PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à École normale supérieure - PSL

Développement et application de nouvelles méthodes pour le bootstrap conforme numérique

Soutenue par
Benoit Sirois
Le 21 septembre 2023

École doctorale n°564
Physique en Île-de-France

Spécialité
Physique statistique / des hautes énergies

Composition du jury :

Kay J. WIESE Directeur de recherche, École normale supérieure	*Président*
David POLAND Associate Professor, Université Yale	*Rapporteur*
Alessandro VICHI Associate Professor, Université de Pise	*Rapporteur*
Yifei HE Chargée de recherches, École normale supérieure	*Examinateur*
Marco Serone Professeur, SISSA	*Examinateur*
Slava Rychkov Professeur permanent, IHES	*Directeur de thèse*

1794
ENS | PSL

Contents

Abstract

For more than a decade, the numerical conformal bootstrap has proved to be an invaluable tool for the study of higher-dimensional Conformal Field Theories (CFTs). The conformal bootstrap program being so young, each year more and more intricate and interesting questions are tackled thanks to the consistent improvement of its arsenal of numerical methods. We propose in this thesis the development of a novel numerical method and in tandem, we make use of it to study questions previously out of reach. We introduce so-called "navigator functions". These functions are defined so as to give a measure of how allowed or disallowed a putative subset of CFT data is. We propose that this measure should be used to orient the search for allowed regions of CFT data, leading in most, if not all cases, to drastic improvements in efficiency.

We first use the 3d critical Ising model as a testing ground. We show how the search for any allowed point, or for an extremal point along the boundary of an allowed region, can be translated to an optimization problem for the navigator function. To solve this problem efficiently, we adapt existing optimization algorithms for the needs of the bootstrap. Any efficient optimization algorithm requires the computation of the function to be optimized, as well as some of its derivatives. We show that the value of a navigator function at any point in the space of CFT data is obtained by solving a semidefinite program (SDP) of the same type as is customary in the bootstrap, and that derivatives may be obtained, for free, through a clever analysis of the perturbation of the optimal solution to an SDP. One peculiar thing is that each navigator function naturally defines a "most allowed" point for a particular bootstrap setup. We flow to this point in the case of the mixed-correlator bootstrap of σ and ϵ, and observe that it sits much closer to the true location of the Ising model than a generic allowed point.

Many interesting questions can also be asked not about a single CFT, but about a *family* of CFTs. A prototypical family of CFTs is the critical $O(N)$

model, and we choose it in the second part of this thesis to demonstrate how the navigator function can be used to study families of CFTs more efficiently. One can use the navigator function and its derivatives at the minimal navigator point of a particular member of the family to predict the location of the minimal navigator point of a neighboring member of the family. This allows to efficiently follow allowed islands through N (and also any other external parameters, for example the spacetime dimension d). Using this trick, we observe, to our knowledge for the first time in the numerical bootstrap program, the clear disappearance of an allowed island. We see that the 3d $O(N)$ model disappears as $N \to 1^+$, and we are able to understand this disappearance as caused by a severe loss of unitarity below $N = 1$. We are able to match this observation with perturbative field theory methods like the ϵ-expansion.

We finish this thesis by tackling a decades-old problem in the theory of classical phase transitions: what is the nature of the phase transition (whether first or second order) of the stacked triangular antiferromagnet, perhaps the simplest of all frustrated magnets? The critical modes of this model should be described by a multiscalar Landau-Ginzburg-Wilson Lagrangian with $O(N) \times O(2)$ symmetry, where N corresponds to the number of spin components. Near its upper critical dimension, it is well known that this model exhibits a conformal window, where a nontrivial CFT with $O(N) \times O(2)$ global symmetry exists only down to some critical value $N_c(d)$. With the help of the navigator function, we are able to efficiently follow the point of disappearance of the unitary solution to the bootstrap crossing equations accross d. We are thus able to obtain a full critical curve, which we compare with those obtained from other analytical and numerical approaches like the ϵ-expansion, the fixed-dimensional RG, Monte Carlo and the Exact Renormalization Group (ERG).

Résumé

Pour plus d'une décennie, le bootstrap conforme numérique s'est avéré être un outil indispensable à l'étude des théories des champs conformes (CFTs). La méthode étant encore dans ses jeunes années, des problèmes de plus en plus difficiles et intéressants peuvent être abordés chaque année grâce à l'amélioration constante de son arsenal de méthodes numériques. Nous proposons dans cette thèse le développement d'une nouvelle méthode numérique, que nous appliquons ensuite à l'étude de questions qui auparavant auraient été pénibles à analyser. Nous introduisons des "fonctions de navigation". Ces fonctions donnent une mesure du degré auquel un ensemble de *CFT data* présumé est permis ou exclu. On propose que cette mesure devrait orienter la recherche de régions permises de *CFT data*, ce qui devrait mener dans la plupart, sinon tous les cas, à des améliorations drastiques en performance.

Nous utilisons pour commencer le modèle d'Ising critique en 3d comme modèle jouet. Nous montrons comment la recherche d'un point permis, ou celle d'un point aux extrêmes de la frontière de la région permise, peut être traduite en un problème d'optimisation de la fonction de navigation. Pour résoudre ce problème, nous adaptons des algorithmes existants aux besoins du bootstrap. Quelconque algorithme d'optimisation efficace requerra la détermination de la valeur de la fonction à optimiser ainsi que certaines de ses dérivées. Nous démontrons que le calcul de la fonction de navigation est équivalent à la résolution d'un programme semidéfini (SDP) typique dans le bootstrap conforme, et que les dérivées peuvent être obtenues gratuitement par l'analyse de la perturbation de solutions optimales d'un SDP. Nous notons aussi que chaque fonction de navigation définie naturellement un point "le plus permis" à l'intérieur d'une région permise. Nous naviguons jusqu'à ce point pour le bootstrap mixte des opérateurs σ et ϵ, et nous observons qu'il se trouve bien plus près du modèle d'Ising qu'un point permis générique.

Plusieurs questions intéressantes peuvent concerner non pas une seule CFT,

mais une *famille* de CFTs. L'exemple typique d'une telle famille est le modèle $O(N)$ critique, et nous l'utilisons dans la deuxième partie de cette thèse pour démontrer comment la fonction de navigation peut être utilisée pour rendre l'étude de familles de CFTs plus efficace. La valeur de la fonction de navigation et de ses dérivées en son minimum pour un membre de la famille peuvent être utilisées pour prédire la position du minimum d'un membre voisin de la famille. Ceci permet de suivre efficacement la position de régions permises comme fonction de N (ou de quelconque autre paramètre externe, par exemple la dimensionalité de l'espace d). En utilisant ce truc, nous observons, à notre connaissance pour la première fois dans l'histoire du bootstrap conforme, la disparition d'une île permise. Nous observons que le modèle $O(N)$ critique en 3d disparaît lorsque $N \to 1^+$, phénomène qu'on explique par la perte sévère d'unitarité dans ce régime. Cette observation est en accord avec des méthodes perturbatives comme l'ϵ-*expansion*.

Nous finissons cette thèse en s'attaquant à un problème non résolu depuis des décennies dans le domaine des transitions de phase classiques: quelle est la nature (de premier ou second ordre) de la transition de phase des *stacked triangular antiferromagnets*, possiblement les plus simples de tous les matériaux magnétiques frustrés? Les fluctuations critiques de ce modèle devraient être décrites par un lagrangien LGW avec symétrie globale $O(N) \times O(2)$, où N réfère au nombre de composantes des spins. Proche de la dimension critique supérieure, ce modèle possède une fenêtre conforme, où une CFT non triviale existe seulement au-dessus d'une valeur critique $N_c(d)$. Avec l'aide de la fonction de navigation, nous sommes capables de suivre efficacement le point de disparation de la solution unitaire aux équations de croisement à travers d. Nous sommes ainsi capables de générer une courbe critique qui peut être comparée à celles obtenues par d'autres approches analytiques et numériques comme l'ϵ-*expansion*, le RG en dimension fixe, Monte Carlo ou le RG exact.

Remerciements

Je tiens pour commencer bien sûr à remercier Slava. Sa connaissance encyclopédique de chaque racoin de la physique m'impressionnera toujours: pas un lunch ne se déroule entre les membres de son groupe sans qu'on soit introduit à toutes les subtilités d'un nouveau problème. Il est rare que j'aie une question auquel il ne peut répondre, et encore plus rare que j'aie raison sur une question où nos intuitions divergent. Par-dessus tout, je lui suis reconnaissant pour avoir été un superviseur hors-pair. Slava collabore et s'investit réellement avec chacun de ses étudiants. J'ai été lancé tête première dans un projet passionnant dès mon arrivée, et malgré mon peu de connaissance, je m'y suis senti membre à part entière. Il s'est assuré, chaque semaine pendant trois ans déjà, de dédier au moins un moment pour qu'on puisse discuter de l'avancement de nos projets. C'est de cette implication que je me rappellerai le plus. Merci encore, et j'espère que tu accepteras un jour de mettre du sirop d'érable sur tes crêpes au lieu de le manger à la cuillère, c'est vraiment meilleur.

Je tiens ensuite à remercier tous les gens avec qui j'ai eu la chance de collaborer ou de discuter de physique: merci à Marten, Jiaxin, Junchen, Aleix, Nikolay, Gela, Johan, Balt van Rees et David Simmons-Duffin, et j'en oublie fort probablement. Je dois un merci particulier à Ning, sans qui il m'aurait pris un temps exponentiellement plus long à m'acclimater aux méthodes numériques du bootstrap conforme.

J'aimerais aussi remercier Andres, Ludovic, Hugo, et tout le monde que je suis toujours heureux de voir quand je passe par le Québec, quand ils passent par la France, ou par quelconque autre forme de communication lorsqu'on est séparé.

Je remercie le Fonds de Recherche du Québec – Nature et technologies, la *Simons Collaboration on the Nonperturbative Bootstrap* et *Mitsubishi Heavy Industries*, à travers une chaire MHI-ENS, pour leur soutien financier.

Je dois finalement remercier ma famille. À Jean, Monique, et Émile, merci. Je

dois bien sûr, de par leur amour et support inconditionnels, tout à mes parents. Nos appels familiaux quasi-quotidiens égaient toujours mes journées, même si à chaque fois ils me rappellent la distance qui me sépare de chez moi.

Introduction

It has now been almost exactly half a decade since Polyakov's proposal of the use of "bootstrapy" ideas in Conformal Field Theory [1], inspired by the earlier advent of the bootstrap philosophy in S-matrix theory [2–4]. In Polyakov's own words, he envisioned a program which

> *consists in calculating all the functions C (to be defined) to within a few constants, substituting the operator expansion into the four-point function and finding the unknown constants from the crossing-symmetry requirement. On the basis of this program, a system of equations for the anomalous dimensions and the interaction constants at short distances will be obtained directly [1].*

The requirement of consistency of correlation functions with general principles like crossing-symmetry and the operator product expansion could be translated into a computational framework for determining conformal data distinct from traditional perturbative field theory methods. As Polyakov admitted, the equations "*unfortunately (. . .) are found to be extremely complicated* [1]". Nevertheless, the first success of the program would come not long after, for conformal field theories in two dimensions [5]. One could harness the power of the Virasoro symmetry (special to two dimensions) to reduce in special cases to a finite set of simple bootstrap equations one could solve exactly. It was well understood at that time that "*in the multidimensional theory ($D > 2$), (the whole) system turns out to be too complicated to be solved exactly* [5]". Because of this, it took several decades for a conformal bootstrap program in higher dimensions to flourish, starting with [6]. The previously missing ingredient was the realization that the complicated system of bootstrap equations could be efficiently approximated and solved numerically.

Since this revival, the conformal bootstrap has likely surpassed Polyakov's original vision and grown to a fully fledged industry which tackles questions about disparate subjects like the structure of the spectrum of conformal field theories [7, 8],

supersymmetric field theories [9] or holography [10][1]. Advances are plentiful both on the analytical [12] and numerical fronts [13]. The bootstrap philosophy also transcends the correlation function bootstrap in conformal field theories, with the modular bootstrap [14–16], the revival of the S-matrix bootstrap [17], the quantum mechanical bootstrap [18], or the bootstrap of lattice models [19, 20].

We will be concerned in this thesis with the application of the conformal bootstrap to the study of critical phenomena, a subject also appearing in the original work of BPZ [5]. Famously, equilibrium classical statistical mechanical models (as well as *some* quantum and non-equilibrium models) that undergo continuous phase transitions exhibit a scale invariance isotropic in space(time) at their critical point which is generically enhanced to conformal invariance[2]. Experiments on critical systems can suffer from many ailments such as impurity of samples, large corrections-to-scaling, or weakly first-order behavior, to name only a few. Furthermore, the boom in experiments that surrounded the birth of the Renormalization Group [23–26] has died down significantly, which has left many interesting puzzles with relatively old experimental data. From the theoretical point of view, standard perturbative field theory methods often fail to clarify experiments since they require ill-controlled extrapolations (e.g. $\epsilon \to 1$ in the ϵ-expansion) and are just inadequate at reproducing nonperturbative behavior (an example very relevant to this thesis is the merger and annihilation of fixed points). The numerical conformal bootstrap emerges as a worthwhile tool to investigate critical phenomena, as the method does involve truncations, *but* is entirely nonperturbative and rigorous[3]. To date, its track record is quite impressive in this domain: notably, it has proved the instability of the $O(3)$ critical point to cubic perturbations [27], and has resolved a long-standing discrepancy between Monte Carlo simulations of the $O(2)$ model and ultra-precise experiments on the superfluid transition in ^{4}He performed in space [28, 29]! It has also produced impressively precise determinations of the main critical exponents of the Ising and $O(N)$ universality classes [30], and more recently, of the $O(N)$ GNY universality class [31], all relevant to the critical behavior of various statistical and quantum systems, from binary mixtures to polymers and superconductors [31, 32].

[1] and all that, regardless of its validity as a scientific mode of enquiry [11]!

[2] See [21] for a short discussion, and [22] for a longer one.

[3] A rigor which is unique to the bootstrap when compared with other numerical methods like Monte Carlo and the Nonperturbative Renormalization Group.

How does the bootstrap work?

Conformal transformations in d dimensions are those which preserve angles between intersecting curves. Locally, they look like compositions of translations, rotations, scale transformations (or dilatations) and so-called special conformal transformations. In Conformal Field Theories, fields transform under given irreps of $SO(d)$ and scale under dilatations with given exponents Δ called *scaling dimensions*. Conformal symmetry furthermore arranges fields into families. These are comprised of one *primary* field and *descendant* fields, obtained from their associated primary by differentiation. A natural quantity of interest in field theories are correlation functions. Those of descendants can simply be derived from those of their associated primary, so one can concentrate only on the latter. 2 and 3-point functions are actually fixed by conformal symmetry up to numerical factors referred to as the *CFT data*. For example, for scalar primaries they assume the form

$$\langle \phi_i(x_1)\phi_j(x_2)\rangle = \frac{\delta_{ij}}{x_{12}^{2\Delta_i}} \quad , \quad x_{ij} = |x_i - x_j| \tag{1a}$$

$$\langle \phi_i(x_1)\phi_j(x_2)\phi_k(x_3)\rangle = \frac{\lambda_{ijk}}{x_{12}^{\Delta_{ijk}} x_{23}^{\Delta_{jki}} x_{31}^{\Delta_{kij}}} \quad , \quad \Delta_{ijk} = \Delta_i + \Delta_j - \Delta_k \tag{1b}$$

The CFT data is thus comprised of the scaling dimensions previously mentioned as well as the *structure constants* λ_{ijk}. As we will see shortly, any N-point function is a functional of this CFT data, and so "solving a CFT" is equivalent to determining its operator content and the associated CFT data.

Higher point functions are less constrained: they are fixed up to arbitrary functions of conformally-invariant cross-ratios[4]. For example, the 4-point function of identical scalar primaries takes the form

$$\langle \phi(x_1)\phi(x_2)\phi(x_3)\phi(x_4)\rangle = \frac{g(u,v)}{x_{12}^{2\Delta_\phi} x_{34}^{2\Delta_\phi}} \quad , \tag{2}$$

where the function g depends on the two cross-ratios $u = \frac{x_{12}^2 x_{34}^2}{x_{13}^2 x_{24}^2}$, $v = \frac{x_{14}^2 x_{23}^2}{x_{13}^2 x_{24}^2}$. Conformal symmetry would in principle allow any such function. The idea of the conformal bootstrap is to impose another condition, crossing-symmetry, on the correlation function in order to constrain possible values of $g(u,v)$. Crossing-symmetry is simply the statement that (2) is invariant under permutation of the

[4]There are $\frac{n(n-3)}{2}$ such cross-ratios for an n-point function [33] in large enough d.

