality and the computational resources needed soon become intractable even for high performance clusters. In order to bound the space of CFT data to higher precision, we must find efficient numerical methods to search over the physical parameters and solve the SDPs.

The second phase of my graduate research focused on developing such a method, which we call the skydiving algorithm, as will be elaborated on in Chapter 3. We integrated the idea of the navigator function of [12] into the solver SDPB. A navigator function is a continuous function of the physical parameter of the CFTs we wish to explore. It assigns a number to any point in the parameter space that indicates whether it is allowed or ruled out by the constraints of the target CFTs. However, the original navigator function requires solving $O(2^{\# \text{ of parameters}})$ many SDPs which is very costly. Instead, we implemented our algorithm so that the navigator function information can be used efficiently without calculating it in details. At the end of this project we achieved greater than 80% improvement in efficiency compared to previous methods.

The main body of this thesis is divided into two chapters, devoted to the two projects mentioned above respectively. Each chapter consists of a more

detailed introduction to the background and the state of the art. Then the methods and results will be presented.

Chapter 2

The Gross-Neveu-Yukawa Archipelago

2.1 Introduction

The conformal bootstrap [21, 22, 23] has emerged as a powerful tool to rigorously¹ constrain CFT data of strongly-coupled fixed points. Based solely on unitarity, symmetry, and assumptions about gaps in the spectrum of scaling dimensions, the bootstrap has produced stringent bounds on critical exponents of several universality classes describing real-world statistical and quantum phase transitions. These include the 3d critical Ising model [3, 4, 5, 1, 6, 7] which describes liquid-vapor transitions and uniaxial magnets, and the $O(N)$ models [8, 9, 6] — in particular the $O(2)$ model [10] which describes the superfluid transition in ^4He , and the $O(3)$ model [11] which describes classical Heisenberg ferromagnets.

The Ising and $O(N)$ models are perhaps the simplest 3d universality classes that can be reached via a renormalization group (RG) flow from a scalar theory. In this work, we focus on perhaps the simplest 3d universality class involving *fermions*: the $O(N)$ -symmetric Gross-Neveu-Yukawa (GNY) model. The GNY model contains N Majorana fermions ψ_i transforming in the vector representation of $O(N)$, interacting with an $O(N)$ -singlet pseudoscalar ϕ [24].

¹Throughout this paper, when stating that our bootstrap results are rigorous we mean that they do not rely on any unstated assumptions about QFTs. However, the bounds are not completely rigorous in the mathematical sense since they rely on some technical assumptions about our search algorithms (for example, over OPE space).

The Lagrangian is²

$$\mathcal{L}_{\text{GNY}} = -\frac{1}{2}(\partial\phi)^2 - i\frac{1}{2}\psi_i\rlap{/}\partial\psi_i - \frac{1}{2}m^2\phi^2 - \frac{\lambda}{4}\phi^4 - i\frac{g}{2}\phi\psi_i\psi_i. \quad (2.1)$$

This theory has a critical value of m^2 below which ϕ spontaneously gets a nonzero vacuum expectation value (VEV), giving a mass to ψ_i and, consequently, breaking parity. Above the critical value of m^2 , the VEV of ϕ vanishes, parity is preserved, and the fermions are massless. At the critical value of m^2 , this theory is expected to flow to a CFT with a single relevant parity-even $O(N)$ -singlet scalar operator $\epsilon \sim \phi^2$.

Beyond serving as one of the simplest models of scalar-fermion interactions in quantum field theory, the GNY universality classes have been proposed to describe a variety of quantum phase transitions in condensed matter systems with emergent Lorentz symmetry. For example, this model and some of its variations have been proposed to describe phase transitions in graphene (using $N = 8$) [26, 27, 28, 29], time-reversal symmetry breaking in d-wave superconductors (also for $N = 8$) [30, 31], and time-reversal-symmetry breaking transition of edge-modes in topological superconductors (for several low values of N) [32]. For the special case $N = 1$, the GNY critical point is expected to exhibit emergent supersymmetry at the transition [33].

The GNY models have been studied previously with the conformal bootstrap in [25, 34]. Those works performed a bootstrap analysis of a single four-point correlator of fermionic operators $\langle\psi_i\psi_j\psi_k\psi_l\rangle$.³ The resulting bounds exhibited a sequence of kinks on the boundary of the space of allowed CFT data, which showed good agreement with perturbative estimates of the scaling dimensions of the GNY critical points (such as the ϵ [35, 36, 37, 38, 39, 29, 40, 41] and large- N expansions [42, 43, 44, 45, 46, 25, 39, 47, 48]). However, bootstrapping the four-fermion correlator was not enough to constrain the GNY theories to lie within isolated islands.

In this work we expand this study to include a mixed system of scalar and fermionic operators, characterized by their representations under parity and

²Here we are following the conventions in appendix A of [25] and contracting the indices of the two components of the Majorana fermions by $\psi_i\psi_i = \Omega^{\alpha\beta}\psi_{i,\alpha}\psi_{i,\beta}$. Note that $\psi_i\psi_i$ is parity-odd — hence the Yukawa term $\phi\psi_i\psi_i$ preserves parity, since ϕ is a pseudoscalar.

³To be more precise, in [25] no global symmetry was assumed, while [34] considered external fermions that transformed in the vector representation of an $O(N)$ global symmetry as in the GNY theories.

	Δ_ψ	Δ_σ	Δ_ϵ	η_ψ	η_ϕ	ν^{-1}
N = 2						
$n_{\max} = 18, \Delta_{\sigma'} > 2.5$	1.0672(25)	0.657(13)	1.74(4)	0.134(5)	0.313(25)	1.26(4)
$n_{\max} = 18, \Delta_{\sigma'} > 3$	1.06861(12)	0.6500(12)	1.725(7)	0.13722(24)	0.3000(23)	1.275(7)
ϵ -exp w/DREG ₃ [41]	1.07(2)	0.6467(21)	1.724(15)	0.1400(39)	0.2934(42)	1.276(15)
Monte Carlo [49]	1.068(3)	0.655(5)	1.81(3)	0.136(5)	0.31(1)	1.19(3)
N = 4						
$n_{\max} = 18, \Delta_{\sigma'} > 3$	1.04356(16)	0.7578(15)	1.899(10)	0.08712(32)	0.5155(30)	1.101(10)
ϵ -exp w/DREG ₃ [41]	1.051(6)	0.744(6)	1.886(33)	0.102(12)	0.487(12)	1.114(33)
Monte Carlo* [50]	–	0.755(15)	1.876(13)	–	0.51(3)	1.124(13)
N = 8						
$n_{\max} = 18, \Delta_{\sigma'} > 3$	1.02119(5)	0.8665(13)	2.002(12)	0.04238(11)	0.7329(27)	0.998(12)
ϵ -exp w/DREG ₃ [41]	1.022(6)	0.852(8)	2.007(27)	0.043(12)	0.704(15)	0.993(27)
Monte Carlo* [51]	1.025(10)	0.79(1)	2.0(1)	0.05(2)	0.59(2)	1.0(1)

Table 2.1: A summary of the bootstrap estimates obtained in this paper for the three external operator that we study. Error bars in **bold** are rigorous.¹ We compare these result to those obtained from the ϵ -expansion and previous Monte Carlo studies. Methods denoted by * indicate that they study the closely related chiral Ising fixed point as opposed to the model studied in this work. The ϵ -expansion work [41] relies on the DREG₃ prescription to analytically continue spinors away from $d = 4$.

the global $O(N)$ symmetry. We focus on the critical points with $N = 1, 2, 4$, and 8, which capture the experimentally-relevant transitions described above and are outside the perturbative control of the large- N expansion. The four-point functions we analyze include all combinations allowed by the symmetries of ψ_i (the lowest dimension fermion in the vector representation of $O(N)$), $\sigma \sim \phi$ (the lowest dimension, parity-odd, $O(N)$ singlet scalar), and $\epsilon \sim \phi^2$ (the lowest dimension, parity-even, $O(N)$ singlet scalar). We refer to σ, ψ_i , and ϵ as “external operators” (as opposed to the “internal operators” that appear in their OPEs). By simultaneously imposing crossing symmetry and unitarity for all four-point functions of external operators, we show that each GNY critical point lies within an isolated island that severely constrains its low-lying scaling dimensions.

Our bounds on the scaling dimensions of σ, ψ_i , and ϵ for the theories with $N = 2, 4$, and 8 are illustrated in figures 2.1 and 2.2, and we give a numerical summary in Table 2.1. Results for the $N = 1$ theory using similar methods are shown later in Figure 2.8. For all values of N , our results are close to perturbative estimates from resummations of the ϵ -expansion. For $N = 8$, the large- N estimates are also close to the island that we find. Monte Carlo estimates for some of the scaling dimensions are also available for $N = 2, 4, 8$ [49, 55, 51];

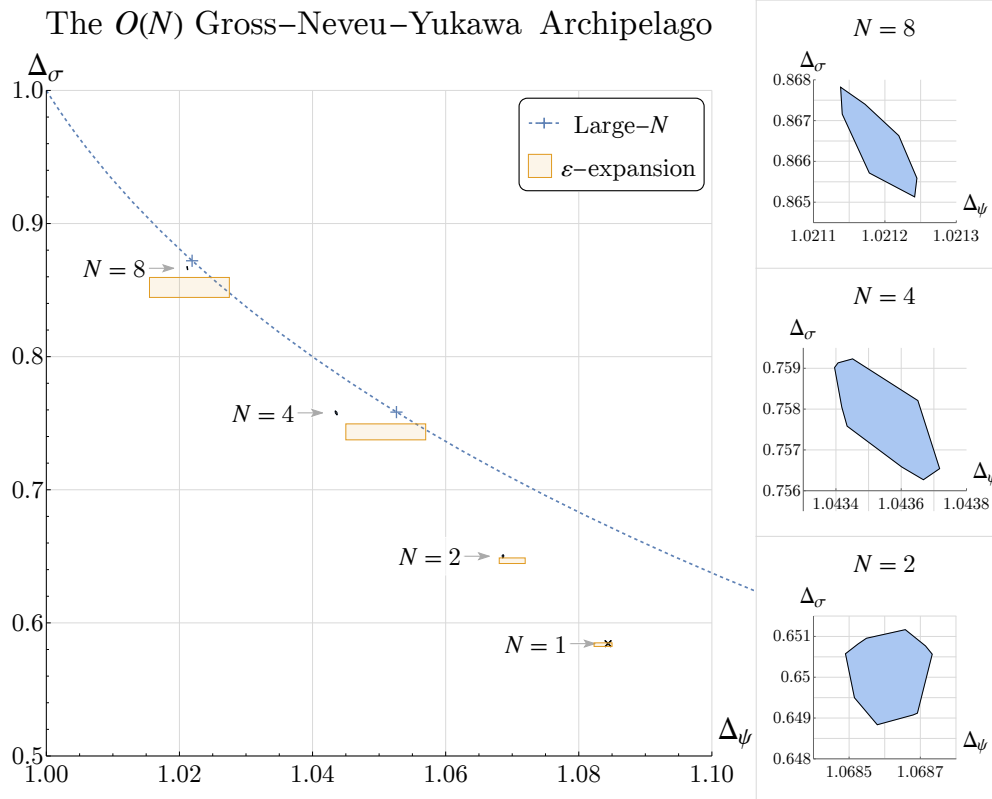


Figure 2.1: A compilation of our $N = 2, 4, 8$ islands at $n_{\max} = 18$, projected onto the $(\Delta_\psi, \Delta_\sigma)$ plane, compared against the perturbative estimates in the large- N expansion (represented by the dotted blue curve), Borel-resummations of the $(4-\epsilon)$ -expansion [41] (represented by the orange boxes), and the location of the $N = 1$ island from the $\mathcal{N} = 1$ supersymmetric Ising bootstrap [52, 53, 54] (represented by the x).

these results, while close to our islands, have error bars that, in most cases, are disallowed by the rigorous bounds obtained from the bootstrap (see section 2.4). In all cases, our work presents a significant jump in the precision of scaling dimension determinations.

An important subtlety is that there are in fact two different GNY models that are often confused in the literature, having the same number of fermions but different global symmetry groups. In addition to the $O(N)$ GNY models discussed above, there are also “chiral” GNY models that possess an $O(N/2)^2 \rtimes \mathbb{Z}_2$ global symmetry. These models, sometimes referred to as being in the “chiral Ising” universality class, are nearly degenerate with the $O(N)$ GNY models for the most common low-lying operators, being only distinguishable at high perturbative order. The Monte Carlo estimates mentioned above at

The $O(N)$ Gross–Neveu–Yukawa Archipelago

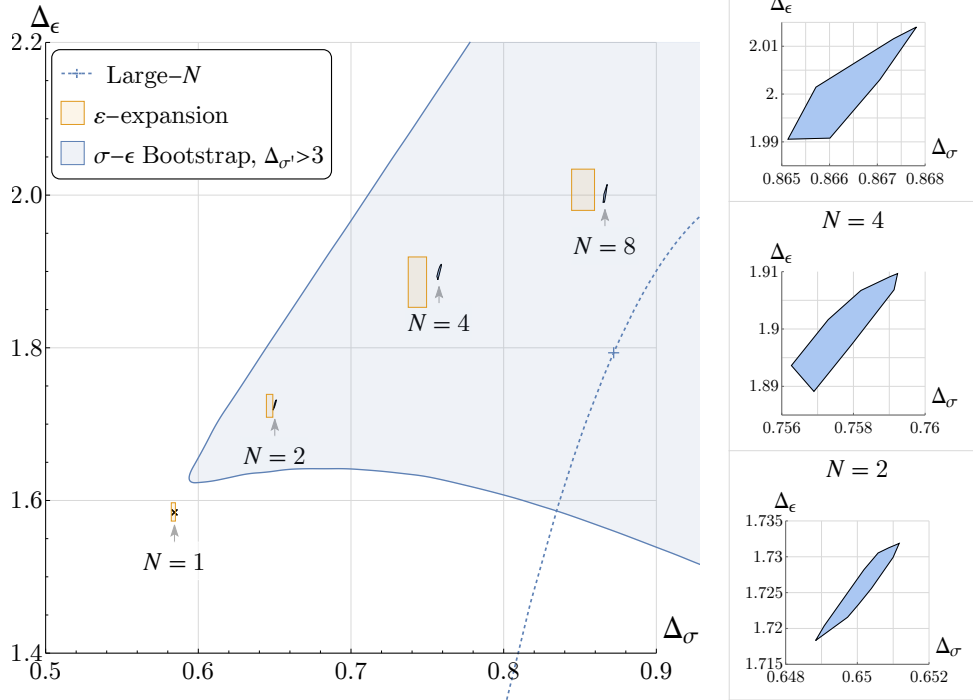


Figure 2.2: A compilation of our $N = 2, 4, 8$ islands at $n_{\max} = 18$, projected onto the $(\Delta_\sigma, \Delta_\epsilon)$ plane, compared against the perturbative estimates in the large- N expansion (represented by the dotted blue curve), Borel-resummations of the $(4-\epsilon)$ -expansion [41] (represented by the orange boxes), and the location of the $N = 1$ island from the $\mathcal{N} = 1$ supersymmetric Ising bootstrap [52, 53, 54] (represented by the x). We also superpose the general σ - ϵ bootstrap bounds with the assumption $\Delta_{\sigma'} > 3$ from [53].

$N = 4, 8$ are believed to fall in this class. However, due to the expected near-degeneracy, we posit that our bootstrap results for the leading operators also provide good (albeit non-rigorous) estimates of the scaling dimensions in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY models. In this work we review some of the existing perturbative estimates for scaling dimensions in both models and provide a number of new ones that will be useful in our bootstrap study. We will also do our best to differentiate which of the two critical models is known to describe various phase transitions discussed in the condensed matter literature.

Another interesting case is the $N = 1$ GNY model which, as previously mentioned, is believed to have emergent supersymmetry at criticality. The island that we find with mild assumptions about gaps in various sectors of this model, shown later in Figure 2.8, is fully consistent with this picture. Without making any assumptions about supersymmetry in the bootstrap setup, we find:

(i) The island lies right on a line along which the low-lying scaling dimensions are related due to supersymmetry, $\Delta_\epsilon = \Delta_\psi + \frac{1}{2} = \Delta_\sigma + 1$. (ii) The lowest dimension operator with spin-3/2 is very close to the unitarity bound across the entire island, suggesting the existence of (at least) an approximate supercurrent for any theory that lies within it. (iii) The $\mathcal{N} = 1$ super-Ising CFT, whose scaling dimensions have been tightly constrained in previous bootstrap studies by a priori assuming supersymmetry, can also be seen to live right at a tip of the island. These computations provide a nice consistency check of our implementation and suggest that there are no non-supersymmetric fixed points at $N = 1$ (consistent with our gap assumptions).

Our work is a culmination of a series of developments in the numerical bootstrap that extend practical limits to allow studies of a wider set of 3d CFTs. To set up the crossing equations, we used the formalisms developed in [14, 15] and the conformal block computation algorithms developed and implemented in `blocks_3d` [16, 17, 15, 18]. We also implemented several Haskell libraries (described in appendix 2.6.1) to efficiently and robustly set up systems of mixed correlators and compute the resulting bounds. We explored the space of external scaling dimensions and external OPE coefficients using the Delaunay search and “cutting surface” OPE search algorithms developed in [10]. Finally, we solved large-scale SDPs using the solver `SDPB` [1, 2]. We hope that these technologies can be used to place comparable constraints on more complicated 3d CFTs with fermionic degrees of freedom, including extensions of the GNY models with different global symmetries or Chern-Simons matter theories whose monopole operators carry half-integer spin.

The structure of this paper is as follows. In section 2.2, we give theoretical background, including details about the perturbative expansions of the GNY models. We also discuss the gap assumptions that we impose as well as the differences between the $O(N)$ GNY models and the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY models. In section 2.3, we discuss the numerical setup of our bootstrap problem; the reader interested solely in results for the GNY model can continue straight to the next section. There, in section 2.4 we discuss the main results of the paper and show a series of bounds on scaling dimensions of the low-lying operators in the theory. We discuss possible future directions in section 2.5.

2.2 Theoretical background and spectrum assumptions

Operator	Parity	$O(N)$	Δ at large N .	Δ in ϵ -exp.
ψ_i	+	V	$1 + \frac{4}{3\pi^2 N} + \frac{896}{27\pi^4 N^2} + \frac{\#}{N^3} + \dots$	$\frac{3}{2} - \frac{N+5}{2(N+6)}\epsilon + \dots$
$\psi'_i \sim \phi^2 \psi_i$	+	V	$3 + \frac{100}{3\pi^2 N} + \dots$	-
$\chi_i \sim \phi^3 \psi_i$	-	V	$4 + \frac{292}{3\pi^2 N} + \dots$	-
$\sigma \sim \phi$	-	S	$1 - \frac{32}{3\pi^2 N} + \frac{32(304-27\pi^2)}{27\pi^4 N^2} + \dots$	$1 - \frac{3}{N+6}\epsilon + \dots$
$\epsilon \sim \phi^2$	+	S	$2 + \frac{32}{3\pi^2 N} - \frac{64(632+27\pi^2)}{27\pi^4 N^2} + \dots$	$2 + \frac{\sqrt{N^2+132N+36}-N-30}{6(N+6)}\epsilon + \dots$
$\sigma' \sim \phi^3$	-	S	$3 + \frac{64}{\pi^2 N} - \frac{128(770-9\pi^2)}{9\pi^4 N^2} + \dots$	$3 + \frac{\sqrt{N^2+132N+36}-N-30}{6(N+6)}\epsilon + \dots$
$\epsilon' \sim \phi^4$	+	S	$4 + \frac{448}{3\pi^2 N} - \frac{256(3520-81\pi^2)}{27\pi^4 N^2} + \dots$	-
ϕ^k	$(-1)^k$	S	$k + \frac{16(3k-5)k}{3\pi^2 N} - \frac{\#}{N^2} + \dots$	-
$\sigma_T \sim \psi_{(i}\psi_{j)}$	-	T	$2 + \frac{32}{3\pi^2 N} + \frac{4096}{27\pi^4 N^2} + \dots$	-
$J^\mu \phi^2 \partial_\mu \partial^2 \phi$	-	A	$8 + \dots$	-

Table 2.2: Large- N [42, 43, 44, 45, 46, 25, 39, 47, 48] and ϵ -expansion estimates [35, 36, 37, 38, 39, 29, 40, 41] for the scaling dimensions in the GNY model. The results for $\Delta_{\psi'}$ and Δ_χ are new and derived in appendix 2.6.2. The numerators denoted by # indicate known expressions that have been omitted here for concision. Δ_ψ has a positive correction at $O(1/N^3)$ that can be found in [44] and Δ_{ϕ^k} has a negative correction (for $k > 1$) at $O(1/N^2)$ that can be found in [47].

2.2.1 Large- N and ϵ -expansion

Since we need a baseline expectation for the scaling dimensions of the external operators σ, ψ_i, ϵ , and we must also impose gaps for some low-lying internal operators, we collect the leading estimates for scaling dimensions of the $O(N)$ GNY model obtained from the large- N expansion and the ϵ -expansion in Table 2.2. In addition to the known perturbative results, we give new calculations of the leading correction at large- N for $\Delta_{\psi'_i}$ and Δ_{χ_i} in appendix 2.6.2. Some of these scaling dimensions have been computed to higher order, primarily in the context of the closely related $O(N/2)^2 \times \mathbb{Z}_2$ GNY models (discussed further below). However, the results for the leading scaling dimensions $\Delta_\psi, \Delta_\sigma, \Delta_\epsilon$ are degenerate between the $O(N)$ theory and the $O(N/2)^2 \times \mathbb{Z}_2$ theory up to 3-loop order. For large- N estimates see [42, 43, 44, 45, 46, 25, 39, 47, 48], for ϵ -expansion estimates see [35, 36, 37, 38, 39, 29, 40, 41], or see [39] for a two-sided Padé expansion in the $2 + \epsilon$ and $4 - \epsilon$ expansion. We will primarily compare our results to the 4-loop ϵ -expansion resummations performed in [41], done using a computation scheme that is believed to be applicable to the $O(N)$ GNY models.

2.2.2 The $N = 1$ theory and emergent supersymmetry

We also get additional information for our gap assumptions from the $N = 1$ case which is of interest by itself. This fixed point is expected to exhibit emergent supersymmetry in the IR and, while this has not been rigorously checked, there have been numerous perturbative tests of this proposal. The basic argument relies on the $N = 1$ critical point having a single relevant singlet scalar in the IR. That is, we expect that only a single coupling — the mass m — needs to be tuned in (2.1) to $m = m_*(g, \lambda)$ in order to reach the $O(1)$ GNY IR fixed point, and the same fixed point is reached regardless of the values of g and λ . On the other hand, for $N = 1$ and $\lambda = g^2/2$, the interaction in (2.1) is explicitly $\mathcal{N} = 1$ supersymmetric. After tuning the scalar mass to the critical point the RG flow will preserve $\mathcal{N} = 1$ supersymmetry.⁴ Since, as all other critical RG flows of (2.1), it terminates at the $O(1)$ GNY CFT, this implies that this CFT has $\mathcal{N} = 1$ supersymmetry. This argument has been more concretely probed by observing the expected supermultiplet relations $\Delta_\epsilon = \Delta_\psi + 1/2 = \Delta_\sigma + 1$ between the ϵ -expansion results for Δ_σ , Δ_ψ , and Δ_ϵ obtained at four loop order [40]. These relations were also probed using a 2-sided Padé approximation in the $2 + \epsilon$ and $4 - \epsilon$ expansions [39].

The emergence of supersymmetry was also probed non-perturbatively with the conformal bootstrap in [34], where, without explicitly imposing supersymmetry, a kink was seen close to the line relating $\Delta_\psi = \Delta_\sigma + 1/2$ when imposing the appropriate bound for $\Delta_{\sigma'}$. Nevertheless, since the CFT associated to the $O(1)$ GNY critical point has no theoretical guarantee to live at the kink and since the location of the kink in [34] changed heavily with the imposed gap on $\Delta_{\sigma'}$, it is interesting to try to constrain the fixed point to lie within an island through which the line $\Delta_\epsilon = \Delta_\psi + 1/2 = \Delta_\sigma + 1$ would pass. In this paper we will indeed find such a bound by making some mild assumptions about the gaps in the various sectors of the theory, providing additional evidence for supersymmetry in $N = 1$ case; see section 2.4 for details.

By using assuming $\mathcal{N} = 1$ supersymmetry for several scaling dimensions and

⁴This is slightly more subtle than the Lagrangian (2.1) being supersymmetric. Depending on the sign of the scalar mass term, we have either a phase with spontaneously broken supersymmetry and a massless goldstino, or a phase with spontaneously broken time-reversal symmetry and a mass gap. Supersymmetry is spontaneously broken in the $\langle \phi \rangle = 0$ phase because the fermion remains massless while the scalar gets a mass. The RG flow is not supersymmetric in this case, and so it is important to tune the effective scalar mass to zero to obtain a time-reversal invariant and supersymmetric IR theory.

OPE coefficients and using a mixed system of scalar operators [52, 53, 54] found highly accurate estimates for the CFT data in the $\mathcal{N} = 1$ super-Ising critical theory, the fixed point to which it was suggested that the $O(1)$ GNY model flows in the IR. For instance, the scaling dimension of Δ_σ was found to be [54], $\Delta_\sigma = 0.5844435(83)$, from which the scaling dimensions of Δ_ϵ and Δ_ψ can also be found. By imposing relations between scaling dimensions within the same supermultiplet, [54] also found accurate estimates for the scaling dimensions of $\Delta_{\psi'} = 3.3869(25)$, $\Delta_\chi = 4.88$, $\Delta_{\epsilon'} = 3.8869(25)$, and $\Delta_{\sigma'} = 2.8869(25)$. We will use these results to inform our gap assumptions for other values of N as we describe below.

2.2.3 Gap assumptions at $N = 1, 2, 4, 8$

We now discuss what gaps we can reasonably assume in the spectrum when trying to find islands for the GNY fixed points. Given the abundance of evidence in favor of supersymmetry for $N = 1$ it will be convenient to assume it when studying the theories with other values of N . Specifically, we can combine the highly accurate superconformal bootstrap results for the scaling dimensions of $N = 1$ theory with the results from the large- N expansion in a two-sided Padé approximation as shown in figures 2.3 and 2.4. This allows us to better estimate the scaling dimensions of the various low-lying operators in the theory and to better assess what gap assumptions we can make in the various sectors of the theory for the other values of N that we study. We want to stress that while these approximations are by no means rigorous, we only use them to motivate our gap assumptions. Given the gap assumptions, which we state explicitly below, our bounds are rigorous.

The plots in figures 2.3 and 2.4, together with the ϵ -expansion and other large- N results, lead us to make the following gap assumptions:

- Since ϵ , σ and ψ_i will serve as the external operators in our bootstrap search, we will not make any assumption about their scaling dimensions.
- We assume that ϵ is the only relevant neutral scalar operator in the theory. Consequently, we will assume $\Delta_{\epsilon'} > 3$. This assumption is substantiated by the Padé approximation obtained in the bottom-right plot in Figure 2.3 as well as by ϵ -expansion estimates.

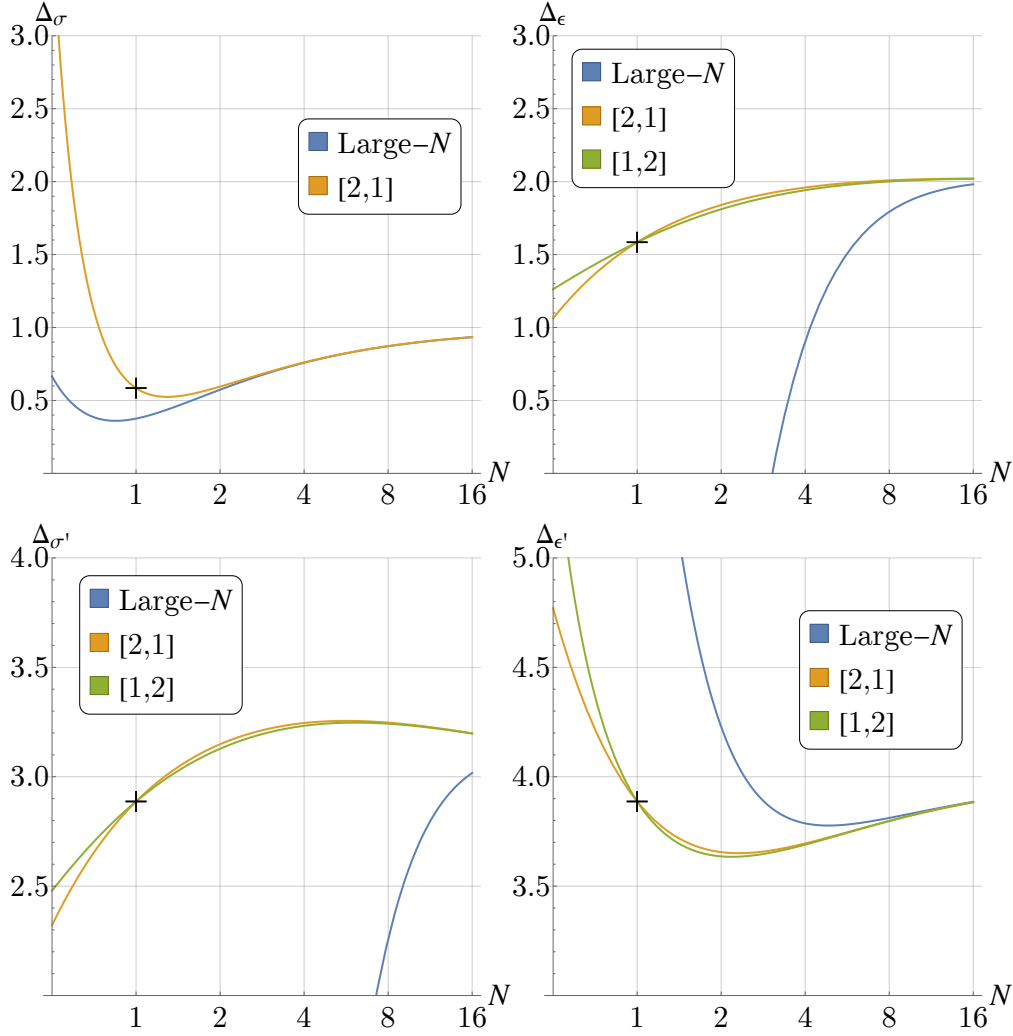


Figure 2.3: Two sided-Padé approximation for the scaling dimensions of low-lying scalars, singlets under the $O(N)$ global symmetry. The results are found using the results for the $\mathcal{N} = 1$ super-Ising model and the large- N estimates for the GNY models.

- Motivated by the large- N equation of motion which removes the operator $\psi_i \psi_i$ from the spectrum of primaries, it will also be useful to impose bounds on $\Delta_{\sigma'}$. The Padé approximation for this scaling dimension is shown in the bottom-left plot of figure 2.3. It suggests that this operator is always irrelevant for $N \geq 2$. For most of our plots we will make the assumption of irrelevance $\Delta_{\sigma'} > 3$, but for $N = 2$ we will also study the more conservative assumption $\Delta_{\sigma'} > 2.5$.
- For operators that are not part of the spectrum for $N = 1$, such as

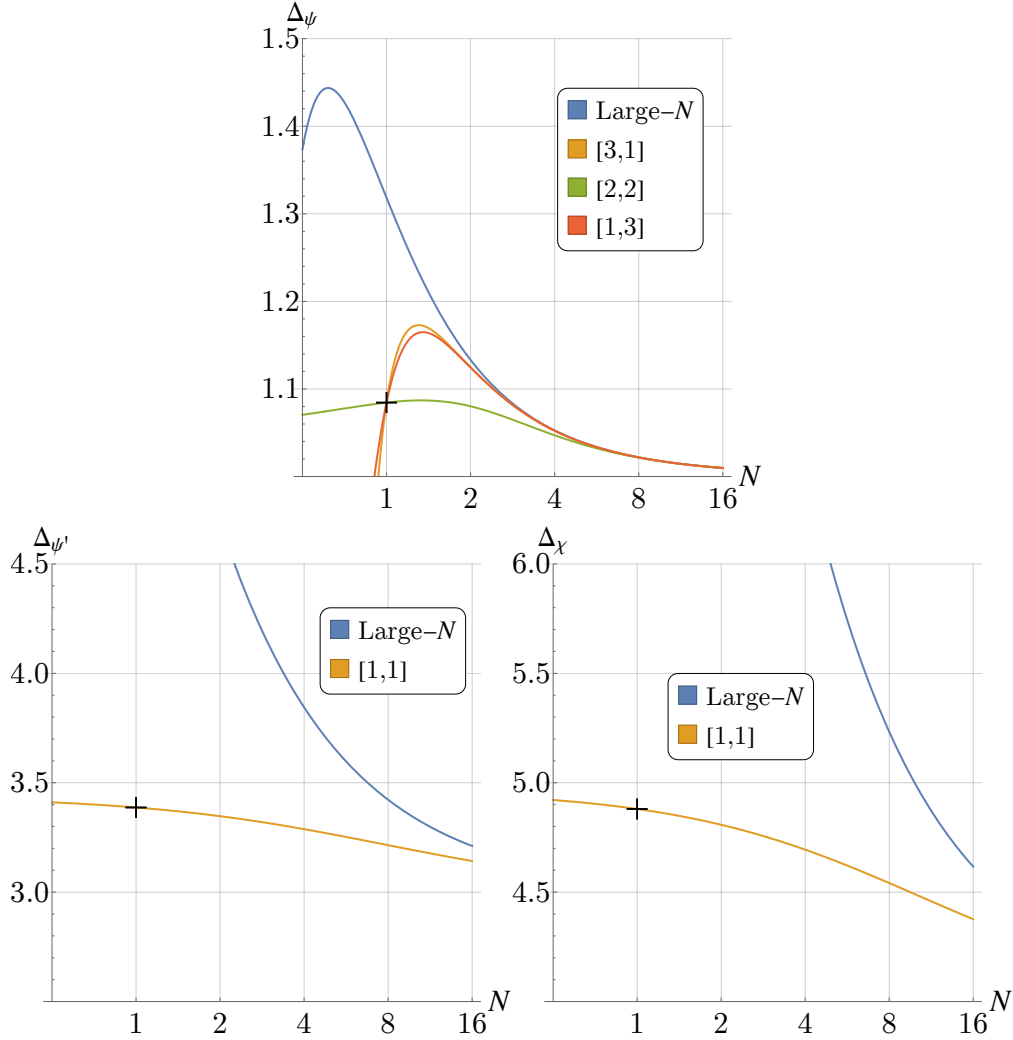


Figure 2.4: Two sided-Padé approximation for the scaling dimensions of low-lying fermionic operators, in the vector representation of the $O(N)$ global symmetry.

σ_T , we can no longer rely on the two-sided Padé approximations and will instead use the estimates from the large- N expansion as well as the past bootstrap results [34], which showed a kink in the $(\Delta_\psi, \Delta_{\sigma_T})$ at the expected location for the GNY critical point. There it was found that $\Delta_{\sigma_T} > 2$ for all N which, emboldened by the large- N estimates, we will take to be our gap value in this sector.

- We will also assume gaps for the low-lying fermionic operators. In particular, due to the equation of motion $\not{\partial}\psi_i = -g\phi\psi_i$,⁵ χ_i has a larger

⁵In mean field theory one would have $\chi_i \propto \phi\psi_i$. However, the equation of motion shows

scaling dimension than could be naïvely expected. We will conservatively assume that $\Delta_\chi > 3.5$. This assumption is well within the expectations from the two-sided Padé interpolation that is shown in the lower-right plot of figure 2.4, which in fact estimates that $\Delta_\chi > 4$ for all $N \geq 1$.

- We will also assume $\Delta_{\psi'} > 2$, based on the two-sided Padé approximation shown in the lower-left plot of figure 2.4.
- For the computations at $N = 1$ we will use a similar set of gap assumptions, taking $\Delta_{\sigma'} > 2.5$, $\Delta_{\epsilon'} > 3$, $\Delta_{\psi'} > 2$, and $\Delta_\chi > 3.5$.
- We also assume in all cases a small twist gap of 10^{-6} to improve numerical stability of the semidefinite programming algorithm. That is, we assume that all operators except the identity, the stress tensor, and the conserved $O(N)$ current have twist at least 10^{-6} higher than allowed by the unitarity bound. This a safe assumption because, based on the existing estimates of the scaling dimensions of σ and ψ , we expect the smallest twists to be on the order of 10^{-2} or more above the unitarity bound [56, 57].

Let us briefly comment on another interpretation of these gaps and why some of them may be helpful for isolating the GNY model. One can consider nonlocal deformations of the GNY fixed point, obtained by coupling its low-dimension operators to generalized free fields. Similar nonlocal deformations of the GNY models were discussed in [58] and are generalizations of the description of the long-range Ising model developed in [59, 60, 61], e.g. one can couple ψ to a generalized free field χ_{GFF} of dimension $\sim 3 - \Delta_\psi \approx 2$ and potentially flow to a nearby nonlocal fixed point. Our gap $\Delta_\chi > 3.5$ then excludes any such nearby solution to the bootstrap equations. Similarly, one could couple σ to a generalized free field σ_{GFF} of dimension $\sim 3 - \Delta_\sigma \approx 2.2 - 2.4$. Our gaps $\Delta_{\sigma'} > 2.5$ or 3 similarly exclude these possible solutions. One could also consider nonlocal deformations involving the ϵ , σ_T , or $\not\partial\psi$ operators, which would also be excluded by our gap assumptions. It would be interesting to give a more systematic study of the nonlocal fixed points (and their dualities) that could be reached by deforming the GNY models.

that $\phi\psi_i$ is removed from the primary spectrum, and therefore χ_i in interacting theory is expected to have a larger scaling dimension.

2.2.4 A distinction between two GNY fixed points

Before diving into the detailed analysis of the condensed matter applications of the $O(N)$ GNY model in (2.1), we would like to compare this model to another theory, referred to in the literature as the “chiral” GNY model, which possesses an $O(N/2)^2 \rtimes \mathbb{Z}_2$ symmetry. The distinction between the two models and physical examples of each universality class is not always clearly described in the literature. Below we shall show that while the two models flow to different fixed points, many scaling dimensions and OPE coefficients at the two fixed points agree up to a high order in a perturbative expansion. To make the distinction clear, we will henceforth refer to the two theories by their global symmetry groups.⁶

In the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY model there are two species of two-component Majorana fermions, ψ_i^L and ψ_i^R , such that each species has $N/2$ flavor components. Both species have a Yukawa coupling to a pseudoscalar ϕ , but with opposite signs:

$$\begin{aligned} \mathcal{L}_{O(N/2)^2 \rtimes \mathbb{Z}_2 \text{ GNY}} = & -\frac{1}{2}(\partial\phi)^2 - i\frac{1}{2}\psi_i^A \not{\partial}\psi_i^A - \frac{1}{2}m^2\phi^2 - \frac{\lambda}{4}\phi^4 - i\frac{g}{2}\phi(\psi_i^L\psi_i^L - \psi_i^R\psi_i^R) \\ & \text{(not what we study)}. \end{aligned} \quad (2.2)$$

Here $i = 1 \dots \frac{N}{2}$ and $A = L, R$. Each species of fermion has its own $O(N/2)$ symmetry. Additionally, there is a discrete \mathbb{Z}_2 “chiral” symmetry of $\psi_i^L \leftrightarrow \psi_i^R$, $\phi \rightarrow -\phi$ which exchanges the fermion species. Note that this symmetry is not really chiral in (2+1)d since there is no notion of “left” or “right” fermions.⁷ In total, the flavor symmetry is $O(N/2)^2 \rtimes \mathbb{Z}_2$. When the fermions spontaneously generate mass due to ϕ getting a VEV, they preserve a parity and a time-reversal symmetry but break the \mathbb{Z}_2 symmetry. The \mathbb{Z}_2 symmetry breaking of this theory is characteristic of the so-called *chiral Ising* universality classes, and it has been studied extensively [36, 29, 39, 64, 65, 66, 67, 68, 55, 69, 70, 42, 71, 72, 73, 44, 43, 74, 47, 35, 75, 76, 40, 41, 50, 51].

⁶In discussing the related gauged QED₃-GN(Y) theories, there is some literature which describes a similar distinction with different nomenclature. The $\frac{SU(N_f) \times U(1)_{\text{top}}}{\mathbb{Z}_{N_f}} \rtimes \mathbb{Z}_2^C$ -symmetric case is referred to as QED₃-GN(Y)₊ and the $\frac{(SU(N_f/2)^2 \times U(1)_b \times U(1)_{\text{top}}) \rtimes \mathbb{Z}_2^e}{\mathbb{Z}_{N_f}} \rtimes \mathbb{Z}_2^C$ -symmetric case as QED₃-GN(Y)₋ [62, 63]. As far as we are aware, the analogous notation has not been used regularly in the gauge-free GN(Y) theories.

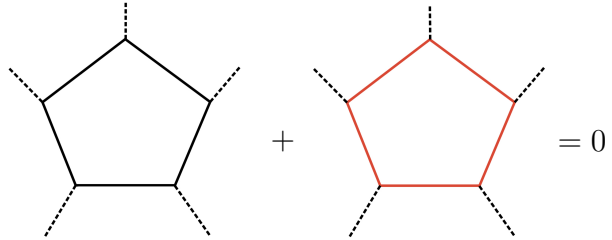
⁷It is worth noting that the chiral symmetry is indeed related to the spacetime chiral symmetry of a (3+1)d fermionic theory with 4-component fermions. This connection is laid out explicitly in appendix 2.6.3.

At the critical value of m^2 perturbative calculations show that the model (2.2) should also be described by a CFT whose scaling dimensions precisely agree at low perturbative orders with that of the critical model (2.1). Due to this seeming coincidence of the perturbative estimates of the CFT data, the distinction between the $O(N)$ GNY universality class and the $O(N/2)^2 \times \mathbb{Z}_2$ GNY a.k.a. chiral Ising universality class has been unresolved.⁸ We will now clarify this ambiguity and show that the two models are distinct when computing observables to higher perturbative orders.

Let us thus compare the two models in the large- N expansion. To compute correlators in the two models, we consider the Feynman diagrams with the leading order propagators of ϕ and ψ_i (or $\psi_i^{L,R}$) denoted by

$$\begin{aligned}
 O(N) \text{ GNY:} & \quad \langle \psi_i(x) \psi_j(y) \rangle = \delta_{ij} \text{ ————— } , \\
 O(N/2)^2 \times \mathbb{Z}_2 \text{ GNY:} & \quad \langle \psi_i^L(x) \psi_j^L(y) \rangle = \delta_{ij} \text{ ————— } , \quad \langle \psi_i^R(x) \psi_j^R(y) \rangle = \delta_{ij} \text{ ————— } , \\
 \text{In both:} & \quad \langle \phi(x) \phi(y) \rangle = \text{-----} . \tag{2.3}
 \end{aligned}$$

The only relevant interactions in the two models are of the form $\phi\psi\psi$ and $\phi(\psi^L\psi^L - \psi^R\psi^R)$, respectively, and corrections to the two-point function of ϕ^k are given by fermion loops whose vertices involve such interactions. Since in a fermionic loop, both the fermions ψ^L and ψ^R can always propagate, the only distinction between the large- N Feynman diagrams of the two models can come from fermionic loops with an odd number of vertices. In the $O(N/2)^2 \times \mathbb{Z}_2$ GNY fixed point, the contribution of such loops is always vanishing since for each loop in which L fermions propagate, there is a loop in which R fermions propagate that has the opposite sign,

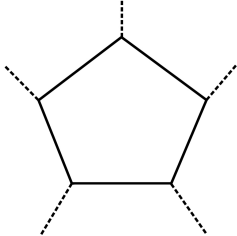


$$\text{Diagram 1} + \text{Diagram 2} = 0 . \tag{2.4}$$

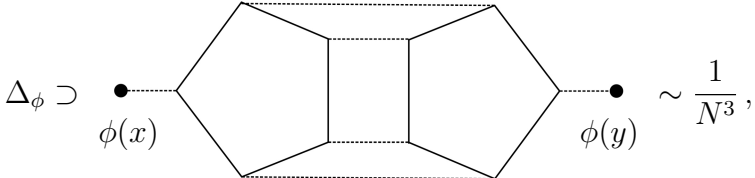
This cancellation is due to the presence of the chiral \mathbb{Z}_2 symmetry under which the field ϕ is charged. In the $O(N)$ model, however, such loops are not guaranteed to vanish when the odd number k of fermions propagating

⁸In fact it has been stated that the two models can be related through a field redefinition of ψ^L and ψ^R [27]. However, while that redefinition makes the Yukawa interaction terms in (2.2) to be the same as in (2.1), the fermionic kinetic terms then differ.

through the loop is $k \geq 5$.⁹ This leads to a five-point point function for the pseudoscalar ϕ

$$\langle \phi(x_1)\phi(x_2)\phi(x_3)\phi(x_4)\phi(x_5) \rangle = \text{Diagram} , \quad (2.5)$$


that can be explicitly checked to be non-zero at order $1/N^{3/2}$.¹⁰ Such higher k -point functions of pseudoscalars are in principle non-zero in 3d CFTs due to the existence of parity-odd k -point structures for $k \geq 5$ [14]. On the other hand, due to (2.4) the five-point point function of ϕ in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY models vanishes. While this already proves that the fixed points in the two models are distinguishable, we would like to see how these differences are manifest in more commonly discussed observables such as the scaling dimensions of ψ_i , ϕ , or ϕ^2 . For this we simply have to find the leading diagram that includes such loops with an odd number of fermion vertices. For instance, we find

$$\Delta_\phi \supset \text{Diagram} \sim \frac{1}{N^3}, \quad (2.6)$$


where we show an example diagram (rather than all the diagrams) contributing to the leading order at which the distinction between the two theories is present.

Consequently, even in these low-lying scaling dimensions the two models are different at a fairly high order in $1/N$. We can similarly determine the large- N expansion for all the other various operators in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY theory. Since these calculations require a lengthy discussion of the irreducible representation of $O(N/2)^2 \rtimes \mathbb{Z}_2$ — which is not the symmetry for the theory we rigorously constrain in this paper using the bootstrap method — we discuss these calculations in appendix 2.6.3.

⁹For $k = 3$ such loops can explicitly be shown to vanish due to $\text{Tr}(\not{x}_{12}\not{x}_{23}\not{x}_{31}) = 0$. Alternatively one can use the fact that the three-point function of pseudo-scalars in 3d is always vanishing to arrive at the same conclusion.

¹⁰We have checked this numerically, in momentum space, for fixed external momenta.

Similar logic shows that the models differ only at high order in the ϵ -expansion. For instance, it was noted in [40] for the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY model at 4-loop order that the ϵ -expansion, continued to $N = 1$, is inconsistent with the expected emergence of supersymmetry; the authors of [40] found that manually adding a 5-fermion loop diagram contribution restored the superscaling relation. As in the large- N expansion, in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY model, loops with an odd number of propagating fermions have vanishing contributions owing to the sign difference in the Yukawa coupling between fermion species. Conversely, these odd-fermion loops should be generically nonzero for the $O(N)$ theory starting at loops with 5-fermions (such as (2.5)). Such a prescription of adding back diagrams with an odd-number of propagating fermions can in principle be used when computing the scaling dimension of any operator in the theory and was named DREG₃ by the authors of [40]. Since the scaling dimensions are only affected at high loop order, the differences between the estimates of $\{\Delta_\psi, \Delta_\sigma, \Delta_\epsilon\}$ for the two models are very small ($\lesssim 3 \times 10^{-6}$) for all values of N that we consider.¹¹ Nevertheless, when comparing our results to those from the ϵ -expansion we will use the estimates from [41], which rely on the DREG₃ prescription.

Materially for this paper, we see it fit to compare this work's results with previous results for the chiral Ising universality class, since the scaling dimensions are expected to be close to those of the $O(N)$ GNY models even at low values of N . Nevertheless, our bounds will not be able to *rigorously* constrain the chiral Ising GNY models. Rigorous bounds for these models, in which the $O(N/2)^2 \rtimes \mathbb{Z}_2$ global symmetry would be explicitly implemented in the bootstrap equations, represent a separate target for the bootstrap to be studied in the future.

2.2.5 Condensed matter applications

As mentioned in the introduction, the universality class of the GNY model is used to describe a variety of quantum phase transitions in condensed matter systems. In this subsection we will list some of the proposals in the literature. Since there can be confusion regarding the two universality classes associated to the $O(N)$ GNY critical point and the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY critical point, we will revisit this point for each quantum phase transition.

¹¹We thank Michael Scherer for providing us with explicit calculations of the differences in the ϵ -expansion at four loop order.

D-wave superconductors [30, 31]: In [30], the possible quantum critical points in d -wave superconductors were classified according to their order parameter in an effort to describe anomalous behavior in cuprate superconductors. The two transitions relevant to the discussion in this paper are the transition to $d_{x^2-y^2} + is$ pairing which is described by the universality class with symmetry $O(4)^2 \times \mathbb{Z}_2$ (the chiral GNY model for $N = 8$) and the transition to $d_{x^2-y^2} + id_{xy}$ pairing which can be seen to correspond to the universality class with symmetry $O(8)$ (the non-chiral $N = 8$ GNY model). As shown in [30], these are the only two transitions that have a nodal quasiparticle momentum distribution curve with a width proportional to $k_B T$. The additional requirement that the superconductor exhibits negligible scattering along the $(1, 0)$ and $(0, 1)$ directions uniquely isolates the transition $d_{x^2-y^2} + id_{xy}$ which thus underpins the importance of the $O(8)$ GNY model for phase transitions in such superconductors.

Chern insulators, and topological superconductors [33]: There are more systems like the d -wave superconductors which have come into focus in the condensed matter community in recent years, which all possess the common factor of time-reversal symmetry breaking (TRSB). TRSB has been known to the condensed matter community for decades, and systems with a broken time-reversal symmetry are known to have the integer quantum anomalous Hall effect. Prototypical examples of these systems include the Haldane model (a Chern insulator in Cartan symmetry class A) [77]. The $O(N)$ -symmetric critical point represents a phase transition between the time-reversal-preserving class DIII and the time-reversal-breaking class D, which in the case $N = 1$ is discussed in [33]. More specifically, the universality class of the $O(1)$ GNY model is expected to describe a quantum phase transition, with emergent supersymmetry, at the boundary of a topological superconductor where the superconducting Majorana edge modes begin to gap out. Phases associated to this transition are expected to be found for a thin film of superfluid He₃-B.

In general, it is understood that the cases $N > 1$ also should have the same phase transition between symmetry classes DIII and D, though we are not aware of any proposed experimental design for empirical observation.¹²

¹²The notion of an interaction spontaneously modifying topological order is a matter of great interest, as it is believed that interactions break the class D free fermion classification of the topological invariant from \mathbb{Z} to \mathbb{Z}_{16} [78]. In other words, the topological invariants of

Graphene [26, 27, 28, 29]: The distinction between the $O(N)$ and $O(N/2)^2 \times \mathbb{Z}_2$ theories is also apparent in the case of graphene lattices. In [27] the authors exhaustively enumerated the various order parameters that are expected to exist in graphene lattice theories. It's understood that the chiral Ising universality class, with symmetry $O(N/2)^2 \times \mathbb{Z}_2$, describes the semimetal to charge density wave insulator transition on a honeycomb lattice [26]; specifically for $N = 8$ it describes a theory of spinful fermions and for $N = 4$ it describes spinless fermions. In these two cases, time-reversal symmetry is preserved, but the fermions are gapped out, leading to an insulating phase. The corresponding time-reversal-breaking $O(N)$ case is also expected to exist, but with a different order parameter which does break time-reversal symmetry.

2.3 Numerical setup

2.3.1 SDP formulation of general crossing equations

The system of crossing equations that we study in this paper is fairly complicated: it involves many correlation functions, several of which involve operators with non-trivial spin and/or flavor charges. On top of it, we are working with fermions and need to keep track of lots of minus signs associated with permutations. As a result, rewriting our system of equations in an SDP form suitable for numerical analysis is a rather non-trivial task and is especially prone to human error.

Motivated by this, and also with a view towards future applications, we developed a computer code `bootstrap-bounds` (see appendix 2.6.1) which handles most of the bookkeeping associated with passing from physically-transparent crossing equations to the SDP form. In this section we describe the basic algorithm that it uses, in the context of a general conformal bootstrap problem.

Let us consider a general conformal bootstrap problem for a set of external primary operators $\mathcal{O}_1, \dots, \mathcal{O}_n$. We allow these operators to have arbitrary spins and flavor symmetry representations, and package their dependence on space-time coordinates and the various polarization indices into an abstract argument \mathbf{p} , i.e. we write $\mathcal{O}_i(\mathbf{p}_i)$. In particular we assume that the action of all known symmetries on \mathcal{O}_i is expressed in terms of the action on \mathbf{p}_i . For

the $O(N)$ -symmetric, time-reversal-breaking theories should have topological order defined by $\nu = N \bmod 16$, since there is an adiabatic way to get from a nontrivial invariant with $\nu = 16$ to a trivial invariant with $\nu = 0$. However, this procedure requires breaking the $O(N)$ symmetry while deforming the theory from the two topological phases.

simplicity, we assume that all operators can be chosen to be Hermitian in Lorentzian signature (as is the case in our setup),

$$(\mathcal{O}_i(\mathbf{p}_i))^\dagger = \mathcal{O}_i(\mathbf{p}_i). \quad (2.7)$$

Furthermore, we introduce the notation $\mathcal{O}_{\Delta,\rho}$ for the primary operators that can appear in the OPE of \mathcal{O}_i but are not among the \mathcal{O}_i , with Δ denoting the scaling dimension, and ρ all the other quantum numbers (such as spin, flavor symmetry representation, space parity, etc.).

First, we consider the three-point functions. We will need the following two types,

$$\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_k(\mathbf{p}_3) \rangle \quad \text{and} \quad \langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_{\Delta,\rho}(\mathbf{p}_3) \rangle. \quad (2.8)$$

Let us focus on the former. It can be written as

$$\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_k(\mathbf{p}_3) \rangle = \sum_a l_{ijk}^a Q_{a,ijk}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3), \quad (2.9)$$

where $Q_{a,ijk}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ is some basis of three-point tensor structures which we are free to choose. Intuitively, this statement is clear, but it turns out we need to formalize this a little in order to have a well-defined algorithm. Formally, choosing a basis of $Q_{a,ijk}$ amounts to the following:

- *For any choice of the ordered triple i, j, k and the index a , specify a function $Q_{a,ijk}$ in three variables $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, which is invariant under all the available symmetries when transformed with the quantum numbers of \mathcal{O}_i at \mathbf{p}_1 , of \mathcal{O}_j at \mathbf{p}_2 , and of \mathcal{O}_k at \mathbf{p}_3 .*

Note that the order of the arguments $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ is fixed: for a function, we know what is the first, what is the second, and what is the third argument. This means that for fixed i, j, k the functions $Q_{a,ijk}, Q_{a,jik}, Q_{a,kji}, \dots$ all require separate choices. But of course, the corresponding physical correlation functions

$$\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_k(\mathbf{p}_3) \rangle, \quad \langle \mathcal{O}_j(\mathbf{p}_1)\mathcal{O}_i(\mathbf{p}_2)\mathcal{O}_k(\mathbf{p}_3) \rangle, \quad \langle \mathcal{O}_k(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_i(\mathbf{p}_3) \rangle, \dots \quad (2.10)$$

are all related to each other in the obvious way (taking into the account fermionic permutation signs), and this induces a relation between the corresponding OPE coefficients $l_{ijk}^a, l_{jik}^a, l_{kji}^a, \dots$. We then demand, as is always

possible to do, that the functions $Q_{a,ijk}$ are chosen so that the OPE coefficients l_{ijk}^a are functions of the *unordered triple* (i, j, k) . In other words, so that

$$l_{ijk}^a = l_{jik}^a = l_{kji}^a = \dots \quad (2.11)$$

In terms of $Q_{a,ijk}$ this means

$$Q_{a,ijk}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \pm Q_{a,jik}(\mathbf{p}_2, \mathbf{p}_1, \mathbf{p}_3) = \dots, \quad (2.12)$$

where \pm account for fermionic permutation signs. Since these signs are included here, (2.11) is true even when some \mathcal{O}_i are fermions. This choice of $Q_{a,ijk}$ is provided to the algorithm by the user; in this way, the algorithm does not have to know about the permutation properties of the operators, and can reason in terms of the simple coefficients l_{ijk}^a . Furthermore, we require $Q_{a,ijk}$ to be chosen so that $l_{ijk}^a \in \mathbb{R}$, which is again always possible to ensure.

Next we consider the three-point functions

$$\langle \mathcal{O}_i(\mathbf{p}_1) \mathcal{O}_j(\mathbf{p}_2) \mathcal{O}_{\Delta,\rho}(\mathbf{p}_3) \rangle, \quad (2.13)$$

where the convention is the same. Concretely, we write

$$\langle \mathcal{O}_i(\mathbf{p}_1) \mathcal{O}_j(\mathbf{p}_2) \mathcal{O}_{\Delta,\rho}(\mathbf{p}_3) \rangle = \sum_a l_{ij;\Delta,\rho}^a Q_{a,ij;\Delta,\rho}. \quad (2.14)$$

Now we only have to worry about the ordering in the pair i, j : we agree to always keep the generic operator $\mathcal{O}_{\Delta,\rho}$ at \mathbf{p}_3 .¹³ So in this case we need to provide only two structures for each pair i, j : $Q_{a,ij;\Delta,\rho}$ and $Q_{a,ji;\Delta,\rho}$. We again choose them in a way such that

$$l_{ij;\Delta,\rho}^a = l_{ji;\Delta,\rho}^a \in \mathbb{R}. \quad (2.15)$$

The above convention allows us to interface the general conformal block code `blocks_3d` [18] from our algorithm. In order to produce a conformal block for the four-point function $\langle \mathcal{O}_i(\mathbf{p}_1) \mathcal{O}_j(\mathbf{p}_2) \mathcal{O}_k(\mathbf{p}_3) \mathcal{O}_l(\mathbf{p}_4) \rangle$ for the exchange of $\mathcal{O}_{\Delta,\rho}$,¹⁴ this code requires the user to specify the three-point structures for

$$\langle \mathcal{O}_i(\mathbf{p}_1) \mathcal{O}_j(\mathbf{p}_2) \mathcal{O}_{\Delta,j}(\mathbf{p}_3) \rangle \quad \text{and} \quad \langle \mathcal{O}_l(\mathbf{p}_1) \mathcal{O}_k(\mathbf{p}_2) \mathcal{O}_{\Delta,j}(\mathbf{p}_3) \rangle. \quad (2.16)$$

¹³Recall that we have agreed not to use the label $\mathcal{O}_{\Delta,\rho}$ for any of the \mathcal{O}_i .

¹⁴The code `blocks_3d` can only compute the 3d conformal blocks; the flavor structure is then added by an additional layer of code.

Our algorithm can simply look up the functions $Q_{a,ij;\Delta,\rho}$ and $Q_{b,lk;\Delta,\rho}$ and pass them to `blocks_3d`.

As a result of our conventions, the expansion of the four-point function becomes

$$\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_l(\mathbf{p}_4) \rangle = \sum'_{\Delta,\rho} \sum_{a,b} l_{ij;\Delta,\rho}^a l_{kl;\Delta,\rho}^b G_{ab,ijkl,\Delta,\rho}(\mathbf{p}_1, \dots, \mathbf{p}_4), \quad (2.17)$$

where $G_{ab,ijkl,\Delta,\rho}$ is the block returned by `blocks_3d`, and $\sum'_{\Delta,\rho} = \sum_{\Delta,\rho} + \sum_n$ denotes the sum over $\mathcal{O}_{\Delta,\rho}$ appearing in $\mathcal{O}_i \times \mathcal{O}_j$ OPE plus the sum over the \mathcal{O}_n appearing in the same OPE.

The above describes the conventions for the three-point functions, but we also need to specify the crossing equations. Any crossing equation involves only one four-point function, expanded in different channels. For a given four-point function $\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2)\mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_l(\mathbf{p}_4) \rangle$, there are at most 3 distinct channels, depending on which operator out of $\mathcal{O}_j, \mathcal{O}_k, \mathcal{O}_l$ we take the OPE of \mathcal{O}_i with. In any case, an equality of two channels in a four-point function takes the form

$$\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2) | \mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_l(\mathbf{p}_4) \rangle \pm \langle \mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_j(\mathbf{p}_2) | \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_l(\mathbf{p}_4) \rangle = 0, \quad (2.18)$$

where we used $|$ to separate the groups of operators between which we take the OPE. The \pm sign is chosen based on the statistics of the operators. Note that

$$\langle \mathcal{O}_j(\mathbf{p}_1)\mathcal{O}_i(\mathbf{p}_2) | \mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_l(\mathbf{p}_4) \rangle \pm \langle \mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_j(\mathbf{p}_1) | \mathcal{O}_i(\mathbf{p}_2)\mathcal{O}_l(\mathbf{p}_4) \rangle = 0 \quad (2.19)$$

expresses the same equality. We will use the convention in which we order operators in such a way that the two terms in the crossing equation differ by swapping the operators at \mathbf{p}_1 and \mathbf{p}_3 , like in (2.18). We choose some complete and independent set of crossing equations, written in this convention.

We can expand each four-point function in a basis of four-point tensor structures,

$$\langle \mathcal{O}_i(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2) | \mathcal{O}_k(\mathbf{p}_3)\mathcal{O}_l(\mathbf{p}_4) \rangle = \sum_I g_{ijkl}^I(z, \bar{z}) T_{I,ijkl}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4), \quad (2.20)$$

$$\langle \mathcal{O}_k(\mathbf{p}_1)\mathcal{O}_j(\mathbf{p}_2) | \mathcal{O}_i(\mathbf{p}_3)\mathcal{O}_l(\mathbf{p}_4) \rangle = \sum_I g_{kjil}^I(z, \bar{z}) T_{I,kjil}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4). \quad (2.21)$$

The conformal block expansions take the form

$$g_{ijkl}^I(z, \bar{z}) = \sum'_{\Delta, \rho} \sum_{a, b} l_{ij; \Delta, \rho}^a l_{kl; \Delta, \rho}^b G_{ab, ijkl, \Delta, \rho}^I(z, \bar{z}), \quad (2.22)$$

$$g_{kjil}^I(z, \bar{z}) = \sum'_{\Delta, \rho} \sum_{a, b} l_{kj; \Delta, \rho}^a l_{il; \Delta, \rho}^b G_{ab, ijkl, \Delta, \rho}^I(z, \bar{z}), \quad (2.23)$$

where G denotes the blocks computed by `blocks_3d`.

Again, we have a choice to make for the functions $T_{I,ijkl}$. We don't constrain it in any particular way. However, since the functions above differ only by the order of operators, we must have

$$T_{I,kjil}(\mathbf{p}_3, \mathbf{p}_2, \mathbf{p}_1, \mathbf{p}_4) = \sum_J M_I^J T_{J,ijkl}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) \quad (2.24)$$

for some matrix M_I^J (which depends on the choice of i, j, k, l). The crossing equation then takes the form

$$g_{ijkl}^I(z, \bar{z}) \pm \sum_J M_J^I g_{kjil}^J(1-z, 1-\bar{z}) = 0. \quad (2.25)$$

The appearance of $1-z$ is due to our convention for the crossing equation and the choice of cross-ratios.

We can furthermore expand the crossing equations (2.25) in a power series around $z = \bar{z} = \frac{1}{2}$. In general, not all the Taylor coefficients are going to be linearly-independent [14, 79, 80]. We therefore choose some independent set, with a cut-off n_{max} on the order. We then introduce a combined label \mathbf{I} which runs over all crossing equations, all four-point tensor structures I and the independent Taylor coefficients up to the cutoff n_{max} . We write $\hat{g}_{ijkl}^{\mathbf{I}}$ for the contribution of the four point function OPE channel $\langle \mathcal{O}_i \mathcal{O}_j | \mathcal{O}_k \mathcal{O}_l \rangle$, in this specific ordering of indices, to the scalar crossing equation labeled by \mathbf{I} .¹⁵ For example, if \mathbf{I} corresponds to $I = 0$ and the $(z - \frac{1}{2})^m (\bar{z} - \frac{1}{2})^n$ term in the equation (2.25), then we have

$$\hat{g}_{ijkl}^{\mathbf{I}} = \frac{1}{m!n!} \partial_z^m \partial_{\bar{z}}^n g_{ijkl}^0(z, \bar{z}) \Big|_{z=\bar{z}=\frac{1}{2}}, \quad (2.26)$$

$$\hat{g}_{kjil}^{\mathbf{I}} = \frac{(-1)^{m+n}}{m!n!} \sum_J M_J^0 \partial_z^m \partial_{\bar{z}}^n g_{kjil}^J(z, \bar{z}) \Big|_{z=\bar{z}=\frac{1}{2}}, \quad (2.27)$$

¹⁵Note that for some orderings of $ijkl$ it might be that $\hat{g}_{ijkl}^{\mathbf{I}}$ vanishes for all \mathbf{I} due to the choice of the crossing equations. Furthermore, some four-point functions vanish by symmetry and the corresponding $\hat{g}_{ijkl}^{\mathbf{I}}$ are also 0.

with all other $\widehat{g}^{\mathbf{I}}$ vanishing. With this notation, the full set of crossing equations takes the form

$$\forall \mathbf{I} : \sum_{ijkl} \widehat{g}_{ijkl}^{\mathbf{I}} = 0, \quad (2.28)$$

where we sum over all choices of $ijkl$. Ultimately, the coefficients $\widehat{g}_{ijkl}^{\mathbf{I}}$ are provided to the algorithm by the user. In our implementation the user provides the equations (2.25) and the list of Taylor coefficients that need to be considered, from which the code reads off the $\widehat{g}_{ijkl}^{\mathbf{I}}$.

We can extend this notation in the obvious way to G , so that

$$\widehat{g}_{ijkl}^{\mathbf{I}} = \sum'_{\Delta, \rho} \sum_{a,b} l_{ij; \Delta, \rho}^a l_{kl; \Delta, \rho}^b \widehat{G}_{ab, ijkl, \Delta, \rho}^{\mathbf{I}}. \quad (2.29)$$

This leads to the following expansion of the crossing equations,

$$0 = \sum_{ijkl} \sum'_{\Delta, \rho} \sum_{a,b} l_{ij; \Delta, \rho}^a l_{kl; \Delta, \rho}^b \widehat{G}_{ab, ijkl, \Delta, \rho}^{\mathbf{I}}. \quad (2.30)$$

We now consider three types of contributions to (2.30)

Generic contributions We start with the contributions of generic operators $\mathcal{O}_{\Delta, \rho}$. After imposing a cut-off on the spin of $\mathcal{O}_{\Delta, \rho}$ there are finitely many distinct ρ appearing in (2.30). We call such ρ “operator channels.” To each operator channel we associate the set

$$\mathcal{I}_{\rho} = \{(a, (i, j)) | Q_{a, ij; \Delta, \rho} \neq 0\}, \quad (2.31)$$

where (i, j) denotes an unordered pair. In other words, the elements of \mathcal{I}_{ρ} label the OPE coefficients $l_{ij; \Delta, \rho}^a$ which are allowed by symmetries.

We now introduce the PSD matrix $(\mathcal{P}_{\Delta, \rho})^{\alpha\beta}$ with $\alpha, \beta \in \mathcal{I}_{\rho}$

$$(\mathcal{P}_{\Delta, \rho})^{\alpha\beta} = \sum_{\text{degeneracies}} l_{ij; \Delta, \rho}^a l_{kl; \Delta, \rho}^b \succeq 0, \quad (2.32)$$

where $\alpha = (a, (i, j))$, $\beta = (b, (k, l))$, and we sum over all the contributions with given Δ, ρ (accounting for possible degeneracies in the spectrum). Similarly, we introduce

$$(\widehat{\mathcal{G}}_{\Delta, \rho}^{\mathbf{I}})_{\alpha\beta} = \frac{1}{2} \left(\widehat{G}_{ab, ijkl, \Delta, \rho}^{\mathbf{I}} + \widehat{G}_{ba, klij, \Delta, \rho}^{\mathbf{I}} \right) + \text{permutations of } (ij) \text{ and } (kl), \quad (2.33)$$

with the same α, β as above.

With this notation and using (2.11), the contribution of generic operators to the crossing equations is

$$0 = \sum_{\rho} \sum_{\Delta} \text{Tr} \left(\mathcal{P}_{\Delta, \rho} \widehat{\mathcal{G}}_{\Delta, \rho}^{\mathbf{I}} \right) + \dots . \quad (2.34)$$

External contributions We now focus on the contribution to the crossing equations of \mathcal{O}_i themselves. We define the index set

$$\mathcal{E} = \{(a, (i, j, k)) | Q_{a,ijk} \neq 0\}, \quad (2.35)$$

where (i, j, k) denotes unordered triples. That is, \mathcal{E} labels the OPE coefficients l_{ijk}^a which are not forced to be 0 by symmetries. We now define the rank-1 PSD matrix, assuming no degeneracies among the quantum numbers of \mathcal{O}_i ,

$$(\mathcal{P}_{ext})^{\alpha\beta} \equiv l_{ijn}^a l_{klm}^b, \quad (2.36)$$

where $\alpha = (a, (i, j, n))$ and $\beta = (b, (k, l, m))$.

We define the symmetric matrix $\widehat{\mathcal{G}}_{ext}^{\mathbf{I}}$ by the requirement that the contribution of the external operators to the crossing equation (2.30) is (recall (2.11))

$$0 = \text{Tr} \left(\mathcal{P}_{ext} \widehat{\mathcal{G}}_{ext}^{\mathbf{I}} \right) + \dots . \quad (2.37)$$

The matrix $\widehat{\mathcal{G}}_{ext}^{\mathbf{I}}$ has a straightforward expression in terms of $\widehat{G}_{ab,ijkl,\Delta,\rho}^{\mathbf{I}}$ which is however awkward to describe.