N fields. For the 4 point function above, the only useful constraint turns out to arise from the $x_1 \leftrightarrow x_3$ permutation, which gives

$$g(u,v) = \left(\frac{u}{v}\right)^{\Delta_\phi} g(v,u) \quad . \tag{3}$$

Similar constraints can be derived from the permutation symmetry of higher point functions, although as will become clear why in just a moment, the bootstrap usually restricts to the study of 4 point functions[5]. One more ingredient is needed in order to translate these into constraints on the CFT data. Away from other operators, the product of two operators can be written as an infinite linear combination of single operator insertions. This is the *Operator Product Expansion (OPE)*. It reads [36]

$$\mathcal{O}_i(x_1)\mathcal{O}_j(x_2) = \sum_{\mathcal{O}_k \in \mathcal{O}_i \times \mathcal{O}_j} \lambda_{ijk} \hat{f}(x_1, x_2, \partial_2)\mathcal{O}_k(x_2) \quad , \tag{4}$$

where λ_{ijk} are the structure constants of (1b)[6] and $\hat{f}$ is a differential operator that packages the contribution of the primary $\mathcal{O}_k$ and all of its descendants. Using the 12 and 34 OPEs inside of (2), and using the diagonal structure of the two point functions (1a), we can express the arbitrary function $g(u,v)$ as a single infinite sum over all primaries [37]:

$$g(u,v) = \sum_{\mathcal{O} \in \phi \times \phi} \lambda^2_{\phi\phi\mathcal{O}}\, g_{\mathcal{O}}(u,v) \quad , \tag{5}$$

where the $g_{\mathcal{O}}$s are special functions called *conformal blocks*, which depend on the quantum numbers of the primary operator $\mathcal{O}$. These are known functions[7], which means that using this decomposition in (3) gives explicit constraints on OPE coefficients and scaling dimensions.

There is little hope of solving this equation exactly since it involves an infinite sum over operators of the CFT. The typical way to make progress is to instead try to exclude incompatible CFT data. In this simple example, there is only one nontrivial crossing equation, but in general when one may consider theories with global symmetries or constraints from multiple correlators, one typically gets a set

[5] Some attention has been given in recent years to higher point functions [34, 35], as they repackage constraints on scaling dimensions and structure constants, and thus make certain information more easily accessible.

[6] From now on, we will also refer to them as *OPE coefficients*.

[7] They are the solution to a specific PDE [38].

of crossing equations which can be cast into the general form[8]

$$\sum_{\mathcal{O}} \boldsymbol{\lambda}_{\mathcal{O}}^{T} \boldsymbol{\mathcal{F}}_{\mathcal{O}}(u,v)\, \boldsymbol{\lambda}_{\mathcal{O}} = \mathbf{0} \quad . \tag{6}$$

$\boldsymbol{\lambda}_{\mathcal{O}}$ is a vector of OPE coefficients involving the operator $\mathcal{O}$, and $\boldsymbol{\mathcal{F}}_{\mathcal{O}}(u,v)$ is a vector of matrices, each element of these matrices being a certain transformation (called "convolution" in the bootstrap literature [40]) of conformal blocks. To exclude incompatible data, one makes some assumptions on what operators can enter the sum and verifies if any set of CFT data respecting these assumptions can solve (6). Crucially, to make any progress, one needs the additional ingredient of *unitarity*. In unitary CFTs, entries of $\boldsymbol{\lambda}_{\mathcal{O}}$ are real. Using this fact, the aim is to transform (6) into a sum of positive terms: if we can find a vector $\boldsymbol{\alpha} = \alpha_i$ of functionals on the space of matrices of functions of (u,v) which applied to $\boldsymbol{\mathcal{F}}_{\mathcal{O}}(u,v)$ gives a positive semidefinite matrix for all $\mathcal{O}$ respecting our assumptions, and a strictly positive definite matrix for one operator known to be part of the theory of interest (e.g. the identity, or the stress-tensor for a local CFT), (6) cannot be satisfied and we can exclude this set of assumptions. This is the essence of the modern numerical bootstrap.

In practice, each α_i is restricted to a finite subspace of all functionals which can naturally be enlarged to obtain stronger and stronger constraints. This subspace is usually taken as the space of derivative expansions around the point $(z,\bar{z}) = (\frac{1}{2},\frac{1}{2})$[9]:

$$\alpha_i = \sum_{m+n\leqslant\Lambda} \alpha^i_{m,n} \partial_z^m \partial_{\bar{z}}^n|_{z=\bar{z}=\frac{1}{2}} \quad . \tag{7}$$

These derivatives acting on elements of $\boldsymbol{\mathcal{F}}_{\mathcal{O}}(u,v)$ are well approximated by positive factors $\times$ polynomials in $\Delta_{\mathcal{O}}$ [41]. This fact can be used to translate the search for coefficients $\alpha^i_{m,n}$ invalidating our assumptions into a so-called Polynomial Matrix Program, and eventually into a semidefinite program. This type of problem can be solved by `SDPB`, a semidefinite program solver built specifically to solve the semidefinite programs encountered in the bootstrap [42, 43].

As an example of what a bootstrapper may want to do, he/she may assume there exists a certain operator with a given scaling dimension in his/her theory of interest, and scan over possible values of this dimension to check which is disallowed by (6). Each point in this scan corresponds to an independent search for

[8] We are stealing here the notation of [39].

[9] The $(z,\bar{z})$ coordinates are related to (u,v) by $u = z\bar{z}$, $v = (1-z)(1-\bar{z})$.

a functional, and at the end of each search the bootstrapper gains only the binary information "allowed/disallowed". We referred in [44] to this scanning-based method as bootstrapping in oracle mode.

Why the need for a navigator function?

The methods laid out above have successfully led to the determination of CFT data for a number of interesting theories, possibly the most famous example being the rigorous determination of the main critical exponents of the 3D Ising model from a mixed-correlator bootstrap involving the lowest-lying $\mathbb{Z}_2$-odd and even scalars σ and ϵ [45]. In that case, finding an allowed island in $(\Delta_\sigma, \Delta_\epsilon)$-space associated to the Ising model could easily be achieved in oracle mode: even a fine two-dimensional grid will have relatively few points, and the location of the island was well known to high precision from other methods like resummations of high-loop perturbative expansions [46] or Monte Carlo [47]. It is clear though that we might not always get so lucky:

- Ideally, we would like to use the bootstrap to solve theories that are not already convincingly solved by other methods. We may want to study theories where perturbation theory is only known to a few orders, or want to bootstrap nonperturbative phenomena, like the collision of fixed points, which would not be captured by perturbation theory. In that case, we would have only a vague idea of where we want to look.

- We may be interested in "more complicated" CFTs. For theories with a richer global symmetry structure, for example the zoo of multiscalar theories [48, 49], or theories involving a mix of bosonic and fermionic operators [31], isolating into small allowed regions may require bootstrapping more correlators, or imposing assumptions in more channels, or both. This will usually translate into searching in a larger-dimensional space.

- Mapping out the full shape of an allowed region has recently been made more efficient thanks to useful tricks like Delaunay triangulation and the cutting surface algorithm [29]. Most times however, we are not interested in plotting out the whole island. Maybe we want only to know if there is *any* allowed point, or we want a *specific* one, e.g. an extremal one on the boundary of the allowed region. In those cases, even with the tricks above, oracle mode bootstrap seems overkill.

Gliozzi first proposed an alternative way to bootstrap by truncating the crossing equation and directly looking for solutions [50]. Further refinements have included the use of Monte Carlo [51] or Machine Learning [52] to search more efficiently for (approximate) solutions of the truncated crossing equation. While these methods can help tackle some of the points above, sometimes impressively so for the second point, they unfortunately part ways with the *rigor* that makes the traditional bootstrap so unique in the first place. We will in this thesis present a new method for the traditional "bounds-based" bootstrap which also addresses the points above.

Frustrated magnets

As promised, the final part of this thesis will concern the long-standing problem of the order of phase transitions in classical frustrated noncollinear magnets. The theoretical study of noncollinear magnets dates all the way back to the 1970s[10]. Since then, they have been discussed countless times in the literature, and for this reason, myself and the rest of the authors of [60] had decided to postpone a thorough exposition of the more basic concepts until here.

The archetypal magnetic phase transition is the paramagnetic-to-ferromagnetic transition. To model it, we can place N-component spins σ_i at each vertex i of a lattice and make them interact with each other according to the Hamiltonian

$$\mathcal{H}[\{\sigma_i\}] = -J(T) \sum_{\langle ij \rangle} \sigma_i \cdot \sigma_j \quad , \tag{8}$$

with $J(T) > 0$ a positive, temperature-dependent coupling. This system will display distinct behaviors at low and high temperatures. At low temperatures, the system wants to minimize energy, leading to a trivially ordered state where all spins point in the same direction. At high temperature, the system wants to maximize entropy, leading to a disordered state. For every $d > 2$, a continuous phase transition takes place at some T_c, whose critical behavior falls under the standard $O(N)$ universality class.

In many magnetic systems however, frustration at low temperatures leads to nontrivially ordered states. The phase transition of such systems will generically fall under a universality class distinct from $O(N)$.

Frustration can be achieved by introducing some degree of antiferromagnetism.

[10] See [53–58] for a nonexhaustive list of foundational works, and/or [59] for a nice review.

Changing $J \to -J$ in (8) will favor anti-alignment of spins at low temperatures. On the hypercubic lattice $\mathbb{Z}^d$, every spin can be exactly anti-aligned with its neighbors, so there is no frustration and the phase transition is still in the $O(N)$ universality class. One type of frustrated magnet which exhibits distinct critical behavior is the so-called *Stacked Triangular Antiferromagnet* (STA). To model it, spins are placed at the vertices of planar triangular lattices stacked on top of one another in the remaining $d-2$ directions, with antiferromagnetic nearest-neighbor intraplane interactions. For continuous spins, the ordered state will have the 120° structure in each plane shown in Fig. 1. Frustration can also be obtained directly in $\mathbb{Z}^d$ by adding to (8) a competing antiferromagnetic next-to-nearest neighbor interaction along a given direction. This construction models so-called *helical magnets*, and together with STAs, they have been referred to in the literature as *noncollinear magnets*.

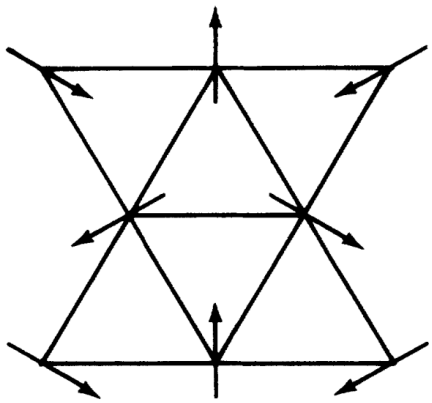


Figure 1: Graphical depiction of the 120° structure of the ordered state of an STA (taken from [58]).

Both systems near criticality enjoy the same continuum description. The procedure to obtain the correct Landau-Ginzburg-Wilson (LGW) Hamiltonian is straightforward and quite short (see Appendix A of [58]). One must first perform a Hubbard-Stratonovich transformation from the original lattice Hamiltonian to obtain a Hamiltonian in terms of new spins $\phi_i \in \mathbb{R}^N$. Throwing away terms higher than quartic in ϕ_i and expanding around every minimum of the free part of the Hamiltonian density in momentum space then yields $\mathcal{H}_{LGW}$. In the case of standard collinear magnets described by (8), there is only one minimum in momentum

space, leading to the well known ϕ^4-theory in terms of a *single* N-component field. For noncollinear magnets, there are *two* inequivalent minima, resulting in the $O(N) \times O(2)$ invariant LGW Hamiltonian

$$\mathcal{H}_{LGW} = \frac{1}{2}\left((\partial_\mu a)^2 + (\partial_\mu b)^2 + r_0\left(a^2+b^2\right) + u\left(a^2+b^2\right)^2 + v\left((a\cdot b)^2 - a^2b^2\right)\right) \tag{9}$$

in terms of *two* N-component fields $a(x)$ and $b(x)$.[11] At the mean-field level, we observe two different phase transitions to distinct ordered states depending on the sign of v_0. As the mass r_0 is tuned, (9) undergoes a phase transition from the disordered state with $\langle a\rangle = \langle b\rangle = 0$ to a non-trivially order state with $a \perp b$ for $v > 0$ or $a \parallel b$ for $v < 0$. The first ordered state reproduces the noncollinear behavior of Fig. 1, while the second describes so-called spin-density waves. The picture beyond mean-field is extremely rich, and we will attempt at the end of this thesis to better understand the nature of the phase transition taking place for $v > 0$ at the nonperturbative level.

[11] It may seem surprising that such anisotropic systems as the STAs and helimagnets are described near criticality by the isotropic effective Hamiltonian (9). Let us consider for example the case of STAs. After performing every step listed above in going from the microscopic Hamiltonian to $\mathcal{H}_{LGW}$, one ends up with a kinetic term of the form $\int_{\mathbb{R}^d} \mathrm{d}q\,(q_\parallel^2 + \frac{4}{3}\left|\frac{J_\perp}{J_\parallel}\right| q_\perp^2)\,(a(p)\cdot a(-p) + b(p)\cdot b(-p))$, where the $\parallel$ directions are those inside of the triangular lattice planes, and $\perp$ refers to the remaining $d-2$ directions [61]. We can see there remains an anisotropy depending on the relative strength of the intra-plane antiferromagnetic coupling $J_\parallel$ and inter-plane coupling $J_\perp$. However, as long as $J_\perp \neq 0$, this anistropy can be scaled away by a change of variables. We thank João Penedones for making us ponder about this point.

Chapter 1

Navigator Function for the Conformal Bootstrap*

* Published in essentially identical form as
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

1.1 Introduction and summary

Over the last decade, the numerical conformal bootstrap program[1] has relied on the idea [6] that for any point in CFT parameter space it is possible to check if the point is allowed or excluded by constructing positive linear functionals. In this chapter we will dramatically upgrade this idea, replacing the binary information "allowed/excluded" by a continuous measure of success, called a "navigator function." For excluded points, the navigator function will tell us how far we are from the allowed region. Minimizing the navigator, we will be able to quickly find the allowed region, starting from an excluded point. For allowed points, the navigator will tell us how far inside the allowed region we are, and navigator minima will be excellent predictors for the position of an actual CFT.

To describe what we have in mind in some detail, let X be an infinite-dimensional vector containing all parameters characterizing a CFT (i.e. all operator dimensions and OPE coefficients, bundled together). We split it as $X = (x, y)$ where $x \in \mathbb{R}^k$ are parameters we are especially interested in, and y contains all the rest. We also select a finite subset of the infinitely many bootstrap equations.

Most bootstrap computations performed so far proceeded in what one may call "oracle mode."[2] One picks a sequence of trial vectors $x_1, x_2, \ldots$ and asks for each of them if there is any y such that $X = (x_i, y)$ satisfies the selected subset of bootstrap equations. A bootstrap solver such as SDPB [42, 43] provides an answer: "allowed" or "excluded". By trying many x_i's, one maps out the allowed region.[3] Thus, we compute the characteristic function χ_R of the allowed region R (i.e. $\chi_R(x) = 1$ for $x \in R$ and $\chi_R(x) = 0$ otherwise). Experience shows that the boundary of the allowed region ∂R is typically smooth, apart from isolated points (kinks). This can guide the choice of future trial points and speed up the computation.[4] By trying many points, one zooms in on the boundary ∂R of the allowed region. Importantly, a single oracle query does not provide any information about whether one is close to or far from ∂R. Rather, one knows that one is close to ∂R if one can find two nearby trial points x_i and $x_{i'}$ such that they are on two

[1] See [36] for a thorough review, and [37, 62] for pedagogical introductions.

[2] In technical jargon referred to as "feasibility mode."

[3] Other typical bootstrap computations are OPE coefficient optimizations. Sometimes these computations allow to zoom in on actual CFTs, as e.g. c-minimization is conjectured to lead to the 3d Ising CFT [63].

[4] Other speed-up tricks include the cutting surface algorithm [29], which allows in some cases to use a single oracle computation to rule out not just one point but a large swath of the parameter space.

different sides of the boundary.

We will modify this setup so that a single SDPB run computes a continuous function $\mathcal{N}(x)$, called a *navigator*, which will give a more nuanced measure of success than simply "allowed/excluded." To be maximally useful, the navigator should have the following properties:

- $\mathcal{N}(x)$ is continuous and differentiable;
- $\mathcal{N}(x) > 0$ outside the allowed region R, and $\mathcal{N}(x) < 0$ inside R. In particular, $\mathcal{N}(x) = 0$ on the boundary ∂R;[5]
- $\mathcal{N}(x)$ should be defined not just in a tiny neighborhood of the allowed region but globally;
- The allowed region R should be a basin of attraction of the navigator function from a sizable neighborhood of R.

Assuming these nice properties, the navigator value will allow us to guess how far we are from the allowed region. We will also be able to reach the allowed region by starting from some initial trial point x_0 and by minimizing the navigator until we reach a point with negative $\mathcal{N}(x)$. We'd like to be optimistic and hope that the navigator has no local minima away from the allowed region where such a search may get stuck.

The idea of replacing the binary information of "oracle mode" with continuous information from solving an optimization problem is not completely new [64, 65]. Notably Ref. [65] emphasized the power of this idea to quickly determine the boundary of the allowed region once its approximate position is known, replacing bisection with the secant method.[6] A crucial difference here is our requirement that the navigator should be defined in a wide region and not only near the boundary, which will greatly increase the list of potential applications. This requirement is non-trivial and the early navigator avatars [64, 65] don't satisfy it (see Section 1.2.1).

In this chapter we will lay down the systematic theory of navigator functions by showing three important results:

[5] In the Level Set Method of computational geometry, such functions are called "level set functions" or "level set fields". Closely related are also "boundary defining functions" of differential geometry, which however are only required to be defined near the boundary.

[6] We will see below that the navigator derivative can be evaluated "for free," allowing to replace the secant method with the even faster Newton method.