Special operators There are sometimes special exchanged operators such as the stress-tensor $T^{\mu\nu}$ or the identity operator $\mathbb{1}$. These operators contribute to the crossing equations as

$$0 = \widehat{\mathcal{G}}^{\mathbf{I}} + \dots , \quad (2.38)$$

where $\widehat{\mathcal{G}}^{\mathbf{I}}$ is determined by the special block for the given operator. It may or may not depend on parameters such as OPE coefficients or some scaling dimensions. For example, for the identity operator the contribution

$$0 = \widehat{\mathcal{G}}_{\mathbb{1}}^{\mathbf{I}} + \dots \quad (2.39)$$

is completely fixed by the normalization of two-point functions of \mathcal{O}_i .

Final crossing equation The final crossing equation then takes the form

$$0 = \widehat{\mathcal{G}}_{\mathbf{1}}^{\mathbf{I}} + \text{Tr} \left(\mathcal{P}_{ext} \widehat{\mathcal{G}}_{ext}^{\mathbf{I}} \right) + \sum_{\rho} \sum_{\Delta} \text{Tr} \left(P_{\Delta, \rho} \widehat{\mathcal{G}}_{\Delta, \rho}^{\mathbf{I}} \right), \quad (2.40)$$

and potentially additional contributions from T or other special operators. Here, $\mathcal{P}_{ext}, \mathcal{P}_{\Delta, \rho} \succeq 0$. So, for example, in a feasibility study we look for functionals $F_{\mathbf{I}}$ such that

$$\sum_{\mathbf{I}} F_{\mathbf{I}} \widehat{\mathcal{G}}_{\mathbf{1}}^{\mathbf{I}} = 1, \quad (2.41)$$

$$\sum_{\mathbf{I}} F_{\mathbf{I}} \widehat{\mathcal{G}}_{ext}^{\mathbf{I}} \succeq 0, \quad (2.42)$$

$$\sum_{\mathbf{I}} F_{\mathbf{I}} \widehat{\mathcal{G}}_{\Delta, \rho}^{\mathbf{I}} \succeq 0. \quad (2.43)$$

Crucially, these conditions can be formed automatically once the user provides the three-point structures Q and the crossing equations in the form (2.25).

2.3.2 Three-point functions

Returning to the bootstrap problem for the GNY model, each local primary operator is characterized by the scaling dimension and three other quantum numbers, namely, spin j , space parity P , and an $O(N)$ irreducible representation μ .¹⁶ We consider the mixed system of the lowest dimension operators ψ , ϵ and σ , where their quantum numbers (j, P, μ) are

$$\sigma : (0, \text{odd}, \bullet) \quad (2.44)$$

$$\epsilon : (0, \text{even}, \bullet) \quad (2.45)$$

$$\psi : \left(\frac{1}{2}, \text{even}, \square\right). \quad (2.46)$$

When $N = 1$, the global symmetry becomes $O(1) = \mathbb{Z}_2$. Since the non-trivial element sends $\psi \rightarrow -\psi$ and $\phi \rightarrow \phi$, it coincides with $(-1)^F$ and should not be considered as a separate global symmetry. Hence, in this case the operators are labeled by (j, P) only.¹⁷ Consequently, there are no flavor structures to consider. There is a corresponding reduction in the number of equations and OPE channels.

¹⁶Note that the parity makes sense both for integer and half-integer j (we use the transformation defined in [14]), although for the latter the notions of “even” and “odd” parities can be exchanged by redefinition of the parity transformation by $(-1)^F$. This allows us to choose the parity for one fermionic operator at will, so we choose ψ to be parity-even.

¹⁷In our discussion, which is valid for generic N , one can take $\bullet \rightarrow \bullet$ and $\square \rightarrow \bullet$ while all other representations are omitted.

The crossing equations under study involve the following set of four point functions

$$\{\langle\psi\psi\psi\psi\rangle, \langle\epsilon\epsilon\epsilon\epsilon\rangle, \langle\sigma\sigma\sigma\sigma\rangle, \langle\psi\psi\epsilon\epsilon\rangle, \langle\psi\psi\sigma\sigma\rangle, \langle\sigma\epsilon\psi\psi\rangle, \langle\sigma\sigma\epsilon\epsilon\rangle\}. \quad (2.47)$$

The conformal block expansions of these correlation functions involve OPEs between all combinations of ψ, ϵ, σ .

We list the tensor structures appearing in relevant three-point functions in Table 2.3 and Table 2.4. We choose to represent the conformal structures in the $\text{SO}(3)_r$ basis following the convention in [15, 18],¹⁸ and we define the $O(N)$ flavor structures as follows,

$$T_{\bullet\bullet\bullet} = 1, \quad (2.48)$$

$$T_{\square}^{i,j} = \delta^{ij}, \quad (2.49)$$

$$T_{\square\square}^{ij,(kl)} = \frac{1}{2} (\delta^{ik}\delta^{jl} + \delta^{il}\delta^{jk}) - \frac{1}{N}\delta^{ij}\delta^{kl}, \quad (2.50)$$

$$T_{\square}^{ij,[kl]} = \frac{1}{2} (\delta^{ik}\delta^{jl} - \delta^{il}\delta^{jk}). \quad (2.51)$$

Some conformal structures might disappear for low spin of the exchanged operator \mathcal{O} . The selection rule is that $|j_{12}, j_{123}\rangle$ is present in $\langle\mathcal{O}_1\mathcal{O}_2\mathcal{O}\rangle$ if $j_{12} \in j_1 \otimes j_2$ and $j_{123} \in j_{12} \otimes l$, where j_1 and j_2 are the spins of \mathcal{O}_1 and \mathcal{O}_2 . In practice, to obtain the conformal structures, we did calculations in the q -basis [14] and then converted them to the $\text{SO}(3)_r$ basis as discussed in [18].

Tensor structures for three external operators are obtained from those in tables 2.3 and 2.4 by restricting \mathcal{O} to the relevant special case. The only exception is the structure for $\langle\psi\psi\sigma\rangle$ which we take to be $-|1, 1\rangle$ (differs by a factor of -1 from table 2.4).

Our choice of the tensor structure basis ensures the equality of the following OPE coefficients, as required by the algorithm in section 2.3.1:

$$\lambda_{\psi\psi\sigma} = \lambda_{\psi\sigma\psi} = \lambda_{\sigma\psi\psi}, \quad (2.52)$$

$$\lambda_{\psi\psi\epsilon} = \lambda_{\psi\epsilon\psi} = \lambda_{\epsilon\psi\psi}, \quad (2.53)$$

$$\lambda_{\sigma\sigma\epsilon} = \lambda_{\sigma\epsilon\sigma} = \lambda_{\epsilon\sigma\sigma}, \quad (2.54)$$

and similarly $l_{\psi\sigma\mathcal{O}} = l_{\sigma\psi\mathcal{O}}$ etc. A slight difference from 2.3.1 is that the OPE coefficients for our structures are purely imaginary if they involve fermionic operators. This difference is accounted for in the software.

OPE	$\mathcal{O} \in (l, P, \mu)$	$\langle \mathcal{O}_a \mathcal{O}_b \mathcal{O}_c \rangle$	Structures
$\sigma \times \sigma$ $\epsilon \times \epsilon$	$(l \in 2\mathbb{Z}, \text{even}, \bullet)$	$\langle \sigma \sigma \mathcal{O} \rangle$ $\langle \epsilon \epsilon \mathcal{O} \rangle$	$ 0, l\rangle$
$\sigma \times \epsilon$	$(l \in 2\mathbb{Z}, \text{odd}, \bullet)$	$\langle \sigma \epsilon \mathcal{O} \rangle$ $\langle \epsilon \sigma \mathcal{O} \rangle$	$ 0, l\rangle$ $(-1)^l 0, l\rangle$
$\sigma \times \psi$	$(l \in \mathbb{Z} + \frac{1}{2}, \text{even}, \square)$	$\langle \sigma \psi^i \mathcal{O}^j \rangle$ $\langle \psi^i \sigma \mathcal{O}^j \rangle$	$(-1)^{l+\frac{1}{2}} \delta^{ij} \frac{1}{2}, l + \frac{1}{2}\rangle$ $\delta^{ij} \frac{1}{2}, l - \frac{1}{2}\rangle$
	$(l \in \mathbb{Z} + \frac{1}{2}, \text{odd}, \square)$	$\langle \sigma \psi^i \mathcal{O}^j \rangle$ $\langle \psi^i \sigma \mathcal{O}^j \rangle$	$(-1)^{l-\frac{1}{2}} \delta^{ij} \frac{1}{2}, l - \frac{1}{2}\rangle$ $\delta^{ij} \frac{1}{2}, l + \frac{1}{2}\rangle$
$\epsilon \times \psi$	$(l \in \mathbb{Z} + \frac{1}{2}, \text{even}, \square)$	$\langle \epsilon \psi^i \mathcal{O}^j \rangle$ $\langle \psi^i \epsilon \mathcal{O}^j \rangle$	$(-1)^{l-\frac{1}{2}} \delta^{ij} \frac{1}{2}, l - \frac{1}{2}\rangle$ $\delta^{ij} \frac{1}{2}, l + \frac{1}{2}\rangle$
	$(l \in 2\mathbb{Z} + \frac{1}{2}, \text{odd}, \square)$	$\langle \epsilon \psi^i \mathcal{O}^j \rangle$ $\langle \psi^i \epsilon \mathcal{O}^j \rangle$	$(-1)^{l+\frac{1}{2}} \delta^{ij} \frac{1}{2}, l + \frac{1}{2}\rangle$ $\delta^{ij} \frac{1}{2}, l - \frac{1}{2}\rangle$

Table 2.3: Tensor structures appearing in the OPEs of mixed scalar-fermion operators.

OPE	$\mathcal{O} \in (l, P, \mu)$	$\langle \psi^i \psi^j \mathcal{O}^a \rangle$ Structures
$\psi \times \psi$	$(l \in 2\mathbb{Z}, \text{even}, \mu \in \{\bullet, \square\square\})$	$T_\mu^{ija} 0, l\rangle$ $T_\mu^{ija} 1, l\rangle$
	$(l \in 2\mathbb{Z} + 1, \text{even}, \mu = \square)$	$T_\mu^{ija} 0, l\rangle$ $T_\mu^{ija} 1, l\rangle$
	$(l \in 2\mathbb{Z}, \text{odd}, \mu \in \{\bullet, \square\square\})$	$T_\mu^{ija} (\sqrt{l+1} 1, l+1\rangle - \sqrt{l} 1, l-1\rangle)$
	$(l \in 2\mathbb{Z} + 1, \text{odd}, \mu \in \{\bullet, \square\square\})$	$T_\mu^{ija} (\sqrt{l+1} 1, l-1\rangle + \sqrt{l} 1, l+1\rangle)$
	$(l \in (2\mathbb{Z})_{\geq 2}, \text{odd}, \mu = \square)$	$T_\mu^{ija} (\sqrt{l+1} 1, l-1\rangle + \sqrt{l} 1, l+1\rangle)$
	$(l \in 2\mathbb{Z} + 1, \text{odd}, \mu = \square)$	$T_\mu^{ija} (\sqrt{l+1} 1, l+1\rangle - \sqrt{l} 1, l-1\rangle)$

Table 2.4: Tensor structures appearing in the OPE of the fermionic operators.

The stress-tensor T and the conserved $O(N)$ current J transform with (l, P, μ) equal to $(2, \text{even}, \bullet)$ and $(1, \text{even}, \square)$. Their OPE coefficients are constrained

¹⁸In [18] it was called the $SO(3)$ basis.

by Ward identities as

$$f_{\sigma\sigma\hat{T}}^1 = -\sqrt{\frac{3}{2}} \frac{\Delta_\sigma}{4\pi\sqrt{C_T}}, \quad f_{\epsilon\epsilon\hat{T}}^1 = -\sqrt{\frac{3}{2}} \frac{\Delta_\epsilon}{4\pi\sqrt{C_T}}, \quad (2.55)$$

$$f_{\psi\psi\hat{T}}^1 = i \frac{\sqrt{3}}{4\pi} \frac{\Delta_\psi}{\sqrt{C_T}}, \quad f_{\psi\psi\hat{T}}^2 = -i \frac{3}{4\sqrt{2}\pi} \frac{1}{\sqrt{C_T}}, \quad (2.56)$$

$$f_{\psi\psi\hat{J}}^1 = i \frac{1}{\sqrt{2}\pi} \frac{1}{\sqrt{C_J}}, \quad f_{\psi\psi\hat{J}}^2 \text{ not constrained}, \quad (2.57)$$

where $\hat{T} = C_T^{-\frac{1}{2}} T$ and $\hat{J} = C_J^{-\frac{1}{2}} J$ are canonically normalized in the conventions of [15, 18], which are

$$\langle \hat{T}\hat{T} \rangle = \frac{H_{12}^2}{X_{12}^5}, \quad \langle \hat{J}^{[ij]} \hat{J}^{[kl]} \rangle = \frac{1}{2} (\delta^{ik} \delta^{jl} - \delta^{il} \delta^{jk}) \frac{H_{12}}{X_{12}^3}. \quad (2.58)$$

In practice, the Ward identity for J does not affect the numerics because it only gives an interpretation of an OPE coefficient in terms of C_J . The Ward identity for T doesn't affect the numerics unless a gap above T is assumed. As explained in section 2.2.3, we do assume a gap of 10^{-6} above T . However, since this gap is small, it is likely that imposing the T Ward identity doesn't have a significant effect on our results.

2.3.3 Four-point functions and crossing equations

We now study the four-point functions of our mixed system. Following the procedures outlined by (2.18), (2.20), and (2.21), we construct the four-point tensor structures $T_{I,ijkl}$ and find the crossing equations in the form of (2.25).

In general, one single four-point structure is the product of a flavor and a conformal structure. The full structure should be invariant under kinematic permutations, which is the group that preserves all conformal cross-ratios formed by the coordinates of the four operators. We refer to [14] for the details on this group.

When the four operators are identical, $\rho_1 = \rho_2 = \rho_3 = \rho_4$ and the kinematic permutation group is $\mathbb{Z}_2 \times \mathbb{Z}_2 = \{e, (12)(34), (13)(24), (14)(23)\}$. The irreducible representations of $\mathbb{Z}_2 \times \mathbb{Z}_2$ can be labelled by $(++)$, $(--)$, $(+-)$, $(-+)$ where the signs stand for the eigenvalues under $(12)(34)$ and $(13)(24)$ respectively. On the other hand, for four-point functions with two pairs of identical operators the kinematic permutation group is \mathbb{Z}_2 . For example, if $\rho_1 = \rho_2$ and $\rho_3 = \rho_4$ then the kinematic permutation group is just $\mathbb{Z}_2 = \{e, (12)(34)\}$. The irreducible representations of \mathbb{Z}_2 can be labelled simply by $+$ and $-$.

To ensure that the full structure is invariant under the kinematic permutations, the flavor and conformal structures should transform in the same irrep of the kinematic permutation group.

There are two additional requirements for the four-point tensor structures. Firstly, the structures must have space parity consistent with the parity of the operators in the four-point function. And secondly, they must have definite parity under the transformation $z \leftrightarrow \bar{z}$, which we refer to as the t-parity transformation.¹⁹ This is needed to simplify the corresponding symmetry of the coefficient functions $g(z, \bar{z})$, see [14].

2.3.3.1 $\langle \psi\psi\psi\psi \rangle$

We first consider the four point function of four fermions

$$\langle \psi^i \psi^j \psi^k \psi^l \rangle = \sum_{I,a} t_I T_a^{ijkl} g_{\psi\psi\psi\psi}^{I,a}(z, \bar{z}), \quad (2.59)$$

where T_a stands for the flavor structures and t_I for the conformal structures. It participates in the crossing equations of the form

$$\langle \psi\psi|\psi\psi \rangle = -\langle \psi\psi|\psi\psi \rangle, \quad (2.60)$$

understood in the sense of equation (2.18).

There are three possible flavor structures for $\langle \psi^i \psi^j \psi^k \psi^l \rangle$, namely $\delta^{ij} \delta^{kl}$, $\delta^{ik} \delta^{jl}$, and $\delta^{il} \delta^{jk}$. They all are invariant under the kinematic permutation group $\mathbb{Z}_2 \times \mathbb{Z}_2$, i.e. are all in the $(++)$ representation. For future convenience, we define the following linear combinations of these structures,

$$T_+^{ijkl} = \delta^{ij} \delta^{kl} + \delta^{il} \delta^{jk}, \quad T_3^{ijkl} = \delta^{ik} \delta^{jl}, \quad T_-^{ijkl} = \delta^{ij} \delta^{kl} - \delta^{il} \delta^{jk}. \quad (2.61)$$

We also notice that under crossing symmetry permutation (13), T_+ and T_3 are symmetric while T_- is anti-symmetric.

In order to construct the conformal structures, we use the q -basis defined in [14]. We look for conformal structures that are $\mathbb{Z}_2 \times \mathbb{Z}_2$ invariant, parity-

¹⁹Not to be confused with time reversal.

even, and have definite parity under the t-parity transformation:²⁰

$$\langle \uparrow\uparrow\uparrow\uparrow \rangle^\pm = \langle \uparrow\uparrow\uparrow\uparrow \rangle \pm \langle \downarrow\downarrow\downarrow\downarrow \rangle, \quad (2.62)$$

$$\langle \uparrow\uparrow\downarrow\downarrow \rangle^+ = \langle \uparrow\uparrow\downarrow\downarrow \rangle + \langle \downarrow\downarrow\uparrow\uparrow \rangle, \quad (2.63)$$

$$\langle \uparrow\downarrow\uparrow\downarrow \rangle^+ = \langle \uparrow\downarrow\uparrow\downarrow \rangle + \langle \downarrow\uparrow\downarrow\uparrow \rangle, \quad (2.64)$$

$$\langle \downarrow\uparrow\uparrow\downarrow \rangle^+ = \langle \downarrow\uparrow\uparrow\downarrow \rangle + \langle \uparrow\downarrow\downarrow\uparrow \rangle. \quad (2.65)$$

Here \pm denotes the t-parity, which acts on the individual fermionic spins by $[\uparrow] \rightarrow i[\downarrow]$ and $[\downarrow] \rightarrow i[\uparrow]$. We use \uparrow to denote $+\frac{1}{2}$ and \downarrow to denote $-\frac{1}{2}$.

From the results of [14] we can determine the phase factor picked up by the conformal structures under crossing symmetry²¹

$$(13) : \langle q_1 q_2 q_3 q_4 \rangle \rightarrow (-1)^{q_1+q_2+q_3-q_4} \langle q_3 q_2 q_1 q_4 \rangle. \quad (2.66)$$

Multiplied by the extra factor of (-1) coming from the fermion exchange, we find that all the conformal structures are invariant under the (13) permutation.

Combining conformal and flavor structures, we form the following two sets of linear combinations that are crossing-symmetric and crossing-antisymmetric respectively,

$$\text{symmetric: } \mathcal{G}_{\langle \uparrow\uparrow\uparrow\uparrow \rangle^\pm, T_+}, \quad \mathcal{G}_{\langle \uparrow\uparrow\downarrow\downarrow \rangle^+, T_+} + \mathcal{G}_{\langle \downarrow\uparrow\uparrow\downarrow \rangle^+, T_+}, \quad \mathcal{G}_{\langle \uparrow\downarrow\uparrow\downarrow \rangle^+, T_+}, \quad (2.67)$$

$$\mathcal{G}_{\langle \uparrow\uparrow\uparrow\uparrow \rangle^\pm, T_3}, \quad \mathcal{G}_{\langle \uparrow\uparrow\downarrow\downarrow \rangle^+, T_3} + \mathcal{G}_{\langle \downarrow\uparrow\uparrow\downarrow \rangle^+, T_3}, \quad \mathcal{G}_{\langle \uparrow\downarrow\uparrow\downarrow \rangle^+, T_3}, \quad (2.68)$$

$$\mathcal{G}_{\langle \uparrow\uparrow\downarrow\downarrow \rangle^+, T_-} - \mathcal{G}_{\langle \downarrow\uparrow\uparrow\downarrow \rangle^+, T_-}, \quad (2.69)$$

$$\text{anti-symmetric: } \mathcal{G}_{\langle \uparrow\uparrow\downarrow\downarrow \rangle^+, T_+} - \mathcal{G}_{\langle \downarrow\uparrow\uparrow\downarrow \rangle^+, T_+}, \quad (2.70)$$

$$\mathcal{G}_{\langle \uparrow\uparrow\downarrow\downarrow \rangle^+, T_3} - \mathcal{G}_{\langle \downarrow\uparrow\uparrow\downarrow \rangle^+, T_3}, \quad (2.71)$$

$$\mathcal{G}_{\langle \uparrow\uparrow\uparrow\uparrow \rangle^\pm, T_-}, \quad \mathcal{G}_{\langle \uparrow\uparrow\downarrow\downarrow \rangle^+, T_-} + \mathcal{G}_{\langle \downarrow\uparrow\uparrow\downarrow \rangle^+, T_-}, \quad \mathcal{G}_{\langle \uparrow\downarrow\uparrow\downarrow \rangle^+, T_-}, \quad (2.72)$$

where we used the simplified notation

$$g_{I,a} \equiv g_{\psi\psi\psi\psi}^{I,a}. \quad (2.73)$$

The above functions satisfy crossing equations with the sign determined by whether they fall in the ‘‘symmetric’’ or ‘‘anti-symmetric’’ category above.

²⁰This example has been worked out in detail in [14] and [18].

²¹Equation (4.44) in [14] assumes $\rho_1 = \rho_3$ and includes an extra (-1) for fermionic permutations. The general result (2.66) can be derived from appendix B of [14].

For example, we have

$$g_{\langle \uparrow \uparrow \uparrow \rangle -, T_+}(z, \bar{z}) = g_{\langle \uparrow \uparrow \uparrow \rangle -, T_+}(1 - z, 1 - \bar{z}), \quad (2.74)$$

while

$$\begin{aligned} & g_{\langle \uparrow \uparrow \downarrow \downarrow \rangle +, T_+}(z, \bar{z}) - g_{\langle \downarrow \uparrow \uparrow \downarrow \rangle +, T_+}(z, \bar{z}) \\ &= - \left(g_{\langle \uparrow \uparrow \downarrow \downarrow \rangle +, T_+}(1 - z, 1 - \bar{z}) - g_{\langle \downarrow \uparrow \uparrow \downarrow \rangle +, T_+}(1 - z, 1 - \bar{z}) \right). \end{aligned} \quad (2.75)$$

When Taylor-expanding these equations, one should keep in mind the t-parity of the conformal structures which determines the parity of the above functions under $z \leftrightarrow \bar{z}$.

Finally, the crossing equations for $g_{\langle \uparrow \uparrow \uparrow \rangle +, T_a}(z, \bar{z})$ at $z = \bar{z}$ are redundant with other crossing equations and should not be imposed in order to avoid numerical instabilities (see appendix A of [14]). In practice this is done by requiring that $n > 0$ derivatives are taken in the direction orthogonal to $z = \bar{z}$ for this structure.

2.3.3.2 $\langle \psi \psi \sigma \sigma \rangle$ and $\langle \psi \psi \epsilon \epsilon \rangle$

Mixed four-point functions containing both σ and ψ give rise to two crossing channels in the sense of (2.18),

$$\langle \sigma \sigma | \psi \psi \rangle = \langle \psi \sigma | \sigma \psi \rangle, \quad (2.76)$$

$$\langle \psi \sigma | \psi \sigma \rangle = - \langle \psi \sigma | \psi \sigma \rangle. \quad (2.77)$$

We expand each ordering of the four point function as

$$\begin{aligned} \langle \sigma \sigma \psi^i \psi^j \rangle &= \sum_{I,a} t_I T_a^{ij} g_{\sigma \sigma \psi \psi}^{I,a}(z, \bar{z}), \\ \langle \psi^i \sigma \sigma \psi^j \rangle &= \sum_{I,a} t_I T_a^{ij} g_{\psi \sigma \sigma \psi}^{I,a}(z, \bar{z}), \\ \langle \psi^i \sigma \psi^j \sigma \rangle &= \sum_{I,a} t_I T_a^{ij} g_{\psi \sigma \psi \sigma}^{I,a}(z, \bar{z}). \end{aligned} \quad (2.78)$$

We are slightly abusing the notation since the conformal structures are different for each of the three orderings (since different operators are inserted at different points x_i).

Only one flavor structure $T_a^{ij} = \delta^{ij}$ exists for each of these four-point functions, and it is kinematic-symmetric. It is also mapped to itself under the crossing

permutation (13). Hence, we will ignore the flavor structure in the following discussion. Furthermore, since the products $\sigma\sigma$ and $\epsilon\epsilon$ have the same parity, the analysis below is the same for the correlation functions $\langle\psi\psi\sigma\sigma\rangle$ and $\langle\psi\psi\epsilon\epsilon\rangle$. We will focus on the correlator $\langle\psi\psi\sigma\sigma\rangle$ for concreteness.

The two parity-even conformal structures for the ordering $\langle\sigma\sigma\psi\psi\rangle$ are $[0, 0, \frac{1}{2}, \frac{1}{2}]$ and $[0, 0, -\frac{1}{2}, -\frac{1}{2}]$, and similar expressions apply for the other two orderings. These structures are automatically kinematically symmetric. We will simplify the notation and write them as $[\uparrow\uparrow]$ and $[\downarrow\downarrow]$ for each of the orderings (i.e. omitting the 0 charges), and define the structures with definite t-parity as

$$[\uparrow\uparrow]^\pm = [\uparrow\uparrow] \mp [\downarrow\downarrow]. \quad (2.79)$$

Applying (2.66) and factors from fermion permutations, we find that conformal structures with the same label are mapped into each other under the (13) permutation. Hence, we can form crossing-symmetric and anti-symmetric combinations as

$$\text{symmetric: } g_{[\uparrow\uparrow]^\pm}^{\sigma\sigma\psi\psi} + g_{[\uparrow\uparrow]^\pm}^{\psi\sigma\sigma\psi}, \quad g_{[\uparrow\uparrow]^\pm}^{\psi\sigma\psi\sigma}, \quad (2.80)$$

$$\text{antisymmetric: } g_{[\uparrow\uparrow]^\pm}^{\sigma\sigma\psi\psi} - g_{[\uparrow\uparrow]^\pm}^{\psi\sigma\sigma\psi}. \quad (2.81)$$

The convention for the crossing equations is the same as in the previous subsection. This time, there are no redundancies between the crossing equations.

2.3.3.3 $\langle\sigma\epsilon\psi\psi\rangle$

The independent crossing equations in this case are

$$\langle\sigma\epsilon|\psi\psi\rangle = \langle\psi\epsilon|\sigma\psi\rangle, \quad (2.82)$$

$$\langle\psi\sigma|\psi\epsilon\rangle = -\langle\psi\sigma|\psi\epsilon\rangle. \quad (2.83)$$

Similar to the case above, we have a trivial flavor structure that can be ignored. However, the overall parity $\langle\sigma\epsilon\psi\psi\rangle$ is now odd and so a separate analysis of the conformal structures is required.

For all of the orderings of the operators, the parity-odd conformal structures are $[\uparrow\downarrow]$ and $[\downarrow\uparrow]$, using the same notation as in the previous subsection. There is no kinematic symmetry to consider. To form structures with definite t-parity, we write

$$[\uparrow\downarrow]^\pm = [\uparrow\downarrow] \mp [\downarrow\uparrow] = \mp[\downarrow\uparrow]^\pm. \quad (2.84)$$

Applying (2.66) and taking into account the factors from fermion permutation, we can form the crossing-symmetric and crossing-antisymmetric linear combinations as before,

$$\text{symmetric: } g_{[\uparrow\downarrow]^\pm}^{\sigma\epsilon\psi\psi} - g_{[\uparrow\downarrow]^\pm}^{\psi\epsilon\sigma\psi}, \quad g_{[\uparrow\downarrow]^+}^{\psi\sigma\psi\epsilon}, \quad (2.85)$$

$$\text{antisymmetric: } g_{[\uparrow\downarrow]^\pm}^{\sigma\epsilon\psi\psi} + g_{[\uparrow\downarrow]^\pm}^{\psi\epsilon\sigma\psi}, \quad g_{[\uparrow\downarrow]^-}^{\psi\sigma\psi\epsilon}. \quad (2.86)$$

A slight subtlety in this case is that the structure $[\uparrow\downarrow]^\pm$ for $\langle\psi\sigma\psi\epsilon\rangle$ ordering is mapped by (13), up to a phase, to $[\downarrow\uparrow]^\pm$, and so we need to use (2.84) to reduce it back to the $[\uparrow\downarrow]^\pm$ basis.

2.3.3.4 Scalar four-point functions

Since both flavor and conformal structures are trivial, it is a straightforward exercise to form crossing-symmetric and crossing-antisymmetric functions,

$$\text{symmetric: } g^{\sigma\sigma\sigma\sigma}, g^{\epsilon\epsilon\epsilon\epsilon}, g^{\sigma\epsilon\sigma\epsilon}, g^{\sigma\sigma\epsilon\epsilon} + g^{\epsilon\sigma\sigma\epsilon}, \quad (2.87)$$

$$\text{antisymmetric: } g^{\sigma\sigma\epsilon\epsilon} - g^{\epsilon\sigma\sigma\epsilon}. \quad (2.88)$$

The t-parity of all these functions is +1.

In total, we end up with 38 crossing equations for $N \geq 2$ and 28 equations for $N = 1$.

2.3.4 Numerical computations

After setting up the crossing equations and OPE channels, our workflow is automated by softwares described in appendix 2.6.1 to search for the functionals $F_{\mathbf{I}}$ to satisfy (2.41–2.43).

The space of CFT data that we searched over is 6-dimensional, but is better understood as two separate searches done together. We have three scaling dimensions and three OPE coefficient ratios that we search over; the latter is done by a cutting-surface search algorithm, and the former is done by a Delaunay mesh search algorithm. We will now briefly go over these search algorithms.

2.3.4.1 OPE scan

As noted in [6], by including assumptions of OPE coefficient ratios in our bootstrap computations, the allowed region in the space of scaling dimensions

can be improved at the cost of also having to search over those OPE coefficient ratios. To that end, we employ the algorithm described in [10] to include constraints imposed by the OPE coefficients involving the only the external operators, λ_{ext} . The non-vanishing OPE coefficients are

$$\vec{\lambda}_{\text{ext}} = \begin{pmatrix} \lambda_{\psi\psi\sigma} \\ \lambda_{\psi\psi\epsilon} \\ \lambda_{\sigma\sigma\epsilon} \\ \lambda_{\epsilon\epsilon\epsilon} \end{pmatrix}. \quad (2.89)$$

With (2.42), we require that the contribution of external scalar OPE coefficients to the crossing equation has a definite sign after applying the functional, independent of the values of those coefficients.

$$\sum_{\mathbf{I}} F_{\mathbf{I}} \widehat{\mathcal{G}}_{\text{ext}}^{\mathbf{I}} \succeq 0 \quad \Rightarrow \quad \sum_{\mathbf{I}} \text{Tr} \left(M_{\text{ext}} F_{\mathbf{I}} \widehat{\mathcal{G}}_{\text{ext}}^{\mathbf{I}} \right) \geq 0. \quad (2.90)$$

However, (2.42) is strong enough that the conclusion above stands for any matrix with the decomposition $M_{\text{ext}} = A^\dagger A$. We, on the other hand, are only interested in the case $M_{\text{ext}} = \mathcal{P}_{\text{ext}} = \vec{\lambda}_{\text{ext}} \vec{\lambda}_{\text{ext}}^T$, which is a rank-1 matrix. Therefore, we instead look for functionals $F_{\mathbf{I}}$'s that satisfy

$$\vec{\lambda}_{\text{ext}}^T \left(\sum_{\mathbf{I}} F_{\mathbf{I}} \widehat{\mathcal{G}}_{\text{ext}}^{\mathbf{I}} \right) \vec{\lambda}_{\text{ext}} \geq 0 \quad (2.91)$$

for each $[\lambda_{\text{ext}}] \in \mathbb{RP}^3$ (independent of the magnitude or sign of the vector), along with (2.41) and (2.43). A point in the dimensional space $(\Delta_\psi, \Delta_\sigma, \Delta_\epsilon)$ is ruled out if such $F_{\mathbf{I}}$'s exist for all $[\lambda_{\text{ext}}]$, and is allowed otherwise. Hence, by scanning over the OPE space, we obtain a union of allowed regions in dimension space, each permitted by some OPE direction:

$$\bigcup_{[\lambda_{\text{ext}}] \in \mathbb{RP}^3} \mathcal{D}_{\lambda_{\text{ext}}}. \quad (2.92)$$

Note that this union of allowed regions is contained in the allowed region obtained by imposing the stronger (2.42), as this is the special case when imposing the condition that M_{ext} is rank-1. We should note that for more than one OPE coefficient ratio, the cutting surface algorithm is non-rigorous [10].

For each point in the dimension that the cutting surface algorithm doesn't exclude, it outputs a direction in the OPE space that could not be excluded

by our constraints. Combined over all the allowed points in the dimension space, this produces a list of OPE coefficient ratios with a relatively small variance. We report the full range of values of the OPE coefficient ratios that we found in our searches in section 2.4 as estimates of the real ratios. It should be noted however that these estimates are not rigorous. In particular, we cannot exclude a systematic error that could arise from the way the cutting surface algorithm samples the OPE ratios. Similarly, the error bars are not rigorous.

2.3.4.2 Delaunay mesh search

The islands that we show were computed by the Delaunay mesh search algorithm; we will refer for details on the algorithm to the original paper [10]. However, in order to properly interpret how we have chosen to represent our results, we will provide a brief qualitative review of this method.

The principle of the algorithm is to divide the search space into a suitable simplicial complex, known as a Delaunay mesh, where each vertex is a point that has already been computed as being allowed or disallowed. A simplex that has entirely allowed or disallowed vertices is assumed to be completely on the interior or exterior of our island. If a simplex has both an allowed and disallowed vertex, then the simplex is deemed to be on the boundary of the island; we can think of the volume of the simplex as the region of uncertainty between allowed and disallowed. This implied boundary can be further refined by computing the feasibility of the point at the simplex's centroid.²² Thus, with each iteration we get an increasingly refined mesh.

After a search has finished, we can determine the island's boundary in one of two ways. The first is to take the centroids of the boundary simplices and compute their convex hull. This method produces fairly smooth islands, but whose bounds are not strictly rigorous. The second, more conservative, approach is to take the convex hull of all boundary simplices, which is equivalent to taking the convex hull of all vertices that neighbor allowed vertices. This includes the interiors of all boundary simplices into our island so we can be sure that the area outside this hull is strictly disallowed by our bootstrap con-

²²The Delaunay search algorithm selects the centroid, which is also the mean of the vertices, as the next point to be computed in the numerical bootstrap [10]. Interpreting the simplex as the uncertain region of the boundary, the centroid is the mean of that uncertain region.

straints, up to assumptions of convexity. Thus, the latter method produces islands and bounds that are rigorous. However, it should be noted that the latter method's islands tend to be more jagged than islands computed with the centroid method given the same set of points. This distinction is heightened when working with relatively small numbers of points computed. In this work, we have opted to take the more conservative approach.

2.4 Results

In this section, we present the results of our numerical bootstrap computations for various values of N . While the space of CFT data that we searched over is 6-dimensional (as discussed in section 2.3), in most cases, we have projected the results into the planes $(\Delta_\psi, \Delta_\sigma)$ and $(\Delta_\sigma, \Delta_\epsilon)$ for ease of visualization.

We first discuss the bounds obtained for $N = 2, 4$, and 8 and compare them with the existing studies from ϵ -expansions (after Borel resummation) and from Monte Carlo simulations [41, 49, 50, 51]. We then focus on the $N = 1$ theory and discuss how our results, without assuming supersymmetry a priori, are strong evidence for the emergence of supersymmetry in the IR in the $N = 1$ critical GNY model.

2.4.1 $N = 2, 4$, and 8

For $N = 2$ at $n_{\max} = 18$ we can report rigorous estimates of our external scaling dimensions of $\Delta_\psi = 1.06861(\mathbf{12})$, $\Delta_\sigma = 0.6500(\mathbf{12})$, $\Delta_\epsilon = 1.725(\mathbf{7})$, given the assumptions as discussed in section 2.2.3 of $\Delta_{\epsilon'} > 3$, $\Delta_{\sigma'} > 3$, $\Delta_{\sigma_T} > 2$, $\Delta_{\psi'} > 2$, $\Delta_\chi > 3.5$. Figure 2.5 shows the allowed regions²³ after projections to the $(\Delta_\sigma, \Delta_\epsilon)$ and $(\Delta_\psi, \Delta_\sigma)$ planes. As shown in the $(\Delta_\sigma, \Delta_\epsilon)$ -plane, the bootstrap results exclude the reported error bars from earlier Monte Carlo studies from [49]; as can be seen in both plots, the bootstrap results also marginally exclude the reported error bars from the ϵ -expansion results after Borel resummation [41]. For both studies, the bootstrap results improve the precision of some of these estimates by orders of magnitude.

²³Readers familiar with bootstrap results may be concerned as to the jagged nature of the islands reported. The $n_{\max} = 18$ computations are very computationally expensive, so we only have a relatively small number of allowed points in each island. However, because we have a lot of information as to the disallowed points from lower n_{\max} , and because the precision of these islands are still improved, we have elected to report the superficially more jagged islands as they represent our best results. We have taken pains to ensure that these results are rigorous, as outlined in section 2.3.4.2.

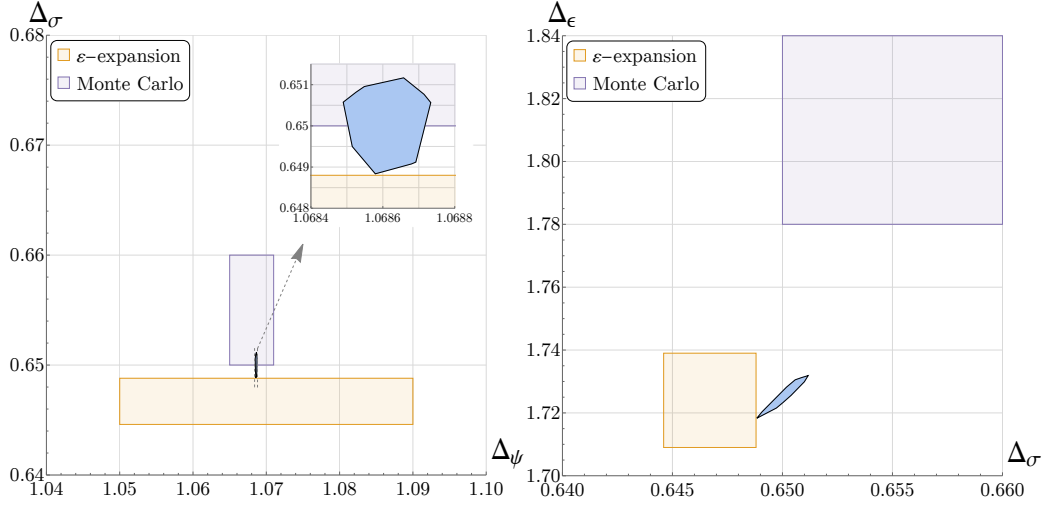


Figure 2.5: The allowed region for the $N = 2$ critical GNY model when imposing that $\Delta_{\sigma'} > 3$ computed at $n_{\max} = 18$, projected to the $(\Delta_\sigma, \Delta_\epsilon)$ and $(\Delta_\psi, \Delta_\sigma)$ planes. This should be compared to the Borel re-summed result obtained from the ϵ -expansion [41] (shown in orange) and with the Monte Carlo results from [49] (shown in purple).

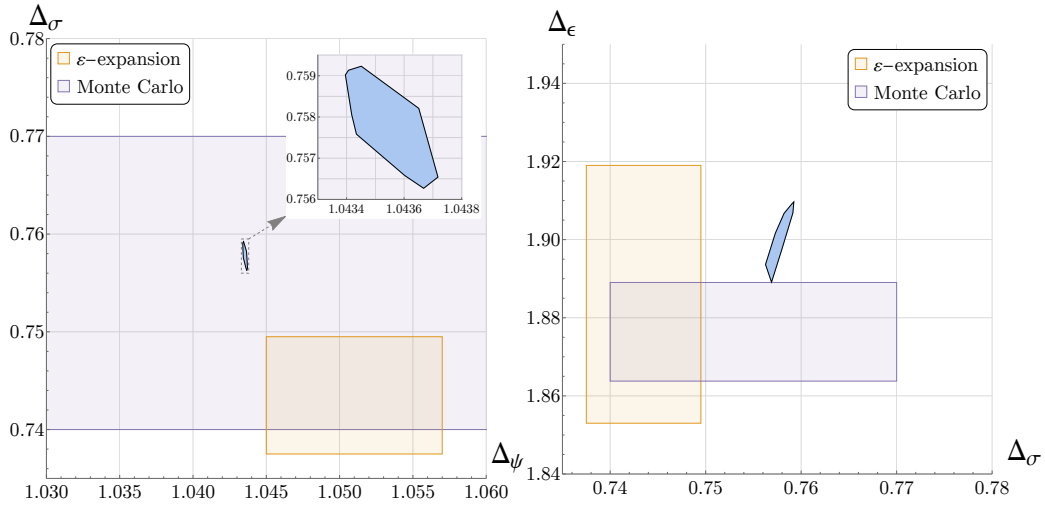


Figure 2.6: The allowed region for the $N = 4$ critical GNY model when imposing that $\Delta_{\sigma'} > 3$ computed at $n_{\max} = 18$, projected to the $(\Delta_\psi, \Delta_\sigma)$ and $(\Delta_\sigma, \Delta_\epsilon)$ planes. This should be compared to the Borel re-summed result obtained from the ϵ -expansion [41] obtained using the DREG₃ regularization scheme (shown in orange) and with the Monte Carlo results from [50] (shown in purple). Results for Δ_ψ are not available from the Monte Carlo study cited.

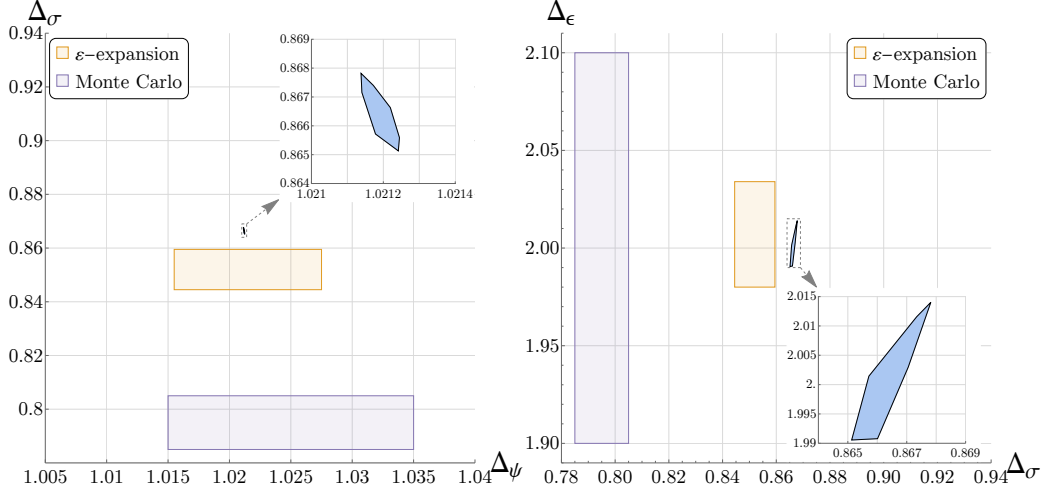


Figure 2.7: The allowed region for the $N = 8$ critical GNY model when imposing that $\Delta_{\sigma'} > 3$ computed at $n_{\max} = 18$, projected to the $(\Delta_\sigma, \Delta_\epsilon)$ and $(\Delta_\psi, \Delta_\sigma)$ planes. This should be compared to the Borel re-summed result obtained from the ϵ -expansion [41], once again obtained using the DREG₃ regularization scheme (and shown in orange), and with the Monte Carlo results from [51] (shown in purple).

We also report (nonrigorous) estimates at $N = 2$ at $n_{\max} = 14^{24}$ of the OPE coefficient ratios of $\frac{\lambda_{\psi\psi\sigma}}{\lambda_{\sigma\sigma\psi}} = 0.5087(10)$, $\frac{\lambda_{\psi\psi\epsilon}}{\lambda_{\sigma\sigma\psi}} = 0.2392(6)$, and $\frac{\lambda_{\epsilon\epsilon\epsilon}}{\lambda_{\sigma\sigma\psi}} = 1.629(13)$. Note that these estimates are nonrigorous as discussed in section 2.3.4.1; for the reader's convenience they are also reported in Table 2.5.

As noted in section 2.2.3, the assumption of $\Delta_{\sigma'} > 3$ for $N = 2$ is perhaps at risk of being too strong because of the $N = 1$ value which violates this assumption. We note that despite this assumption we are still able to find feasible points. Moreover, the two-sided Padé interpolations shown in figure 2.3 support this assumption, so we think this gap is justified. In future work we plan to study the CFT data in the σ' sector more rigorously using the navigator method [12]. For now, we report our estimates using both $\Delta_{\sigma'} > 3$ as well as the more conservative assumption of $\Delta_{\sigma'} > 2.5$ in tables 2.1 and 2.5.

For $N = 4$ at $n_{\max} = 18$, we can report rigorous estimates of our external scaling dimensions of $\Delta_\psi = 1.04356(\mathbf{16})$, $\Delta_\sigma = 0.7578(\mathbf{15})$, $\Delta_\epsilon = 1.899(\mathbf{10})$, given the assumptions as discussed in section 2.2.3 of $\Delta_{\epsilon'} > 3$, $\Delta_{\sigma'} > 3$, $\Delta_{\sigma_T} >$

²⁴We report our $n_{\max} = 14$ estimates because we ran out of computing resources while computing $n_{\max} = 18$. While we have enough data to feel confident in our rigorous scaling dimension estimates, we deferred to our lower n_{\max} results for the OPE coefficient ratio estimates.

2, $\Delta_{\psi'} > 2$, $\Delta_\chi > 3.5$. Figure 2.6 shows the allowed regions after projections. In this case, the ϵ -expansion estimates [41] are excluded by the conformal bootstrap.²⁵ On the other hand, the existing Monte Carlo results [50] give no estimates on Δ_ψ , and the reported error bars are excluded by the conformal bootstrap in the $(\Delta_\sigma, \Delta_\epsilon)$ plane. A subtlety to be noticed is that the MC estimates shown in the plots were obtained based on the $O(2)^2 \times \mathbb{Z}_2$ “chiral” GNY model as discussed in section 2.2.4, whose CFT data for $\{\Delta_\psi, \Delta_\sigma, \Delta_\epsilon\}$ is expected to be slightly different from that of the $O(4)$ GNY model that we implemented in the conformal bootstrap.

We can also report estimates at $N = 4$ of OPE coefficient ratios of $\frac{\lambda_{\psi\psi\sigma}}{\lambda_{\sigma\sigma\psi}} = 0.4386(6)$, $\frac{\lambda_{\psi\psi\epsilon}}{\lambda_{\sigma\sigma\psi}} = 0.15530(19)$, and $\frac{\lambda_{\epsilon\epsilon\epsilon}}{\lambda_{\sigma\sigma\psi}} = 1.682(18)$. Note that these estimates are nonrigorous as discussed in section 2.3.4.1; for the reader’s convenience they are also reported in Table 2.5.

For $N = 8$ and $n_{\max} = 18$, we can report rigorous estimates of our external scaling dimensions of $\Delta_\psi = 1.02119(\mathbf{5})$, $\Delta_\sigma = 0.8665(\mathbf{13})$, $\Delta_\epsilon = 2.002(\mathbf{12})$, given the assumptions as discussed in section 2.2.3 of $\Delta_{\epsilon'} > 3$, $\Delta_{\sigma'} > 3$, $\Delta_{\sigma_T} > 2$, $\Delta_{\psi'} > 2$, $\Delta_\chi > 3.5$. Figure 2.7 shows the allowed regions after projections. The reported error bars for the Δ_σ estimates of the ϵ -expansion [41] and the Monte Carlo results [51] are excluded by the conformal bootstrap, while the other estimates show good agreement. Precision is considerably improved for all scaling dimensions, especially for Δ_ψ . It should be noted again, the MC estimates were obtained by studying the chiral theory, discussed in section 2.2.4.

We can also report estimates at $N = 8$ of OPE coefficient ratios of $\frac{\lambda_{\psi\psi\sigma}}{\lambda_{\sigma\sigma\psi}} = 0.3322(8)$, $\frac{\lambda_{\psi\psi\epsilon}}{\lambda_{\sigma\sigma\psi}} = 0.08082(12)$, and $\frac{\lambda_{\epsilon\epsilon\epsilon}}{\lambda_{\sigma\sigma\psi}} = 1.71(4)$. Note that these estimates are nonrigorous as discussed in section 2.3.4.1; for the reader’s convenience they are also reported in Table 2.5.

2.4.2 $N = 1$ and emergent supersymmetry

We also computed islands at $n_{\max} = 6, 10$ for the $N = 1$ case, shown in Figure 2.8. Shown also in the plot is the expected relation between scaling dimensions for an $\mathcal{N} = 1$ SCFT; our three external operators are expected to all be in a supermultiplet with each other. We can see that the very tip of

²⁵It’s worth noting that the ϵ -expansion results encounter a pole for $N \simeq 2$ which appears to distort some of their resummations [41].

	Δ_ψ	Δ_σ	Δ_ϵ	$\lambda_{\psi\psi\sigma}/\lambda_{\sigma\sigma\epsilon}$	$\lambda_{\psi\psi\epsilon}/\lambda_{\sigma\sigma\epsilon}$	$\lambda_{\epsilon\epsilon\epsilon}/\lambda_{\sigma\sigma\epsilon}$
N = 2						
$n_{\max} = 18, \Delta_{\sigma'} > 2.5$	1.0672(25)	0.657(13)	1.74(4)	0.5071(15)	0.2347(35)	1.636(17)
$n_{\max} = 14, \Delta_{\sigma'} > 3$	1.06860(16)	0.6498(14)	1.724(8)	0.5087(10)	0.2392(6)	1.629(13)
$n_{\max} = 18, \Delta_{\sigma'} > 3$	1.06861(12)	0.6500(12)	1.725(7)	–	–	–
N = 4						
$n_{\max} = 18, \Delta_{\sigma'} > 3$	1.04356(16)	0.7578(15)	1.899(10)	0.4386(6)	0.15530(19)	1.682(18)
N = 8						
$n_{\max} = 18, \Delta_{\sigma'} > 3$	1.02119(5)	0.8665(13)	2.002(12)	0.3322(8)	0.08082(12)	1.71(4)

Table 2.5: A summary of the results of this work with estimates of all six search parameters, compiled here for the reader’s convenience. We do not report the $n_{\max} = 18$ OPE coefficient ratio estimates, as we did not have sufficient statistics. Note that we have reported the scaling dimension estimates as only scaling dimensions, and we have only included this work’s results. For critical exponents and comparisons, see Table 2.1.

the island indeed is consistent with the assumption of supersymmetry. There is a long tail, however, which seems to get cut away as n_{\max} is increased. The intersection of the tip of the island with the supersymmetric line is very narrow, and at $n_{\max} = 10$ we can report a rigorous estimate of Δ_σ of 0.58444(**8**), which is both completely consistent though roughly an order of magnitude less precise than the $n_{\max} = 30$ superconformal bootstrap results reported in [54].²⁶

One advantage of our mixed fermion bootstrap setup is that we now have access to half-integer spin exchange channels. In this case, the parity-odd $\ell = 3/2$ channel would be expected to have a supercurrent at the unitarity bound of $\Delta = 2.5$ if the solution to crossing corresponds to a supersymmetric CFT. We can therefore investigate whether a given solution to crossing in our $N = 1$ island *must* have supersymmetry by imposing gaps in the supercurrent channel and seeing what the upper bound of that gap is. If the feasibility of a solution to crossing is sensitive to this gap assumption, we can say that in that solution there must be a low-lying operator in that spectrum.

Specifically, we can perform a binary search in the supercurrent gap to find precisely what is the upper bound of the scaling dimension of the leading operator in that channel. Scanning along an axis of our $n_{\max} = 10$ island, which goes from the tip through the center (shown in Figure 2.9), we find that for the entire scan $\Delta_{\text{SC}} < 2.54$. In particular, at the tip of the island

²⁶While the n_{\max} values reported here are quite different, we should note that the system of crossing equations studied in [54] has only 4 crossing equations while our $N = 1$ system has 28.

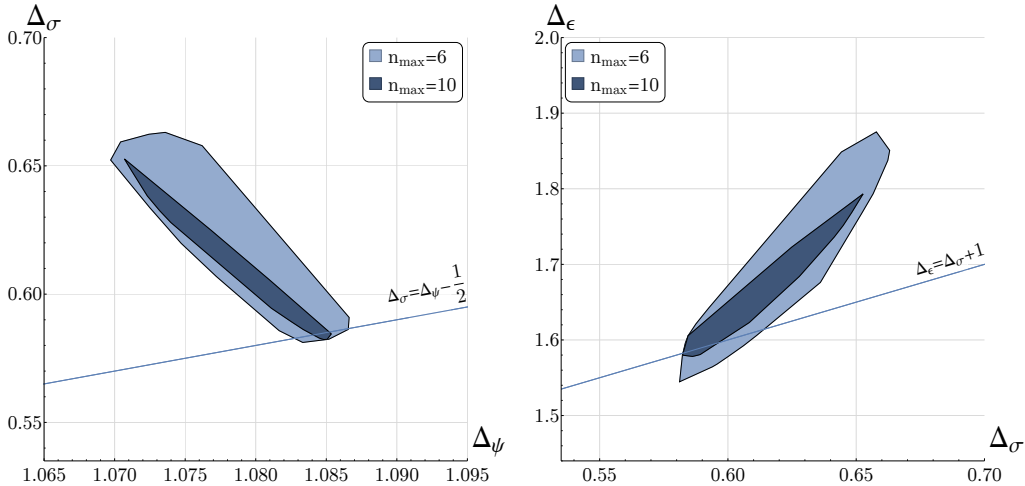


Figure 2.8: The allowed region for the $N = 1$ critical GNY model when imposing that $\Delta_{\sigma'} > 2.5$, projected to the $(\Delta_\psi, \Delta_\sigma)$ and $(\Delta_\sigma, \Delta_\epsilon)$ planes. The solid lines capture the expected relation between scaling dimensions in an $\mathcal{N} = 1$ SCFT. The tip intersects with the supersymmetric constraint on scaling dimensions, though note that due to the projection and visualization method (described in section 2.3.4.2) that the extent of that intersection is exaggerated, as the line and island are actually in 3-dimensional space. We separately computed a binary search along the supersymmetric line in all three external dimensions and found a rigorous estimate of $\Delta_{\sigma'} = 0.58444(8)$. This agrees exactly with those found for the $\mathcal{N} = 1$ super-Ising model in [54].

(corresponding to the $\mathcal{N} = 1$ super-Ising model), the upper bound drops to $\Delta_{\text{SC}} < 2.5003219$. This implies that any CFT with these parameters must be, to a high degree of precision, supersymmetric.²⁷

2.5 Discussion

In this work we have obtained the first rigorous and precise islands for the conformal data of the 3d $O(N)$ Gross-Neveu-Yukawa fixed points from the conformal bootstrap. Much like the 3d Ising and $O(N)$ vector models, these theories appear to be readily amenable to bootstrap methods. In particular, we have shown that one can obtain small islands in the parameter space of CFT data after studying the crossing relations for the operators $\{\psi, \sigma, \epsilon\}$ and

²⁷We also did a preliminary exploration of how the bound on lowest spin-3/2 operator changes as a function of the gap in $\Delta_{\sigma'}$. As this gap is increased towards the value that it takes in the $\mathcal{N} = 1$ super-critical Ising model, we saw that the upper-bound on the dimension of the spin-3/2 operator becomes even stronger, as the right part of the plot (the tail of the island) shrinks away.

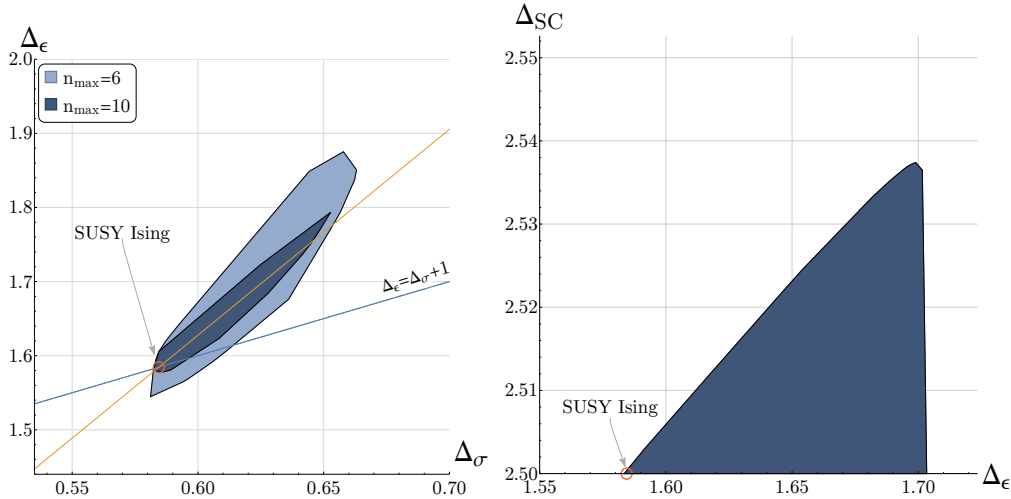


Figure 2.9: On the right we show the upper-bound at $n_{\max} = 10$ on the scaling dimension of the lowest spin-3/2 operator as a function of Δ_ϵ obtained along an axis of the $N = 1$ island when imposing $\Delta_{\sigma'} > 2.5$. The axis was selected such that it interpolates between the location of the $\mathcal{N} = 1$ super-Ising model as determined in previous literature [52, 53, 54] and the center of the $n_{\max} = 10$ island. We show this axis projected into $(\Delta_\sigma, \Delta_\epsilon)$ space in orange in the figure on the left. We see that all points along this axis are forced to have a spin-3/2 operator that is close to the unitarity bound, where such operator becomes a supercurrent. The red circle in the plot on the right gives the location of the critical $\mathcal{N} = 1$ super-ising model as determined from the $\mathcal{N} = 1$ superconformal bootstrap [54].

imposing gaps in the spectrum which isolate the leading scalar and spin-1/2 operators. The gaps we have chosen are motivated by perturbative calculations in the large- N and ϵ -expansions. They can also be viewed as being necessary in order to exclude potential nearby nonlocal fixed points obtained by coupling the leading operators in the GNY theories to generalized free fields. In the χ and σ' sectors the gaps can be viewed as imposing a consequence of the equations of motion for the fundamental fields. We saw a particular sensitivity of our results to the gap in $\Delta_{\sigma'}$, and establishing its irrelevance down to $N = 2$ appears to be an important open problem. In future work we plan to study the allowed values of this gap more rigorously using navigator methods [12] and use the results to further improve our islands.

In the present study, we focused on the $O(N)$ -invariant GNY theories with $N = 1, 2, 4, 8$, but it is clear that the results can be extended to any N . In the case of $N = 1$, the fixed point is believed to have emergent 3d $\mathcal{N} = 1$ su-

persymmetry. If one assumes supersymmetry then one can perform a precise bootstrap of this model using only external scalar fields as was demonstrated in [54]. If one does not assume supersymmetry, then we have seen in the present work that the numerical bootstrap with external fermions still forces the solution to be approximately supersymmetric, requiring a spin-3/2 “supercurrent” operator in the spectrum that is near the unitarity bound.

In this paper we also highlighted the distinction between the $O(N)$ GNY fixed points and the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY fixed points. For the leading operators $\{\psi, \sigma, \epsilon\}$, differences in their scaling dimensions only show up at 4-loop order in the ϵ and large- N expansions and they are expected to be extremely close to each other. On the other hand, the models have more significant differences in other parts of the spectrum, e.g. in the number of conserved currents and spectrum of fermion bilinear operators. In the future it will be interesting to study the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY fixed points using numerical bootstrap methods and see if we can clearly resolve the difference between these models. It will additionally be interesting to study bounds on OPE coefficients and central charges in both models. These fixed points can also readily be generalized to models with multiple scalar fields, e.g. the chiral-XY and chiral-Heisenberg GNY fixed points, which will also be interesting targets for the bootstrap.

Our work was the first significant application of the new software tool `blocks_3d` [18], which enables systematic and efficient calculations of 3d conformal blocks with arbitrary spinning operators. Along with it, we have developed an extensible and modular software stack, as described in appendix 2.6.1. The success of our study makes it clear that this approach can be used in other studies of interest, e.g. mixed correlators containing various combinations of scalars, fermions, currents, and stress tensors. We hope that future systematic studies of the bootstrap constraints for such correlators will lead to the discovery of exciting new islands in the vast ocean of possible CFTs.

2.6 Appendix

2.6.1 Software

For this work, we implemented and used several software packages, which we briefly describe in this appendix.²⁸ We indicate with a “★” the packages that were newly written for this work. Other packages have been re-used (and in

²⁸Note: many of these packages are works in progress, and their names and APIs are subject to change.

some cases modified) from other projects. The profusion of libraries is because we have made an effort to split them up into an orthogonal set of features.

- SDPB (<https://github.com/davidsd/sdpb>): a C++ program for solving semidefinite programs [1, 2].
- `blocks_3d` (https://gitlab.com/bootstrapcollaboration/blocks_3d): a C++ program for computing spinning conformal blocks in 3d [15, 18].
- `scalar_blocks` (https://gitlab.com/bootstrapcollaboration/scalar_blocks): a C++ program for computing scalar conformal blocks in general d .
- `hyperion` (<https://github.com/davidsd/hyperion>): a Haskell framework for concurrent computations on an HPC cluster.
- `hyperion-bootstrap` (<https://gitlab.com/davidsd/hyperion-bootstrap>): A Haskell library for computing numerical bootstrap bounds on an HPC cluster using `hyperion`.
- `sdpb-haskell` (<https://gitlab.com/davidsd/sdpb-haskell>): A Haskell interface to SDPB.
- ★ `bootstrap-build` (<https://gitlab.com/davidsd/bootstrap-build>): A Haskell “build system” for building objects and their dependencies.
- `bootstrap-math` (<https://gitlab.com/davidsd/bootstrap-math>): A Haskell math library containing datatypes and algorithms useful for bootstrap computations.
- ★ `blocks-core` (<https://gitlab.com/davidsd/blocks-core>): Core Haskell datatypes and functions for conformal blocks in bootstrap computations.
- ★ `blocks-3d` (<https://gitlab.com/davidsd/blocks-3d>): A Haskell interface to the C++ program `blocks_3d`, built on `blocks-core`.
- ★ `scalar-blocks` (<https://gitlab.com/davidsd/scalar-blocks>): A Haskell interface to the C++ program `scalar_blocks`, built on `blocks-core`.
- ★ `bootstrap-bounds` (<https://gitlab.com/davidsd/bootstrap-bounds>): A Haskell library for setting up crossing equations using information

about three- and four-point structures. This library implements the algorithm described in section 2.3.

- **quadratic-net** (<https://gitlab.com/davidsd/quadratic-net>): A search algorithm for solving non-convex quadratic constraints in low numbers of dimensions, used in the OPE scan algorithm of [10].
- ★ **scalars-3d** (<https://gitlab.com/davidsd/scalars-3d>): An implementation of several bootstrap bounds on scalar theories in 3d, using the libraries listed here.
- ★ **fermions-3d** (<https://gitlab.com/davidsd/fermions-3d>): An implementation of several bootstrap bounds on GNY models in 3d, using the libraries listed here. This is the main “umbrella” package for the computations in this work.

2.6.2 Δ_χ and $\Delta_{\psi'}$ at large N

To better isolate the $O(N)$ GNY-model using the conformal bootstrap it is useful to get a better estimate for the scaling dimension of various operators in the theory, especially for those that are involved in the equation of motion or in the new fermionic channels that are involved in a fermionic-scalar OPE. In particular, we would like to compute the $1/N$ correction to the dimension of the fermionic operators $\phi^2\psi_i$ (the lowest dimensional operator above ψ_i) and $\phi^3\psi_i$ (the lowest dimensional primary in its parity sector and $O(N)$ representation). We will determine these scaling dimension by extracting the logarithmic divergence coefficient in the propagator of the more generic operator $\mathcal{O}_i = \phi^k\psi_i$. The bare dimension of this operator is $\Delta_{\phi^k\psi_i} = k+1+O(1/N)$. We will follow a similar logic to that used to determine the dimension of ϕ^k in appendix B of [25].

Up to order $1/N$, this propagator (in the concrete, but generalizable, example

with $k = 2$) is given by the following diagrams:

$$\begin{aligned}
 D_{\mathcal{O}_i}(p) = & \mathcal{O}_i(p) \bullet \text{---} \text{---} \text{---} \bullet \mathcal{O}_i(-p) + k \left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right] \\
 & + \frac{k(k-1)}{2} \left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right] + \left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right] + k \left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right].
 \end{aligned}
 \tag{2.93}$$

The first diagram gives the leading bare propagator. The next two sub-leading terms (i.e. the one proportional to k and the one proportional to $k(k-1)/2$) capture the anomalous dimension of ϕ^k , which was found to be $\delta_{\phi^k} = \frac{1}{N} \frac{16k(3k-5)}{3\pi^2}$ [25]. The next term yields the $1/N$ correction to the fermionic propagator and gives the contribution of the anomalous dimension of the fermionic field, $\delta_{\psi_i} = \frac{1}{N} \frac{4}{3\pi^2}$. Finally, the last term is a new diagram which yields a correction $k\eta$ to the anomalous dimension. Thus, the scaling dimension of the operator is given by,

$$\Delta_{\phi^k \psi_i} = k + 1 + \delta_{\phi^k} + \delta_{\psi_i} + k\eta.
 \tag{2.94}$$

To determine η we note that in the special case when $k = 1$, following from the equations of motion, the operator \mathcal{O}_i is a descendant and therefore, $\Delta_{\phi \psi_i} = 1 + \Delta_{\psi_i}$. This implies that the logarithmic divergences for the following diagrams should cancel each other:

$$\left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right] + \left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right] + \left[\text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \right] = 0 \log \Lambda + \dots,
 \tag{2.95}$$

from which we conclude that $\eta = -\delta_\phi = \frac{32}{3\pi^2 N}$. Consequently, we find:

$$\Delta_{\phi^k \psi_i} = k + 1 + \frac{1}{N} \left(\frac{16k(3k-5)}{3\pi^2} + \frac{4}{3\pi^2} + \frac{32k}{3\pi^2} \right) = k + 1 + \frac{48k(k-1) + 4}{3\pi^2 N} + O\left(\frac{1}{N^2}\right). \quad (2.96)$$

In particular, we find

$$\Delta_{\psi'_i} = 3 + \frac{100}{3\pi^2 N} + O\left(\frac{1}{N^2}\right), \quad (2.97)$$

$$\Delta_{\chi_i} = 4 + \frac{292}{3\pi^2 N} + O\left(\frac{1}{N^2}\right). \quad (2.98)$$

2.6.3 More on the differences between the $O(N)$ and $O(N/2)^2 \times \mathbb{Z}_2$ GNY models

2.6.3.1 2- vs 4-component

We will note that in the literature sometimes authors will describe the theories discussed in this paper using the language of 4-component fermions as opposed to the natively 3-dimensional 2-component language which we use in this paper. This is especially true in the case of $4 - \epsilon$ calculations. A Lagrangian may look like

$$\mathcal{L}_\chi = -\frac{1}{2}(\partial\phi)^2 - i\bar{\Psi}_i \not{\partial} \Psi_i - \frac{1}{2}m^2\phi^2 - \frac{\lambda}{4}\phi^4 - ig\phi\bar{\Psi}_i\Psi_i, \quad (2.99)$$

where Ψ_i are 4-component Dirac spinors²⁹ and $i = 1 \dots N_4$, $N_4 \equiv N/4$. Notably, the Lagrangian has an additional discrete ‘‘chiral’’ symmetry of $\Psi_i \rightarrow \gamma^5 \Psi_i$, $\phi \rightarrow -\phi$, inherited from the four-dimensional theory.³⁰ Models of this form have been studied extensively to understand the chiral Ising universality class, which for $N = 4, 8$ (i.e. $N_4 = 1, 2$) describe spinless/spinful critical points for the semimetal-to-CDW transition in graphene. The critical points describe breaking of the symmetry while preserving time-reversal symmetry [41, 40].

To make the distinction between this theory more clear from the theory that is the focus of our paper, it is helpful to consider decomposing our 4-component Dirac spinors into 2-component Majorana spinors. In a suitably convenient

²⁹Here we define $\bar{\Psi}_i \equiv \Psi_i^\dagger \tilde{\gamma}^0$, where $\tilde{\gamma}^0$ is defined in 2.100. The Clifford algebra satisfies $\{\tilde{\gamma}^\mu, \tilde{\gamma}^\nu\} = 2\eta^{\mu\nu}$, just like the 2×2 gamma matrices defined in [25] that we use in this paper.

³⁰We denote the ‘‘chiral’’ symmetry in quotes as there is no inherent spacetime notion of chirality in three (or indeed, any odd number of) dimensions. This symmetry amounts to an internal flavor symmetry of the fermions.

4×4 basis of the gamma matrices, such as the following defined in [63, 81] in terms of the 2×2 γ^μ we have used throughout this paper (defined in [25]), we have:

$$\tilde{\gamma}^\mu = \begin{pmatrix} \gamma^\mu & 0 \\ 0 & -\gamma^\mu \end{pmatrix}, \quad \mu = 0, 1, 2; \quad \tilde{\gamma}^3 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (2.100)$$

We should note that the $\tilde{\gamma}^3$ will not play a role in the 3d theory; thus this gamma matrix basis reduces to a block-diagonal form in 3d. Therefore, the 4-component spinors can be broken down into 2-component spinors as

$$\Psi_i = \begin{pmatrix} \psi_i^L \\ \psi_i^R \end{pmatrix} = \begin{pmatrix} \psi_i \\ \psi_{i+N_4} \end{pmatrix}. \quad (2.101)$$

We get

$$\mathcal{L}_\chi = -\frac{1}{2}(\partial\phi)^2 - i\bar{\psi}_i \not{\partial} \psi_i - \frac{1}{2}m^2\phi^2 - \frac{\lambda}{4}\phi^4 - ig\phi(\bar{\psi}_i^L \psi_i^L - \bar{\psi}_i^R \psi_i^R). \quad (2.102)$$

(This can be further decomposed into the requisite Majorana spinors without significant difference in form, save for the index i going from 1 to $2N_4$.) In this form, we see that we have two distinct fermion species with opposite signs on their Yukawa couplings; the chiral symmetry becomes $\psi_i^L \leftrightarrow \psi_i^R$, $\phi \rightarrow -\phi$. We can see that, if there are N Majorana fermions total, the symmetry of this Lagrangian is $O(N/2)^2 \rtimes \mathbb{Z}_2$, where \mathbb{Z}_2 is the chiral symmetry. For completeness, we will note that the 4-component spinor notation for the fermion bilinear which appears in (2.1) is $i\bar{\Psi}_i \gamma^3 \gamma^5 \Psi_i$.