1. First, we will show that navigators satisfying all of the above properties can indeed be found for a generic bootstrap problem. We will present both the general principle of their existence, and several explicit constructions (see Section 1.2). Please scroll down to Fig. 1.1 for a concrete navigator example in the mixed σ-ϵ bootstrap setup used to isolate the 3d Ising model. It has all the nice properties, and in particular a single minimum (within the range we show), located within the 3d Ising island. See Section 1.3 for more beautiful navigator plots.

2. Our navigators can be evaluated using standard conformal bootstrap software such as SDPB. In practical applications that we have in mind, it's important to know not just the navigator but also its gradient. Our second important result is a general "SDP gradient formula," Eq. (1.48). This formula shows that navigator gradient can be evaluated essentially for free once the navigator value has been computed using SDPB.

3. We foresee that one of the most important navigator applications will be to quickly look for allowed points, i.e. to "sail towards the Ising island," by minimizing the navigator. Naive minimization strategies, such as the gradient descent, are inefficient, getting stuck in narrow "valleys" of the navigator surface. Our third important result is to demonstrate how a quasi-Newton method—the BFGS algorithm [66]—successfully overcomes these difficulties (Section 1.5). This algorithm finds first the allowed region, and then the navigator minimum, in a relatively small number of steps.

The chapter is structured as follows. Section 1.2 will explain our two main navigator constructions: the GFF-navigator and the Σ-navigator. (A third construction is in App. A.2). In Section 1.3 we will show various plots of these navigators, to gain intuition about their shape. In Section 1.4 we will derive the SDP gradient formula. In Section 1.5 we will describe the BFGS algorithm and its bounding-box modification, to look for an allowed point and the navigator minimum, and show that it performs well in realistic multiple-correlator setups. In Section 1.6 we describe another possible navigator application: extremizing operator dimension within the allowed region. This represents an attractive alternative to the `tiptop` algorithm recently introduced for this purpose in the feasibility setup [27]. In Section 1.7 we conclude. Appendix A.3 shows how one can also evaluate the navigator Hessian, in addition to the gradient, provides numerical tests of these procedures.

1.2 Navigator function

Our motivation to look for the navigator function, and its desired properties, have already been described in the introduction. The crucial requirement is that the navigator should be *finite*. Indeed, a navigator which is negative inside the allowed region and equals $+\infty$ outside would be rather useless for the purposes we have in mind, such as looking for an allowed point starting from an excluded one. Furthermore, once a finite navigator is constructed, other nice properties turn out to also be satisfied.

How to get a robustly finite navigator is one of the main ideas of this chapter (see Section 1.2.1 for an account of naive attempts which fail). Although the idea is general, we will start in Section 1.2.1 by presenting it in the simplest single-correlator setup. We will then move on to more realistic multiple-correlator problems.

1.2.1 Single-correlator problems

Consider the simplest bootstrap setup: scalar gap maximization in a single 4pt function of four identical scalars [6]. Thus we are solving the bootstrap equation

$$F_{0,0}(u,v) + \sum_{(\Delta,\ell)\in S(\Delta_*)} p_{\Delta,\ell} F_{\Delta,\ell}(u,v) = 0, \qquad p_{\Delta,\ell} \geqslant 0 \tag{1.1}$$

where $F_{\Delta,\ell}(u,v) = v^{\Delta_\phi} g_{\Delta,\ell}(u,v) - u^{\Delta_\phi} g_{\Delta,\ell}(v,u)$. Here Δ_ϕ is the external scalar dimension which for simplicity is considered fixed (although see footnote 7). The set $S(\Delta_*)$ is given by:

$$S(\Delta_*) = \{(\Delta,\ell) : \ell = 0 \text{ and } \Delta \geqslant \Delta_*, \text{ or } \ell = 2,4,\dots \text{ and } \Delta \geqslant \ell + d - 2\}\,. \tag{1.2}$$

The variables to be solved for in (1.1) are the set of appearing pairs (Δ,ℓ) and the corresponding coefficients $p_{\Delta,\ell}$. We are interested to know what is the maximal Δ_* such that (1.1) has a solution.

We would like to define a navigator function $\mathcal{N}(\Delta_*)$ such that it is negative if a solution exists and is positive if it does not exist. To this end we will consider a modified problem of the form

$$F_{0,0}(u,v) + \lambda M(u,v) + \sum_{(\Delta,\ell)\in S(\Delta_*)} p_{\Delta,\ell} F_{\Delta,\ell}(u,v) = 0, \qquad p_{\Delta,\ell} \geqslant 0, \tag{1.3}$$

We just added an extra term in the l.h.s. with a fixed function $M(u,v)$ and a new parameter λ. The function $M(u,v)$ will be chosen so that the following crucial property holds:

$$\bigstar \text{ For any } \Delta_*, \text{ problem (1.3) has a solution with } \textit{some } \lambda = \lambda_0(\Delta_*) > 0. \quad (1.4)$$

Given this property, the navigator function will be defined as the *minimal* value of λ such that (1.3) has a solution:[7]

$$\mathcal{N}(\Delta_*) = \min \lambda \text{ such that (1.3) has a solution.} \quad (1.5)$$

Property (1.4) then guarantees that the navigator is bounded from above, as we have $\mathcal{N}(\Delta_*) \leqslant \lambda_0(\Delta_*)$. We also see that the navigator is monotonically non-decreasing in the Δ_* direction, negative in the allowed region and positive outside.[8]

This described construction does not formally guarantee other nice properties of the navigator that we wish to have (that $\mathcal{N}(\Delta_*)$ is differentiable, strictly negative in the allowed region, has no local minima outside the allowed region where minimization can get stuck etc.) It also does not guarantee that the navigator is finite inside the allowed region (it may be $-\infty$ there). Nevertheless, explicit navigator functions constructed below using this idea will have all these additional nice properties, by inspection.

We will now give two examples of functions $M(u,v)$ that have the required property (1.4).

GFF-navigator

We know that for any Δ_ϕ, Eq. (1.1) has a Generalized Free Field (GFF) solution with the spectrum $\Delta = 2\Delta_\phi + 2n + \ell$, $n \geqslant 0$, $\ell = 0, 2, 4, \ldots$, corresponding to operators of schematic form $\phi\partial^\ell\Box^n\phi$. The GFF-navigator is obtained by taking

[7] Although in this section we consider Δ_ϕ fixed, it is trivial to relax this and consider the navigator as a function of both Δ_ϕ and Δ_*, defined by the same Eq. (1.5). The zero set of $\mathcal{N}(\Delta_\phi, \Delta_*)$ is then a curve which is the upper bound on Δ_* as a function of Δ_ϕ. We will not develop this idea further here but we will encounter analogous situations below in the multiple-correlator context.

[8] Note that for any Δ_* the set of λ's for which (1.3) has a solution is a connected subset of the real axis. This follows from the fact that a convex linear combination of solutions is again a solution.

$M(u,v)$ to be the first term in this solution:

$$M_{\text{GFF}}(u,v) = 2F_{2\Delta_\phi,0}(u,v). \tag{1.6}$$

Here 2 is the square of the GFF OPE coefficient in the OPE $\phi \times \phi \ni \sqrt{2}\mathcal{O}$, where $\mathcal{O} = \frac{1}{\sqrt{2}}\phi^2$ is unit-normalized. The GFF solution to crossing provides a solution to (1.3) with $\lambda = 1$ as long as all GFF operators besides ϕ^2 belong to $S(\Delta_*)$, which will be the case for $\Delta_* \leqslant 2\Delta_\phi + 2$. Hence $\mathcal{N}(\Delta_*) \leqslant 1$ for any Δ_* in this range.

Note that having a finite navigator in the range $\Delta_* \leqslant 2\Delta_\phi + 2$ is sufficient for the problem at hand, since the boundary of the allowed region for (1.1) is known to satisfy this condition. Alternatively, higher GFF operators which do not satisfy gap assumptions may be added to the r.h.s. of Eq. (1.6). See App. A.1 for this tweak of the GFF-navigator, important for bootstrap problems with additional gaps in the spectrum.

Σ-navigator

Another possibility, called the Σ-navigator, results from choosing:

$$M_\Sigma(u,v) = -\sum_{i=1}^{n} c_i F_{\Delta_i,\ell_i}(u,v)\,, \tag{1.7}$$

where (Δ_i, ℓ_i) are any n spectrum points in $S(\Delta_*)$, $c_i > 0$ some fixed positive coefficients, and n is a sufficiently large number. Since the coefficients c_i are, apart from being positive, essentially arbitrary, there is a lot of freedom in choosing the Σ-navigator.

Consider Eq. (1.3) with this $M(u,v)$. In practice, in the numerical conformal bootstrap we analyze this equation in Taylor expansion around some point, i.e. we replace functions of u, v by vectors of Taylor coefficients of some finite length n_0. Denoting vectors by boldface symbols, we have

$$\mathbf{F}_{0,0} + \lambda \mathbf{M}_\Sigma + \sum_{(\Delta,\ell)\in S(\Delta_*)} p_{\Delta,\ell}\mathbf{F}_{\Delta,\ell} = 0, \qquad p_{\Delta,\ell} \geqslant 0\,. \tag{1.8}$$

We claim that this equation will generically have a solution with some positive λ as long as the number of terms n in (1.7) is $n \geqslant n_0$. Indeed, generically the vectors

$\mathbf{F}_{\Delta_i,\ell_i}$ are not expected to be linearly independent. Thus the equation

$$\mathbf{F}_{0,0} + \sum_{i=1}^{n} x_i \mathbf{F}_{\Delta_i,\ell_i} = 0, \tag{1.9}$$

will have a solution as longs as x_i are allowed to have either sign. We rewrite this solution as

$$\mathbf{F}_{0,0} + \lambda \mathbf{M}_\Sigma + \sum_{i=1}^{n} (x_i + \lambda c_i) \mathbf{F}_{\Delta_i,\ell_i} = 0, \tag{1.10}$$

For sufficiently large positive $\lambda = \lambda_0$ all the coefficients $x_i + \lambda_0 c_i \geqslant 0$ so this is a solution to (1.8), proving the above claim. Hence, by the general arguments, the navigator is bounded from above by λ_0.

In the described construction the number of terms n in (1.7) may have to be increased with the number of conformal block derivatives used in the numerical analysis. Alternatively, we may replace the sum in (1.7) by an integral with a positive continuous measure in some interval of Δ's. Then the same navigator may be used independently of the number of derivatives.

Dual picture

In the dual approach to the numerical conformal bootstrap, the problem of computing the navigator (1.5) is formulated as follows:

$$\begin{aligned} &\mathcal{N}(\Delta_*) = \max \alpha(F_{0,0}) \text{ over all linear functionals } \alpha \text{ such that} \\ &\quad \alpha(M) = -1 \\ &\quad \alpha(F_{\Delta,\ell}) \geqslant 0 \text{ for all } (\Delta,\ell) \in S(\Delta_*) \end{aligned} \tag{1.11}$$

Our construction guarantees that the choices (1.6) or (1.7) lead to this problem having a solution bounded from above for any Δ_*.

From this dual formulation we can see that the Σ-navigator is guaranteed to be finite also in the allowed region (i.e. it cannot be $-\infty$ there). That's because for any Δ_* there is always some functional which satisfies the positivity condition in (1.11). Rescaling this functional we may make it also satisfy the normalization condition. This provides a finite lower bound for the Σ-navigator. For the GFF-navigator this argument clearly fails if $\Delta_* \leqslant 2\Delta_\phi$. In this case there is no functional α satisfying both the normalization and the positivity conditions. Thus the GFF-

navigator equals $-\infty$ for $\Delta_* \leqslant 2\Delta_\phi$.[9] This is not so problematic in practice, since this range is anyway deep inside the allowed region for the single-correlator problem. In principle the GFF-navigator could become $-\infty$ even for Δ_* somewhat above $2\Delta_\phi$, but we have not seen this happen.

It is instructive to compare the above dual formulation with how one computes the maximal allowed value $p^{\max}_{\Delta_0,\ell_0}$ of the squared OPE coefficient for an operator (Δ_0, ℓ_0) present in the spectrum [64, 67]:

$$\begin{aligned} p^{\max}_{\Delta_0,\ell_0} = &- \max \alpha(F_{0,0}) \text{ over all linear functionals } \alpha \text{ such that} \\ &\alpha(F_{\Delta_0,\ell_0}) = 1 \\ &\alpha(F_{\Delta,\ell}) \geqslant 0 \text{ for all } (\Delta,\ell) \in S(\Delta_*) \end{aligned} \tag{1.12}$$

Comparing (1.12) with (1.11), one may wonder if one could perhaps define a navigator simpler than in our proposals, namely as

$$\mathcal{N}(\Delta_*) = -p^{\max}_{\Delta_0,\ell_0} \qquad (?) \tag{1.13}$$

for some appropriate choice of (Δ_0, ℓ_0) in $S(\Delta_*)$. E.g. what if one tries $\ell_0 = 0$ and Δ_0 a little above the boundary of the allowed region? It turns out however that such simple-minded choices of functional normalization are inadequate. Namely, they give a finite navigator only in a rather small neighborhood of the boundary of the allowed region, which moreover gets smaller and smaller as one increases the number of derivatives used in the conformal bootstrap computation.[10] If one already knows quite well where the boundary is (e.g. via bisection), then using this navigator one can quickly determine it even more precisely. But if one starts far away from the boundary, this navigator would not help. Our Σ-navigator proposal shows that to get a robustly bounded navigator one needs to modify this idea by normalizing not on a single conformal block in the allowed region as in (1.12) but on a positive linear combination of many blocks as in (1.7).

Analogously, one could have hoped to get a bounded navigator by normalizing the functional to -1 on a single conformal block in the region outside $S(\Delta_*)$. But

[9] This is also obvious from the primal definition (1.6).

[10] Ref. [65] considered an early version of navigator function corresponding to normalizing one particular component of the functional to 1. This navigator prototype suffered from the same problem of being finite only in a small region. We are grateful to Tom Hartman and Amir Tajdini for enlightening communications concerning their findings, which sparked our search for a robust navigator function.

again, one finds that choosing $\ell_0 = 0$ and Δ_0 a little below the boundary of the allowed region gives a navigator which is finite only in a small neighborhood of the boundary of the allowed region. Instead, our GFF-navigator proposal shows that if Δ_0 is lowered all the way to $2\Delta_\phi$, which is quite a bit lower than the boundary of the allowed region, then the navigator becomes robustly bounded from above.

1.2.2 Multiple-correlator problems

We will now discuss how the navigator function construction generalizes to bootstrap problems involving several correlation functions. The main idea will be the same: we just need to add a new term so that crossing can always be obeyed, and minimize its coefficient.

We will consider the example of three 4pt functions $\langle\sigma\sigma\sigma\sigma\rangle$, $\langle\sigma\sigma\epsilon\epsilon\rangle$ and $\langle\epsilon\epsilon\epsilon\epsilon\rangle$ where σ and ϵ are an odd and even scalars in a $\mathbb{Z}_2$-invariant CFT (such as the critical 3d Ising model). This system of correlators leads to 5 independent crossing relations [45]:

$$\sum_{\mathcal{O}^+} \mathrm{Tr}\left[P_{\mathcal{O}}\vec{V}_{+,\Delta,\ell}\right] + \sum_{\mathcal{O}^-} p_{\mathcal{O}}\vec{V}_{-,\Delta,\ell} = 0\,, \tag{1.14}$$

$$P_{\mathcal{O}} = \begin{pmatrix}\lambda_{\sigma\sigma\mathcal{O}} & \lambda_{\epsilon\epsilon\mathcal{O}}\end{pmatrix} \otimes \begin{pmatrix}\lambda_{\sigma\sigma\mathcal{O}} \\ \lambda_{\epsilon\epsilon\mathcal{O}}\end{pmatrix}, \qquad p_{\mathcal{O}} = \lambda^2_{\sigma\epsilon\mathcal{O}}\,, \tag{1.15}$$

where $\vec{V}_{-,\Delta,\ell}$ is a 5-vector of functions while $\vec{V}_{+,\Delta,\ell}$ is a 5-vector of 2×2 symmetric

matrices of functions of u, v:

$$\vec{V}_{+,\Delta,\ell} = \begin{pmatrix} \begin{pmatrix} F^{\sigma\sigma,\sigma\sigma}_{-,\Delta,\ell} & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & F^{\epsilon\epsilon,\epsilon\epsilon}_{-,\Delta,\ell} \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{2}F^{\sigma\sigma,\epsilon\epsilon}_{-,\Delta,\ell} \\ \frac{1}{2}F^{\sigma\sigma,\epsilon\epsilon}_{-,\Delta,\ell} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{2}F^{\sigma\sigma,\epsilon\epsilon}_{+,\Delta,\ell} \\ \frac{1}{2}F^{\sigma\sigma,\epsilon\epsilon}_{+,\Delta,\ell} & 0 \end{pmatrix} \end{pmatrix}, \quad \vec{V}_{-,\Delta,\ell} = \begin{pmatrix} 0 \\ 0 \\ F^{\sigma\epsilon,\sigma\epsilon}_{-,\Delta,\ell} \\ (-1)^\ell F^{\epsilon\sigma,\sigma\epsilon}_{-,\Delta,\ell} \\ -(-1)^\ell F^{\epsilon\sigma,\sigma\epsilon}_{+,\Delta,\ell} \end{pmatrix}. \tag{1.16}$$

See [45] for the expressions of the functions $F^{ij,kl}_{\pm,\Delta,\ell}(u,v)$. The first sum in (1.14) runs over the $\mathbb{Z}_2$-even operators $\mathcal{O}^+$ in the OPEs $\sigma \times \sigma$ and $\epsilon \times \epsilon$ (whose spin is necessarily even), while the second sum in (1.14) is over all $\mathbb{Z}_2$-odd operators $\mathcal{O}^-$ in the OPE $\sigma \times \epsilon$ (which can have any spin).

As usual, we will treat separately the unit operator contribution

$$\vec{V}_{0,0} = \mathrm{Tr}\Big[P_{0,0}\vec{V}_{+,0,0}\Big], \qquad P_{0,0} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{1.17}$$

Furthermore, we will group the contributions of ϵ and σ using the relation $\lambda_{\sigma\sigma\epsilon} = \lambda_{\sigma\epsilon\sigma}$. We will work in $d = 3$ and assume that all other scalars apart from ϵ and σ are irrelevant, so all remaining $\mathcal{O}^\pm$ will satisfy the spectrum restrictions:

$$S_+ = \{(\Delta, 0) : \Delta \geqslant 3\} \cup \{(\Delta, \ell) : \ell = 2, 4, 6, \dots \text{ and } \Delta \geqslant \ell + 1\} \tag{1.18}$$

$$S_- = \{(\Delta, 0) : \Delta \geqslant 3\} \cup \{(\Delta, \ell) : \ell = 1, 2, 3, \dots \text{ and } \Delta \geqslant \ell + 1\} \tag{1.19}$$

Then we can write (1.14) as

$$\vec{V}_{0,0} + \mathrm{Tr}\left[P_{\Delta_\epsilon,0}\left(\vec{V}_{+,\Delta_\epsilon,0} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\vec{V}_{-,\Delta_\sigma,0}\right)\right] + \sum_{(\Delta,\ell)\in S_+} \mathrm{Tr}\left[P_{\Delta,\ell}\vec{V}_{+,\Delta,\ell}\right] + \sum_{(\Delta,\ell)\in S_-} p_{\Delta,\ell}\vec{V}_{-,\Delta,\ell} = 0\,. \quad (1.20)$$

If the point $(\Delta_\sigma, \Delta_\epsilon)$ is allowed, this equation must have a solution with $P_{\Delta_\epsilon,0}, P_{\Delta,\ell} \succcurlyeq 0$, $p_{\Delta,\ell} \geqslant 0$. As discovered in [45],[11] this condition gives rise to an allowed region in the $(\Delta_\sigma, \Delta_\epsilon)$ plane consisting of a small island containing the 3d Ising CFT and a larger detached "continent." We will first discuss how this can be reproduced using a two-parameter navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$. See Section 1.2.2 below for how to include the third parameter θ parametrizing the ratio of the OPE coefficients $\lambda_{\sigma\sigma\epsilon}/\lambda_{\epsilon\epsilon\epsilon}$.

Analogously to (1.3), we consider the modification of (1.20) adding to the l.h.s. an extra term $\lambda\vec{M}$ where $\lambda \in \mathbb{R}$ and $\vec{M}$ is a particular 5-vector of functions of u, v:

$$\vec{V}_{0,0} + \lambda\vec{M} + \mathrm{Tr}\left[P_{\Delta_\epsilon,0}\left(\vec{V}_{+,\Delta_\epsilon,0} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\vec{V}_{-,\Delta_\sigma,0}\right)\right] + \sum_{(\Delta,\ell)\in S_+} \mathrm{Tr}\left[P_{\Delta,\ell}\vec{V}_{+,\Delta,\ell}\right] + \sum_{(\Delta,\ell)\in S_-} p_{\Delta,\ell}\vec{V}_{-,\Delta,\ell} = 0\,. \quad (1.21)$$

In general $\vec{M}$ will also have some dependence on Δ_σ and Δ_ϵ (just like all the other vectors in the equation). We will be looking for solutions of (1.21) with $P_{\Delta_\epsilon,0}, P_{\Delta,\ell} \succcurlyeq 0$ and $p_{\Delta,\ell} \geqslant 0$. Analogously to (1.4) and (1.5), the navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ is defined as the minimal λ such that a solution exists:

$$\mathcal{N}(\Delta_\sigma, \Delta_\epsilon) = \min \lambda \text{ such that (1.21) has a solution}, \quad (1.22)$$

while $\vec{M}$ has to be chosen such that there is always some solution for a sufficiently large λ. This then provides an upper bound for the navigator and in particular guarantees that $\mathcal{N} < +\infty$.

The GFF-navigator idea from Section 1.2.1 generalizes to the present multiple-

[11] Ref. [45] did not impose the constraint $\lambda_{\sigma\sigma\epsilon} = \lambda_{\sigma\epsilon\sigma}$ so their allowed region was somewhat larger than the one we will find. See [30], Eq. (2.3) for the setup we are describing here.

correlator setup. Indeed, we always have a GFF solution to crossing in which σ and ϵ are independent GFFs. The vector $\vec{M}$ is constructed from the contributions of (unit-normalized) operators $\frac{1}{\sqrt{2}}:\sigma^2:\in\sigma\times\sigma$, $\frac{1}{\sqrt{2}}:\epsilon^2:\in\epsilon\times\epsilon$, $:\sigma\epsilon:\in\sigma\times\epsilon$:

$$\vec{M}_{\text{GFF}}=\text{Tr}\left[\begin{pmatrix}2&0\\0&0\end{pmatrix}\vec{V}_{+,2\Delta_\sigma,0}\right]+\text{Tr}\left[\begin{pmatrix}0&0\\0&2\end{pmatrix}\vec{V}_{+,2\Delta_\epsilon,0}\right]+\vec{V}_{-,\Delta_\sigma+\Delta_\epsilon,0}\,. \tag{1.23}$$

With this $\vec{M}$, Eq. (1.21) has a solution with $\lambda=1$, $P_{\Delta_\epsilon,0}=0$ and $P_{\Delta,\ell}$ and $p_{\Delta,\ell}$ coming from the rest of the GFF spectrum in the $\sigma\times\sigma$, $\epsilon\times\epsilon$, $\sigma\times\epsilon$ OPE. This guarantees that $\mathcal{N}_{\text{GFF}}(\Delta_\sigma,\Delta_\epsilon)\leqslant 1$.[12]

To describe Σ-navigators we choose two finite sets $R_\pm\subset S_\pm$ of (Δ,ℓ) pairs, and the linear equation

$$\vec{\mathbf{V}}_{0,0}+\sum_{(\Delta,\ell)\in R_+}\text{Tr}\Big[X_{\Delta,\ell}\vec{\mathbf{V}}_{+,\Delta,\ell}\Big]+\sum_{(\Delta,\ell)\in R_-}x_{\Delta,\ell}\vec{\mathbf{V}}_{-,\Delta,\ell}=0\,, \tag{1.24}$$

where the variables $X_{\Delta,\ell}$ and $x_{\Delta,\ell}$ don't have to satisfy any positivity requirement. As in Section 1.2.1, the boldface symbols mean that we have switched to working at some finite order in Taylor expansion. Taking into account the structure of $\vec{V}_{0,0}$, $\vec{V}_{\pm,\Delta,\ell}$, $\vec{V}_{+,\Delta,\ell}$, and the fact that the functions $F^{ij,kl}_{\pm,\Delta,\ell}(u,v)$ are generically linearly independent (as follows from their expressions in [45]), Eq. (1.24) has a solution as long as $R_\pm$ include sufficiently many points.[13] We won't need to know anything about the solution apart from the fact that it exists.

So let us pick any two such sets $R_\pm$ with sufficiently many points, and define

$$\vec{M}_\Sigma=-\sum_{(\Delta,\ell)\in R_+}\text{Tr}\Big[C_{\Delta,\ell}\vec{V}_{+,\Delta,\ell}\Big]-\sum_{(\Delta,\ell)\in R_-}c_{\Delta,\ell}\vec{V}_{-,\Delta,\ell}\,, \tag{1.25}$$

with some strictly positive fixed coefficients $C_{\Delta,\ell}\succ 0$, $c_{\Delta,\ell}>0$. For any such $\vec{M}_\Sigma$, Eq. (1.21) has a solution with some positive λ, by the same argument as

[12] We used here the fact that all the GFF operators apart from σ^2, ϵ^2, $\sigma\epsilon$ satisfy the $S_\pm$ constraints, assuming as we are that $\Delta_\sigma,\Delta_\epsilon\geqslant 1/2$. This is obvious for operators of spin $\ell\geqslant 1$ where we only impose the unitarity bounds. In the scalar sector, the next GFF operators are schematically $\sigma\Box\sigma$, $\epsilon\Box\epsilon$ and $\sigma\Box\epsilon$, all of which have dimension above 3. If there were additional GFF operators violating gap assumptions, their contributions would have to be added to (1.23). See App. A.1 for an example. There it is also explained how to deal with the case where the navigator function depends on the magnitude of a squared OPE coefficient.

[13] Generically it will suffice to take $|R_+|=\min(t_1,t_2,t_4+t_5)$, $|R_-|=t_3$, where t_i is the number of Taylor coefficients retained for line $i=1\ldots5$ of the original equation (1.14).

in Section 1.2.1. Hence the corresponding Σ-navigator defined via (1.22) will be bounded from above.

As a final comment, we would like to recall another problem with the feasibility-mode searches which is resolved by our navigators. Feasibility-mode SDPB runs may not converge due to precision issues for points that can already be excluded using the bootstrap of crossing equations involving only a subset of the correlators [68]. E.g. this sometimes happens for points outside the 3d Ising island which are excluded by a single-correlator constraint. The navigators presented in this section converge in all the cases we tested, including the exact Ising setup that does exhibit this problem when run in feasibility-mode. Thus, navigators also provide a more robust method of checking the feasibility of any point.

Including the angles

As shown in [30], the allowed region in the 3-correlator bootstrap can be further reduced by treating the $P_{\Delta_\epsilon,0}$ term in (1.20) differently from the other $P_{\Delta,\ell}$. This is possible since we are assuming ϵ is non-degenerate. Writing $\lambda_{\sigma\sigma\epsilon} = \lambda_\epsilon \cos\theta$, $\lambda_{\epsilon\epsilon\epsilon} = \lambda_\epsilon \sin\theta$, $p_\epsilon = \lambda_\epsilon^2 \geqslant 0$, we can then specialize Eq. (1.20) as

$$\vec{V}_{0,0} + p_\epsilon \vec{V}_\epsilon(\theta) + \sum_{(\Delta,\ell)\in S_+} \mathrm{Tr}\Big[P_{\Delta,\ell}\vec{V}_{+,\Delta,\ell}\Big] + \sum_{(\Delta,\ell)\in S_-} p_{\Delta,\ell}\vec{V}_{-,\Delta,\ell} = 0\,, \tag{1.26}$$

$$\vec{V}_\epsilon(\theta) = \mathrm{Tr}\left[\begin{pmatrix} c_\theta^2 & c_\theta s_\theta \\ c_\theta s_\theta^2 & s_\theta \end{pmatrix}\vec{V}_{+,\Delta_\epsilon,0} + \begin{pmatrix} c_\theta^2 & 0 \\ 0 & 0 \end{pmatrix}\vec{V}_{-,\Delta_\sigma,0}\right]. \tag{1.27}$$

The original numerical implementation of this setup [30] involved scanning over the angle θ in addition to Δ_σ and Δ_ϵ, which was computationally laborious. Significant progress in reducing the computational cost has been recently achieved via the cutting surface algorithm [29].

In this chapter we will show how this setup can be analyzed even more efficiently using the navigator function. The construction is almost the same as above. We simply add to the l.h.s. of (1.20) the term $\lambda\vec{M}$ and define the navigator $\mathcal{N}(\Delta_\sigma,\Delta_\epsilon,\theta)$ as the minimal value of λ for which the so modified equation has a solution with $p_\epsilon \geqslant 0$, $P_{\Delta,\ell} \succcurlyeq 0$, $p_{\Delta,\ell} \geqslant 0$. We can choose $\vec{M}_{\mathrm{GFF}}$ as in (1.23), or $\vec{M}_\Sigma$ as in (1.25), with $R_\pm \subset S_\pm$. The numerical results will be shown below.

Dual picture

The primal definition of the navigator function given above was convenient for clarifying the condition under which the navigator is bounded from above. For the actual numerical computation, we translate the primal definition to an equivalent dual formulation. As an example, for the 2-parameter navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$, Eq. (1.22), the dual definition takes the form:

$$\begin{aligned}
&\mathcal{N}(\Delta_\sigma, \Delta_\epsilon) = \max\ \vec{\alpha}\cdot\vec{V}_{0,0} \text{ over all linear functionals } \vec{\alpha} \text{ such that} \\
&\quad \vec{\alpha}\cdot\vec{M} = -1\,, && (1.28)\\
&\quad \vec{\alpha}\cdot\left(\vec{V}_{+,\Delta_\epsilon,0} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\vec{V}_{-,\Delta_\sigma,0}\right) \succcurlyeq 0\,, && (1.29)\\
&\quad \vec{\alpha}\cdot\vec{V}_{+,\Delta,\ell} \succcurlyeq 0 \text{ for all } (\Delta,\ell)\in S_+\,, && (1.30)\\
&\quad \vec{\alpha}\cdot\vec{V}_{-,\Delta,\ell} \geqslant 0 \text{ for all } (\Delta,\ell)\in S_-\,. && (1.31)
\end{aligned}$$

For the 3-parameter navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, \theta)$ from Section 1.2.2 we have to simply replace condition (1.29) with (see (1.27))

$$\vec{\alpha}\cdot\vec{V}_\epsilon(\theta) \geqslant 0\,. \tag{1.32}$$

We recall that the above dual problems can be then transformed into a polynomial matrix problem using rational approximations of conformal blocks expanded up to some finite derivative order around the $z = \bar{z} = 1/2$ point. This polynomial matrix problem is then transformed into a semidefinite programming problem, which can be solved by SDPB [42, 43].

In App. A.2 we describe an alternative construction of the navigator function, which turns the feasibility problem into an optimization problem not at the level of crossing equations, but after the problem has already been dualized and translated into an SDP. We have not used that construction in this chapter, but it may turn out useful in future applications.

1.3 Visualizing the GFF-navigator

In the previous section we provided a formal definition of navigator functions. Their actual numerical evaluation can be performed using SDPB. Since navigator

evaluation involves maximization, it will be comparable in cost to an OPE coefficient maximization, and more expensive than say testing feasibility of a point. Of course, we hope that this extra cost will be offset due to additional information provided by the navigator. And indeed, in subsequent sections we will see that complicated bootstrap tasks can be achieved with relatively few navigator evaluations.

Before we go to those applications, in this section we will explicitly visualize the various navigator functions of Section 1.2. We will do this to get some intuition about their "shape," and to check that they are sufficiently well behaved to allow application of minimization algorithms. Visualization will be done by performing fine scans in all variables. We emphasize again that in realistic applications we will not need to perform such expensive visualization scans.

We will focus on the 2- and 3-parameter GFF-navigators $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ and $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, \theta)$ from Sections 1.2.2 and 1.2.2. Numerical evaluation is done using the dual formulations given in Section 1.2.2, where we need to put $\vec{M} = \vec{M}_{\text{GFF}}$ from Eq. (1.23). We will not show plots for the Σ-navigators, although we have checked that they behave similarly to the GFF-navigators.

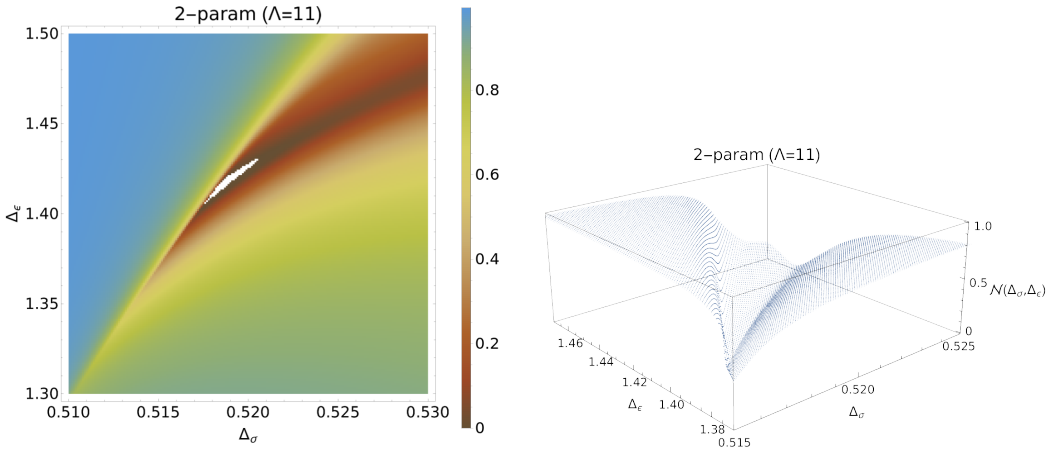


Figure 1.1: Example of a navigator function $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ for the 3d Ising setup. *Left:* Heat map of the navigator function. The negative region, corresponding to the Ising model island, is depicted in white. (Note that this image, and similarly other heat maps in this chapter, appears pixelated due to the finite resolution of our scan. The actual island has a piecewise smooth boundary.) *Right:* Surface plot of the navigator function.