2.6.3.2 Irreps of $O(N/2)^2 \rtimes \mathbb{Z}_2$

In the next subsection we will classify the various low-lying primary operators of the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY theory. To attempt such a classification, we should first discuss the irreps of $O(N/2)^2 \rtimes \mathbb{Z}_2$. We will start with a more general discussion. Given a compact simple Lie group G , let $H = (G \times G) \rtimes \mathbb{Z}_2$. The group H is generated by $(g_L, g_R) \in G \times G$, together with an element s such that $s^2 = 1$ and

$$s(g_L, g_R) = (g_R, g_L)s. \quad (2.103)$$

For each irrep ρ of H , we can consider its restriction to $G \times G$. The irreps of $G \times G$ have the form $\rho_1 \boxtimes \rho_2$ where ρ_1, ρ_2 are irreps of G . The symbol \boxtimes means

we take a tensor product as vector spaces, but not as G representations. The first G acts on the left-tensor factor and the second G acts on the right tensor factor.

Suppose the restriction of ρ contains the $G \times G$ irrep $\rho_1 \boxtimes \rho_2$. That is, we have a $G \times G$ homomorphism

$$\phi \in \text{Hom}_{G \times G}(\rho_1 \boxtimes \rho_2, \rho). \quad (2.104)$$

Now consider $\phi' : \rho_2 \boxtimes \rho_1 \rightarrow \rho$ given by

$$\phi'(v_2 \otimes v_1) \equiv s\phi(v_1 \otimes v_2), \quad (2.105)$$

where $s\phi(v_1 \otimes v_2)$ denotes the action of $s \in H$ on the vector $\phi(v_1 \otimes v_2) \in \rho$. We have

$$\begin{aligned} \phi'((g_L, g_R)(v_2 \otimes v_1)) &= \phi'(g_L v_2 \otimes g_R v_1) \\ &= s\phi(g_R v_1 \otimes g_L v_2) \\ &= s(g_R, g_L)\phi(v_1 \otimes v_2) \\ &= (g_L, g_R)\phi'(v_2 \otimes v_1), \end{aligned} \quad (2.106)$$

so we see that ϕ' is a $G \times G$ homomorphism from $\rho_2 \boxtimes \rho_1$ to ρ :

$$\phi' : \text{Hom}_{G \times G}(\rho_2 \boxtimes \rho_1, \rho). \quad (2.107)$$

We have two cases to consider.

- Suppose that ρ_1, ρ_2 are distinct. Clearly $\phi(\rho_1 \boxtimes \rho_2) + \phi'(\rho_2 \boxtimes \rho_1)$ is an H -invariant subspace of ρ , and thus must be all of ρ . Since each of ϕ, ϕ' is a $G \times G$ isomorphism onto its image, this furnishes an isomorphism

$$\rho \cong (\rho_1 \boxtimes \rho_2) \oplus (\rho_2 \boxtimes \rho_1) \quad (2.108)$$

as $G \times G$ representations. Furthermore, the two summands are swapped by the action of s . We denote the corresponding H -representation by $\rho = \langle \rho_1, \rho_2 \rangle$.

- Suppose that $\rho_1 = \rho_2$. In this case, we claim that $\rho_{\pm} = (\phi \pm \phi')(\rho_1 \boxtimes \rho_1)$ are H -invariant subspaces of ρ . Clearly they are $G \times G$ -invariant subspaces. To prove they are also H -invariant, note that

$$\begin{aligned} s(\phi \pm \phi')(u \otimes v) &= s(\phi(u \otimes v) \pm s\phi(v \otimes u)) \\ &= s\phi(u \otimes v) \pm \phi(v \otimes u) \\ &= \pm(\phi \pm \phi')(v \otimes u) \in \rho_{\pm}. \end{aligned} \quad (2.109)$$

By irreducibility, one of the ρ_{\pm} must be all of ρ , and the other must vanish. When $\rho = \rho_+$ or $\rho = \rho_-$, we denote the H -representation by $\langle \rho_1, \rho_1 \rangle_{\pm}$. A basis of $\langle \rho_1, \rho_1 \rangle_{\pm}$ is given by $u \otimes v$ with $u, v \in \rho_1$, with the s -action

$$s(u \otimes v) = \pm v \otimes u. \quad (2.110)$$

Below, when listing the perturbative estimates for the scaling dimensions for some of the low-lying operators in the theory we will be interested in $\rho_i \in \{\bullet, \square, \square\square, \square\square\square\}$ which corresponds to the singlet, vector, symmetric traceless tensor and antisymmetric representations.

2.6.3.3 Large- N computations in the $O(N/2)^2 \times \mathbb{Z}_2$ GNY model

As explained in section 2.2.4, the large- N Feynman diagrams in the $O(N)$ GNY model and $O(N/2)^2 \times \mathbb{Z}_2$ GNY model only differ by diagrams that contain fermionic loops with five or more fermion propagators. For instance, in the GNY model we have that

$$\begin{aligned} \Delta_{\psi_i} \supset & \quad \sim \frac{1}{N^4}, \\ \Delta_{\phi^2} \supset & \quad \sim \frac{1}{N^3}, \end{aligned} \quad (2.111)$$

where we have again listed examples of diagrams (rather than all the diagrams) contributing to the leading non-vanishing order. As explained in section 2.2, these diagrams cancel in the $O(N/2)^2 \times \mathbb{Z}_2$ GNY models. More generally, since such loops only contribute at higher order we see that at leading order

$$\Delta_{\mathcal{O}_\rho}^{O(N)} = \Delta_{\mathcal{O}_{(\rho, \bullet)}}^{O(N/2)^2 \times \mathbb{Z}_2} + O\left(\frac{1}{N^3}\right), \quad (2.112)$$

where \mathcal{O} can represent an operator with any spin and parity representation, ρ can be one of the $\{\square, \square\square, \square\square\square\}$ irreps of $O(N)$ or $O(N/2)$, and we have

added the $O(N)$ and $O(N/2)^2 \rtimes \mathbb{Z}_2$ superscripts to differentiate between the two models.

If $\rho = \bullet$, then we have to consider the $O(N/2)^2 \rtimes \mathbb{Z}_2$ irreps $\langle \bullet, \bullet \rangle_{\pm}$. For operators that have the same representation under parity as under the chiral \mathbb{Z}_2 , their scaling dimensions match that of the operator with the same parity in the $O(N)$ GNY model for low enough order in the large- N expansion: for instance, $\Delta_{\sigma_{\langle \bullet, \bullet \rangle_-}^{O(N/2)^2 \rtimes \mathbb{Z}_2}} \approx \Delta_{\sigma}^{O(N)}$ and $\Delta_{\epsilon_{\langle \bullet, \bullet \rangle_+}^{O(N/2)^2 \rtimes \mathbb{Z}_2}} \approx \Delta_{\epsilon}^{O(N)}$. If that is not the case, a more elaborate analysis is needed. For example, consider the operator $\sigma_{\langle \bullet, \bullet \rangle_+} \sim \psi_i^L \psi_i^L + \psi_i^R \psi_i^R$.³¹ The diagrams contributing to the two-point function of this operator are precisely the same as those contributing to the two-point function of $\sigma_{\langle \square\square, \bullet \rangle}$. The fact that the $O(N/2)$ indices are contracted differently does not matter at low enough order when computing the scaling dimension of the operator and only affects the overall normalization of the two-point function. Therefore, we find that

$$\Delta_{\sigma_{\square\square}}^{O(N)} \approx \Delta_{\sigma_{\langle \square\square, \bullet \rangle}^{O(N/2)^2 \rtimes \mathbb{Z}_2}} \approx \Delta_{\sigma_{\langle \bullet, \bullet \rangle_+}^{O(N/2)^2 \rtimes \mathbb{Z}_2}}. \quad (2.113)$$

This approximate relation between the scaling dimensions in the singlet sector and those in the $\langle \square\square, \bullet \rangle$ irrep can be extended to the parity even sector. For instance, consider the operator $\epsilon_{\sigma_{\square\square}} \sim \phi \psi_{(i} \psi_{j)}$ in the $O(N)$ model and the operators $\epsilon_{\langle \square\square, \bullet \rangle} \sim \phi \psi_{(i}^{L,R} \psi_{j)}^{L,R}$ and $\epsilon_{\langle \bullet, \bullet \rangle_-} \sim \phi(\psi_i^L \psi_i^L + \psi_i^R \psi_i^R)$ in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ model. Once again the large- N Feynman diagrams of the above operators are identical and we find

$$\Delta_{\epsilon_{\square\square}}^{O(N)} \approx \Delta_{\epsilon_{\langle \square\square, \bullet \rangle}^{O(N/2)^2 \rtimes \mathbb{Z}_2}} \approx \Delta_{\epsilon_{\langle \bullet, \bullet \rangle_-}^{O(N/2)^2 \rtimes \mathbb{Z}_2}}. \quad (2.114)$$

Next, we discuss some of the operators in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ model whose scaling dimensions are not found among operators in the $O(N)$ model even at leading order in the large- N expansion. One example of such an operator is $\sigma_{\langle \square, \square \rangle_+} \sim \psi_i^L \psi_j^R + \psi_i^R \psi_j^L$. The two-point function of such an operator up to

³¹This is in a different irrep than ϕ itself which as mentioned above lies in $\langle \bullet, \bullet \rangle_-$, but transforms in the same way under parity.

Operator	Spin	Parity	Global irrep.	Δ at large N	GNY op.
$\psi_{\langle \square, \bullet \rangle} \sim \psi_i^{L,R}$	$\frac{1}{2}$	+	$\langle \square, \bullet \rangle$	$1 + \frac{4}{3\pi^2 N} + \dots$	ψ_{\square}
$\psi'_{\langle \square, \bullet \rangle} \sim \phi^2 \psi_i^{L,R}$	$\frac{1}{2}$	+	$\langle \square, \bullet \rangle$	$3 + \frac{100}{3\pi^2 N} + \dots$	ψ'_{\square}
$\chi_{\langle \square, \bullet \rangle} \sim \phi^3 \psi_i$	$\frac{1}{2}$	-	$\langle \square, \bullet \rangle$	$4 + \frac{292}{3\pi^2 N} + \dots$	χ_{\square}
$\sigma_{\langle \bullet, \bullet \rangle_-} \sim \phi$	0	-	$\langle \bullet, \bullet \rangle_-$	$1 - \frac{32}{3\pi^2 N} + \dots$	σ_{\bullet}
$\epsilon_{\langle \bullet, \bullet \rangle_+} \sim \phi^2$	0	+	$\langle \bullet, \bullet \rangle_+$	$2 + \frac{32}{3\pi^2 N} + \dots$	ϵ_{\bullet}
$\sigma'_{\langle \bullet, \bullet \rangle_-} \sim \phi^3$	0	-	$\langle \bullet, \bullet \rangle_-$	$3 + \frac{64}{\pi^2 N} + \dots$	σ'_{\bullet}
$\epsilon'_{\langle \bullet, \bullet \rangle_+} \sim \phi^4$	0	+	$\langle \bullet, \bullet \rangle_+$	$4 + \frac{448}{3\pi^2 N} + \dots$	ϵ'_{\bullet}
ϕ^k	0	$(-1)^k$	$\langle \bullet, \bullet \rangle_{(-1)^k}$	$k + \frac{16k(3k-5)}{3\pi^2 N} + \dots$	ϕ^k
$\sigma_{\langle \square, \square \rangle} \sim \psi_{(i)}^{L,R} \psi_{(j)}^{L,R}$	0	-	$\langle \square, \square \rangle$	$2 + \frac{32}{3\pi^2 N} + \dots$	σ_{\square}
$\sigma_{\langle \bullet, \bullet \rangle_+} \sim \psi_i^L \psi_i^L + \psi_i^R \psi_i^R$	0	-	$\langle \bullet, \bullet \rangle_+$	$2 + \frac{32}{3\pi^2 N} + \dots$	σ_{\square}
$\sigma_{\langle \square, \square \rangle_+} \sim \psi_i^L \psi_j^R + \psi_i^R \psi_j^L$	0	-	$\langle \square, \square \rangle_+$	$2 - \frac{16}{3\pi^2 N} + \dots$	-
$\sigma_{\langle \square, \bullet \rangle} \sim j_{\langle \square, \bullet \rangle}^{\mu} \phi^2 \partial_{\mu} \partial^2 \phi$	0	-	$\langle \square, \bullet \rangle$	$8 + \dots$	σ_{\square}
$j_{\langle \square, \bullet \rangle}^{\mu} \sim \psi_{[i]}^{L,R} \gamma^{\mu} \psi_{[j]}^{L,R}$	1	+	$\langle \square, \bullet \rangle$	2	j_{\square}^{μ}
$j_{\langle \square, \square \rangle_-}^{\mu} \sim \psi_i^L \gamma^{\mu} \psi_j^R - \psi_i^R \gamma^{\mu} \psi_j^L$	1	+	$\langle \square, \square \rangle_-$	$2 + \frac{16}{3\pi^2 N} + \dots$	-

Table 2.6: Estimates for the large- N scaling dimensions at the $O(N/2)^2 \times \mathbb{Z}_2$ GNY critical point. The last column shows which scaling dimensions in the $O(N)$ GNY critical point match the dimensions of some operator in the $O(N/2)^2 \times \mathbb{Z}_2$ GNY critical point at low orders in the large- N expansion.

order $O(1/N^2)$ is given by

$$\begin{aligned}
D_{\sigma_{\langle \square, \square \rangle_+}}(0, x) = & \text{Diagram 1} + \text{Diagram 2} + \text{Diagram 3} + \text{Diagram 4} \\
& + O\left(\frac{1}{N^2}\right). \tag{2.115}
\end{aligned}$$

To extract the anomalous dimension of the operator, we need to read-off the coefficient of the logarithmic divergence in (2.115) for the last three diagrams $\delta_{\sigma_{\langle \square, \square \rangle_+}} = 2\delta_{\psi} - \eta_{\text{vertex}}$ which are responsible for the $1/N$ correction. The first two diagrams contribute the same amount since $\delta_{\psi} = \delta_{\psi^L} = \delta_{\psi^R}$, while the last diagram was computed in appendix B of [25] in order to compute the anomalous dimension $\delta_{\epsilon_{\square}}$ in the $O(N)$ model. The calculation is almost identical to there, the only difference is that the last diagram in (2.115) has the opposite sign due to the difference in sign between the Yukawa coupling of ψ^L and ψ^R . Therefore,

$$\Delta_{\sigma_{\langle \square, \square \rangle_+}} = 2 + \delta_{\sigma_{\langle \square, \square \rangle_+}} = 2 - \frac{16}{3\pi^2} \frac{1}{N} + O\left(\frac{1}{N^2}\right). \tag{2.116}$$

Similarly, we see that the diagrams needed to compute the scaling dimension of $\epsilon_{\langle \square, \square \rangle_-} \sim \phi(\psi_i^L \psi_j^R + \psi_i^R \psi_j^L)$ and $\epsilon_{\langle \square, \square \rangle_+} \sim \sigma_+(\psi_i^L \psi_j^R + \psi_i^R \psi_j^L)$ from the $O(N/2)^2 \rtimes \mathbb{Z}_2$ model are not among those found in the $O(N)$ model.

Both the $O(N)$ GNY and $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY models have conserved spin-one currents, with the $O(N)$ model having $j_{\square}^\mu \sim \psi_{[i} \gamma^\mu \psi_{j]}$ in the antisymmetric irrep of $O(N)$ and $j_{\langle \square, \bullet \rangle}^\mu \sim \psi_{[i}^{L,R} \gamma^\mu \psi_{j]}^{L,R}$ in the antisymmetric representation of each $O(N/2)$ subgroup. However, to further distinguish the two models and to ensure that the $O(N/2)^2 \rtimes \mathbb{Z}_2$ does not have a symmetry enhancement in the IR to a Lie group with a greater number of generators, we can compute the anomalous dimension of the bifundamental current $j_{\langle \square, \square \rangle_-}^\mu \sim \psi_i^L \gamma^\mu \psi_j^R - \psi_i^R \gamma^\mu \psi_j^L$. The computation for the anomalous dimension of this spin-1 operator follows from the computation which shows that the conserved current $j_{\langle \square, \bullet \rangle}^\mu$ has no anomalous dimension. Specifically,

$$\delta_{j_{\langle \square, \bullet \rangle}^\mu} = 0 = 2\delta_\psi + \tilde{\eta}_{\text{vertex}}, \quad \delta_{j_{\langle \square, \square \rangle_-}^\mu} = 2\delta_\psi - \tilde{\eta}_{\text{vertex}}, \quad (2.117)$$

from which it follows that

$$\Delta_{j_{\langle \square, \square \rangle_-}^\mu} = 2 + \frac{16}{3\pi^2} \frac{1}{N} + O\left(\frac{1}{N^2}\right). \quad (2.118)$$

To summarize, while operators in the $O(N)$ GNY model have associated operators in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ GNY model that have the same scaling dimension at low orders in the large- N expansion, the reverse is not true. We review all discussed large- N estimates in the $O(N/2)^2 \rtimes \mathbb{Z}_2$ model in Table 2.6.

Chapter 3

Skydiving to Bootstrap Islands

3.1 Introduction

The numerical conformal bootstrap (see [19, 20] for recent reviews) has revealed that small sets of crossing equations can encode detailed information about a CFT. But how does this information scale as we study more crossing equations? Unfortunately, larger systems of crossing equations pose new numerical challenges. More equations depend on more parameters — for example the scaling dimensions and OPE coefficients of “external” operators entering the given correlators.¹ In order to build exclusion plots in the space of CFT data, we must find efficient ways to search over these parameters.

The navigator function of [12] is a useful tool for this problem. It provides a notion of “height” on the space of CFT data. By following the navigator function to its minima, we can find allowed points in this space. In this work, we propose an efficient method for using the navigator function without computing it in detail. Our approach is analogous to skydiving onto a landscape instead of walking across it.

Let us recall the idea behind the navigator function in more detail. Numerical bootstrap problems depend on a parameter space \mathcal{P} that can include external scaling dimensions, gap assumptions, OPE coefficients, etc. There exists a binary function on \mathcal{P} , indicating whether or not a point $p \in \mathcal{P}$ is consistent

¹We refer to an operator as “external” if it appears in a four-point correlator in a given bootstrap problem. For example, when bootstrapping $\langle \mathcal{O}_1 \mathcal{O}_2 \mathcal{O}_3 \mathcal{O}_4 \rangle$, the operators $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4$ are all “external.” We refer to their scaling dimensions and the OPE coefficients among them as “external” dimensions and OPE coefficients. By contrast, an “internal” operator is one that appears in OPEs of the external operators, but is not itself external.

with crossing symmetry. In practice, this function is computed by solving a semidefinite program (SDP) [13, 5, 1]. If a so-called “dual” solution of the SDP exists, the point is inconsistent with crossing symmetry, and we deem it *infeasible*. If no “dual” solution exists, the point may be consistent with crossing symmetry, and we deem it *feasible*. The ultimate goal is to carve away all the infeasible points until only a small feasible region remains, thus determining the possible values of the physical parameters with an accuracy given by the size of the feasible region.

The navigator function of [12] replaces this binary function (*feasible* vs. *infeasible*) with a real-valued function that is positive for infeasible points and negative for feasible points. Crucially, the navigator function is continuous and locally differentiable, and its gradient provides information about where to navigate in \mathcal{P} . For example, a simple numerical minimization algorithm can be used to navigate towards a feasible point. It is also easy to find a feasible point with, say, the largest value of a given coordinate in \mathcal{P} . As discussed in [12], this is a straightforward equality-constrained optimization problem.

The searches discussed in [12] involve two steps for each point $p \in \mathcal{P}$. First, one computes the navigator function $\mathcal{N}(p)$ and its gradient $\nabla\mathcal{N}(p)$ by solving an SDP to high precision. Then, one uses these values to determine a step $p \rightarrow p + \delta p$ in \mathcal{P} . If we denote the internal variables of the SDP solver as ξ , this two-step process changes ξ and p in an alternating fashion. This is suboptimal, in particular because the optimal ξ variables are solved from scratch for each p (and the usual SDP algorithms do not allow for efficient warm-starting from a previous solution). Instead, it is much more natural to update both p and ξ simultaneously at every step, which is what we do in this work.

Beginning with initial values of the solver variables ξ and external parameters p , we compute a simultaneous step in both variables, and follow it to a new value of ξ and p . As the optimization continues, we converge simultaneously to an optimal value of ξ that solves the SDP and the desired value of p that optimizes some user-defined criterion. Thus, finding an optimal point in \mathcal{P} space occurs over a comparable time scale to the solution of a *single* SDP. This is a substantial speedup over traditional approaches, which typically require solving tens, hundreds, or even thousands of auxiliary SDPs, see figure 3.1.

For fixed p , the space of allowed ξ is convex, and thus ξ can be optimized deterministically using interior point methods. However, once we allow p to

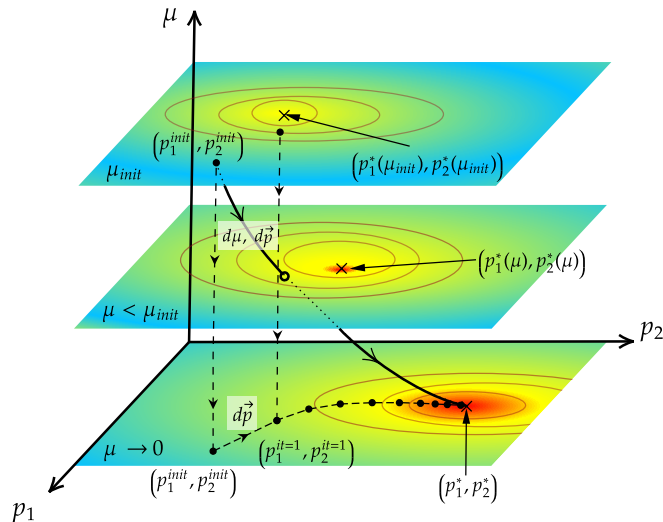


Figure 3.1: A comparison of (1) repeatedly computing the navigator function vs. (2) skydiving. Here, the variable μ measures the distance to a solution of an SDP. The SDP is solved when $\mu \rightarrow 0$. The vector $p = (p_1, p_2)$ represents external parameters. The traditional navigator approach solves an SDP for each p , beginning at large μ and converging along the dashed vertical lines towards $\mu = 0$ each time. Meanwhile, skydiving moves in p and μ simultaneously, following the solid black curve.

vary, our optimization problem is no longer convex (though violations of convexity occur in a relatively small number of dimensions). We therefore find ourselves in the less-familiar world of non-convex constrained optimization, or more precisely non-linear semidefinite programming. In this work, we focus on families of SDPs that arise in typical conformal bootstrap problems — in particular families of Polynomial Matrix Problems (PMPs) [1] depending on a small number of parameters. We find solution methods for these problems that work effectively in many examples, including ways to mitigate issues like stalling described below. Although our strategy shows promising results in the examples we have considered, we cannot claim it to be optimal or universally applicable. Continuing to explore and develop other strategies remains an interesting problem for future research.

This paper is organized as follows. In section 3.2, we review the framework of semidefinite programming and explain how to incorporate external parameters. In section 3.3, we introduce the skydiving algorithm for solving families of SDPs. We discuss details of our implementation of skydiving in section 3.4. In section 3.5, we present examples of applying skydiving to various bootstrap

problems, including the 3d Ising model and 3d O(3) model, comparing with previous methods. Finally, we conclude in section 3.6 with a discussion of some current limitations and potential improvements to skydiving, and possible future applications.

3.2 Non-linear semidefinite programming problems

3.2.1 Review of semidefinite programming

3.2.1.1 Problem formulation

Using the methods of [13, 5], a numerical conformal bootstrap problem can be formulated as a semidefinite program (SDP). Let us briefly review the solution procedure used by the semidefinite program solver SDPB [1, 2]. Following the notation of [1], an SDP takes the form:

$$\begin{aligned} & \text{maximize } b^T y \text{ over } y \in \mathbb{R}^n, Y \in \mathcal{S}^K \\ & \text{such that } Y \succeq 0 \quad \text{and} \quad \text{(dual),} \quad (3.1) \\ & \quad \quad \quad By + \text{Tr}(A_* Y) = c \end{aligned}$$

where \mathcal{S}^K denotes the space of symmetric matrices of size K . Here $c \in \mathbb{R}^P$ is a vector, $B \in (\mathbb{R}^n)^P$ is a rectangular matrix, and $A_* = (A_1, \dots, A_P) \in (\mathcal{S}^K)^P$ is a vector of matrices. The program (3.1) is called the *dual* program. The corresponding *primal* program is:

$$\begin{aligned} & \text{minimize } c^T x \quad \text{over } x \in \mathbb{R}^P \\ & \text{such that } X(x) := x^T A_* \succeq 0 \quad \text{and} \quad \text{(primal),} \quad (3.2) \\ & \quad \quad \quad B^T x = b \end{aligned}$$

where $x^T A_* \equiv \sum_{p=1}^P x_p A_p \in \mathcal{S}^K$.

A vector x is called *primal feasible* if the conditions in (3.2) are satisfied, even if optimality is not necessarily achieved. Similarly, a pair (y, Y) is *dual feasible* if it obeys the conditions in (3.1), without necessarily achieving optimality. The *duality gap* is the difference between the primal and dual objectives:

$$D(x, y) := c^T x - b^T y. \quad (3.3)$$

Positive semidefiniteness of X and Y implies that the duality gap is nonnegative when x is primal feasible and (y, Y) is dual feasible:

$$D(x, y) = \text{Tr}((x^T A_*) Y) = \text{Tr}(XY) \geq 0, \quad (3.4)$$

Let us denote the values at optimality of the primal and dual problems by (x_*, X_*) and (y_*, Y_*) , respectively. It is a standard result that, for a generic SDP, the duality gap *vanishes*:

$$c^T x_* = b^T y_* \quad (3.5)$$

This, in turn, implies the *complementarity condition*:

$$X_* Y_* = 0, \quad (3.6)$$

which follows from $\text{Tr}(X_* Y_*) = 0$ and positive semidefiniteness of X_* and Y_* .

3.2.1.2 Solution algorithm

In SDPB, SDPs are solved using a *primal-dual interior point algorithm*. This algorithm solves both the primal and dual problems simultaneously and approaches optimality from the interior of the cone of positive semidefinite matrices, meaning $X, Y \succ 0$ at every step. An efficient way to describe the algorithm is via the Lagrange function

$$L(x, y, X, Y) = c^T x + b^T y - x^T B y + \text{Tr}((X - x^T A_*)Y) - \mu \log \det X, \quad (3.7)$$

where μ is a parameter multiplying a logarithmic *barrier function* $-\log \det X$ that is meant to help X stay within the cone of positive semidefinite matrices. The stationarity equations of this Lagrange function with respect to x , y , and Y yield the primal and dual feasibility conditions, while stationarity with respect to X yields:

$$XY = \mu I, \quad (3.8)$$

which is a finite- μ version of the complementarity condition (3.6).

The *central path* is the μ -dependent line of stationary points of the Lagrange function (3.7). The solution to the original SDP is obtained by taking $\mu \downarrow 0$ along the central path. A standard Newton step (dx, dX, dy, dY) towards this central path can be computed by solving the linearized equations:²

²As explained in [1], the natural way to solve these equations leads to an update dY that is not manifestly symmetric. In practice Y is therefore updated with the symmetrized version of the dY computed from the Newton step equations. This will be left implicit in our notation.

$$\begin{aligned}
dx^T A_* - dX &= x^T A_* - X \\
B^T dx &= b - B^T x \\
Bdy + \text{Tr}(A_* dY) &= c - By - \text{Tr}(A_* Y) \\
dXY + XdY &= R := \mu I - XY.
\end{aligned} \tag{3.9}$$

These are solved in practice with the following procedure. First one solves the linear system:

$$\begin{pmatrix} S & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x + dx \\ y + dy \end{pmatrix} = \begin{pmatrix} By - c + \text{Tr}(A_* Y) + \mu \text{Tr}(A_* X^{-1}) \\ b \end{pmatrix}, \tag{3.10}$$

and then one can compute the updates:

$$\begin{aligned}
dX &= (x^T + dx^T)A_* - X \\
dY &= X^{-1}(R - dXY).
\end{aligned} \tag{3.11}$$

Here the matrix S is defined element-wise as $S_{ij} = \text{Tr}(A_i X^{-1} A_j Y)$. It is easily verified that this procedure is a simple rewriting of the system (3.9). One important problem is however that this linear system (generically) becomes singular at optimality. First of all, X and Y become singular because $XY = 0$ and $X, Y \succeq 0$, and S becomes singular because $x^T S x = \text{Tr}(XY) = 0$. In practice, the singularity of the linear system *at* optimality can cause serious numerical instabilities *near* optimality.

The finite- μ version of the problem has the advantage that X and Y are (again, generically) strictly positive definite even at optimality, since they now obey equation (3.8). This makes the Newton steps towards the modified complementarity condition (3.8) well-defined. But of course we would be solving the wrong problem if we kept μ fixed throughout the run. In practice one therefore gradually decreases μ . At every step one estimates a ‘current’ value of μ as

$$\mu_{\text{est}} = \text{Tr}(XY) / \dim(X), \tag{3.12}$$

and then one takes the above Newton step with

$$\mu \leftarrow \beta \mu_{\text{est}}, \tag{3.13}$$

with $\beta \in (0, 1)$ chosen by the user; a typical value is 0.3. In this way, one obtains a trajectory where each step is aimed towards an ever lower point along the central path.

The Newton step itself is not guaranteed to maintain positivity of the matrices X and Y . In practice one therefore chooses an update

$$\begin{aligned}(X, x) &\leftarrow (X, x) + \alpha_P(dX, dx), \\ (Y, y) &\leftarrow (Y, y) + \alpha_D(dY, dy),\end{aligned}\tag{3.14}$$

where α_P and α_D are chosen in $(0, 1]$ to ensure that X and Y (and therefore also S) remain inside the cone of positive semidefinite matrices.

The essential balance of this algorithm arises from the desire to decrease μ quickly while maintaining reasonable α_P and α_D . If α_P and α_D are too close to zero, then progress slows to a crawl. This can happen if the barrier function does not sufficiently punish points near the boundary of the cone. This issue can be mitigated by increasing μ , which here means choosing β closer to 1. However, then we will descend more slowly along the central path. A judicious choice of β at each step is therefore essential for success of the algorithm.

3.2.1.3 Corrector step

For any step length between 0 and 1, the Newton step is guaranteed to decrease the error in the linear equality constraints, but not necessarily in the non-linear equation (3.8). SDPB tries to mitigate this issue by applying a predictor-corrector trick [82] which works as follows.

One first computes the initial ‘‘predictor’’ step (dx_p, dy_p, dX_p, dY_p) as discussed above. If one were to simply take this step, the residue $R := \mu I - XY$ would become $-dX_p dY_p$, whereas we would like it to be zero. To improve this state of affairs somewhat, we compute another ‘‘corrector’’ step (dx_c, dy_c, dX_c, dY_c) which is obtained by replacing the residue R in the last equation in (3.9) by

$$R \rightarrow \mu_c I - XY - dX_p dY_p\tag{3.15}$$

and keeping the other equations the same. Then one updates the original variables with the corrector step, using the step length choice discussed above. Notice that in (3.15) we allowed for a new value μ_c ; SDPB’s choice of μ_c is modeled on that of SDPA [83, 84, 85] and can be found in detail in [1].

We note that it is comparatively cheap to compute the corrector step once the predictor step has been computed. This is because both computations reuse the same matrices S and $Q = B^T S^{-1} B$, whose formation and Cholesky factorization are the most expensive operations in the solver.

3.2.1.4 Aside: iterated corrector step

SDPB, following SDPA, simply takes one predictor-corrector step for each iteration of the algorithm. We however found that it might be more efficient to repeat the corrector step multiple times, even for standard SDPB runs.

Considering a general setup where we search for the stationary point of a function $L(x)$, if we write the standard Newton step (with unit step length for simplicity) as:

$$x_{k+1} = x_k - H_{xx}^{-1}(x_k) \cdot \nabla_x L(x_k) \quad (3.16)$$

then the added corrector step would simply be:

$$x_{k+2} = x_{k+1} - H_{xx}^{-1}(x_k) \cdot \nabla_x L(x_{k+1}), \quad (3.17)$$

which importantly “recycles” the old Hessian matrix. Repeating the corrector step multiple times (using $H_{xx}^{-1}(x_k)$ for every step) is therefore akin to simple gradient descent, and we can at best expect linear convergence as opposed to the superlinear convergence of proper Newton steps. But computationally speaking it might very well be worthwhile if the computation of H_{xx}^{-1} is expensive, which happens to be the case for our SDPs.

After some experimentation we found that significant speedups could indeed be obtained using an iterative-corrector algorithm for SDPB. We propose a minimal modification of the SDPB algorithm as outlined in algorithm 1. Each update (dx, dy, dX, dY) is now computed in two stages: the first stage is the usual predictor step and the second stage is to compute the corrector step iteratively with some general convergence criterion and a limit on the maximum number of steps.

We have explored two types of convergence criterion. The first strategy uses the residue R . We consider a solution to be converging if R continues to decrease with each iteration. The second strategy uses the step length parameters $\min(\alpha_P, \alpha_D)$ as an indicator of convergence. As long as $\min(\alpha_P, \alpha_D)$ is increasing, we consider the solution to be converging. Both the residue R and the step length parameters can function as convergence criteria, but the former has lower computational cost.

In general, we have observed that the iterative corrector steps converge efficiently when the initial point ξ_0 is sufficiently close to the true solution for a

Algorithm 1: The iterative corrector step. `PredictorStep`, `CorrectorStep`: standard Interior Point Method subroutines.

```

Input  $\xi_0 = (x_0, y_0, X_0, Y_0)$ ;
Compute  $d\xi_p \leftarrow \text{PredictorStep}(R)$ ;
Initialize  $d\xi_c = d\xi_p$ ;
while (convergence criterion) & (it  $\leq$  MAX) do
    | Store  $d\xi_c \leftarrow d\xi_c$ ;
    |  $R_c \leftarrow \mu_c I - X_0 Y_0 - dX_c dY_c$ ;
    | Update  $d\xi_c \leftarrow \text{CorrectorStep}(R_c)$ ;
    | it  $\leftarrow$  it + 1;
end
return  $d\xi_c$ ;

```

fixed μ . We have observed linear convergence in actual SDPB runs, in agreement with the above general analysis. It is worth mentioning that there is potential to enhance efficiency by reducing μ_c during each iterative step.³ We leave the problems of analyzing the convergence rate with varying μ , establishing criteria for how fast to decrease μ , and determining when to stop the corrector iterations for future work.