2-parameter case. We start with Fig. 1.1 showing $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ in an extended region around the 3d Ising island at the derivative order $\Lambda = 11$. We can see

from it that the region of negative navigator value matches in size the $\Lambda = 11$ allowed region of [45], Figs. 3 and 4.[14] On this scale the navigator is observed to be smooth (see however below) and approaching its predicted asymptotic value $\mathcal{N}_{\max} = 1$ far away from allowed regions. There is clearly a valley coming from the top right of Fig. 1.1(left), narrowing to a tight gorge as it approaches its minimum inside the island. The surface has only one local minimum located in the plotted region and, as expected, it is inside the island. This feature will be essential when we discuss navigator minimization strategies in Section 1.5. Indeed, local minima in the disallowed region would have required more computationally expensive optimization methods than the BFGS algorithm discussed there.

In addition to the island, the allowed region found in [45] also included a detached "continent" at larger values of Δ_σ, beyond the range of Fig. 1.1. This continent is of course also found to be a region of negative navigator. Our navigator minimization strategies will use a bounding box, see Section 1.5.2, to make sure that we sail to the island and not to the continent.

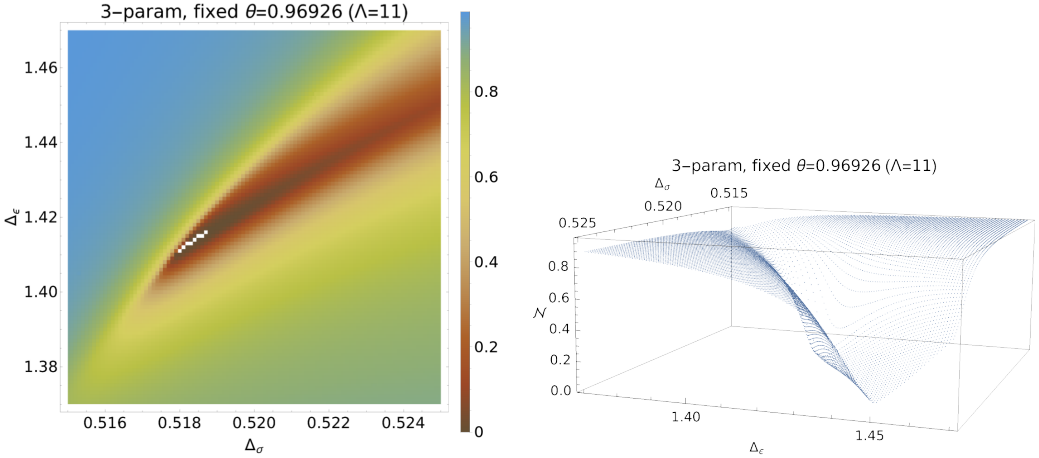


Figure 1.2: $(\Delta_\sigma, \Delta_\epsilon)$ slice of the 3-parameter GFF navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, \theta = 0.96926)$ at $\Lambda = 11$. *Left:* Heat map of this 2d slice. *Right:* Surface plot of the 2d slice.

3-parameter case. To get an idea of the shape of $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, \theta)$, we will show two-dimensional slices for fixed values of one of the 3 parameters. Thus, in Fig. 1.2 we fix $\theta = 0.96926$ (the central value from [30]), and let $(\Delta_\sigma, \Delta_\epsilon)$ vary in a region close to the navigator minimum. The surface shape is similar to the two-parameter

[14] Our $\Lambda = 11$ corresponds to $n_{\max} = 6$ in [45]. The slight difference in shape between our island and that of [45] is because we have imposed the OPE equality $\lambda_{\sigma\sigma\epsilon} = \lambda_{\sigma\epsilon\sigma}$ in our setup, see footnote 11.

navigator surface in Fig. 1.1.[15]

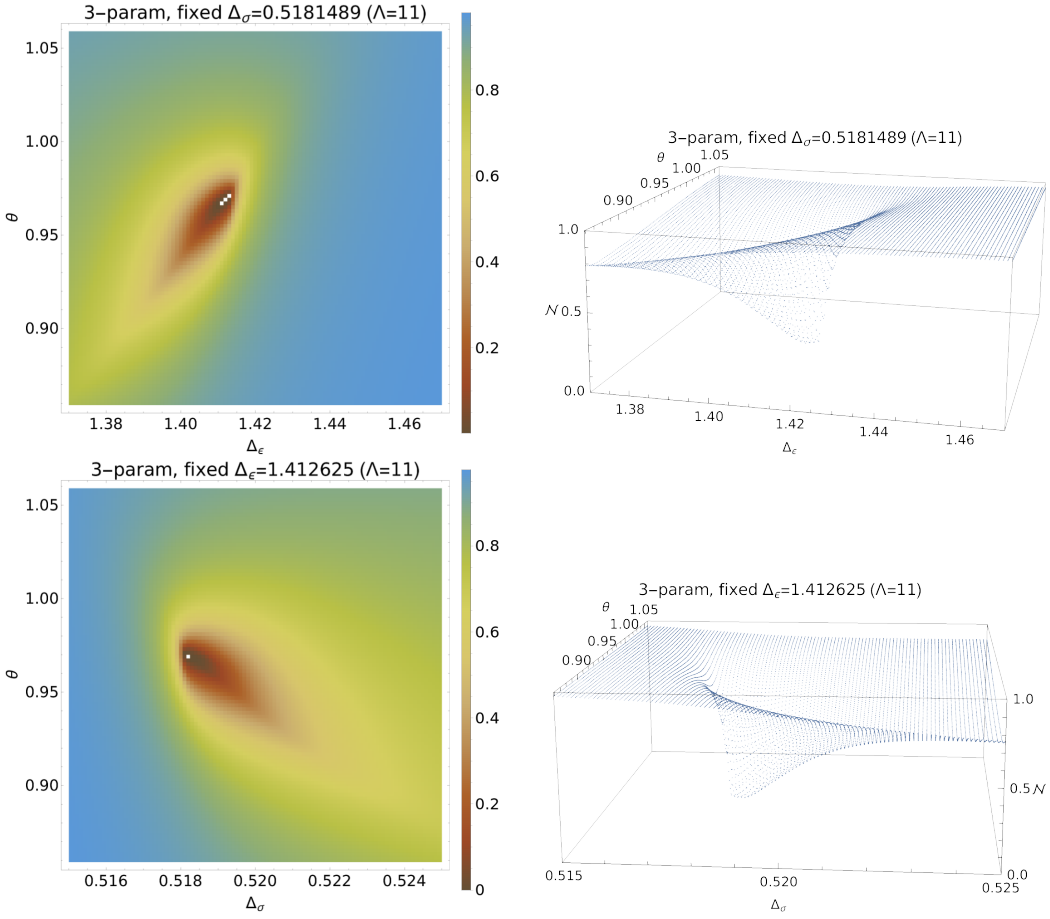


Figure 1.3: *Top row:* 2d slice of the 3-parameter GFF-navigator for fixed $\Delta_\sigma = 0.5181489$ around the Ising island at $\Lambda = 11$. *Bottom row:* Same, but for fixed $\Delta_\epsilon = 1.412625$.

Furthermore, in Fig. 1.3 we show 2d slices of the 3-parameter navigator arising for a fixed Δ_σ and Δ_ϵ. Although the precise shapes here are somewhat different, all three 2d slice surfaces are found to be smooth at this scale and free of local minima in the disallowed region (i.e. where the navigator is positive). This is a good sign that optimization algorithms should be able to quickly converge towards the Ising island given a reasonably precise initial guess.

Variation with Λ. Here we will explore how navigator shape changes with the derivative order Λ. By design, the navigator function monotonically increases

[15]The surface plot in Fig. 1.2 is rotated opposite to Fig. 1.1, to facilitate comparison to Fig. 1.4 below.

pointwise with Λ, i.e. $\mathcal{N}_{\Lambda_2}(x) \geqslant \mathcal{N}_{\Lambda_1}(x)$ for $\Lambda_2 > \Lambda_1$. This generalizes the fact that the allowed region shrinks with Λ. It is interesting to know *how* this increase happens. E.g. does the navigator surface move up with Λ uniformly or not? To answer this question, we show in Fig. 1.4 the 2d slice of the 3-parameter navigator at fixed $\theta = 0.96926$ with $\Lambda = 19$, comparing it to $\Lambda = 11$ from Fig. 1.2. We see that the navigator surface has indeed moved up, but in non-uniform fashion. Most notably, the surface along one of the nearly flat "valley" directions gets lifted up much more than near the minimum. As a result, the minimum became more pronounced, which is a good sign.

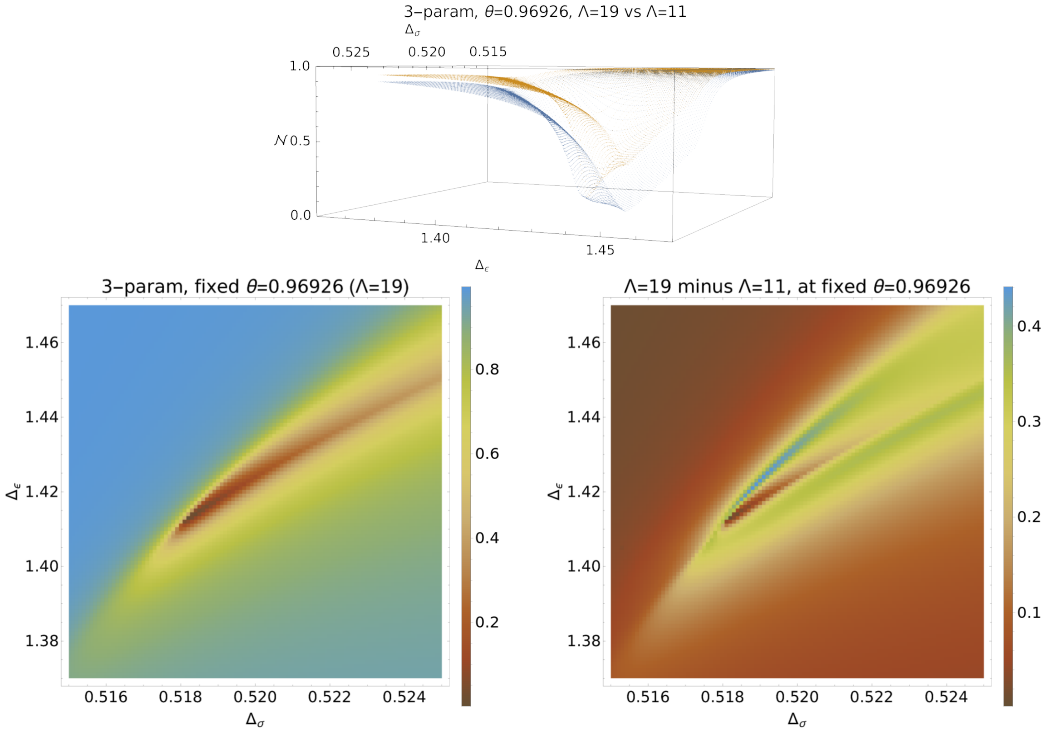


Figure 1.4: *Top:* Surface plot of the $\Lambda = 19$ 2d slice (orange) compared to the $\Lambda = 11$ 2d slice from Fig. 1.2 (blue) *Bottom left:* Heat map of the $(\Delta_\sigma, \Delta_\epsilon)$ slice of the 3-parameter GFF navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, \theta = 0.96926)$ around the Ising island at $\Lambda = 19$. *Bottom right:* Heat map of the difference between $\Lambda = 19$ and $\Lambda = 11$.

1.3.1 Derivative of the navigator

The visualizations show navigator functions that are seemingly smooth and free of local minima. Both these properties would be very helpful for the numerical

minimization algorithms, but they did not automatically follow from the definition of the navigator functions and we cannot guarantee that they hold in other setups. In fact, in the course of our investigations we found that even the navigator function under consideration is not *entirely* smooth: more precisely, we believe that it is not everywhere C^2.

Our evidence is provided in figure 1.5. In this figure we consider a GFF navigator function with $\Lambda = 11$ for $\Delta_\sigma = 0.51831848513294$, as a function of Δ_ϵ. (The chosen values of Δ_σ and Δ_ϵ are in the vicinity of the minimum that we found using the techniques described below. Notice that the navigator is negative along the entirety of the cross-section in figure 1.5 and so we are inside the Ising island. We also imposed the OPE relation $\lambda_{\sigma\sigma\epsilon} = \lambda_{\sigma\epsilon\sigma}$ but left the ratio $\lambda_{\epsilon\epsilon\epsilon}/\lambda_{\sigma\epsilon\sigma}$ unspecified.) We plot both the navigator function itself as well as its first derivative in the Δ_ϵ direction. The kink in the latter plot strongly suggests that there is a discontinuity in the second derivative of the navigator. Indeed, the straight lines on either side of the kink allow us to reliably estimate the second derivative with finite differences: we find the value to be 767.762901557722(1) on the left and 219229.421457(1) on the right. Furthermore, using the two points closest to the kink we can estimate that the third derivative would have to be at least 10^{23} if the navigator function were smooth, which seems highly unlikely.

Although we have only shown a single cross section plot, it is likely that the non-smoothness persists along a line (segment) in the $(\Delta_\sigma, \Delta_\epsilon)$ plane. It would be interesting to understand its origin and whether there is a connection with the physics of the problem. Some preliminary investigations indicate that the discontinuity might be due to rearrangements of the extremal spectrum, but a detailed investigation is beyond the scope of this work.

Fortunately we will see below that the jump in the second derivative does not appear to inhibit the functioning of our minimization algorithm. We will comment more on this in the section 1.5.3.

1.4 Gradient at primal-dual optimality

In order to find points x where $\mathcal{N}(x) < 0$ we will use a numerical minimization algorithm. The convergence rate of such algorithms is significantly improved if we also provide it with derivative information. In this section we therefore outline a procedure to compute the gradient $\nabla\mathcal{N}(x)$.

Naively, one might think that gradient evaluation would involve computational

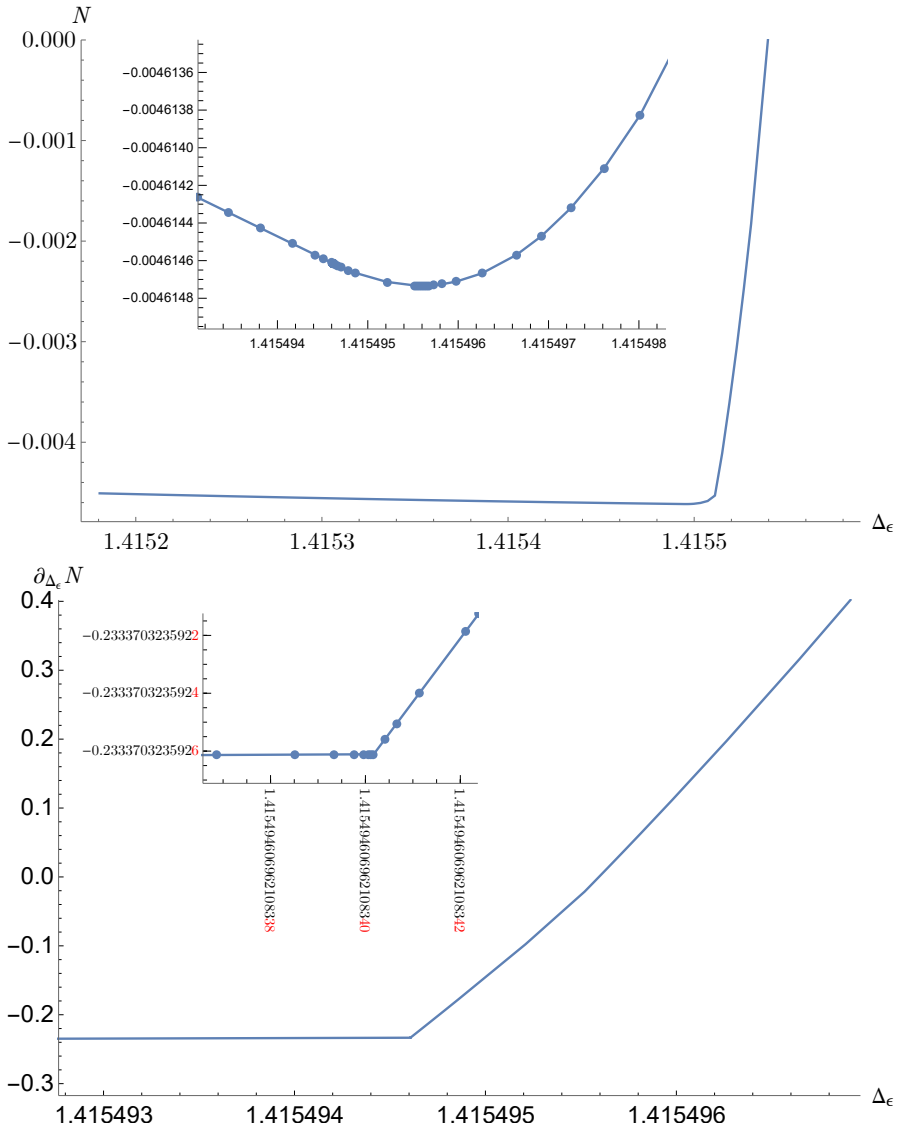


Figure 1.5: *Top:* Plot of $N(\Delta_\sigma, \Delta_\epsilon)$ v.s. Δ_ϵ where N is the navigator function, and a zoom-in plot around the minimum. *Bottom:* Plot of the derivative of the navigator function with respect to $\partial_{\Delta_\epsilon} N(\Delta_\sigma, \Delta_\epsilon)$ as a function of Δ_ϵ and a zoom-in plot in the scale of 10^{-18} around the kink.

overhead. For example, evaluating it via finite differences would require k additional SDPB runs where k is the number of variables on which the navigator depends. However this naive expectation is wrong: the main result of this section will be that $\nabla\mathcal{N}(x)$ can be evaluated at negligible computational cost if we have already evaluated the function $\mathcal{N}(x)$ itself. The underlying reason is that the evaluation of $\mathcal{N}(x)$ is an extremization problem, and at extremality the first-order variation can be computed using only the original, unperturbed solution. This remains true even for constrained minimization problems, as is the case for us, when solved via primal-dual algorithms such as in SDPB, because primal and dual variables play the role of each other's Lagrange multipliers. To explain this in more detail we first have to introduce the semidefinite programming problem that underlies the computation of $\mathcal{N}(x)$.

1.4.1 Semidefinite programming reminder

Now we will explain how to compute the gradient of the objective in the above setup. As mentioned above, the evaluation of $\mathcal{N}(x)$ is computationally analogous to an OPE extremization problem that is often encountered in numerical bootstrap studies. Let us recall that, using a rational approximation for conformal blocks [69], these extremization problems become semidefinite programs with a particular structure of the constraint matrices. We will use the notation of [42], using which the problem can be written as:

$$\begin{aligned} \mathcal{D}: \text{ maximize } & b^T y \quad \text{over} \quad y \in \mathbb{R}^n, Y \in \mathcal{S}^K \\ \text{such that } & Y \succeq 0 \quad \text{and} \\ & By + \mathrm{Tr}(A_* Y) = c\,, \end{aligned} \tag{1.33}$$

with $\mathcal{S}^K$ the space of symmetric matrices of size K. Note that $c \in \mathbb{R}^P$ is a vector, $B \in (\mathbb{R}^n)^P$ a rectangular matrix, and the $A_* = (A_1, \ldots, A_P) \in (S^K)^P$ is a vector of matrices.[16]

In the language of convex optimization the program (1.33) is called a *dual*

[16] Although this notation suffices for our purposes, in actuality the matrices involved all have a block structure and the number of non-zero components is significantly lower than a naive counting would suggest.

program ($\mathcal{D}$), and the corresponding *primal* program $\mathcal{P}$ is given by:[17]

$$\begin{aligned} \mathcal{P}: \text{ minimize } & c^T x \quad \text{over} \quad x \in \mathbb{R}^P \\ \text{such that } & X(x) := x^T A_* \succeq 0 \quad \text{and} \\ & B^T x = b\,. \end{aligned} \tag{1.34}$$

Note that $x^T A_* \equiv \sum_{p=1}^P x_p A_p$, so that $X(x) \in \mathcal{S}^K$.

We need a few more definitions. A vector x is said to be *primal feasible* if all the conditions in (1.34) are obeyed, even if optimality is not necessarily achieved. In the same vein a pair (y, Y) can be *dual feasible* if it obeys all the conditions in (1.33). The *duality gap* is defined as the difference between the objectives:

$$D(x, y) := c^T x - b^T y\,. \tag{1.35}$$

If x is primal feasible and (y, Y) is dual feasible, then the duality gap is nonnegative:

$$D(x, y) = \mathrm{Tr}\Big((x^T A_*)\, Y\Big) = \mathrm{Tr}(XY) \geqslant 0\,, \tag{1.36}$$

by the positive semidefiniteness of X and Y. So for any primal feasible point x the value of $c^T x$ provides an upper bound for the dual optimum, and similarly for any dual feasible point (y, Y) the value of $b^T y$ provides a lower bound for the primal optimum.

Now suppose one finds primal and dual feasible points with $D(x, y) = 0$. Then clearly both the primal and dual problem have been solved and brought to extremality, because neither objective has any room left to improve. It is a nontrivial fact of life that this condition is not only sufficient but also necessary for optimality in a generic semidefinite program (see [42] and references therein for details). In other words, rather than solving the primal or dual extremization

[17] We have opted to keep in this section the notation of [42] (excepting setting $C = 0$ in Eq. (2.3) and (2.21) of [42]). This unfortunately produces a clash of notation: in this section x denotes the vector of free variables in the primal semidefinite program, whereas in the rest of the chapter x is the argument of the navigator function. We stress that these are unrelated quantities.

problem, we can equivalently solve

$$\begin{aligned} &\mathrm{Tr}\,(A_* Y) + By = c\,, \\ &B^T x = b\,, \\ &X = x^T A_*\,, \\ &XY = 0\,, \\ &X, Y \succeq 0\,, \end{aligned} \tag{1.37}$$

and then the optimal value of (1.33) and (1.34) is given by $b^T y = c^T x$. Notice that the fourth equation in (1.37) states that $XY = 0$ as a matrix equation. We call this the *complementarity condition*, and it follows from the vanishing duality gap, i.e. $\mathrm{Tr}(XY) = 0$, together with $X, Y \succeq 0$.

1.4.2 SDP gradient formula

Suppose we have found a primal-dual optimal point (x, y, X, Y) such that the equations (1.37) are solved. To compute the gradient of the objective we change the parameters in the problem a little bit,

$$(b, c, B, A_*) \to (b, c, B, A_*) + (db, dc, dB, dA_*)\,, \tag{1.38}$$

and ask how the objective will change. So we need to investigate the corresponding linearized problem. The change in the solution

$$(x, y, X, Y) \to (x, y, X, Y) + (dx, dy, dX, dY) \tag{1.39}$$

must obey the linearized version of the optimality equations (1.37):

$$\begin{aligned} &\mathrm{Tr}(dA_*\, Y) + \mathrm{Tr}(A_*\, dY) + B\, dy + dB\, y = dc\,, \\ &dB^T x + B^T dx = db\,, \\ &dX = dx^T A_* + x^T dA_*\,, \\ &dX\, Y + X\, dY = 0\,, \\ &X + dX,\, Y + dY \succeq 0\,. \end{aligned} \tag{1.40}$$

Our goal will be to compute the change in the dual objective, which is given by:

$$d(b^T y) = db^T y + b^T dy\,. \tag{1.41}$$

In fact, since the duality gap remains zero we find $d(c^T x) = d(b^T y)$ and one could equally well have computed the change in the primal objective.

We start by showing a useful auxiliary result. The $dX\,Y + X\,dY = 0$ in (1.40) implies of course that $\mathrm{Tr}(dX\,Y) + \mathrm{Tr}(X\,dY) = 0$. We claim that a stronger result is true, namely that the two terms vanish independently:

$$\mathrm{Tr}(dX\,Y) = \mathrm{Tr}(X\,dY) = 0\,. \tag{1.42}$$

The proof is as follows. If $XY = 0$ and $X, Y \succeq 0$ then X and Y must have some zero eigenvalues. We can choose a basis where X is an upper block matrix,

$$X = \begin{pmatrix} X_{11} & 0 \\ 0 & 0 \end{pmatrix} \tag{1.43}$$

with $X_{11} \succ 0$. Then any symmetric Y obeying $XY = 0$ must look like

$$Y = \begin{pmatrix} 0 & 0 \\ 0 & Y_{22} \end{pmatrix} \tag{1.44}$$

with $Y_{22} \succeq 0$ because $Y \succeq 0$. If we now write the variations as

$$dX = \begin{pmatrix} dX_{11} & dX_{12} \\ dX_{12}^T & dX_{22} \end{pmatrix}\,, \qquad dY = \begin{pmatrix} dY_{11} & dY_{12} \\ dY_{12}^T & dY_{22} \end{pmatrix}\,, \tag{1.45}$$

then

$$dX\,Y = \begin{pmatrix} 0 & dX_{12}Y_{22} \\ 0 & dX_{22}Y_{22} \end{pmatrix}\,, \qquad X\,dY = \begin{pmatrix} X_{11}dY_{11} & X_{11}dY_{12} \\ 0 & 0 \end{pmatrix}\,. \tag{1.46}$$

Now it becomes clear that the condition $dX\,Y + X\,dY = 0$ implies that $X_{11}dY_{11} = 0$ and $dX_{22}Y_{22} = 0$, which in turn implies (1.42).

Let us return to the change in the dual objective as given in equation (1.41).

Using the linearized optimality equations it can be written as:

$$\begin{aligned} db^T y + b^T dy &= db^T y + x^T B dy \\ &= db^T y + x^T (dc - \mathrm{Tr}(dA_* \, Y) - \mathrm{Tr}(A_* \, dY) - dB \, y) \\ &= db^T y + x^T dc - x^T \, \mathrm{Tr}(dA_* \, Y) - x^T \, \mathrm{Tr}(A_* \, dY) - x^T dB \, y \end{aligned} \tag{1.47}$$

At this point we recall that $x^T A_* = X$. Moreover we have just shown $\mathrm{Tr}(X dY) = 0$. So the term proportional to dY in (1.47) vanishes, and we obtain:

$$\boxed{d(b^T y) = d(c^T x) = db^T y + dc^T x - x^T dB \, y - x^T \, \mathrm{Tr}(dA_* \, Y) \, .} \tag{1.48}$$

This "SDP gradient formula" constitutes one of the main points of this chapter. It shows that the variation of the objective function of semidefinite programs (1.33) and (1.34) can be computed just from the variation of the data (db, dc, dB, dA_*) provided that we know the primal-dual solution (x, y, X, Y). A remarkable fact is that we have eliminated all the dependence on (dx, dy, dX, dY) from this formula.

In this chapter we will apply Eq. (1.48) to the navigator function. Once the navigator has been evaluated for some parameter values, Eq. (1.48) computes the gradient at the same point with negligible extra computational cost (see Section 1.4.2 below for how we organized the computation in practice). It's worth pointing out that this observation holds also for more familiar conformal bootstrap problems such as the OPE coefficient maximization. Such problems have been analyzed for years using primal-dual methods, but the existence of the "SDP gradient formula" has never been suspected by the people in the bootstrap community.

There is one important caveat to the preceding derivation. Although the solution (dx, dy, dX, dY) to the linearized optimality equations does not appear in equation (1.48), we did need to assume that it existed in the intermediate steps. On the other hand, it is not guaranteed that the equations in (1.40) always have a solution. Fortunately this question has been analyzed in the semidefinite programming literature: for example, the paper [70] proves that the linearized equations for the semidefinite programs considered here will have a solution if $X + Y \succ 0$, which in our notation is equivalent to the requirement that $Y_{22} \succ 0$ rather than just $Y_{22} \succeq 0$. The paper [71] shows that this is in fact a generic property of the optimal matrices in semidefinite programs. We therefore expect the navigator functions to be generically C^1, which is also confirmed experimentally by the smooth plots shown in the previous section.

Practical details for navigator gradient evaluation

As was shown in the previous section the change of the objective under a small perturbation of a bootstrap problem can be computed using only the solution to the initial problem (x, y, X, Y) and the differences between the data (db, dc, dB, dA_*) defining the SDP. In this short technical section we describe the precise workflow using available codes. In order to be able to compute the gradient, one should first run SDPB on the original problem specified by (b, c, B, A_*) using the option `-writeSolution="x,y,X,Y"` to save the full solution to a file. The values (db, dc, dB, dA_*) can either be obtained by taking the difference between the perturbed and unperturbed bootstrap problem on the level of the polynomial matrix problem (PMP) and converting that to an SDP using `pvm2sdp` or by first converting both PMP's to SDP's and taking the difference between the resulting (b, c, B, A) and (b', c', B', A'). For the computations in this chapter we did the latter. A dedicated tool called `approx_objective` that takes one optimal checkpoint containing (x, y, X, Y), one unperturbed SDP-file and one perturbed SDP-file[18] as input and outputs the corresponding change in the objective has been packaged with SDPB as of version 2.5.

When converting the PMP to an SDP, we can choose to keep the bilinear basis, sample points, and sample scalings (see [42]) the same for both the perturbed and the unperturbed SDP. With such choices, we automatically have $dA_* = 0$ and the gradient formula simplifies to $d(b^T y) = db^T y + dc^T x - x^T dB\, y$.

1.4.3 Lagrangian perspective

In this section we give an alternative derivation of the SDP gradient formula. This derivation may look like a trick, but it provides an interesting perspective on why we were able to eliminate the variation (dx, dy, dX, dY) from the change $d(b^T y)$ in the dual objective.

Consider the following Lagrange function:

$$L(x, y, X, Y) = c^T x + b^T y - x^T B y + \mathrm{Tr}\Big((X - x^T A_*)Y\Big) - \mu \log\det X \qquad (1.49)$$

with $\mu > 0$ a parameter, and it is understood that $X \succeq 0$. As is readily verified, the stationarity equations of this Lagrangian with respect to x, y and Y yield exactly the primal and dual feasibility conditions, i.e. the first three conditions in

[18] The file is expected to contain (b, c, B, A_*) in the format produced by `pvm2sdp`.

(1.37). Demanding stationarity with respect to X yields:

$$XY = \mu I, \tag{1.50}$$

with I the identity matrix. We can think of the last term $-\mu \log\det X$ in (1.49) as a barrier function that guarantees that $X, Y \succ 0$. In the limit $\mu \downarrow 0$ the barrier disappears and we recover the original complementarity condition.[19] As is well known (e.g. [72]), the barrier function $-\log\det X$ is convex.

We denote by $(x(\mu), y(\mu), X(\mu), Y(\mu))$ the stationary point of the Lagrange function, i.e. the solution of all the feasibility conditions and of the deformed complementarity condition $XY = \mu I$. Apart from degenerate situations, this solution exists; it is also unique.[20] The value of the Lagrange function at this solution is given by:

$$L(\mu) = c^T x(\mu) - \mu \log\det X(\mu) \tag{1.51}$$

since the constraints multiplying y and Y are obeyed by assumption. Furthermore, since the Lagrange function is stationary with respect to (x, y, X, Y), its variation with respect to the parameters (b, c, B, A_*) is immediate:

$$dL(\mu) = dc^T x(\mu) + db^T y(\mu) - x(\mu)^T\, dB\, y(\mu) - x(\mu)^T \operatorname{Tr}(dA_*\, Y(\mu)) \tag{1.52}$$

Now we can ask what happens if we take μ very small. Since the original semidefinite program is assumed to have a solution, we expect $(x(\mu), y(\mu), X(\mu), Y(\mu))$ to smoothly approach that solution as we send $\mu \downarrow 0$. Clearly, the variation of the Lagrangian (1.52) at the stationary point will then approach the right-hand side of our previous result (1.48). What of the value of the Lagrangian (1.51)

[19] The modified complementarity condition (1.50) is also at the heart of primal-dual interior point algorithms as used in SDPB. Keeping μ finite and therefore X, Y strictly positive is useful to avoid getting stuck at the boundary where X and Y are singular. In the course of the algorithm the value of μ is then gradually reduced to zero in order to obtain a solution that obeys the original complementarity condition. See [42] for details.

[20] Let M be the convex set of all x obeying $b^T = x^T B$ and $x^T A_* = X(x) \succ 0$. On this set we consider the convex and smooth function $t(x) = c^T x - \mu \log\det X$. Generically the sublevel sets of this function are bounded. Indeed, any unbounded direction inside M can be parametrized as $x_0 + \lambda\hat{x}$ with $x_0 \in M$, and $\hat{x}^T B = 0$ and with $\lambda \to \infty$. If the original primal minimization problem is bounded, we have $c^T\hat{x} \geqslant 0$ for all directions. Generically we have a stronger condition $c^T\hat{x} > 0$, in which case we eventually exit all sublevel sets for any such direction. Therefore $t(x)$ must have a minimum inside M, and by convexity it is unique. At this point $\nabla t(x) = c - \mu \operatorname{Tr}\big(A_* X^{-1}\big)$ is orthogonal to M. But the directions orthogonal to M are spanned by the gradient of the constraints, so by the columns of the matrix B. There must then exists some coefficients y such that $\nabla t(x) = By$, which solve the last remaining equation.

itself? We know that X becomes singular and so $\det(X)$ will diverge. However, for $Y = \mu X^{-1}$ to remain finite the eigenvalues of X cannot vanish faster than linearly with μ. We conclude that $-\mu \log\det X = O(\mu \log \mu)$, the additional term in equation (1.51) vanishes in the limit, and so $\lim_{\mu\downarrow 0} L(\mu) = c^T x$. Together with equation (1.52) this reproduces (1.48).

This derivation elucidates the absence of (dx, dy, dX, dY) from the variation of the objective. To summarize, the point is to replace the original constrained problem with an unconstrained one, involving a barrier function times a regulator μ. The unconstrained variation involves only variation of the data, and not of the solution itself. The constrained variation is recovered in the $\mu \downarrow 0$ limit and also has this property.

In Appendix A.3 we push this logic one step further and explain how it can be used to compute the second variation of the objective, which one may call the "SDP Hessian formula." Also there we provide numerical tests of the SDP gradient and Hessian formulas. Having the Hessian as opposed to just the gradient could further speed up the minimization algorithms to be described in the next section, allowing to use Newton rather than quasi-Newton methods, but exploring this is postponed to future work.

1.5 Navigator minimization

A central task in the numerical bootstrap is the search for a feasible point. This corresponds to finding a point where the navigator function is negative. In addition, we may also be interested in finding the minimum of the navigator function, since its location might be close to the true CFT (we will show shortly that this indeed seems to be the case).