3.2.2 Semidefinite programs with parameters

As described in the introduction, besides the internal SDP variables

$$\xi := (x, y, X, Y), \quad (3.18)$$

the problems of interest depend on additional external parameters p taking values in a parameter space \mathcal{P} . We will assume that the vector of matrices A_* does not depend on p , as is adequate for most conformal bootstrap problems, but the other ingredients in the SDP do depend on p :

$$(b, c, B) \rightarrow (b(p), c(p), B(p)). \quad (3.19)$$

We furthermore assume that this dependence on p is smooth. Note that this does not imply that the objectives are smooth; see [12] for an example where the objective is not C^2 , even though the inputs are C^∞ .

We consider two types of optimization problems over \mathcal{P} . For fixed p , let us denote the optimal value of ξ by $\xi_*(p)$. We furthermore define the objective

³We have done some preliminary exploration in this direction. In experiments, it was sometimes possible to decrease the duality gap from 10^{-15} to 10^{-30} using only corrector iterations (without ever recomputing the Hessian).

at optimality by

$$\text{obj}(p) := c^T(p)x_*(p) = b^T(p)y_*(p). \quad (3.20)$$

The first type of problem is a simple minimization:

$$\text{minimize } \text{obj}(p) \text{ over } p \in \mathcal{P}. \quad (3.21)$$

The second type is a constrained optimization with a linear objective:

$$\begin{aligned} &\text{maximize } v^T p \text{ over } p \in \mathcal{P} \\ &\text{such that } \text{obj}(p) \leq 0. \end{aligned} \quad (3.22)$$

Both types of problem arise naturally in the navigator approach to the conformal bootstrap [12]. The first type is used to find a (primal) feasible point in \mathcal{P} or, in bootstrap terminology, a point inside a bootstrap “island”. (For this application, the algorithm can actually terminate as soon as a point with $\text{obj}(p) < 0$ is found.) The second type is used to map out the extremal points of bootstrap islands, and consequently provide rigorous upper or lower bounds on the parameters in \mathcal{P} . For this application, it suffices to consider only linear functions $v^T p$ on \mathcal{P} , but the formalism we discuss can straightforwardly be adapted to non-linear objectives.

Both types of problem fall into the category of *non-linear semidefinite programs*, where one optimizes a function $f(p, x)$ subject to a positive semidefiniteness constraint $X(p, x) \succeq 0$, with $X(\cdot, \cdot)$ a non-affine function. The fundamentals of such programs were analyzed in 1997 [86]. To date several solution methods have been proposed, see for example [87, 88], and we are aware of one publicly available solver [89]. A survey of existing numerical methods can be found in [90].

Compared to a general non-linear SDP, our case is special for two reasons. First, our SDPs have additional equality constraints $B^T(p)x = b(p)$, which implies the existence of free variables y on the dual side. Second, our function $X(\cdot, \cdot)$ is non-linear only in the p variables and typically there are significantly fewer of these (about 10-20) than the dimension of x (a few thousand). The numerical experiments reported in this work only pertain to these “mildly non-linear” semidefinite programs.

3.2.2.1 Lagrange function and Newton step

To set up an interior point method for a parameter-dependent SDP, it is natural to start from the same Lagrange function (3.7) as before, now viewed as a function of both p and ξ :

$$L_\mu(\xi, p) = c^T(p)x + b^T(p)y - x^T B(p)y + \text{Tr}((X - x^T A_*)Y) - \mu \log \det X, \quad (3.23)$$

The Newton steps stemming from this function (at finite μ) are easily obtained, but differ slightly for the two types of problem discussed above. The first type just needs stationarity of the Lagrange function:

$$\nabla_\xi L_\mu = 0 \text{ and } \nabla_p L_\mu = 0, \quad (3.24)$$

and the step is therefore computed from:

$$\begin{aligned} H_{pp}\delta p + H_{p\xi}\delta\xi &= -\nabla_p L_\mu \\ H_{\xi p}\delta p + H_{\xi\xi}\delta\xi &= -\nabla_\xi L_\mu. \end{aligned} \quad (3.25)$$

Here, H_{pp} , $H_{p\xi}$, $H_{\xi p}$, $H_{\xi\xi}$ denote schematically the components of the Hessian matrix of second derivatives of the Lagrange function.

An important difference compared to the fixed- p case is that the steps produced by (3.25) are no longer guaranteed to step towards a *minimum* of the Lagrangian. Indeed, suppose we have solved the internal component of the problem and therefore $\nabla_\xi L_\mu = 0$. Then we can eliminate $d\xi$ to find:

$$\delta L_\mu = -\frac{1}{2}\delta p^T (H_{pp} - H_{px}H_{xx}^{-1}H_{xp}) \delta p + \dots \quad (3.26)$$

In this case, we are guaranteed to make progress in the right direction if:

$$H_{pp} - H_{px}H_{xx}^{-1}H_{xp} \succ 0. \quad (3.27)$$

However, this constraint is not automatically satisfied, since we do not assume any particular structure for the hessian and gradient in \mathcal{P} space. If (3.27) is not satisfied, then the naïve Newton step might aim for a local saddle point or maximum rather than a minimum. A common method for dealing with this problem is to replace the non-positive-definite (reduced) Hessian (3.27) with a positive-definite one. We will outline our procedure for performing this replacement in section 3.3.

For the second type of problem, we would like to reach a point with:

$$\nabla_{\xi} L_{\mu} = 0 \text{ and } \lambda v = \nabla_p L_{\mu} \text{ and } L_{\mu} = 0 \text{ and } \lambda > 0, \quad (3.28)$$

where λ is a (scalar) Lagrange multiplier, whose sign is constrained by demanding that we maximize $v^T p$ subject to $L_{\mu} \leq 0$. In [12] it was proposed to take the following step:

$$\begin{aligned} \frac{1}{2} (\delta p H_{pp} \delta p + \delta p H_{p\xi} \delta \xi + \delta \xi H_{\xi p} \delta p + \delta \xi H_{\xi\xi} \delta \xi) + \\ (\nabla_p L_{\mu})^T \delta p + (\nabla_{\xi} L_{\mu})^T \delta \xi = -L_{\mu}, \\ H_{\xi p} \delta p + H_{\xi\xi} \delta \xi = -\nabla_{\xi} L_{\mu}, \\ H_{pp} \delta p + H_{p\xi} \delta \xi = -\nabla_p L_{\mu} + \lambda v. \end{aligned} \quad (3.29)$$

The quadratic term on the first line is not present in a regular Newton step. We added it because it is expected to give a more accurate step at negligible computational cost (since it amounts to solving only a single quadratic equation).

It is again useful to consider the case $\nabla_{\xi} L_{\mu} = 0$ and eliminate $\delta \xi$. The first equation in (3.29) then becomes:

$$\frac{1}{2} \delta p (H_{pp} - H_{p\xi} H_{\xi\xi}^{-1} H_{\xi p}) \delta p + (\nabla_p L_{\mu})^T \delta p = -L_{\mu}. \quad (3.30)$$

The step will always head towards a zero of this quadratic model. (If there is no zero in sight, *i.e.*, if the solution to the quadratic equation (3.30) in δp is complex, then we extremize instead by taking its real part.) The other equations can be combined to give:

$$\lambda v^T \delta p = \frac{1}{2} \delta p^T (H_{pp} - H_{p\xi} H_{\xi\xi}^{-1} H_{\xi p}) \delta p - L_{\mu}. \quad (3.31)$$

Now suppose L_{μ} were negative. Then we should make progress towards maximizing the objective, so we would like to have $v^T \delta p > 0$. But the right-hand side of the above equation is only guaranteed to be positive if:

$$H_{pp} - H_{p\xi} H_{\xi\xi}^{-1} H_{\xi p} \succ 0, \quad (3.32)$$

which is the same constraint we found for the first type of problem. To avoid stepping in an incorrect direction, we should therefore once again modify this matrix to ensure its positivity.

3.2.2.2 Newton steps and stalling

The above equations lead naturally to an interior-point algorithm based on Newton's method. As in the usual algorithm, we first estimate $\mu_{\text{est}} = \text{Tr}(XY) / \dim(X)$ and set the target $\mu \rightarrow \beta\mu_{\text{est}}$. Then we substitute this value of μ to compute a Newton-like step $(\delta\xi, \delta p)$ from equations (3.25) or (3.29). We then update

$$\xi \leftarrow \xi + \alpha\delta\xi, \quad p \leftarrow p + \alpha\delta p, \quad (3.33)$$

where $\alpha = \min(\alpha_P, \alpha_D)$ for dp .

In our initial investigations, we implemented essentially this algorithm (although we estimated $H_{pp} - H_{p\xi}H_{\xi\xi}^{-1}H_{\xi p}$ with a BFGS approach). Unfortunately, it did not perform as well as we hoped. The main issue we encountered was that of *stalling*. Concretely we found that, in the course of running the algorithm, α_P and α_D eventually become too small to produce a meaningful step in parameter space.

As we discussed briefly at the end of section 3.2.1, small values of α_P and/or α_D indicate that the naïve Newton step brings X and/or Y far outside the cone of positive semidefinite matrices. This in turn means that the barrier function $-\mu \log \det(X)$ in (3.23) is not sufficiently strong to steer the step towards the interior of the positive cone. It is therefore natural to try to recover from stalling by increasing μ . Although we found that acceptable step sizes can indeed be obtained in that way, doing so is clearly disadvantageous because we would ultimately like to send $\mu \rightarrow 0$.

These results highlight a basic tension in this algorithm (and in fact in the use of barrier methods more generally). If one decreases μ too quickly, one ends up in a stalled situation with infinitesimal step length, and one needs to backtrack by increasing μ . One can avoid hitting the boundary of the cone by decreasing μ more slowly, but then progress towards the objective might be slow. Although we cannot offer a first-principles derivation of the optimal way to decrease μ and move in parameter space, in the following we provide a somewhat different algorithm that works well in practice.

Finally let us note that the stalling phenomenon is well-known in other scenarios beyond ours. For example it can occur in standard runs of SDPB where β is chosen too close to 0 so that μ is decreased too quickly. Stalling can also occur in *warm-starting*, where one starts SDPB from a checkpoint or even an optimal

solution of an already completed run with similar parameters. In both cases, the problem can be solved in principle by increasing μ , but often at significant computational cost.

3.3 Skydiving algorithm

Our main algorithm was developed through trial and error while exploring several representative numerical conformal bootstrap problems. Its principal advantage is that it appears to resolve the stalling problems that plagued naïve implementations of the combined Newton step (3.33).

Let us first introduce some terminology which we illustrate in figure 3.2.

We define the *central section* $\xi_\mu^*(p)$ as the optimal point in ξ -space for a given (p, μ) . In figure 3.2, the central section is a two-dimensional plane. To avoid clutter, we show it with three solid black lines indicating the intersection of the central section with the three fixed- μ surfaces. A physicist might say that ξ is “on-shell” along the central section, as it satisfies the stationary condition $\nabla_\xi L_\mu = 0$, while p and μ are fixed.

Moving along the central section, we can furthermore optimize over p , still keeping μ fixed. Let the optimal value of p at fixed μ be p_μ^* . The values $(\xi_\mu^*(p_\mu^*), p_\mu^*)$ then trace out a 1-dimensional curve in (ξ, p) -space as a function of μ that we call the *global central path*. The global central path is shown in figure 3.2 as the black curve interpolating through the fixed μ surfaces. The optimal point, in both p - and ξ -space, is reached by sending $\mu \downarrow 0$ along the global central path.

It is also useful to consider a standard primal-dual interior point (PDIP) run in the context of figure 3.2. This would correspond to holding p fixed and moving along the shaded straight vertical plane. The intersection of this plane with the central section is conventionally called the “central path” in the SDP literature. In our context, we call it a “local central path”.

The most important conceptual feature of our algorithm is that it prioritizes proximity to the central section, in the sense that it generically only changes p and μ if the current value $\xi_\mu(p)$ is sufficiently close to $\xi_\mu^*(p)$. The algorithm therefore incorporates two types of steps: the first are so-called *centering* steps where p and μ are held fixed and ξ is updated to approach the central section. The second type are full Newton steps, where the aim is to update both ξ and p . These Newton steps are more delicate, both because they are sensitive to

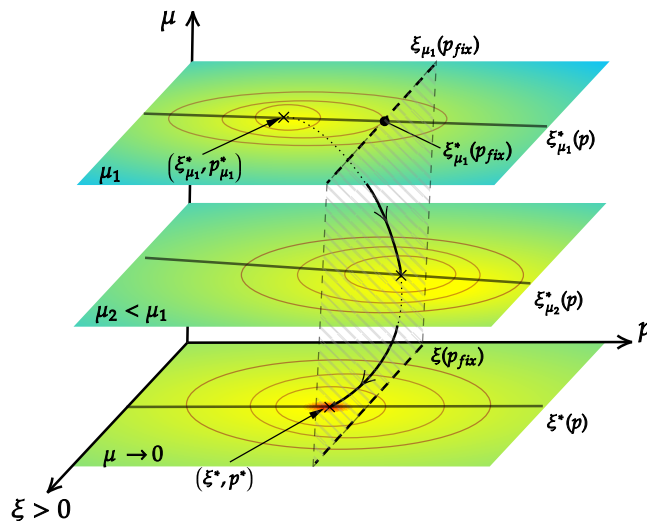


Figure 3.2: Visualization of central section $\xi_\mu(p)$, optimal solution of standard PDIP run $\xi(p_{fix})$ and global central path $\xi_\mu^*(p_\mu^*)$. A contour map is sketched for the Lagrangian $L_\mu(\xi, p)$ for different values of μ . The usual fixed- p SDPB solves the problem on the shaded vertical surface, while the skydiving algorithm prioritizes finding the global central path, shown as a curved path traveling through the horizontal surfaces.

the non-linearity in \mathcal{P} and because μ cannot be decreased too quickly. To avoid stalling, we therefore introduce a new *scanning* technique that we describe in the next subsection.

As we outlined in the introduction, the original navigator approach of [12] solves an SDP all the way to optimality, so $\mu \downarrow 0$, before deciding which step to take in \mathcal{P} . With the skydiving approach we instead work at finite μ . The larger we take μ , however, the more our picture of the navigator function over \mathcal{P} risks being distorted. This implies that our steps in the parameter space risk becoming meaningless if we work at very large μ . For this reason we do not immediately start stepping around in \mathcal{P} . Instead, we initiate our algorithm with a single standard run of the primal-dual interior point (PDIP) algorithm described in the previous section, updating both ξ and μ until the duality gap is sufficiently small (but not zero). Only then do we enter the main component of the algorithm.

The broad-strokes algorithm is sketched in algorithm 2. In words, while the solution has not converged, we first follow the standard PDIP algorithm to update ξ until $\mu_{est} = \text{Tr}(XY)/\text{dim}(X)$ has decreased enough. On the first

Algorithm 2: Skydiving.

```

initialize  $(\xi, p)$ ;
while not converged do
  while duality gap  $> dg_{thr}$  do
    | update  $\xi$  using PDIP step
  end
  while  $\|XY - \mu I\|_{\max} > R_{thr}$  do
    | update  $\xi$  using centering step
  end
  update  $(\xi, p)$  from scanning step
end

```

iteration this typically takes 100-200 PDIP steps, whereas in later iterations the duality gap tends to stay below dg_{thr} . We then enter the second while loop where ξ is updated by a few centering steps until R is small enough. At the end of this loop, $\xi_\mu(p)$ is brought sufficiently close to $\xi_\mu^*(p)$ so that we declare ourselves to (effectively) be at $\xi_\mu^*(p)$. Finally, we take a scanning step to update both p and ξ .

Of course proximity to the global central path also requires that ξ be both primal and dual feasible, i.e. that the linear equality constraints in (3.2) and (3.1) are satisfied. In practice, these constraints are usually obeyed once $R \ll 1$. The reason is that often the solver can take a centering step with step length 1 — i.e. a full Newton step. A full Newton step will exactly solve the primal and dual equality constraints, since they are linear equations. Further centering steps will then stay on the primal and dual feasible locus.⁴

We now elaborate more on the centering step, deferring details of the scanning step to the next subsection. A centering step is the same as a PDIP step except that β is set to 1 so the target value of μ is the same as μ_{est} . Of course μ_{est} can vary slightly from one centering step to the next, due to the non-linearity of the problem, but we found these changes to be immaterial. In our implementation of centering steps, we adapt the iterative corrector procedure described in section 3.2.1.4. In particular, we choose the convergence criterion

⁴If, for some reason, R happened to be small *without* solving the primal dual constraints (for example due to warm-starting with a checkpoint with small R), then R will typically become large again in subsequent iterations if they still don't solve the primal dual constraints. Therefore testing $R \ll 1$ usually suffices. A more robust implementation might test the primal and dual constraints as well.

in Algorithm 1 to be

$$\text{convergence criterion} = (\|R\|_{\max} > R_{\text{thr}}) \ \& \ (\|R\|_{\max} \text{ is decreasing}). \quad (3.34)$$

Here, R_{thr} is the same parameter as in algorithm 2. A typical value for the maximum number of corrector iterations **MAX** is 5.

Figure 3.3 shows an example skydiving run for the 3d Ising mixed correlator problem described below in section 3.5.1. This is a one-dimensional search over $p = \Delta_\sigma$, with fixed Δ_ϵ and $\lambda_{\sigma\sigma\epsilon}/\lambda_{\epsilon\epsilon\epsilon}$. The figure compares the skydiving path (connected lines) with the global central path (individual dots) in the (duality gap)- p plane. (The global central path is obtained by solving equation (3.23) at fixed μ .) We see that, despite early fluctuations in the skydiving path, it in general follows the global central path and eventually converges to it. We observed in experiments that without centering steps, the skydiving path often predicts wrong steps in p so that it deviates from the global central path and eventually stalls.

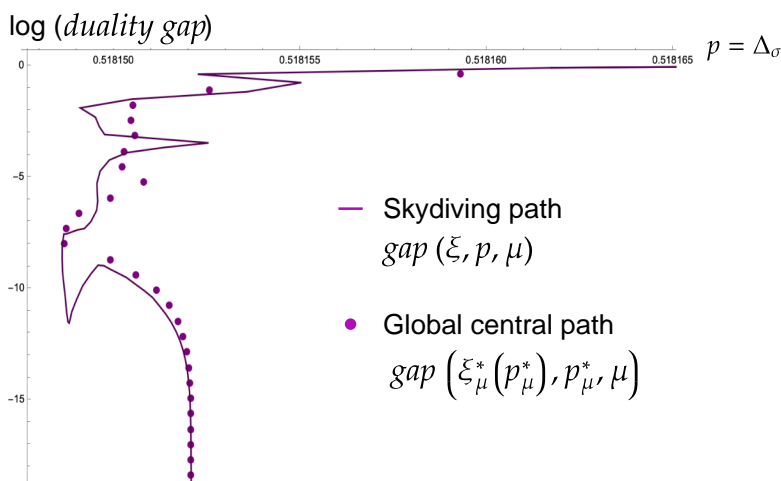


Figure 3.3: An illustration of the proximity between the solver path (ξ, p, μ) and the global central path $(\xi_\mu^*(p_\mu^*), p_\mu^*, \mu)$ in the (duality gap)- p plane for a 3d Ising mixed correlator problem. The reader might be surprised at the rather erratic behavior of the global central path when $\log(\text{duality gap}) > -8$. This is due to the unforgiving nature of the navigator function itself for this problem. As we show below in Figure 3.5 it sometimes has two local minima and also contains a nearly flat region.

3.3.1 Scanning

As described above, once a target value of μ has been set we can solve the linearized equations to obtain a step $(d\xi, dp)$. But is this a good step or are we aiming “too low” and putting ourselves at risk of stalling? Balancing efficiency and robustness, the scanning algorithm serves to find a good step towards the lowest reasonable target value of μ . During scanning, we use the step length

$$\alpha := \min(\alpha_P, \alpha_D) \quad (3.35)$$

as a proxy for whether a step is good or not. The procedure is as follows:

1. As in the PDIP algorithm, the solver begins by choosing a reduction factor β_{\min} . We calculate a step $(d\xi, dp)$ and the corresponding α based on the target value

$$\mu = \beta_{\min} \mu_{\text{est}}. \quad (3.36)$$

We accept this step if α is big enough, that is if $\alpha > \alpha_{\text{thr.max}}$ for some user-defined $\alpha_{\text{thr.max}}$.

2. If α is not big enough, we gradually increase β with step size $\Delta\beta$ up to a value β_{\max} (which is typically 1). In other words we use targets:

$$\mu_i = \beta_i \mu_{\text{est}}, \quad \beta_{\min} \leq \beta_1 < \beta_2 < \dots \leq \beta_{\max}, \quad (3.37)$$

to compute the i 'th step $(d\xi_i, dp_i)$. Again, we accept a step as soon as $\alpha > \alpha_{\text{thr.max}}$. Note that the computation of these steps is cheap because we can recycle the S and Q matrices, just as in the predictor-corrector step.

3. If no step is big enough even after reaching β_{\max} , we decide to tolerate a smaller but not yet problematic step size. To do so, we store the best step during the search (which does not necessarily occur at β_{\max}), and accept it if the corresponding α_{best} is bigger than some user-defined parameter $\alpha_{\text{thr.min}}$. Clearly, this part of the algorithm is only used if $\alpha_{\text{thr.min}} < \alpha_{\text{thr.max}}$.
4. If α is still too small, that is, $\alpha_{\text{best}} < \alpha_{\text{thr.min}}$, then we are at risk of stalling. Based on our experiments, this situation do not occur often, but when it does, we need a recovery mechanism. Our recovery subroutine foregoes the update on p and instead opts for a PDIP step with $\beta =$

$\beta_{\text{climbing}} > 1$ that only updates ξ . Another attempt to update p will be made in the next round. In practice, we typically set $\beta_{\text{climbing}} = 2$.

This scanning procedure is outlined in algorithm 3.⁵

Algorithm 3: The scanning step to compute $d\xi$ and dp .

```

 $\beta \leftarrow \beta_{\min};$ 
 $\alpha_{\text{best}} = 0;$ 
while  $\beta < \beta_{\max}$  do
  compute  $(d\xi, dp)$  from Newton step with target  $\mu = \beta\mu_{\text{est}};$ 
   $\alpha \leftarrow \alpha(d\xi);$ 
  if  $\alpha > \alpha_{\text{thr.max}}$  then
    | return step  $(\alpha d\xi, \alpha dp);$ 
  end
  if  $\alpha > \alpha_{\text{best}}$  then
    |  $d\xi_{\text{best}} \leftarrow d\xi;$ 
    |  $\alpha_{\text{best}} \leftarrow \alpha;$ 
  end
   $\beta \leftarrow \beta + \Delta\beta;$ 
end
if  $\alpha_{\text{best}} > \alpha_{\text{thr.min}}$  then
  | return step  $(\alpha_{\text{best}}d\xi_{\text{best}}, \alpha_{\text{best}}dp_{\text{best}});$ 
end
compute  $d\xi$  using PDIP step with  $\beta = \beta_{\text{climbing}};$ 
return step  $(d\xi, dp = 0);$ 

```

Naively, the gradient $\nabla_p L$ and Hessian H_{pp} can be computed with finite differences. However, this is particularly expensive for the Hessian, which in an n -dimensional parameter space \mathcal{P} would require computing $(b(p), c(p), B(p))$ for a constellation of $n(n+1)/2$ points around the actual value of p .⁶ In practice, we therefore use the BFGS algorithm discussed below to approximate the Hessian.

Note that the algorithm above currently does not tread “carefully” in the parameter space \mathcal{P} , and it is possible for it to wander into problematic regions

⁵If the computed step length α is less than 1 then the step returned by this algorithm would bring us exactly to the point where a primal or dual matrix becomes singular. The issues this may cause are avoided by applying the same step length reduction calculations as in standard SDPB [1].

⁶By a “constellation”, we mean a collection of SDPs at several nearby points $\{p_0, p_0 + \epsilon e_1, \dots, p_0 + \epsilon e_n\}$ where e_i is a basis-vector in the i -th direction and ϵ is a small parameter.

and/or exhibit runaway behavior. One way to remedy runaway behavior would be to include a line search along the computed step $(d\xi, dp)$. Another possibility is to introduce a bounding box in \mathcal{P} -space. In our experiments, these improvements were not needed, but they might be necessary in future computations.

As with any non-convex optimization, our algorithm is not guaranteed to converge. For example, the algorithm might never find a suitable step and continue to increase μ . In this case, centering steps would still take us toward the central section. However, as we explained before, at large μ we would not have a reliable picture of the navigator function, and any calculated step could be meaningless. A practical way to address this issue is to introduce an upper limit on the duality gap, beyond which the solver will no longer increase μ .

The PDIP step in algorithm 3 includes a single corrector step, just like the scanning step discussed above. A similar corrector step is implemented whenever $H_{\xi\xi}^{-1}$ was used, for example for $H_{\xi\xi}^{-1}\nabla_p L$. The iterated corrector procedure described in section 3.2.1.4 is invoked during centering steps only.

Finally, let us note that $\alpha_{\text{thr_max}}$ should not be set to 1 (typically we set it to 0.6). The reason is that when the algorithm is very close to the global central path (at finite μ), it would simply compute that the step $(d\xi, dp) \approx 0$ if it chooses $\beta = 1$ and then take an infinitesimal step with step size $\alpha = 1$. Of course there are many ways to avoid such trivial stalling behavior, but choosing $\alpha_{\text{thr_max}}$ strictly below 1 works well in practice. By choosing $\alpha_{\text{thr_max}}$ closer to 1, we encourage the algorithm to stay closer to the global central path.

3.3.2 Modified BFGS algorithm

For the Newton step in the scanning iterations, we need various gradients and Hessians with respect to p . This requires us to compute the p -dependence of the input data b , c , and B , as exemplified by the computation of $\nabla_p L$, which we recall is given by:

$$dp^T \nabla_p L = dc^T x + db^T y - x^T dB y. \quad (3.38)$$

We currently estimate dB , db , and dc using finite differences, which is costly but acceptable. However, to compute H_{pp}

$$dp^T H_{pp} dp = d^2 c^T x + d^2 b^T y - x^T d^2 B y, \quad (3.39)$$

the cost becomes even less favorable as it scales quadratically with n , the dimension of \mathcal{P} . We therefore use a BFGS-type approximation to avoid direct computation of H_{pp} . In this subsection, we explain the details.

3.3.2.1 Review of standard BFGS

The core idea behind the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is to estimate the Hessian matrix using information about the gradient. To briefly review the procedure, let us consider the optimization of a non-linear function $f(x)$. A (quasi-)Newton method gives a sequence of points x_k , and at each point we have a gradient $\nabla f(x_k)$. Let us construct the differences

$$\begin{aligned} s_k &= x_k - x_{k-1}, \\ y_k &= \nabla f(x_k) - \nabla f(x_{k-1}). \end{aligned} \quad (3.40)$$

The pair (s, y) is insufficient to determine the hessian matrix H . Instead, we have a vectorial equation that constrains H :

$$Hs = y. \quad (3.41)$$

This is called the *secant* equation; it would be exact for infinitesimal step size. BFGS now opts to update a previous estimate H_{k-1} by a linear combination of $y_k y_k^T$ and $s_k s_k^T$, with coefficients determined such that the secant equation holds. This produces

$$H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k}. \quad (3.42)$$

With this estimate the new step is $s_{k+1} = -H_k^{-1} \nabla f(x_k)$. (In practice, one can directly calculate an update for the *inverse* Hessian, see for example [91], but for our purposes this difference is unimportant.)

3.3.2.2 A BFGS-type update for the Hessian

The main difference between our setup and the standard BFGS setup is that we need to estimate only one block of the Hessian matrix. Consider two points (ξ_1, p_1) and (ξ_2, p_2) and suppose we know $\nabla_p L(\xi_1, p_1)$ and $\nabla_p L(\xi_2, p_2)$, the full secant equation reads:

$$\begin{pmatrix} H_{\xi\xi} & H_{\xi p} \\ H_{p\xi} & H_{pp} \end{pmatrix} \begin{pmatrix} p_2 - p_1 \\ \xi_2 - \xi_1 \end{pmatrix} = \begin{pmatrix} \nabla_p L(\xi_2, p_2) - \nabla_p L(\xi_1, p_1) \\ \nabla_\xi L(\xi_2, p_2) - \nabla_\xi L(\xi_1, p_1) \end{pmatrix}. \quad (3.43)$$

Given that we can easily compute $H_{\xi\xi}$, as well as $H_{\xi p}$ and $H_{p\xi}$ using (3.38), only H_{pp} needs to be estimated, and there are two natural ways of doing so.

Estimating H_{pp} directly In the first method, we write a separate secant equation for H_{pp} and use it to calculate a BFGS update. Such a secant equation must be of the form:

$$H_{pp} \cdot (p_2 - p_1) = \nabla_p L(\xi_A, p_2) - \nabla_p L(\xi_A, p_1), \quad (3.44)$$

where A can be either 1 or 2, but it is essential that ξ_A is the *same* for both gradients. Indeed, in the limit where (ξ_2, p_2) and (ξ_1, p_1) are infinitesimally close, say with a distance ϵ , picking ξ_2 for the first gradient and ξ_1 for the second gradient leads to an error of order ϵ on the right-hand side, and therefore an error of order 1 for the estimate of H_{pp} . In contrast, the distinction between A being either 1 or 2 is one order higher in ϵ . So to implement this method we need to compute either $\nabla_p L(p_2, \xi_1)$ or $\nabla_p L(p_1, \xi_2)$, which can be done using the gradient formula (3.38).

Eliminating components A second method is to use our exact knowledge of $H_{\xi p}$ and $H_{\xi\xi}$ to eliminate the irrelevant components from equation (3.43). We then get:

$$N_{pp} \cdot (p_2 - p_1) = -\nabla_p L(\xi_2, p_2) + \nabla_p L(\xi_1, p_1) + H_{p\xi} H_{\xi\xi}^{-1} (\nabla_\xi L(\xi_2, p_2) - \nabla_\xi L(\xi_1, p_1)), \quad (3.45)$$

where we introduced:

$$N_{pp} = H_{pp} - H_{p\xi} H_{\xi\xi}^{-1} H_{\xi p}. \quad (3.46)$$

and on the right-hand side of equation (3.45) we are omitting the dependence on (ξ, p) of the Hessian components because these give rise to subleading effects.

We can now view equation (3.45) as a separate secant equation, and use a BFGS update for N_{pp} directly from this equation. This is *almost* consistent, but one wrinkle needs to be ironed out: over the course of the algorithm we are actually minimizing slightly *different* Lagrangians because μ is decreasing! Re-instating this μ -dependence produces:

$$N_{pp} \cdot (p_2 - p_1) = -\nabla_p L(\xi_2, p_2) + \nabla_p L(\xi_1, p_1) + H_{p\xi} H_{\xi\xi}^{-1} (\nabla_\xi L(p_2, \xi_2, \mu_A) - \nabla_\xi L(p_1, \xi_1, \mu_A)), \quad (3.47)$$

where we ignored the μ -dependence in $H_{\xi\xi}$ which again gives rise to higher-order effects. (We recall that $\nabla_p L$ does not explicitly depend on μ , which is also why the previous method did not have this wrinkle.) We now need to choose a reference value μ_A . To find an efficient value it is worth looking at the specifics of our algorithm to see which gradients are known. Let us take ξ_1, ξ_2 to be the values obtained *after* the centering iterations. Supposing that these iterations succeeded perfectly, we can write:

$$\nabla_\xi L(\xi_1, p_1, \mu_1^{\text{est}}) = 0, \quad (3.48)$$

$$\nabla_\xi L(\xi_2, p_2, \mu_2^{\text{est}}) = 0, \quad (3.49)$$

where we recall that $\mu^{\text{est}} = \text{Tr}(XY)/\dim(X)$ is the estimated value of μ at a point ξ . Furthermore, in the course of the algorithm we had to compute all the gradients that appear on the right-hand side of the Newton step. Among these we have $\nabla_p L(\xi_1, p_1)$ and $\nabla_p L(\xi_2, p_2)$, but also $\nabla_\xi L(\xi_1, p_1, \mu_2)$ for some target value μ_2 determined in the scanning step. The Lagrangian is however linear in μ , so using (3.48) we can immediately find one more gradient:

$$\nabla_\xi L(\xi_1, p_1, \mu_2^{\text{est}}) = \frac{\mu_1 - \mu_2^{\text{est}}}{\mu_1 - \mu_2} \nabla_\xi L(\xi_1, p_1, \mu_2). \quad (3.50)$$

The fraction on the right-hand side, in turn, is roughly equal to the step length α_1 :

$$\frac{\mu_1 - \mu_2^{\text{est}}}{\mu_1 - \mu_2} \approx \alpha_1, \quad (3.51)$$

since both sides indicate (at a linearized level) how much progress was made towards the target μ_2 starting from (ξ_1, p_1) . It therefore makes the most sense to evaluate the secant equation for μ_2^{est} , where we obtain:

$$N_{pp} \cdot (p_2 - p_1) \approx -\nabla_p L(\xi_2, p_2) + \nabla_p L(\xi_1, p_1) - \alpha H_{p\xi} H_{\xi\xi}^{-1} \nabla_\xi L(p_1, \xi_1, \mu_2) \quad (3.52)$$

and all the gradients on the right-hand side are known. The BFGS update for N_{pp} based on this secant equation is the algorithm currently implemented in our code.

We stress that this update procedure is technically only applicable if the centering steps are *complete*, in the sense that any residuals in (3.48) translate directly into a mismatch of the update. Our core algorithm, however, does not require this: the scanning step is OK even if the centering steps are reduced or completely omitted.

3.3.2.3 Ensuring positivity of the Hessian

As discussed in section 3.2.2.1, positivity of $N_{pp} = H_{pp} - H_{p\xi}H_{\xi\xi}^{-1}H_{\xi p}$ would guarantee progress toward optimality.⁷ However, positivity is not a-priori guaranteed, and we need a strategy to ensure it. In our implementation, we choose to update our approximation to N_{pp} (using the BFGS-type update discussed above) only when the new N_{pp} would be positive. If positivity would be violated, we continue using the old (non-updated) Hessian.

The user can initialize the Hessian either by specifying it explicitly or using the default value

$$N_{pp}^{\text{default}} = -H_{p\xi}H_{\xi\xi}^{-1}H_{\xi p}. \quad (3.53)$$

This default value generally has the advantage of being similar in scale to the exact N_{pp} .

3.3.3 Shifting the Lagrange function

We need a further modification of our algorithm for the second type of optimization (3.22), which searches for an extremal point of the feasible region in parameter space \mathcal{P} . In such a search, the goal is to find a point where the navigator function *vanishes*:

$$\lim_{\mu \downarrow 0} L_\mu = 0. \quad (3.54)$$

A natural question is: how should we interpret this equation at finite μ ? In other words, when we work at $\mu > 0$, what is our best guess for the zero locus given by equation (3.54)? For simplicity, in (3.28) and (3.29), we used the simplest finite- μ version of this equation, namely $L_\mu = 0$. However, this is not necessarily the best choice.

⁷This also provides an argument in favor of using the second method for estimating the Hessian of the previous subsection. Namely, if we were to augment our scanning steps with a line search then we can use the standard Wolfe termination conditions. At the moment this line search is not yet implemented.

For the following discussion it is useful to suppose that we are “on-shell” in the ξ variables, so we evaluate the Lagrangian at $\xi_\mu^*(p)$. Let us introduce the corresponding on-shell navigator function:

$$N_\mu(p) := L_\mu(\xi_\mu^*(p), p) = c^T x_\mu^* - \mu \log \det X_\mu^*, \quad (3.55)$$

where the second equality holds because the terms multiplying y and Y in equation (3.7) vanish on-shell. In this language, we are interested in the locus $N_0(p) = 0$. As explained for example in [12], one generically finds that $\lim_{\mu \downarrow 0} \mu \log \det X_\mu^* = 0$ as well (even though X_μ^* becomes singular) so $N_0(p) = c^T x_0^*$ which also equals $b^T y_0^*$ because the duality gap vanishes.

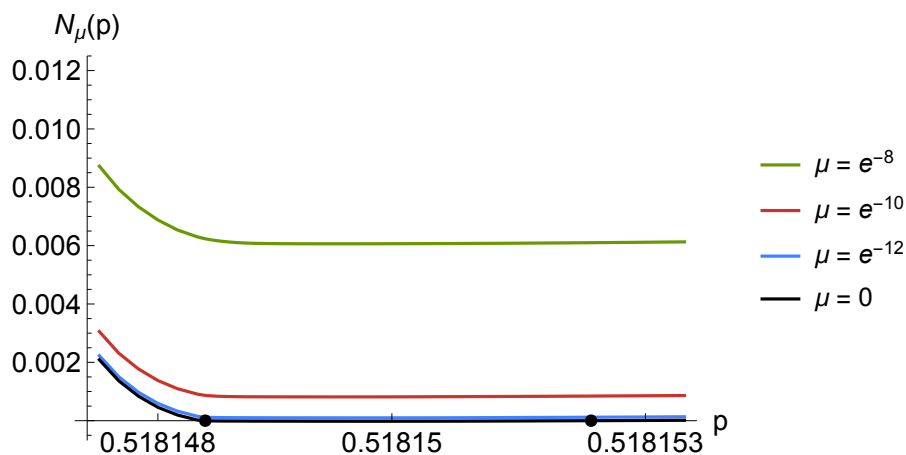


Figure 3.4: The function $N_\mu(p)$ in a representative example for various values of μ . For $\mu = 0$ there exist two zeroes at the indicated points, but for the other values of μ shown here the functions are everywhere positive.

In figure 3.4 we show $N_\mu(p)$ in a one-dimensional example for a few different values of μ . (The problem here is that of the three-dimensional Ising model, which is representative of more general conformal bootstrap problems. The parameter p is Δ_σ .⁸) The black line is $N_0(p)$ and, as can be seen in more detail in figure 3.5, there is actually a small negative region, and therefore there exist two points where $N_0(p) = 0$. At higher values of μ , however, there are no points with $N_\mu(p) = 0$ at all. Therefore it seems inadvisable to aim for $N_\mu(p) = 0$ as a proxy for $N_0(p) = 0$.

⁸Specifically, the problem is same as the problem in section 3.5.1 except we fixed $\Delta_\epsilon = 1.412625$, $\arctan(\lambda_{\epsilon\epsilon}/\lambda_{\sigma\sigma\epsilon}) = 0.9692606$ and used the GFF navigator at $\Lambda = 19$. Here we simply use this one parameter example as a demonstration.

We can improve this state of affairs by working with the primal and dual objectives. Since $\xi_\mu^*(p)$ is optimal at finite μ , it is in particular feasible, which leads immediately to the inequalities:

$$b^T y_\mu^* < b^T y_0^* = c^T x_0^* < c^T x_\mu^*, \quad (3.56)$$

for each $\mu > 0$. We can see this explicitly in figure 3.5. More importantly, we should notice that the vertical axis is zoomed in by a factor 250 compared to figure 3.4 so the primal and dual objective at finite μ provide a much more accurate estimate of $N_0(p)$!

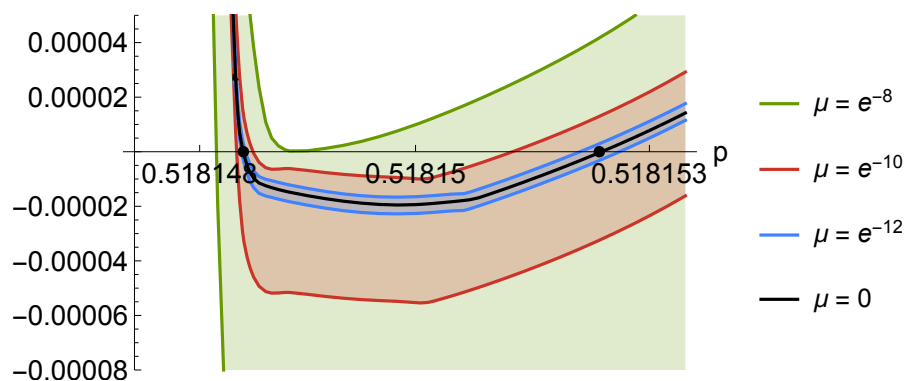


Figure 3.5: We shaded the region allowed by the bound of equation (3.56) for various values of μ . The black line agrees with the function $N_0(p)$ shown in figure 3.4 and, as it should be, lies in between the bounds for each value of μ . Notice that the vertical axis is zoomed in a factor 250 compared to figure 3.4.

Altogether it therefore makes sense to simply replace the condition $L_\mu = 0$ in equation (3.29) with an expression of the form:

$$\tilde{N}_\mu(p) := (1 - \gamma) c(p)^T x_\mu^* + \gamma b(p)^T y_\mu^* = 0 \quad (3.57)$$

for some parameter $\gamma \in [0, 1]$. Solving this equation should bring us much closer to the locus with $N_0(p) = 0$ already at finite μ , and of course for $\mu = 0$ the conditions are equivalent. Using the expression for the duality gap at finite μ

$$c^T x_\mu^* - b^T y_\mu^* = \mu \dim(X), \quad (3.58)$$

we can rewrite the constraint as,

$$\tilde{N}_\mu(p) = c(p)^T x_\mu^* - \gamma \mu \dim(X) = 0. \quad (3.59)$$

Generalizing from this on-shell analysis to an off-shell implementation where $\mu = \beta\mu_{\text{est}}$, we find the function

$$\tilde{L}_\mu(\xi, p) := c(p)^T x - \gamma \mu_{\text{est}} \dim(X) \quad (3.60)$$

We claim that $\tilde{L}_\mu = 0$ provides a better finite- μ approximation for the boundary of the bootstrap island. Finally, comparing $\tilde{L}_\mu(\xi, p)$ with the original Lagrangian (3.23), we have

$$\tilde{L}_\mu(\xi, p) = L_\mu(\xi, p) - \gamma \mu \dim(X) + \mu \log \det X + \text{linear errors}. \quad (3.61)$$

Aside from the negligible linear errors from solving for the SDP constraints, $\tilde{L}_\mu(\xi, p)$ contains a constant shift $-\gamma \mu \dim(X)$, and an X -dependent shift $\mu \log \det X$. We emphasize that both shifts are taken purely to locate the zero locus effectively and they do not affect the first two constraints on the gradients in (3.28). We reiterate the full set of constraints here:⁹

$$\nabla_\xi L_\mu = 0 \text{ and } \lambda v = \nabla_p L_\mu \text{ and } \tilde{L}_\mu = 0 \text{ and } \lambda > 0. \quad (3.62)$$

⁹In the code, this choice is equivalent to specifying the options `--navigatorWithLogDetX False` and `--gradientWithLogDetX True`

3.4 Implementation

3.4.1 Software

We implemented the skydiving algorithm as an open source C++ program called `skydive`. The code is based on `SDPB` and inherits most of `SDPB`'s options. The program `skydive` works on a specific SDP and performs the computation in the body of the first `while` loop (the main loop) of algorithm 2. The input of `skydive` is the SDP at p , a checkpoint ξ , a constellation (see footnote 6) of SDPs used to compute the gradient, and various options that control the behavior of the computation. The program performs one iteration of the main loop and gives the following output: a step dp in the external parameter space, a checkpoint for a new ξ , a gradient of the Lagrangian and a BFGS-updated Hessian.

To set up the full computation, we use external “driver” programs to drive the main loop of algorithm 2 and call `skydive`. The external programs set up the bootstrap problem, produce SDPs, call `skydive`, update p and other parameters, and iterate the main loop until optimality is reached.

The source code of our software is available online:

- The C++ solver `skydive`:
https://github.com/davidsd/sdpb/tree/skydiving_release.
- The Mathematica framework `simpleboot`, which can drive the main loop:
<https://gitlab.com/bootstrapcollaboration/simpleboot>.
- A Haskell package `dynamical-sdp` which can also drive the main loop:
<https://gitlab.com/davidsd/dynamical-sdp>.

Two of the present authors (AL and NS) conducted a mini-course on numerical bootstrap methods at the Perimeter Institute in April 2023, which included tutorials on skydiving and associated software tools. The materials from the mini-course can be found online at the following locations

- Mini course details:
<https://events.perimeterinstitute.ca/event/45/>.
- Slides and tutorial code:
<https://gitlab.com/AikeLiu/Bootstrap-Mini-Course>.

3.4.2 skydive options

In this section, we describe the input parameters of the `skydive` program and their relation to the algorithm presented in section 3.3.

- **newSdpDirs**: Specifies the path containing a “constellation” of SDP files that surround the SDP corresponding to the current value of p . These files are required to compute the derivatives of the Lagrangian $L(\xi, p)$ (using finite differences), including $\nabla_p L_\mu$, $H_{p\xi}$, and possibly H_{pp} (depending on whether the BFGS algorithm is used). The SDP files should be named `plus_i`, `minus_i`, and `sum_i_j`, where $0 \leq i < j < \dim(p)$.
- **externalParamInfinitesimal**: The step size ϵ in p -space between each SDP in the constellation and the current center SDP. For example, if the center SDP is computed at \vec{p} , then `plus_0` corresponds to the SDP computed at $\vec{p} + \epsilon\vec{e}_0$, `minus_0` corresponds to the SDP computed at $\vec{p} - \epsilon\vec{e}_0$, and `sum_0_1` corresponds to the SDP computed at $\vec{p} + \epsilon\vec{e}_0 + \epsilon\vec{e}_1$, where \vec{e}_i are basis vectors.
- **numExternalParams**: $\dim(p)$, the dimension of the parameter space.
- **totalIterationCount**: The total number of PDIP iterations computed in previous skydiving runs. This option does not affect the actual run, but can be useful for bookkeeping.
- **dualityGapUpperLimit**: dg_{thr} in algorithm 2. If the initial duality gap (either from a checkpoint or from primal/dual initial matrices) is larger than `dualityGapUpperLimit`, `skydive` will run PDIP until the duality gap becomes smaller than `dualityGapUpperLimit`.
- **centeringRThreshold**: R_{thr} in algorithm 2. `skydive` will run centering steps at a fixed p until $\|R\|_{\max} \leq \text{centeringRThreshold}$.
- **finiteDualityGapTarget**: This option can be used to find the optimal solution at a chosen value of $\mu > 0$, either to minimize the finite μ navigator function or to extremize p on the finite μ navigator function.
- Parameters of β -scanning, as described in section 3.3.1:
 - **betaScanMin**, **betaScanMax**, **betaScanStep**: β_{\min} , β_{\max} , and $\Delta\beta$ in the scanning routine.

- **stepMaxThreshold**: $\alpha_{\text{thr_max}}$, the threshold to decide whether to accept the step in items 1 and 2.
 - **stepMinThreshold**: $\alpha_{\text{thr_min}}$, the threshold to decide whether to execute item 3.
 - **betaClimbing**: β_{climbing} , decides which higher μ the solver should attempt (climb up to) to find $d\xi$ in item 4.
 - **maxClimbingSteps**: how many times can the solver climb before executing $\xi \rightarrow \xi + d\xi$ in item 4.
- Parameters of the modified BFGS algorithm.
 - **useExactHessian**: A boolean value indicating whether to use the full Newton method or the modified BFGS algorithm which does not require the exact values of the Hessian matrix H_{pp} . If this value is **False**, then the following parameters are required.
 - **prevGradientBFGS**, **prevExternalStep**, **prevHessianBFGS**: These three parameters correspond to $\nabla_p L(\xi_1, p_1) - \alpha H_{p\xi} H_{\xi\xi}^{-1} \nabla_\xi L(p_1, \xi_1, \mu_2)$, $p_2 - p_1$ and N_{pp} in section 3.3.2.2, respectively. If **prevHessianBFGS** is not specified, **skydive** will use N_{pp}^{default} in (3.53) as the default Hessian.
 - Parameters when extremizing p in a given direction.
 - **findBoundaryDirection**: If this option is not specified, the program is set to minimize the navigator function in the space of (ξ, p) . Otherwise, this option should be a vector v in p -space and then the program will maximize $v^T p$.
In practice, if the initial point is far away from the feasible region, it is useful to start from the minimization mode to move to the desired region and turn on **findBoundaryDirection** only after dual objective turns negative.
 - **findBoundaryObjThreshold**: A threshold parameter that terminates the program when $\mathcal{N}(p) \leq \text{findBoundaryObjThreshold}$.
 - **primalDualObjWeight**: The weight parameter $\gamma \in [0, 1]$ of section 3.3.3. Aside from the possible shift of $\mu \log \det X$ term (determined by **navigatorWithLogDetX**), if $\gamma = 0$, the navigator function

matches with the primal objective, and if $\gamma = 1$, the navigator function matches with the dual objective.

- `navigatorWithLogDetX`: A boolean value indicating whether the navigator function is computed with the $\mu \log \det X$ term. This option determines whether we aim at $L_\mu = 0$ or $\tilde{L}_\mu = 0$ during Newton’s steps as described in section 3.3.3.
- `gradientWithLogDetX`: A boolean value indicating whether the gradient of the navigator function is computed with the $\mu \log \det X$ term. This option determines whether we aim at $\nabla_p L_\mu = 0$ or $\nabla_p N_\mu = 0$ during Newton’s steps as described in section 3.3.3, where N_μ is defined as (3.55).

3.5 Example runs

In this section, we showcase the performance of the skydiving algorithm applied to the 3d Ising Model and the 3d $O(3)$ Model.¹⁰

In Table 3.1, we specify the values of the `skydive` parameters that were common throughout the example runs. We provide the values of `centeringRThreshold` and `dualityGapUpperLimit` in the subsections below. Some further comments on parameters will be provided in section 3.5.3.

<code>externalParamInfinitesimal</code>	10^{-40}
<code>betaScanMin</code>	0.1
<code>betaScanMax</code>	1.01
<code>betaScanStep</code>	0.1
<code>stepMinThreshold</code>	0.1
<code>stepMaxThreshold</code>	0.6
<code>maxClimbing</code>	1
<code>betaClimbing</code>	1.5
<code>primalDualObjWeight</code>	0.2
<code>gradientWithLogDetX</code>	True
<code>navigatorWithLogDetX</code>	False

Table 3.1: `skydive` parameters common for all example runs.

¹⁰Some of the runs in this section used older versions of the skydiving algorithm that are slightly different from the one presented in previous sections. However, these differences are immaterial to the results presented here.

3.5.1 3d Ising island

We tested our algorithm on the 3d Ising model mixed bootstrap problem of [5], using four-point functions of the lowest-dimension \mathbb{Z}_2 -odd operator σ and \mathbb{Z}_2 -even operator ϵ . The space \mathcal{P} consists of three parameters: the scaling dimensions Δ_σ and Δ_ϵ , together with the ratio of OPE coefficients $\lambda_{\sigma\sigma\epsilon}/\lambda_{\epsilon\epsilon\epsilon}$. In the notation of [5, 12] the navigator setup can be summarized as follows:

$$\begin{aligned}
\text{parameters: } & p = (\Delta_\sigma, \Delta_\epsilon, x = \lambda_{\sigma\sigma\epsilon}/\lambda_{\epsilon\epsilon\epsilon}), \\
\text{SDP objective: } & \mathcal{N}(p) := \max \vec{\alpha} \cdot \vec{V}_{\Delta=0, \ell=0}^{(+)}, \\
\text{conditions: } & \vec{\alpha} \cdot ((1, x) \cdot \vec{V}^{(\theta)} \cdot (1, x)) \geq 0, \\
& \vec{\alpha} \cdot \vec{V}_{\Delta, \ell=0}^{(+)} \geq 0 \text{ for } \Delta \geq 3, \\
& \vec{\alpha} \cdot \vec{V}_{\Delta, \ell=0}^{(-)} \geq 0 \text{ for } \Delta \geq 3, \\
& \vec{\alpha} \cdot \vec{V}_{\Delta, \ell}^{(+)} \geq 0 \text{ for } \Delta \geq \Delta_{\text{unitary}} \text{ and } \ell = 2, 4, \dots, \\
& \vec{\alpha} \cdot \vec{V}_{\Delta, \ell}^{(-)} \geq 0 \text{ for } \Delta \geq \Delta_{\text{unitary}} \text{ and } \ell = 1, 2, \dots, \\
& \vec{\alpha} \cdot M_\Sigma = 1,
\end{aligned} \tag{3.63}$$

where

$$\vec{V}^{(\theta)} = \vec{V}_{\Delta_\epsilon, 0}^{(+)} + \vec{V}_{\Delta_\sigma, 0}^{(-)} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{3.64}$$

Here $\vec{V}^{(+)}$ and $\vec{V}^{(-)}$ are the \mathbb{Z}_2 -even and \mathbb{Z}_2 -odd crossing vectors introduced in [5] and M_Σ imposes a normalization corresponding to the Σ -navigator described in [12].¹¹ All these conditions depend non-linearly on the parameters p , which is implicit in our notation. A point $p \in \mathcal{P}$ is excluded if the objective is positive and all the conditions are met.

With this setup we performed three runs of the skydiving algorithm, with the following goals:

$$\begin{aligned}
\text{runs 1 and 3: } & \max \Delta_\sigma \text{ such that } \mathcal{N}(p) \leq 0 \\
\text{run 2: } & \min \mathcal{N}(p)
\end{aligned} \tag{3.65}$$

Note that for runs 1 and 3 we aim for the rightmost tip of the Ising island and not the larger ‘‘continent’’ where Δ_σ is unbounded. The minimization of $\mathcal{N}(p)$ for run 2 is less physically relevant, but it might be a good predictor for the true values of the Ising CFT data as discussed in [12].

¹¹The specific choice of the vector does not affect the boundary of the feasible region and has minimal impact on the numerical results.

The details of each run are presented in table 3.2. The most important qualitative difference between runs 1 and 3 is that the former was done at bootstrap derivative order $\Lambda = 19$ and the latter at $\Lambda = 35$.¹² We terminated the runs when the first 12 digits of each component of p_{final} stabilized.

run	1	2	3
initial point	p_1	p_2	p_3
goal	max Δ_σ	min navigator	max Δ_σ
precision	448	448	768
Λ	19	19	35
κ	14	14	32
spins	S_{19}	S_{19}	S_{35}
Initial Hessian	H_{init}	none	none
dualityGapUpperLimit for 1st SDP	10^{-3}	10^{-6}	10^{-15}
dualityGapUpperLimit for the rest	10^{-3}	none	none
centeringRThreshold	10^{-10}	10^{-10}	10^{-10}
final point	$p_{1,\text{final}}$	$p_{2,\text{final}}$	$p_{3,\text{final}}$
total number of skydive calls: $d\xi, dp$	236	170	133
total number of PDIP iterations: $d\xi$	580	411	601
total number of Newton steps	816	581	734

Table 3.2: Individual parameter setup for example runs on 3d Ising model mixed bootstrap problem and the corresponding results. See the main text for the explanation of several the entries.

The symbols in the table correspond to the following values. First, the initial points for each run were

$$p_1 = (0.515, 1.4, 0.5)$$

$$p_2 = (0.51814, 1.4121, 0.686)$$

$$p_3 = (0.5181496477062039, 1.4126328939948094, 0.6863837285229739),$$

for reasons explained below, the initial Hessian for the first run was

$$H_{\text{init}} = \text{diag}(111.7955564168356, 1.184293042423152, 0.8299842690871352), \quad (3.66)$$

and the spin sets S_Λ correspond to

$$S_{19} = \{0, \dots, 26\} \cup \{49, 50\}$$

$$S_{35} = \{0, \dots, 44\} \cup \{47, 48, 51, 52, 55, 56, 59, 60, 63, 64, 67, 68, 71, 72, 75, 76, 79, 80\}. \quad (3.67)$$

¹²See [9] for the definition of the parameter Λ .

As for the results, in the first run we found that Δ_σ is maximized at

$$p_{1,\text{final}} = (0.518193035759, 1.41299963388, 0.686124688310),$$

in the second run that the navigator function is minimized at

$$p_{2,\text{final}} = (0.518134931547, 1.41244135167, 0.686615119182),$$

and in the third run with $\Lambda = 35$ that Δ_σ is maximized at

$$p_{3,\text{final}} = (0.518151700083, 1.41264851228, 0.686376348482). \quad (3.68)$$

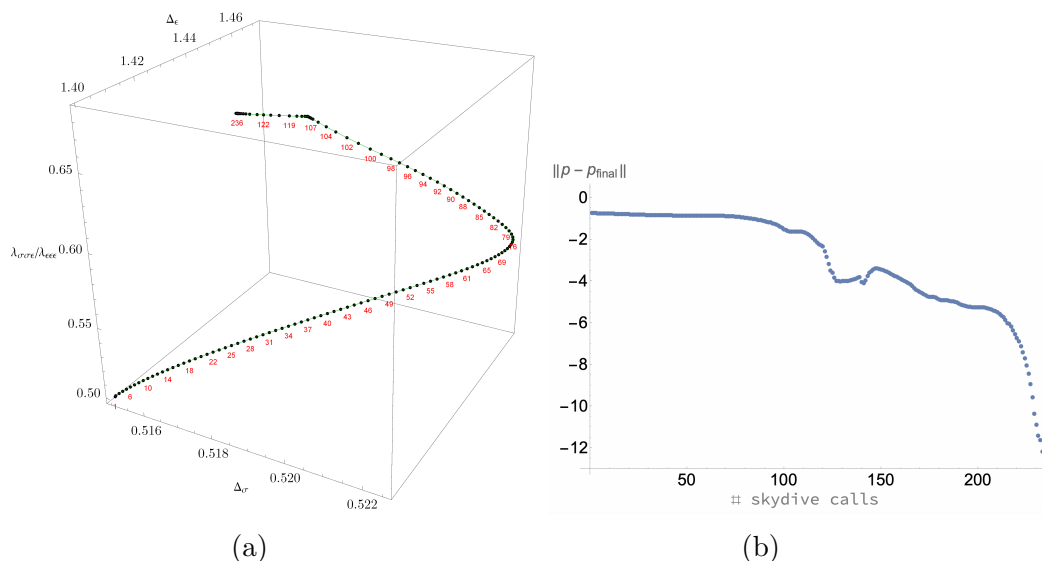


Figure 3.6: The path of a skydiving run 1 in Table 3.2. On the left, the path in the three-dimensional parameter space. On the right, the distance to the final point as a function of the number of `skydive` steps.

Let us offer some further comments on the setup for run 1. Here the initial point p_1 was deliberately chosen to lie rather far from the physically interesting region. In fact, with the standard navigator setup of [12] we were unable to make any run starting at p_1 converge to the Ising island. With our new algorithm, however, this turned out to be possible (and subsequently we could maximize Δ_σ). It did unfortunately still require some trial and error. For example, we chose the initial Hessian matrix to be as in equation (3.66) — but we expect that similar initial values will also work. We also set `dualityGapUpperLimit` to 10^{-3} throughout the run in order to prevent it from increasing μ and diverting to the continent in the initial stages. Altogether these tweaks might be useful in the future, where we would like to better control the behavior of the algorithm as it steps around in \mathcal{P} .

Let us now discuss the progression of the skydiving algorithm during run 1. As indicated in Table 3.2, run 1 took 236 calls to `skydive` to converge. In figures 3.6a, 3.6b, 3.7 and 3.8 we plot various quantities as a function of the number of steps. For brevity, we do not include plots for the other two runs. They were qualitatively similar, up to one exception that we discuss below.

Let us first discuss convergence. Figure 3.6a shows the path taken in the three-dimensional parameter space. The distance to the final point is shown in Figure 3.6b. Finally, Figure 3.7 shows how the duality gap decreases to zero.¹³

For the first 100 iterations or so we observe slow but steady movement towards the Ising island in \mathcal{P} , whereas the duality gap stays essentially flat. During this time, the algorithm would have frequently preferred to take a climbing step (with $\beta > 1$) but it is not allowed to do so because the duality gap has reached our set `dualityGapUpperLimit` of 10^{-3} . It therefore settles instead for a full $(d\xi, dp)$ step, even though the corresponding step length α is very small. We stress that this is exactly the desired behavior at this stage, since the bigger steps in \mathcal{P} do not aid convergence toward the Ising island. It is this initial part that is atypical compared to the other two runs.

After about the 100th iteration, things accelerate. The duality gap begins to decrease and we observe approximate linear convergence. At about the 200th `skydive` call we observe an *increase* in the duality gap for several steps. This is the stalling recovery mechanism at work, where the algorithm chose $\beta > 1$ to ensure a reasonable step size. In the very last stage, we finally observe (slightly) superlinear convergence. Together, these figures vividly illustrate how the different subroutines described in section 3.3 work cooperatively in a complete run of the skydiving algorithm.

The dramatic improvement of the computational cost of our algorithm over a standard navigator function run is visualized in Figure 3.8. To understand it, we should first note that using `skydive` does not seem to require a much larger number of steps dp than the standard navigator approach of [12]. For example, a problem like our run 2 (navigator minimization over 3 parameters at

¹³After we finished computations for those plots, we discovered a small bug in our code: during the scanning step, the code used the value of the navigator function at $\beta = 1$ to solve for the boundary, but in fact it should use the extrapolated value at the scanned β . After fixing this bug, typical runs had about 5% fewer iterations than the ones presented in this paper.

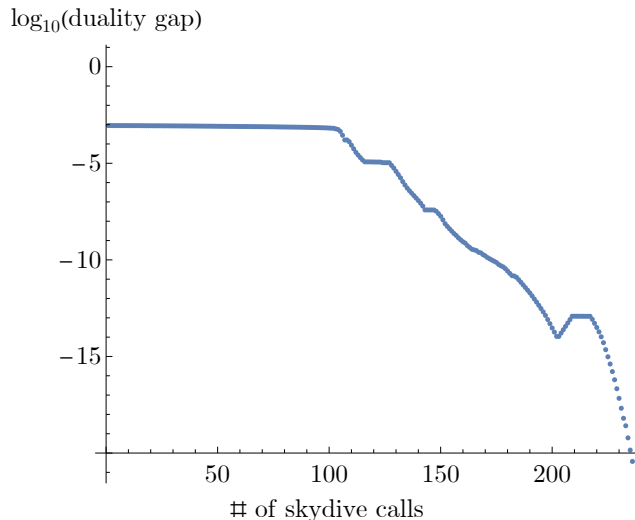


Figure 3.7: Duality gap in each `skydive` call during run 1 in table 3.2 which maximizes Δ_σ from point p_1 at $\Lambda = 19$.

$\Lambda = 19$) would typically have required about 100 steps ([12], figure 22) whereas it took us 173 steps (see Table 3.2). Therefore, to compare the algorithms it is meaningful to just compare the computational cost of obtaining a single dp , i.e., of a single call to `skydive`.

The costliest operation in our algorithm is the inversion of the Hessian $H_{\xi\xi}$ to produce an update $d\xi$. In Figure 3.8 we plot the number of $d\xi$'s computed for each call to `skydive` during run 1. In the first iteration, we begin with a standard PDIP run to bring the duality gap below `dualityGapUpperLimit`, which takes about 200 Hessian inversions. Afterwards, we however only compute $d\xi$ a handful of times for each step dp . More precisely, we compute $d\xi$ once for each of the (typically 2 or 3) centering steps, plus once more in conjunction with dp in the scanning step. In the traditional navigator approach, on the other hand, an entire SDP was solved to optimality before calculating dp . This would mean that the new algorithm reduces about 200 Hessian inversions to just 2 or 3!

The final rows in Table 3.2 show that the same speedup was realized for the other two runs. If we take run 2 as an example: the standard navigator approach would need about $2 \cdot 10^4$ Hessian inversions (200 steps $d\xi$ for each of the 100 steps dp) whereas we reached the same optimal point with only 581 such operations (3% as many).

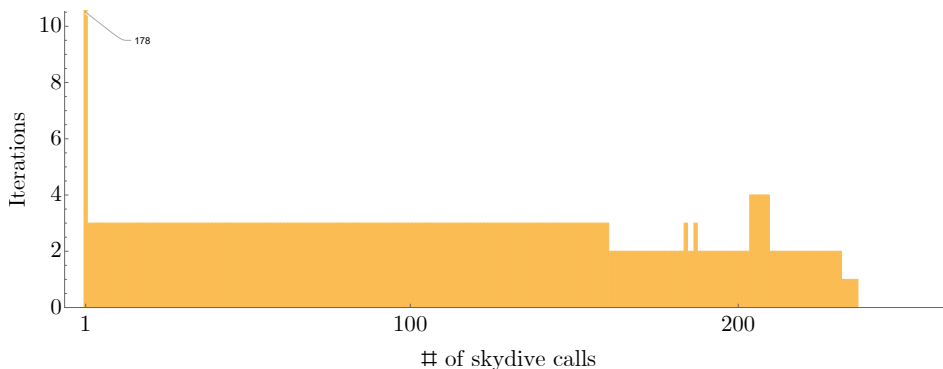


Figure 3.8: The number of Newton iterations spent in each `skydive` call for the skydiving run of p_1 . For each call one Newton iteration is necessary for the scanning step, and the remaining Newton iterations are due to scanning steps or, as in the first `skydive` call, standard PDIP steps used to lower the duality gap.

3.5.1.1 Computational resources

In this section we give more details on the computational resources used for run 1. We performed the computation on a computer with 40 CPU cores.¹⁴ The computation uses `simpleboot` in high efficiency mode, where all heavy computations are done in C++ programs [92]. Within one iteration of the main loop of Algorithm 2, the heavy computations are: (1) computation of convolved conformal blocks (using the `scalar_blocks_mod` program); (2) creation of the files defining SDPs (using the `sdp2input_mod` program); (3) running the `skydive` program. Other than these three C++ programs, the rest costs a negligible amount of time.

During each iteration, there are 12 `scalar_blocks_mod` calls and 4 `sdp2input_mod` calls (corresponding to 4 SDPs). The precision of both programs was set to 448 binary digits. To efficiently use the 40 cores, we launched batches of 6 calls of `scalar_blocks_mod` simultaneously and each call had 7 OpenMP threads. After 12 `scalar_blocks_mod` calls were finished, we launched 4 `sdp2input_mod` calls simultaneously and each call had 10 MPI ranks. We found this is the optimal scheme to produce the SDPs. Finally, `skydive` was launched with 40 MPI ranks. The total computation from p_1 took 13696 seconds, that is, less than four hours, among which 40% is spent on `scalar_blocks_mod`, 30% on `sdp2input_mod`, 30% on `skydive`.

¹⁴The computer is a node with Intel(R) Xeon(R) Gold 6148 2.40GHz CPU on the Symmetry cluster of the Perimeter Institute. We thank the Perimeter Institute for the computational resources.

If we look at the cost of the `skydive` calls alone, on average 48% of the time was spent on reading and writing data from and to disk. Furthermore, we found that CPU efficiency of the total computation is 31% , i.e., 69% of the time the CPUs were idle.

We would like to emphasize that, in this run, we have already carefully optimized the SDP-generating part of the run. However, the total time spent running `skydive` is still less than the time spent on generating the SDPs. Thanks to the skydiving algorithm, solving many different SDPs is no longer the bottleneck of this bootstrap computation!

3.5.2 $O(3)$ tiptop search

In this section, we describe an application of skydiving to the 3d critical $O(3)$ model. A previous bootstrap exploration of this model [93] studied correlation functions of three scalar operators (v, s, t) with lowest dimension in the vector, scalar, and traceless symmetric tensor representations of $O(3)$, respectively. Stability of the $O(3)$ model under RG flow depends on the scaling dimension of the lowest-dimension scalar t_4 in the rank-4 tensor representation of $O(3)$. If $\Delta_{t_4} < 3$, then the $O(3)$ fixed point is unstable.

The authors of [93] obtained an upper bound on Δ_{t_4} only slightly less than 3, thereby rigorously demonstrating instability of the $O(3)$ model. That computation was however very costly. It involved sampling many points in an 8-dimensional parameter space \mathcal{P} . Furthermore, the proximity of the best upper bound to 3 made the search rather delicate. It is therefore an ideal example to test whether skydiving can be more efficient.

The bootstrap problem from [93] is as follows:

$$\begin{aligned}
\text{parameters: } & p = \left(\Delta_v, \Delta_s, \Delta_t, \Delta_{t_4}, \frac{\lambda_{vtv}}{\lambda_{vvs}}, \frac{\lambda_{tts}}{\lambda_{vvs}}, \frac{\lambda_{ttt}}{\lambda_{vvs}}, \frac{\lambda_{sss}}{\lambda_{vvs}} \right), \\
\text{SDP objective: } & \vec{\alpha} \cdot V_{\Delta=0, \ell=0}^{([0,+])}, \\
\text{goal: } & \text{maximize } \Delta_{t_4} \text{ while SDP objective } \leq 0 \\
\text{conditions: } & \vec{\alpha} \cdot ((1, \lambda_{vtv}, \lambda_{tts}, \lambda_{ttt}, \lambda_{sss}) \cdot \vec{V}^{(\theta)} \cdot (1, \lambda_{vtv}, \lambda_{tts}, \lambda_{ttt}, \lambda_{sss})) \geq 0, \\
& \vec{\alpha} \cdot V_{\Delta, \ell=0}^{[1,-]} \geq 0 \text{ for } \Delta \geq 3, \\
& \vec{\alpha} \cdot V_{\Delta, \ell=0}^{[4,+]} \geq 0 \text{ for } \Delta \geq \Delta_{t_4}, \\
& \vec{\alpha} \cdot V_{\Delta, \ell=0}^{[2,+]} \geq 0 \text{ for } \Delta \geq 3, \\
& \vec{\alpha} \cdot V_{\Delta, \ell=0}^{[0,+]} \geq 0 \text{ for } \Delta \geq 3, \\
& \vec{\alpha} \cdot V_{\Delta=3, \ell=2}^{\text{EMT}} \geq 0, \\
& \vec{\alpha} \cdot V^R \geq 0 \text{ for } \Delta \geq \Delta_{\text{unitary}} + \delta \text{ and all other } R \text{ and spins } \ell.
\end{aligned} \tag{3.69}$$

Here $\vec{V}^{(\theta)}$ represents the crossing vector that has the external operators (v, s, t) appearing as an intermediate operator, δ is a small twist gap set to 10^{-7} , and V^R denotes the crossing vector for the representation R . The representations appearing in this setup are

$$[1, -], [4, +], [2, +], [2, -], [3, +], [3, -], [4, +],$$

where the notation $[n, \pm]$ refers to the $O(3)$ rank n traceless symmetric tensor with $O(3)$ parity \pm . V^{EMT} represents the crossing vector for the stress tensor, using OPE coefficient ratios determined by the Ward identity.¹⁵

The only modification between the setup of (3.69) and ours is that we switched from feasibility mode to navigator mode. The problem is then exactly analogous to the Ising case of the previous subsection: we define $\mathcal{N}(p)$ as the maximum of the SDP objective and, after imposing a suitable normalization $\vec{\alpha} \cdot M_\Sigma = 1$, we slightly reformulate the goal as maximizing Δ_{t_4} while $\mathcal{N}(p) \leq 0$.