Given an n-dimensional search space, the search for a local minimum of the navigator function $\mathcal{N}(x)$ is a standard optimization problem. As explained in Section 1.4, the gradient of the navigator function is cheap to compute. Quasi-Newton methods can make good use of this cheap gradient. Recall that Newton's method requires computing a gradient and a Hessian at each point in the search. By contrast, quasi-Newton methods approximate the Hessian using gradient information.[21] In this chapter, and really throughout this thesis, we use the

[21] In Appendix A.3 we explain that it also possible to compute the Hessian of the navigator. However in this work we will only use the gradient information.

Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm ([73], Sec. 6.1) which is a well documented and widely used quasi-Newton method.

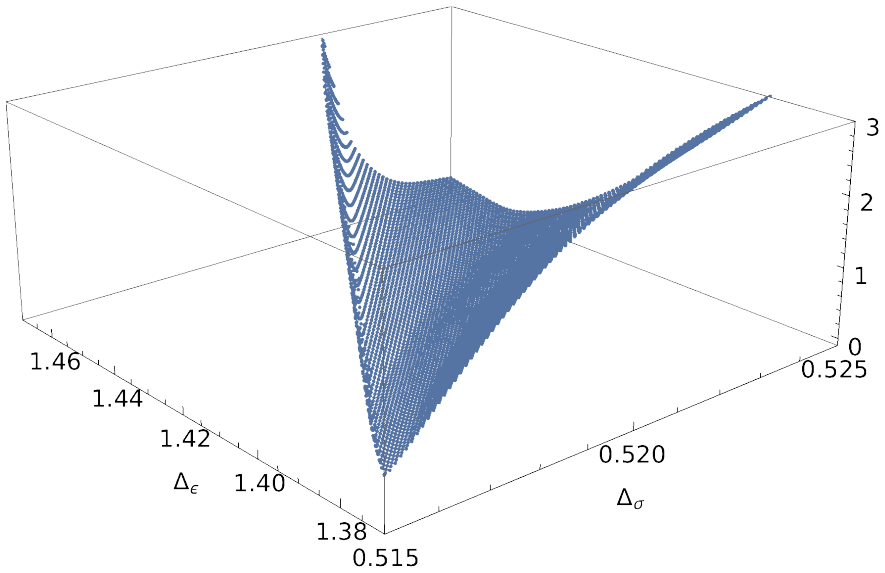


Figure 1.6: Rendering of $f\left(\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)\right)$, i.e. the 2-parameter GFF navigator to which was applied the fractional linear transformation (1.53). The derivative order used here is $\Lambda = 11$.

The BFGS algorithm maintains an approximation to the Hessian, which it updates using gradient information at each step. This update enforces positive-definiteness of the Hessian. Thus, it can only provide a truthful representation of the Hessian if the objective function is convex. In the examples studied in this chapter, we have found that the navigator function is convex close to its minimum. However, this is not true further away from the minimum (for example, the GFF navigator tends to its asymptotic value 1 in a concave manner far away from allowed regions). This can lead to failure of the BFGS algorithm or less than optimal convergence. Therefore it is helpful to compose the navigator function with a monotonic function so that it becomes convex in a larger region but maintains the same minima. For example, if the maximal value of the navigator $\mathcal{N}(x)$ is $\mathcal{N}_{\text{max}}$ (e.g. $\mathcal{N}_{\text{max}} = 1$ by construction for the GFF navigator), we can instead minimize

$$f(x) = \frac{\mathcal{N}(x)}{1 - \mathcal{N}(x)/\mathcal{N}_{\text{max}}} \ . \tag{1.53}$$

Note that $f(x) < 0$ if and only if $\mathcal{N}(x) < 0$, so that the allowed region is unchanged after this transformation. It's also easy to show that $f(x)$ is convex in a larger

region than $\mathcal{N}(x)$.[22] Intuitively, the main idea is that $f(x) \approx \mathcal{N}(x)$ where $\mathcal{N}(x) \approx 0$, while at large x, where $\mathcal{N}(x)$ approaches its asymptotic limit and hence is not convex, $f(x)$ instead grows and has a chance to be convex. E.g. if $\mathcal{N}_{\max} - \mathcal{N}(x) = O(|x|^{-a})$ at large x, then $f(x)$ grows as $|x|^a$, which is convex for $a > 1$.[23]

To see how this works in practice, consider the GFF-navigator plotted in Fig. 1.1, which is clearly not convex. Fig. 1.6 shows the result of applying to it transformation (1.53) with $\mathcal{N}_{\max} = 1$. We can see that the fractional linear transformation indeed improves the convexity of the objective function fed to BFGS. The function in Fig. 1.6 is still not globally convex, but it is locally convex, or close to it, in a much larger region than the original function in Fig. 1.1. We will see below that this transformation indeed results in more appropriate step lengths in the initial line searches and that BFGS has a higher rate of success of finding the Ising model minimum, even when starting in regions of relative flatness of the untransformed navigator.

In our studies we will use the standard implementation of the BFGS algorithm which can be found in the SciPy library [74], with some minor modifications. In Section 1.5.1, we review the BFGS method. We describe our modifications and their motivation in Section 1.5.2. We will see in Section 1.5.3 that the resulting algorithm gives good results when applied to the 3d Ising model case. Finally, we will comment in Section 1.5.4 on further possible improvements on our modified BFGS algorithm.

1.5.1 BFGS algorithm

Let $f(x)$ be the objective function to be minimized. The BFGS algorithm attempts to minimize $f(x)$ by taking successive steps $x_0 \to x_1 \to \ldots \to x_k \to \ldots$, where step k is taken using the information from an approximated quadratic model of the function at x_k. This approximate quadratic model is

$$f(x_k + \Delta x) \approx f(x_k) + \nabla f(x_k)^{\mathrm{T}} \Delta x + \frac{1}{2} \Delta x^{\mathrm{T}} B_k \Delta x \ , \tag{1.54}$$

where B_k is an approximation to the Hessian at x_k. After BFGS takes the k^{th} step $x_k \to x_{k+1}$, it determines the approximate Hessian at x_{k+1} by updating the

[22] For example, in the 1D case, we have $f''(x) \geqslant 0$ iff $\mathcal{N}''(x)[\mathcal{N}_{\max} - \mathcal{N}(x)] + [\mathcal{N}'(x)]^2 \geqslant 0$.

[23] If it turns out e.g. that $\mathcal{N}(x)$ approaches its asymptotic limit as an inverse power of x, but with $a < 1$, one could consider the modified function $f(x) = \frac{\mathcal{N}(x)}{(1-\mathcal{N}(x)/\mathcal{N}_{\max})^k}$ with $k > 1$.

one at x_k using only gradient information at x_k and x_{k+1}. For the full updating formula, see (6.19) of [73], Sec. 6. The minimum of the quadratic model (1.54) is the so-called "Newton step"

$$p_k = -B_k^{-1} \nabla f(x_k) \ . \tag{1.55}$$

In Newton's method, at each iteration the Newton step would be taken, so that $x_{k+1} = x_k + p_k$. In BFGS, the Newton step is replaced by a line search in the direction of p_k. An *exact* line search would correspond to

$$x_{k+1} = \arg \min_{\alpha > 0} \phi(\alpha), \qquad \phi(\alpha) := f(x_k + \alpha p_k) \, . \tag{1.56}$$

In practice, one uses an *inexact* line search, which means that one looks for an "approximate" minimum of $\phi(\alpha)$ at $\alpha > 0$. It turns out that a rather rough approximation is sufficient for good performance of the algorithm. A typical termination criterion is the "strong Wolfe conditions:"

$$\phi(\alpha) \leqslant \phi(0) + \mu \, \alpha \, \phi'(0) \, , \tag{1.57}$$

$$|\phi'(\alpha)| \leqslant \eta \, |\phi'(0)| \ . \tag{1.58}$$

The first condition enforces that the function decreases sufficiently. The parameter μ controlling this is usually chosen to be very small. We used the default value $\mu = 10^{-4}$ implemented in SciPy. The second condition demands that the gradient decreases sufficiently. This is usually called the curvature condition, and it guarantees that the BFGS update to the Hessian maintains positive-definiteness,[24] which in turn implies that p_{k+1} will be a decrease direction, allowing the algorithm to proceed. The parameter η controlling the demanded decrease is usually chosen somewhat below 1. We used the default value $\eta = 0.9$, with satisfactory results. The SciPy BFGS algorithm used in this chapter relies on the Moré-Thuente line search algorithm [76], a standard and robust algorithm for finding points obeying the strong Wolfe conditions.

Once an "accepted" point, i.e. a point obeying these conditions, is found, the

[24] This condition in particular trivially implies $(x_{k+1} - x_k) \cdot (\nabla f(x_{k+1}) - \nabla f(x_k)) > 0$. The latter condition guarantees that the BFGS Hessian update preserve positive definiteness; one should be able to convince oneself that this is the case by inspection of (6.17) in [73]. See [75], Theorem 3.2.2 and the top of p.56 for a proof and discussion.

Hessian is updated,[25] and the BFGS algorithm proceeds with its next step. The algorithm terminates once the norm of the gradient gets smaller than some value g_{tol} supplied by the user (we used $g_{\text{tol}} = 10^{-5}$).

It's worth pointing out that in the line searches, the Newton step $\alpha = 1$ is used as the initial guess. Once the Hessian has been well-approximated (as may happen towards the end of the minimization run), the first step $\alpha = 1$ will usually be accepted, and a convergence similar to that of Newton's method is expected. On the contrary, $\alpha = 1$ may not be a good guess at the beginning of the run unless we have an idea of the typical size of the region in which the minimum is expected to lie. This is provided via a bounding box in our modified BFGS algorithm described below.

1.5.2 Modified BFGS algorithm

The BFGS algorithm requires an initial guess for the Hessian at the first step B_0. This guess is usually taken to be the identity, which does not take into account the different scalings of the different variables. This is often okay because the BFGS algorithm recovers scale information after a sufficient number of steps have been taken, i.e. once the Hessian approximation becomes accurate in all directions. However, we still found that if some idea of the scale of the problem is known, e.g. if we have a vague idea about the location of the allowed region, it is best to incorporate this information into the initial Hessian. By setting a well-scaled initial Hessian, an appropriate step length in the initial line searches can be achieved. (Recall that the initial line search step is always $\alpha = 1$ in the direction of p_k, and the length as well as the direction of this search clearly depends on B_k via (1.55).) This will ensure that the BFGS algorithm explores the vicinity of the starting point rather than a much larger space—a crucial feature in cases where we are interested in one specific nearby local minimum. For example, when we want to study the 3d Ising model, we are not interested in studying the navigator in the big allowed "continent" found at large Δ_σ [45]. We found that the BFGS algorithm may end up in this much larger feasible region unless an appropriately scaled initial Hessian is supplied.

One trick to set an appropriately scaled initial Hessian (based on [73], p.142)

[25] Note that the line search will also involve evaluating the function and its gradient at several intermediate points along the direction p_k, until a point satisfying the strong Wolfe conditions is found. In the BFGS algorithm, information from those intermediate points is not used in any way to improve the Hessian.

is the following: Compute the gradient at the initial point, and set B_0 to

$$B_0 = \|\nabla f(x_0)\| \operatorname{diag}\Big(\frac{1}{\alpha_0^1}, \cdots, \frac{1}{\alpha_0^n}\Big). \tag{1.59}$$

Then, from (1.55), the initial Newton step $\alpha = 1$ will result in probing the function at $x_0 - \operatorname{diag}(\alpha_0^1, \cdots, \alpha_0^n) \cdot \frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}$. Hence the parameters α_0^i have the meaning of the characteristic desired $|\Delta x^i|$ during the initial step of the first line search. Alternatively, one could use the procedure described in Appendix A.3 to explicitly compute the initial Hessian. However we do not advise this, since the Hessian for a point far away from the minimum could very well not provide an accurate scale for the problem, nor is the Hessian far away from the minimum likely to provide a more accurate starting point for the approximation of the Hessian at the minimum than an appropriately scaled diagonal matrix.[26]

Apart from specifying the initial Hessian, some minor modifications have to be made in order to apply the BFGS algorithm to conformal bootstrap problems. Firstly, the navigator function is naturally defined only in certain regions and not globally. Consider for example the case of the 3d Ising model. The navigator function is naturally defined only for Δ_σ and Δ_ϵ above the unitarity bound. Similarly, when demanding the existence of exactly one relevant parity odd and even singlet, we restrict the domain of $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ to values Δ_σ or Δ_ϵ below 3. Additionally, as discussed above, one might only be interested in minima or negative values that are located in a certain region around the starting point.

Hence, we ask the user to provide a bounding box for the search space, past which we do not allow the search to move. This constraint is implemented by altering the line search such that if a step outside of the bounding box would be taken, the maximal step in the same direction within the boundaries is taken instead. If this point on the edge is accepted, i.e. obeys the strong Wolfe conditions, we check whether the new search direction points inside or outside of the bounding box. If the new search direction points outside of the bounding box, but the gradient descent direction lies inside, the search direction is taken to be the gradient descent direction for the next step. If neither the initial search direction nor the negative gradient lie inside the bounding box, the search is terminated. The user

[26] On the contrary, having access to the exact rather than BFGS-approximated Hessian is expected to speed up the last stage of the minimization run, although we have not took advantage of this possibility in this work.

Input: A navigator function $\mathcal{N}(x)$, an initial guess x_0, the bounding box coordinates $b^i_{\min}, b^i_{\max}$ and a value g_{tol}.
Output: The final point x_f and the termination message.
begin
  $f(x) = \frac{\mathcal{N}(x)}{1-\mathcal{N}(x)/\mathcal{N}_{\max}}$
  $\alpha^i_0 = 0.2 \times (b^i_{\max} - b^i_{\min})$
  $B_0 = \|\nabla f(x_0)\| \operatorname{diag}(\frac{1}{\alpha^x_0}, \frac{1}{\alpha^y_0}, \cdots)$
  $p_0 = -B_0^{-1} \nabla f(x_0)$
  while $\|\nabla f(x_k)\| > g_{\text{tol}}$ **do**
    $\alpha = \text{linesearch}(f, x_k, p_k, B_k)$
    $x_{k+1} = x_k + \alpha p_k$
    The hessian B_k is updated to B_{k+1}
    The search direction p_k is updated to p_{k+1}
    if x_{k+1} *is at the boundary* **then**
      if p_{k+1} *points back inside the bounding box* **then**
        continue
      else
        if $-\nabla f(x_{k+1})$ *points back inside the bounding box* **then**
          $p_{k+1} = -\nabla f(x_{k+1})$
        else
          return x_{k+1} and the termination message *"Out of the bounding box"*
        end
      end
    end
    Optional: **if** $f(x_{k+1}) < 0$ **then**
      return x_{k+1} and the termination message *"Found a negative point"*
    end
  end
  return x_k and the termination message *"Minimum found: gradient is smaller than the tolerance"*
end

Algorithm 1.1: Modified BFGS algorithm

should then either try a different initial point or change the bounding box.[27]

The provided bounding box is also used to specify the desired step lengths in the initial Hessian of (1.59). We found satisfactory results by setting the desired step lengths in each direction to be 20% of the supplied bounding box.

It is fair to ask how the user will know which bounding box to specify. We assume that the user has some idea of the range of parameters they want to explore. Results obtained at lower derivative order Λ can also be used for guidance, as well as estimates of CFT data coming from other methods such as RG or Monte Carlo simulations.

The BFGS algorithm including these modifications is summarized as Algorithm 1.1.

1.5.3 Minimization results

To illustrate the effectiveness of our minimization algorithm, we apply it to the classic conformal bootstrap problem of finding an allowed point corresponding to the 3d Ising model using the system of correlators $\{\langle\sigma\sigma\sigma\sigma\rangle, \langle\sigma\sigma\epsilon\epsilon\rangle, \langle\epsilon\epsilon\epsilon\epsilon\rangle\}$, which contains the lowest dimensional $\mathbb{Z}_2$-odd scalar σ and the lowest dimensional $\mathbb{Z}_2$-even scalar ϵ, under the assumption that those operators are the only relevant ones, as described in Section 1.2.2. We will see that the navigator function enables us to locate an allowed point with a relatively small number of SDPB calls. Finding an allowed point naively by checking feasibility for a dense grid of points covering the search space would take orders of magnitude more SDPB calls.

Of course, in a decade of feasibility searches many useful tricks have been found to speed them up.[28] Still, we foresee that navigator-function methods will offer even better performance. They should eventually allow computations in more complicated setups involving an even higher number of parameters to scan over, such as e.g. bootstrapping the full system of $\sigma, \epsilon, \epsilon'$ 4pt functions, which were not possible to treat so far via feasibility-based methods.

[27] Note that if this happens, it may mean that that boundary includes some part of the attraction basin for a minimum that lies outside the bounding box. In this case an alteration of the relevant boundary is probably advised.

[28] E.g. for Ising and $O(N)$ we can use the fact that they live close to the kink in a single correlator bound, for Ising we can use c-minimization [63], OPE scans can be replaced with the cutting surface algorithm [29], etc.