We conducted three **skydive** runs at $\Lambda = 19$, each starting from a different initial point. The parameters and performance statistics are summarized in table 3.3. Again, the symbols in the table correspond to the following values.

¹⁵The specific details of all the crossing vectors may depend on conventions. In practice, we used the **autoboot** [94] package to generate these crossing vectors.

First, the initial points for each run were

$$p_1 = (0.518957, 1.59539, 1.2097, 2.97757, 3.0461, 2.4233, 3.98997, 0.557463)$$

$$p_2 = (0.52, 1.6, 1.21, 3, 3, 2.5, 4, 0.5)$$

$$p_3 = (0.517, 1.6, 1.21, 3, 3, 2.5, 4, 0.5)$$

All three runs reached the optimum point:

$$p_{\text{final}} = (0.519124, 1.59715, 1.21049, 2.99985, 3.04764, 2.42518, 3.99318, 0.560976), \quad (3.70)$$

from which we in particular conclude that

$$\Delta_{t_4} \lesssim 2.99985, \quad (3.71)$$

and therefore the $O(3)$ model is unstable.¹⁶

run	1	2	3
initial point	p_1	p_2	p_3
dualityGapUpperLimit for 1st SDP	0.0001	0.001	0.001
dualityGapUpperLimit for the rest	none	0.001	0.001
centeringRThreshold	10^{-10}	10^{-40}	10^{-10}
total number of <code>skydive</code> calls: $d\xi, dp$	254	401	515
total number of PDIP iterations: $d\xi$	577	1446	1058
total number of Newton steps	831	1847	1573

Table 3.3: Skydiving computations of the $O(3)$ tiptop search starting from different initial points at $\Lambda = 19$. The initial points p_i are given in the main text.

Again, let us provide more details on run 1 as an example.

In figure 3.9, we illustrate how the duality gap changes in each `skydive` iteration. Here we see several climbing phases in order to avoid stalling.

In figure 3.10, the number of updates $d\xi$ conducted in each `skydive` call is shown. These iterations include the pure PDIP steps in the initial `skydive` call to reduce $\mu < \text{dualityGapUpperLimit}$, the centering steps which are PDIPs at $\beta = 1$, and the full step updating both (ξ, p) . As we stressed before, these were computationally the most costly steps because of the required Cholesky

¹⁶Note that this upper bound is weaker than the one in [93]. This is because we have used fewer derivatives ($\Lambda = 19$) as opposed to $\Lambda = 35$ in [93].

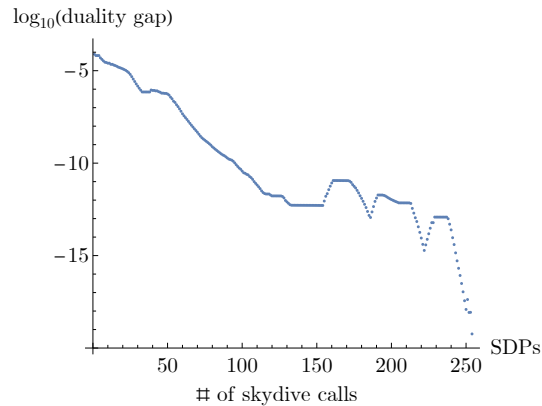


Figure 3.9: Duality gap in each `skydive` call during run 1 in table 3.3 which maximizes Δ_{t_4} from point p_1 at $\Lambda = 19$.

factorization to invert the Hessian matrix $H_{\xi\xi}$ as mentioned in section 3.2.1. Once again we managed to realize a drastic reduction in the number of such iterations, thereby mitigating the bottleneck in standard SDPB computations. We also observe the typical behaviors of skydiving runs: the algorithm spends significantly many PDIP steps in the first `skydive` call (208 in this example), and then the algorithm only requires a few iterations per call.

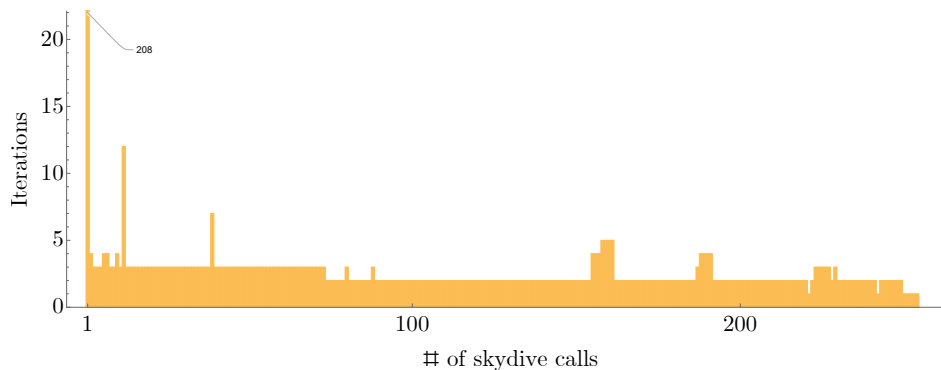


Figure 3.10: The number of Newton iterations spent in each `skydive` call for the skydiving run 1.

We can compare results above with the (navigator + SDPB) method. A navigator computation of the same problem was performed starting from:

(0.51902209, 1.59623576, 1.20998, 2.988, 3.04429644, 2.42004559, 3.98537291, 0.55365625).

which is in between p_1 and p_{final} . This computation took 227 navigator function calls (so steps dp), which is again entirely in the same ballpark as the number of `skydive` calls. However, here one would normally require 240 PDIP

(Newton) iterations to update ξ in between updates of p . We therefore once again find a significant speedup.

As an aside we note that we managed to find some improvement in the standard navigator method through the use of warmstarting: at a point p_k we save a checkpoint about midway through the SDPB run, and then at point p_{k+1} we restart SDPB from this checkpoint rather than completely un-initialized. Normally such warmstarting would quickly result in stalling, but we managed to find a climbing routine (inspired by the scanning subroutine of `skydive`, in particular item 4 on page 77, to recover from such stalling). After implementing the required modifications in SDPB¹⁷ we were able to reduce the number of PDIP steps from 240 to about 60 on average.

We also compared the performance of the skydiving algorithm with the `tiptop` algorithm of [93]. In a specific `tiptop` run at $\Lambda = 19$ ¹⁸, the total number of Newton steps is about 143,910¹⁹. The skydiving run from p_1 has the number of iterations that are 173 times smaller! Also, the result from the skydiving run is almost exactly on the highest feasible value of Δ_{t_4} , whereas the `tiptop` algorithm bisects the Δ_{t_4} until reaching a certain resolution.

3.5.3 Usage tips and current limitations

As we have alluded in previous sections, some of our parameters, `dualityGapUpperLimit`, `centeringRThreshold`, and the initial hessian H_{init} to start the BFGS algorithm, affect the efficiency and the rate of success of skydiving runs and their preferred values depend closely on specific bootstrap problems and initial conditions. In this section, we will elaborate more on their roles during a run and offer some tips on how to choose them. We will also mention some current limitations of the algorithm

¹⁷https://github.com/suning-git/sdpb/tree/sdpb2.4.0_midck_stallingrecover.

¹⁸This computation was done during the project of [93], but was not reported in the paper.

¹⁹The `tiptop` algorithm scanned about 1,755 points in the space of $(\Delta_v, \Delta_s, \Delta_t, \Delta_{t_4})$. Feasibility at each point is determined by the cutting surface algorithm. On average, the cutting surface algorithm scans 41 points in the space of OPE coefficients, and each point corresponds to an SDP in feasibility mode, whose computation requires about 2 Newtonian steps.

3.5.3.1 Running off in parameter space

Let us first discuss the importance of `dualityGapUpperLimit` and how it prevents the algorithm from leaving the region of interest in parameter space. We will use *duality gap* and μ interchangeably in this section as they are proportional to each other according to (3.58).

As we mentioned in the beginning of section 3.3, when the duality gap (or μ) is large, the Lagrangian is not a reliable approximation to the navigator function, and we must avoid taking steps in parameter space until the duality gap falls below `dualityGapUpperLimit`. However, during the scanning steps discussed in section 3.3.1, it is possible for the duality gap to increase again, and potentially move back above `dualityGapUpperLimit`. Should we continue to impose `dualityGapUpperLimit` during scanning to prevent this?

The parameter space of a bootstrap problem can contain multiple isolated feasible regions — either compact regions (islands), or unbounded regions (continents). In a feasible region, the navigator function $\mathcal{N}(p)$ is non-positive. An island will contain at least one local minimum of $\mathcal{N}(p)$. The existence of multiple isolated feasible regions implies the non-convexity of the navigator function. Often, we want to restrict our attention to an island as we did in the previous subsections, but the non-convexity implies that the steps in parameter space \mathcal{P} can run off to a different region. The navigator method can mitigate this issue by penalizing the $\mathcal{N}(p) > 0$ region.

By contrast, the finite- μ Lagrangian makes running-off behaviors more troublesome for a few reasons:

1. The finite- μ navigator generally starts large and gradually decreases. Unlike for the navigator function, there is no simple criterion like $\mathcal{N}(p) \geq 0$ for penalizing an unwanted region.
2. At large μ , the Lagrangian is smoother and barriers between different local regions are lower, so it is easier for the solver to walk from one region to another.
3. If the finite- μ Lagrangian is too different from the navigator function, the solver can be attracted to a region that is different from the target region at $\mu = 0$, and it may not be able to escape when μ gets smaller.

We found that continuing to impose `dualityGapUpperLimit` can help avoid these complications.

If the initial point is sufficiently close to the island, we observed that a good choice of `dualityGapUpperLimit` for the first `skydive` call is the value of the duality gap when the relative gap is $\mathcal{O}(1)$ ²⁰:

$$\frac{|\text{primal objective} - \text{dual objective}|}{\min(|\text{primal objective}|, |\text{dual objective}|)} \sim \mathcal{O}(1). \quad (3.72)$$

In the second and third run in table 3.3, for example, we found that it is necessary to impose `dualityGapUpperLimit` on all `skydive` iterations. The initial points p_2 and p_3 are significantly far from the island and a constant `dualityGapUpperLimit` prevents the solver from moving towards the “continent”.

Another subtlety is that sometimes using N_{pp}^{default} of (3.53) as the initial Hessian can be problematic. In the Ising example, we observed that for the run from p_1 , if we use N_{pp}^{default} as the initial Hessian, the solver makes a bad step towards the continent, even though the gradient points towards the island. Most likely the problem is caused by off-diagonal elements in N_{pp}^{default} . We solved this problem by using the initial Hessian (3.66), which is the diagonal piece of N_{pp}^{default} at p_1 .

3.5.3.2 Failure of centering

Our algorithm 2 assumes that the centering procedure in section 3.3 can bring $\xi_\mu(p)$ sufficiently close to the central section $\xi_\mu^*(p)$. In practice, this means that we assume that we are able to reduce $R_{err} := \|R\|_{\max}$ to arbitrarily small values at any fixed finite μ .

However, in certain semidefinite programs it is not possible to find an optimal solution at finite μ . One important class of examples is when primal or dual feasible points lie exactly on the boundary of the cone of positive semidefinite matrices rather than in its interior. This is a violation of the so-called Slater condition, which demands the existence of at least one feasible point which is strictly positive definite. The logarithmic barrier function introduced above is then infinite at every feasible point and consequently the central path does

²⁰Notice that this choice is problem dependent that might require users to launch some test run to find a good `dualityGapUpperLimit`

not really exist. The standard primal-dual interior point algorithm is then no longer guaranteed to work, although it might still perform well enough in practice.

In bootstrap studies, violation of the Slater condition happens occasionally on the dual side, when some components of the functionals must be exactly zero in order to satisfy the positivity conditions. In the course of an SDPB run, this typically corresponds to the failure to obtain dual feasible “jumps;” instead the dual error reduces gradually to 0 as Y becomes increasingly singular.

Returning to the context of centering steps, R_{err} is a measure of how correctly the constraint $XY = \mu I$ is being solved and it is impossible to make R_{err} zero at finite μ if there is no non-singular Y satisfying the dual constraints. Hence, failing to satisfy the Slater condition manifests itself as a non-convergent R_{err} in our algorithm.

The current skydiving algorithm does not include a solution to this problematic situation. To avoid it, the user may have to modify their SDP, perhaps by discarding crossing equations by hand so that the Slater condition is satisfied. Another option is to bypass centering steps by setting `centeringRThreshold` to a large value. Finding a general strategy to cope with this scenario remains a problem for future research.

3.6 Conclusions and future directions

The core idea of skydiving is simple: combine optimization of a SDP with optimization over the external parameters that it depends on. In theory, this simply requires us to treat the Lagrange function of the SDP as a function of both internal variables ξ and external parameters p . We have found that this idea can work in practice, as long as we supplement it with a few crucial techniques to avoid stalling: (1) centering iterations to stay close to the central section, (2) β -scan to carefully decide how to change μ , (3) climbing in μ to recover from stalling when necessary. Furthermore, we found an efficient method for avoiding expensive computations of the Hessian in p -space using a BFGS-type update. These techniques are implemented in the open-source C++ program `skydive`.

Our tests of `skydive` on various conformal bootstrap problems show promising results. In some cases, the time spent towards solving an SDP is shorter than the time spent generating the SDP (by constructing matrices of confor-

mal blocks and writing them to disk). In these cases, the bottleneck of the bootstrap computation is no longer in optimization and in the future we need to pursue other avenues for improving performance.

While `skydive` has already proved practically useful, it represents only an initial attempt in the direction of combined internal-external optimization. There are several possibilities for further enhancements, some of which we outline below.

The idea of iterating the corrector step has proven valuable to improve convergence at small computational cost. It would be interesting to use this idea in ordinary SDP solutions, as well as to use it to solve (3.25) with fixed dp .²¹ We expect that this could make it relatively cheap to move along the local central path.

In skydiving, there is a basic dilemma: at higher values of μ , the solver can take larger steps in $d\xi$ (since the X, Y matrices are not close to degeneracy), but the prediction for p becomes less accurate (due to the large- μ navigator function being a poor approximation to the true $\mu = 0$ navigator function). Meanwhile, the opposite holds for lower values of μ . In our current algorithm, we seek a balance between these competing effects using the “ β -scan” and “climbing” techniques. However, if it becomes relatively inexpensive to move along the local central path, it might be appealing to implement a strategy where the solver computes the step dp at lower μ but executes the step at higher μ . A natural measure of the accuracy of the navigator function could be the relative duality gap

$$\frac{|\text{primal objective} - \text{dual objective}|}{\min(|\text{primal objective}|, |\text{dual objective}|)}, \quad (3.73)$$

while a natural criterion for determining whether μ is large enough could come from testing the step length.

Our current implementation separates control of the main loop and the computations inside the loop between the external “driver” programs and `skydive`. While this implementation is suitable for experimentation, it suffers from high

²¹In our `skydive` program, we have actually implemented corrector iterations for solving (3.25) with fixed dp . However, this requires saving the Schur complement from the previous SDP and loading it into the new SDP. In the current implementation, this data is processed as text, resulting in slow read/write operations. We find it to be inefficient in practice, but it can certainly be improved in the future.

I/O costs, due to the need to repeatedly generate new SDP files and write them to disk in the driver, and load them from disk in `skydive`. In the future, it would be desirable to develop an integrated program that executes the full algorithm.

A more ambitious goal is to extrapolate bootstrap computations from lower to higher derivative order.²² As the derivative order increases, the data specifying the SDP (b , B , and c) will increase in dimension discontinuously. An important challenge will be to compute an appropriate $d\xi$ to account for this discontinuous change.

²²This idea was made to work for the spinless modular bootstrap in [95].

Bibliography

- [1] D. Simmons-Duffin, *A Semidefinite Program Solver for the Conformal Bootstrap*, *JHEP* **06** (2015) 174, [1502.02033].
- [2] W. Landry and D. Simmons-Duffin, *Scaling the semidefinite program solver SDPB*, 1909.09745.
- [3] S. El-Showk, M. F. Paulos, D. Poland, S. Rychkov, D. Simmons-Duffin and A. Vichi, *Solving the 3D Ising Model with the Conformal Bootstrap*, *Phys. Rev.* **D86** (2012) 025022, [1203.6064].
- [4] S. El-Showk, M. F. Paulos, D. Poland, S. Rychkov, D. Simmons-Duffin and A. Vichi, *Solving the 3d Ising Model with the Conformal Bootstrap II. c -Minimization and Precise Critical Exponents*, *J. Stat. Phys.* **157** (2014) 869, [1403.4545].
- [5] F. Kos, D. Poland and D. Simmons-Duffin, *Bootstrapping Mixed Correlators in the 3D Ising Model*, *JHEP* **11** (2014) 109, [1406.4858].
- [6] F. Kos, D. Poland, D. Simmons-Duffin and A. Vichi, *Precision islands in the Ising and $O(N)$ models*, *JHEP* **08** (2016) 036, [1603.04436].
- [7] D. Simmons-Duffin, *The Lightcone Bootstrap and the Spectrum of the 3d Ising CFT*, *JHEP* **03** (2017) 086, [1612.08471].
- [8] F. Kos, D. Poland and D. Simmons-Duffin, *Bootstrapping the $O(N)$ vector models*, *JHEP* **1406** (2014) 091, [1307.6856].
- [9] F. Kos, D. Poland, D. Simmons-Duffin and A. Vichi, *Bootstrapping the $O(N)$ Archipelago*, *JHEP* **11** (2015) 106, [1504.07997].
- [10] S. M. Chester, W. Landry, J. Liu, D. Poland, D. Simmons-Duffin, N. Su et al., *Carving out OPE space and precise $O(2)$ model critical exponents*, *JHEP* **06** (2020) 142, [1912.03324].

- [11] S. M. Chester, W. Landry, J. Liu, D. Poland, D. Simmons-Duffin, N. Su et al., *Bootstrapping Heisenberg magnets and their cubic instability*, *Phys. Rev. D* **104** (2021) 105013, [2011.14647].
- [12] M. Reehorst, S. Rychkov, D. Simmons-Duffin, B. Sirois, N. Su and B. van Rees, *Navigator Function for the Conformal Bootstrap*, *SciPost Phys.* **11** (2021) 072, [2104.09518].
- [13] D. Poland, D. Simmons-Duffin and A. Vichi, *Carving Out the Space of 4D CFTs*, *JHEP* **05** (2012) 110, [1109.5176].
- [14] P. Kravchuk and D. Simmons-Duffin, *Counting Conformal Correlators*, *JHEP* **02** (2018) 096, [1612.08987].
- [15] R. S. Erramilli, L. V. Iliesiu and P. Kravchuk, *Recursion relation for general 3d blocks*, *JHEP* **12** (2019) 116, [1907.11247].
- [16] P. Kravchuk, *Casimir recursion relations for general conformal blocks*, *JHEP* **02** (2018) 011, [1709.05347].
- [17] D. Karateev, P. Kravchuk and D. Simmons-Duffin, *Harmonic Analysis and Mean Field Theory*, *JHEP* **10** (2019) 217, [1809.05111].
- [18] R. S. Erramilli, L. V. Iliesiu, P. Kravchuk, W. Landry, D. Poland and D. Simmons-Duffin, *blocks_3d: software for general 3d conformal blocks*, *JHEP* **11** (2021) 006, [2011.01959].
- [19] D. Poland, S. Rychkov and A. Vichi, *The conformal bootstrap: Theory, numerical techniques, and applications*, *Reviews of Modern Physics* **91** (jan, 2019) .
- [20] D. Poland and D. Simmons-Duffin, *Snowmass white paper: The numerical conformal bootstrap*, 2022.
- [21] A. M. Polyakov, *Nonhamiltonian approach to conformal quantum field theory*, *Zh. Eksp. Teor. Fiz.* **66** (1974) 23–42.
- [22] S. Ferrara, A. F. Grillo and R. Gatto, *Tensor representations of conformal algebra and conformally covariant operator product expansion*, *Annals Phys.* **76** (1973) 161–188.

- [23] G. Mack, *Duality in Quantum Field Theory*, *Nucl.Phys.* **B118** (1977) 445.
- [24] D. J. Gross and A. Neveu, *Dynamical Symmetry Breaking in Asymptotically Free Field Theories*, *Phys. Rev.* **D10** (1974) 3235.
- [25] L. Iliesiu, F. Kos, D. Poland, S. S. Pufu, D. Simmons-Duffin and R. Yacoby, *Bootstrapping 3D Fermions*, *JHEP* **03** (2016) 120, [1508.00012].
- [26] I. F. Herbut, *Interactions and phase transitions on graphene's honeycomb lattice*, *Phys. Rev. Lett.* **97** (2006) 146401, [cond-mat/0606195].
- [27] I. F. Herbut, V. Juricic and B. Roy, *Theory of interacting electrons on the honeycomb lattice*, *Phys. Rev. B* **79** (2009) 085116, [0811.0610].
- [28] I. F. Herbut, V. Juricic and O. Vafek, *Relativistic Mott criticality in graphene*, *Phys. Rev.* **B80** (2009) 075432, [0904.1019].
- [29] L. N. Mihaila, N. Zerf, B. Ihrig, I. F. Herbut and M. M. Scherer, *Gross-Neveu-Yukawa model at three loops and Ising critical behavior of Dirac systems*, *Phys. Rev. B* **96** (2017) 165133, [1703.08801].
- [30] M. Vojta, Y. Zhang and S. Sachdev, *Quantum Phase Transitions in d-Wave Superconductors*, *Phys. Rev. Lett.* **85** (2000) 4940–4943.
- [31] M. Vojta, *Quantum phase transitions*, *Reports on Progress in Physics* **66** (2003) 2069.
- [32] L. Ziegler, E. Tirrito, M. Lewenstein, S. Hands and A. Bermudez, *Large- N Chern insulators: Lattice field theory and quantum simulation approaches to correlation effects in the quantum anomalous Hall effect*, *Annals Phys.* **439** (2022) 168763, [2111.04485].
- [33] T. Grover, D. Sheng and A. Vishwanath, *Emergent Space-Time Supersymmetry at the Boundary of a Topological Phase*, *Science* **344** (2014) 280–283, [1301.7449].
- [34] L. Iliesiu, F. Kos, D. Poland, S. S. Pufu and D. Simmons-Duffin, *Bootstrapping 3D Fermions with Global Symmetries*, *JHEP* **01** (2018) 036, [1705.03484].

- [35] J. A. Gracey, *Three loop calculations in the $O(N)$ Gross-Neveu model*, *Nucl. Phys.* **B341** (1990) 403–418.
- [36] B. Rosenstein, H.-L. Yu and A. Kovner, *Critical exponents of new universality classes*, *Phys. Lett.* **B314** (1993) 381–386.
- [37] N. Zerf, C.-H. Lin and J. Maciejko, *Superconducting quantum criticality of topological surface states at three loops*, *Phys. Rev.* **B94** (2016) 205106, [1605.09423].
- [38] J. A. Gracey, T. Luthe and Y. Schroder, *Four loop renormalization of the Gross-Neveu model*, *Phys. Rev.* **D94** (2016) 125028, [1609.05071].
- [39] L. Fei, S. Giombi, I. R. Klebanov and G. Tarnopolsky, *Yukawa CFTs and Emergent Supersymmetry*, *PTEP* **2016** (2016) 12C105, [1607.05316].
- [40] N. Zerf, L. N. Mihaila, P. Marquard, I. F. Herbut and M. M. Scherer, *Four-loop critical exponents for the Gross-Neveu-Yukawa models*, *Phys. Rev. D* **96** (2017) 096010, [1709.05057].
- [41] B. Ihrig, L. N. Mihaila and M. M. Scherer, *Critical behavior of Dirac fermions from perturbative renormalization*, *Phys. Rev. B* **98** (2018) 125109, [1806.04977].
- [42] J. A. Gracey, *Anomalous mass dimension at $O(1/N^{**2})$ in the $O(N)$ Gross-Neveu model*, *Phys. Lett. B* **297** (1992) 293–297.
- [43] S. E. Derkachov, N. A. Kivel, A. S. Stepanenko and A. N. Vasiliev, *On calculation in $1/n$ expansions of critical exponents in the Gross-Neveu model with the conformal technique*, [hep-th/9302034](#).
- [44] J. A. Gracey, *Computation of critical exponent η at $O(1/N^{**3})$ in the four Fermi model in arbitrary dimensions*, *Int. J. Mod. Phys. A* **9** (1994) 727–744, [hep-th/9306107].
- [45] A. C. Petkou, *Operator product expansions and consistency relations in a $O(N)$ invariant fermionic CFT for $2 < d < 4$* , *Phys.Lett.* **B389** (1996) 18–28, [hep-th/9602054].
- [46] M. Moshe and J. Zinn-Justin, *Quantum field theory in the large N limit: A Review*, *Phys.Rept.* **385** (2003) 69–228, [hep-th/0306133].

- [47] A. N. Manashov and M. Strohmaier, *Correction exponents in the Gross–Neveu–Yukawa model at $1/N^2$* , *Eur. Phys. J. C* **78** (2018) 454, [1711.02493].
- [48] J. A. Gracey, *Fermion bilinear operator critical exponents at $O(1/N^2)$ in the QED-Gross-Neveu universality class*, *Phys. Rev. D* **98** (2018) 085012, [1808.07697].
- [49] S. M. Tabatabaei, A.-R. Negari, J. Maciejko and A. Vaezi, *Chiral Ising Gross-Neveu Criticality of a Single Dirac Cone: A Quantum Monte Carlo Study*, *Phys. Rev. Lett.* **bfseries 128** (2022) 225701, [2112.09209].
- [50] E. Huffman and S. Chandrasekharan, *Fermion-bag inspired Hamiltonian lattice field theory for fermionic quantum criticality*, *Phys. Rev. D* **101** (2020) 074501, [1912.12823].
- [51] Y. Liu, W. Wang, K. Sun and Z. Y. Meng, *Designer Monte Carlo simulation for the Gross-Neveu-Yukawa transition*, *Phys. Rev. B* **101** (2020) 064308, [1910.07430].
- [52] J. Rong and N. Su, *Bootstrapping the minimal $\mathcal{N} = 1$ superconformal field theory in three dimensions*, *JHEP* **06** (2021) 154, [1807.04434].
- [53] A. Atanasov, A. Hillman and D. Poland, *Bootstrapping the Minimal 3D SCFT*, *JHEP* **11** (2018) 140, [1807.05702].
- [54] A. Atanasov, A. Hillman, D. Poland, J. Rong and N. Su, *Precision Bootstrap for the $\mathcal{N} = 1$ Super-Ising Model*, *JHEP* **08** (2022) 136, [2201.02206].
- [55] E. Huffman and S. Chandrasekharan, *Fermion bag approach to Hamiltonian lattice field theories in continuous time*, *Phys. Rev. D* **96** (2017) 114502, [1709.03578].
- [56] A. L. Fitzpatrick, J. Kaplan, D. Poland and D. Simmons-Duffin, *The Analytic Bootstrap and AdS Superhorizon Locality*, *JHEP* **1312** (2013) 004, [1212.3616].
- [57] Z. Komargodski and A. Zhiboedov, *Convexity and Liberation at Large Spin*, *JHEP* **1311** (2013) 140, [1212.4103].

- [58] N. Chai, S. Chakraborty, M. Goykhman and R. Sinha, *Long-range fermions and critical dualities*, *JHEP* **01** (2022) 172, [2110.00020].
- [59] C. Behan, L. Rastelli, S. Rychkov and B. Zan, *Long-range critical exponents near the short-range crossover*, *Phys. Rev. Lett.* **118** (2017) 241601, [1703.03430].
- [60] C. Behan, L. Rastelli, S. Rychkov and B. Zan, *A scaling theory for the long-range to short-range crossover and an infrared duality*, *J. Phys. A* **50** (2017) 354002, [1703.05325].
- [61] C. Behan, *Bootstrapping the long-range Ising model in three dimensions*, *J. Phys. A* **52** (2019) 075401, [1810.07199].
- [62] S. Benvenuti and H. Khachatryan, *QED's in 2+1 dimensions: complex fixed points and dualities*, 1812.01544.
- [63] R. Boyack, A. Rayyan and J. Maciejko, *Deconfined criticality in the QED3 Gross-Neveu-Yukawa model: The 1/N expansion revisited*, *Phys. Rev. B* **99** (2019) 195135, [1812.02720].
- [64] L. Janssen and I. F. Herbut, *Antiferromagnetic critical point on graphene's honeycomb lattice: A functional renormalization group approach*, *Phys. Rev. B* **89** (2014) 205403, [1402.6277].
- [65] G. P. Vacca and L. Zambelli, *Multimeson Yukawa interactions at criticality*, *Phys. Rev. D* **91** (2015) 125003, [1503.09136].
- [66] S. Chandrasekharan and A. Li, *Quantum critical behavior in three dimensional lattice Gross-Neveu models*, *Phys. Rev. D* **88** (2013) 021701, [1304.7761].
- [67] L. Wang, P. Corboz and M. Troyer, *Fermionic Quantum Critical Point of Spinless Fermions on a Honeycomb Lattice*, *New J. Phys.* **16** (2014) 103008, [1407.0029].
- [68] Z.-X. Li, Y.-F. Jiang and H. Yao, *Fermion-sign-free Majarana-quantum-Monte-Carlo studies of quantum critical phenomena of Dirac fermions in two dimensions*, *New J. Phys.* **17** (2015) 085003, [1411.7383].

- [69] S. Hesselmann and S. Wessel, *Thermal Ising transitions in the vicinity of two-dimensional quantum critical points*, *Phys. Rev. B* **93** (2016) 155157, [1602.02096].
- [70] J. A. Gracey, *Calculation of exponent η to $O(1/N^{**2})$ in the $O(N)$ Gross-Neveu model*, *Int. J. Mod. Phys. A* **6** (1991) 395–408.
- [71] A. N. Vasiliev, S. E. Derkachov, N. A. Kivel and A. S. Stepanenko, *The $1/n$ expansion in the Gross-Neveu model: Conformal bootstrap calculation of the index η in order $1/n^{**3}$* , *Theor. Math. Phys.* **94** (1993) 127–136.
- [72] A. N. Vasiliev and A. S. Stepanenko, *The $1/n$ expansion in the Gross-Neveu model: Conformal bootstrap calculation of the exponent $1/\nu$ to the order $1/n^{**2}$* , *Theor. Math. Phys.* **97** (1993) 1349–1354.
- [73] J. A. Gracey, *Computation of Beta-prime ($g(c)$) at $O(1/N^{**2})$ in the $O(N)$ Gross-Neveu model in arbitrary dimensions*, *Int. J. Mod. Phys. A* **9** (1994) 567–590, [hep-th/9306106].
- [74] J. A. Gracey, *Critical exponent ω in the Gross-Neveu-Yukawa model at $O(1/N)$* , *Phys. Rev. D* **96** (2017) 065015, [1707.05275].
- [75] J. A. Gracey, *Computation of the three loop Beta function of the $O(N)$ Gross-Neveu model in minimal subtraction*, *Nucl. Phys. B* **367** (1991) 657–674.
- [76] C. Luperini and P. Rossi, *Three loop Beta function(s) and effective potential in the Gross-Neveu model*, *Annals Phys.* **212** (1991) 371–401.
- [77] F. D. M. Haldane, *Model for a Quantum Hall Effect without Landau Levels: Condensed-Matter Realization of the 'Parity Anomaly'*, *Phys. Rev. Lett.* **61** (1988) 2015–2018.
- [78] M. A. Metlitski, L. Fidkowski, X. Chen and A. Vishwanath, *Interaction effects on 3d topological superconductors: surface topological order from vortex condensation, the 16 fold way and fermionic kramers doublets*, *arXiv preprint arXiv:1406.3032* (2014) .
- [79] A. Dymarsky, F. Kos, P. Kravchuk, D. Poland and D. Simmons-Duffin, *The 3d Stress-Tensor Bootstrap*, *JHEP* **02** (2018) 164, [1708.05718].

- [80] A. Dymarsky, J. Penedones, E. Trevisani and A. Vichi, *Charting the space of 3D CFTs with a continuous global symmetry*, *JHEP* **05** (2019) 098, [[1705.04278](https://arxiv.org/abs/1705.04278)].
- [81] K.-i. Kubota and H. Terao, *Dynamical symmetry breaking in QED(3) from the Wilson RG point of view*, *Prog. Theor. Phys.* **105** (2001) 809–825, [[hep-ph/0101073](https://arxiv.org/abs/hep-ph/0101073)].
- [82] S. Mehrotra, *On the implementation of a primal-dual interior point method*, *SIAM Journal on Optimization* **2** (1992) 575–601, [<https://doi.org/10.1137/0802028>].
- [83] M. Yamashita, K. Fujisawa and M. Kojima, *Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0)*, *Optimization Methods and Software* **18** (2003) 491–505.
- [84] M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi et al., *A high-performance software package for semidefinite programs: Sdpa 7, Tokyo, Japan* (2010) .
- [85] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata and M. Nakata, *Latest developments in the sdpa family for solving large-scale sdps*, *Handbook on semidefinite, conic and polynomial optimization* (2012) 687–713.
- [86] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, *Mathematical programming* **77** (1997) 301–320.
- [87] F. Leibfritz and E. Mostafa, *An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems*, *SIAM Journal on Optimization* **12** (2002) 1048–1074.
- [88] R. Correa, *A global algorithm for nonlinear semidefinite programming*, *SIAM Journal on optimization* **15** (2004) 303–318.
- [89] M. Kočvara and M. Stingl, *Pennon: A code for convex nonlinear and semidefinite programming*, *Optimization methods and software* **18** (2003) 317–333.
- [90] H. Yamashita and H. Yabe, *A survey of numerical methods for nonlinear semidefinite programming*, *Journal of the Operations Research Society of Japan* **58** (2015) 24–60.

- [91] Wikipedia contributors, *Broyden–fletcher–goldfarb–shanno algorithm* — *Wikipedia, the free encyclopedia*, 2023.
- [92] N. Su, *The Hybrid Bootstrap*, 2202.07607.
- [93] S. M. Chester, W. Landry, J. Liu, D. Poland, D. Simmons-Duffin, N. Su et al., *Bootstrapping heisenberg magnets and their cubic instability*, *Phys. Rev. D* **104** (Nov, 2021) 105013.
- [94] M. Go and Y. Tachikawa, *autoboot: A generator of bootstrap equations with global symmetry*, *JHEP* **06** (2019) 084, [1903.10522].
- [95] N. Afkhami-Jeddi, T. Hartman and A. Tajdini, *Fast Conformal Bootstrap and Constraints on 3d Gravity*, *JHEP* **05** (2019) 087, [1903.06272].