2-parameter searches

We start with the 2-parameter case which is easier to visualize. So we minimize $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ of Section 1.2.2. We use $\Lambda = 11$ and the bounding box $[0.510, 0.530] \times [1.30, 1.50]$, i.e. the same range as in Fig. 1.1. Running our algorithm for 10 different starting points chosen at random within this bounding box, the number FC of function calls to reach a point of negative navigator value was $9 \leqslant FC \leqslant 31$, while $\overline{FC} = 19.3$ on average. All runs terminated at essentially the same point (with an error controlled by $g_{\rm tol}$)

$$\begin{aligned} x_f &= (\Delta_\sigma, \Delta_\epsilon) = (0.5182861212(4), 1.41521640889(6)), \\ N(x_f) &= -0.00267253307546000(2), \end{aligned} \tag{1.60}$$

where the tiny error bars show the largest difference observed between different runs. We conclude that the minimum is unique and all the runs terminate near it.

A representative run is shown in Fig. 1.7, where the numbered points correspond to the path taken by our modified BFGS algorithm. Convergence rate in this run is illustrated in Fig. 1.8(left) where we plot the navigator values $\mathcal{N}_i$ returned by subsequent function calls, until the negative navigator region is reached. This plot can be correlated with the navigator shape in Fig. 1.1, which features an arrow-shaped valley around the Ising island (see Section 1.3). Thus, Fig. 1.8(left) shows a period of modest progress in the minimization of $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$, in some sense looking for the the valley. This is followed by a period of fast decrease once the valley is found (at around $i = 25$).

Another way to evaluate the convergence rate is shown in Fig. 1.8(right), where we plot for the same run the distance $\|x_i - x_f\|$ between the point x_i and the eventually found minimum x_f. This measure of convergence is appropriate also for the region where $\mathcal{N}(x) < 0$. This plots show a period of greatly accelerated convergence towards the end of the run. Indeed, we expect Newton-like, i.e. superlinear,[29] convergence in the final stages of the BFGS algorithm. Similar plots for six more

[29] Recall that superlinear means $\epsilon_{i+1} = o(\epsilon_i)$ where ϵ_i is the error after step i. The Newton method has quadratic convergence, $\epsilon_{i+1} = O(\epsilon_i^2)$, while for the BFGS only weaker theorems showing superlinear convergence are available [73]. One-dimensional bisection in this notation has linear convergence, $\epsilon_{i+1} \leqslant \alpha\epsilon_i$ with $\alpha < 1$.

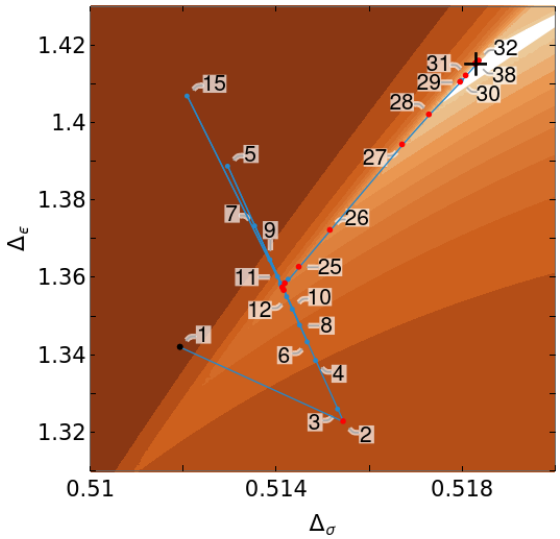


Figure 1.7: A representative run of our algorithm, see Section 1.5.3. Only the relevant part of the bounding box $[0.510, 0.530] \times [1.30, 1.50]$ is shown. *Black dot:* the initial point x_0. *Red dots:* points x_k accepted by the line searches as satisfying the strong Wolfe condition. *Blue dots:* intermediate points where the function was evaluated during the line searches. *Black cross:* position of the found minimum. *Background:* contour plot of the navigator function $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)$ (darker colors correspond to higher function values, and the white area to the negative navigator, i.e. the Ising island). This run took 29 function evaluations to reach the island, and 66 function evaluations to reach the minimum within the specified g_{tol} (see Fig. 1.8). Only the first 38 points are marked, the rest being too closely spaced to be distinguishable.

runs are collected in App. A.5.1.[30]

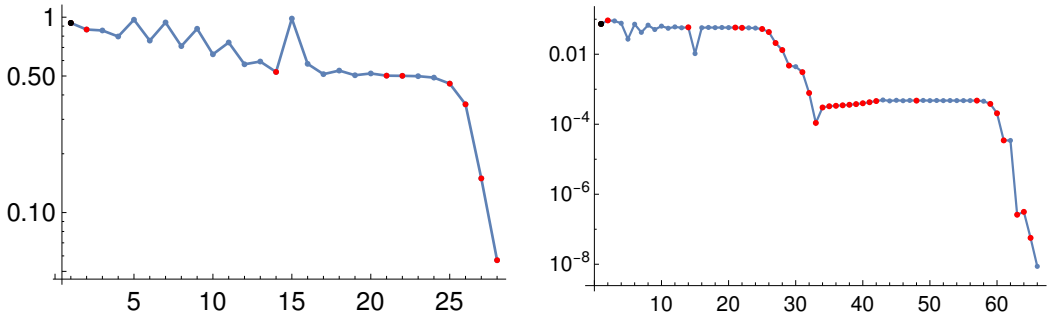


Figure 1.8: These plots refer to the run of our modified BFGS algorithm shown in Fig. 1.7, and use the same color code for the dots. *Left:* Navigator value $\mathcal{N}_i$ at the i-th function call. Only the function calls before reaching the negative navigator region are shown in this logarithmic plot. Naturally, function values decrease monotonically along the red dots (points accepted by the line searches), while this condition does not have to hold for the blue dots. *Right:* Logarithmic plot of $\|x_i - x_f\|$ at the i-th function call.

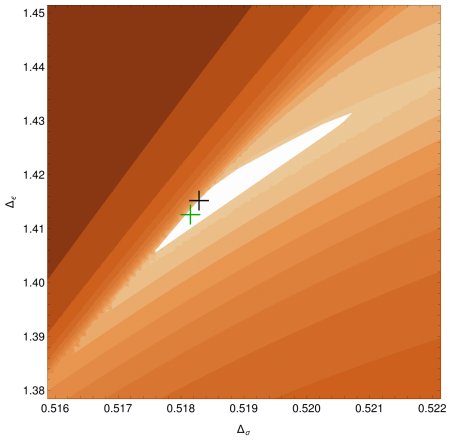


Figure 1.9: This plot shows that minimum (1.60) of the $\Lambda = 11$ navigator (black cross) is very close to the best available estimate of the true location of the Ising model [30] (green cross), considering the size of the $\Lambda = 11$ Ising island (white region).

[30]Once the navigator function is negative, we often observe a bit of a plateau, for example between iterations 30-60 in Fig. 1.8. At that point we are relatively close to the minimum, but the long sequences of blue dots indicate that the BFGS quadratic model of the navigator function is not yet accurate. It is quite possible that this is caused by the non-C^2 locus that we identified above, and it would certainly be interesting to investigate this further. Either way, the algorithm eventually recovers and then continues to converge rapidly to the minimum.

Finally, we observe that minimum (1.60) of the $\Lambda = 11$ navigator function gives a good prediction for the actual location of the Ising model, as compared to a generic point in the Ising island. Indeed, the distance between this minimum and the best prediction from [30] (3-parameter scan at $\Lambda = 43$) is only $\sim 10\%$ of the size of the $\Lambda = 11$ island, see Fig. 1.9.

3-parameter searches

We will present next the tests for the three-parameter navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, \theta)$. We used $\Lambda = 19$, the bounding box $[0.510, 0.530] \times [1.30, 1.50] \times [0.8, 1.1]$, and 20 random initial points within it.

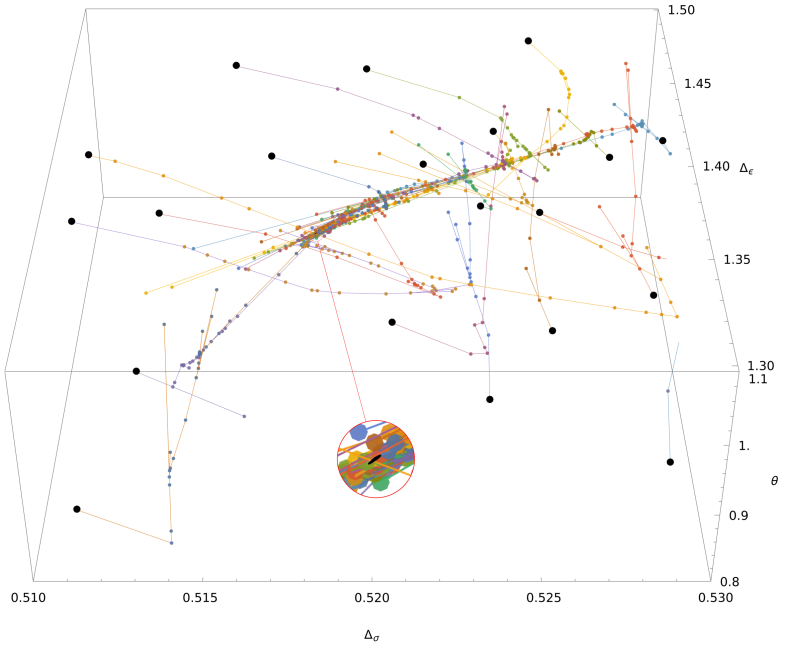


Figure 1.10: BFGS runs starting at 20 random points from of the bounding box $[0.510, 0.530] \times [1.30, 1.50] \times [0.8, 1.1]$, at $\Lambda = 19$. Initial points are black. Except for two runs that terminated at the boundary (one of them is in the lower right), all the others converged to the same minimum inside the Ising island (see the tiny black shape in the the magnified inset).

These runs are shown in Fig. 1.10. Eighteen of them successfully converged to the same minimum inside the $\Lambda = 19$ Ising island:

$$\begin{aligned} &x_f = (\Delta_\sigma, \Delta_\epsilon, \theta) = (0.5181536110(7), 1.412692879(8), 0.969334757(6)), \\ &N(x_f) = -0.0000208827730(5). \end{aligned} \tag{1.61}$$

A typical successful run is shown separately in Fig. 1.11.

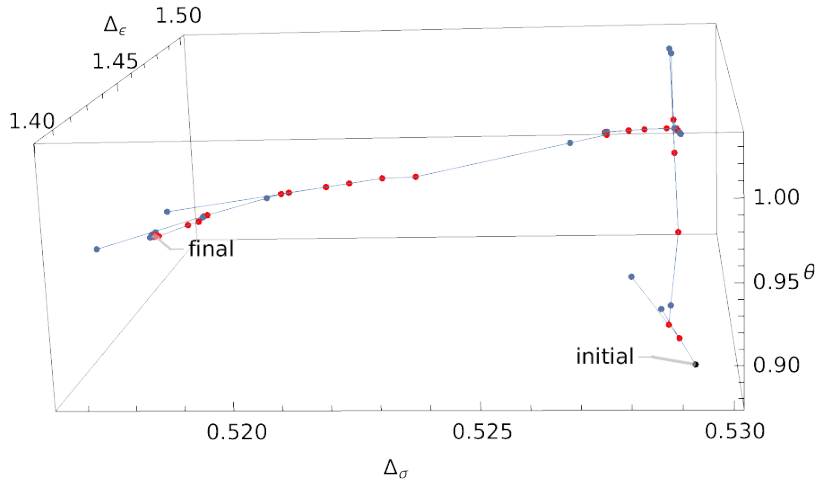


Figure 1.11: A typical BFGS run from Fig. 1.10 (only a part of the bounding box is shown). First, the search is seen to be looking for the "valley", and once it has found it, converges rapidly to the Ising island.

Two runs terminated at the boundary of the bounding box with both the subsequent BFGS search direction and the gradient pointing outside, according to the safe-guarding procedure (see Algorithm 1.1). This suggests that these points were being attracted by a minimum outside the bounding box. By inspection, these runs started close to the edge of the bounding box in regions where the navigator surface is non-convex even after applying transformation (1.53).

Limiting to the successful runs, it took on average 50.3 function calls to reach the negative navigator region. Of course, the $\Lambda = 19$ island is orders of magnitude smaller in all directions than the bounding box. This demonstrates our point that the navigator minimization method is capable of finding a small isolated island given even a rough estimate of its location. We will comment in Section 1.5.4 on an iterative way to speed up high-Λ calculations.

Using the run in Fig. 1.11 as an example, we show its rate of convergence in Fig. 1.12, following the same conventions as in Fig. 1.8. Comparing Figs. 1.11 and 1.12, it's easy to reconstruct what is going on. The initial line searches are spent finding the bottom of the valley. Once this is found, the algorithm quickly manages to follow the valley towards the negative navigator region. Similar plots for six more runs from Fig. 1.10 are collected in App. A.5.2.

It's worth pointing out that in both Fig. 1.8(right) and Fig. 1.12 we see two periods of accelerated convergence: one when the negative region is approached

and another towards the end of the run. The slower rate of convergence in between might be due to the function exhibiting some local concavity, or due to a large change in the local Hessian. We have not investigated this in detail.

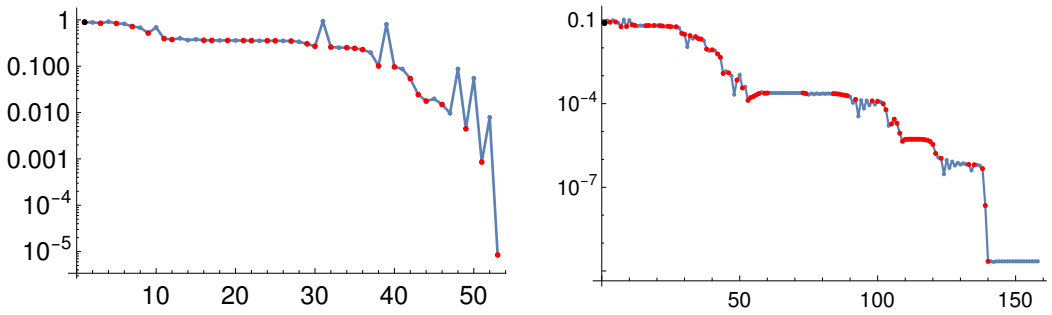


Figure 1.12: These plots refer to the run of our modified BFGS algorithm shown in Fig. 1.11, and follow the same convention as Fig. 1.8, with $\mathcal{N}_i$ on the left and $\|x_i - x_f\|$ on the right.

1.5.4 Other algorithms and possible improvements

We have shown in the previous section that navigator minimization using our modified BFGS algorithm offers a robust and efficient method for finding an allowed point. However, there are bound to be avenues for improvement. We will remark on some potential improvements in this section. We hope that the algorithm presented here sets a good benchmark to which future algorithms will be compared.

In order to efficiently find an allowed point at high values of Λ, one could imagine an iterative procedure where the navigator minimum point at a lower derivative order is used as an initial point for a minimization run at a higher derivative order (perhaps reducing the bounding box, or inheriting the Hessian estimate from the lower Λ BFGS run). This is expected to perform well for two reasons. First, because the navigator minimum provides an excellent estimate of the position of the Ising model, see Section 1.5.3, and hopefully also of other CFTs. Second, because of the accelerated convergence properties of the BFGS algorithm after reaching the convex region around the minimum (see Figs. 1.8 and 1.12). Using the exact Hessian computed as explained in Appendix A.3 may be also especially beneficial in the convex region around the minimum.

In the above, we did not make use of the fact that the minimum of the navigator occurs close to $\mathcal{N}(x) = 0$, and that in some cases, one will only be interested in

reaching any negative point rather than the minimum. This information could be e.g. incorporated in the initial guess for the Hessian, by scaling the identity matrix such that the initial step aims towards zero of $\mathcal{N}(x)$ in a first-order expansion around the initial point (instead of scaling it so that the initial step explores some percentage of the bounding box, as done here)

As discussed before, we have also found that the navigator function is not globally convex. We have found that in our case, this problem can be mitigated by minimizing another related, more convex function instead, Eq. (1.53). Even in regions of non-convexity of this transformed function, we have found that line searches provided robustness to the algorithm. Still, other bootstrap problems may require more care when dealing with non-convexity. In such cases, algorithms where the updating formula for the approximate Hessian does not enforce positive-definiteness could be advantageous, see [77].

We have opted in our algorithm to constrain the search space in a rudimentary way via the bounding box, and found this to be adequate for our needs. With that being said, there exist a myriad of other algorithms for constrained optimization that could offer more robustness with the way they deal with constraints. Here we mention two included in SciPy: L-BFGS-B, a bounded limited memory version of the BFGS method optimized for dealing with problems with search spaces with a large number of dimensions, and SLSQP, allowing general, as opposed to box, constraints. See [73] for more information on constrained optimization.

Similarly, there are many unexplored avenues for parallelization. One could imagine parallelizing the line search, or using an inherently parallel optimization algorithm, in the spirit of particle swarms [78]. Particle swarm algorithms that we have seen do not make use of gradient information. Since we have gradients for free (Section 1.4), it would be desirable to develop a similar algorithm taking advantage of the gradients.

1.6 An application: exploring the tip of an island

In order to connect the Ising island to physical observables it is important to know its extreme points. For example, the left- and rightmost point of the island provide a rigorous lower and upper bound on the critical exponent $\eta = 2\Delta_\sigma + 2 - d$. In previous applications such bounds were often found by simply mapping out the entire island, using a higher-dimensional analogue of a binary search based on a Delaunay triangulation, and then locating its extremal points. A more systematic

triangulation algorithm, suitable for parallelization, was introduced in [27] and used to determine the instability of the $O(3)$ fixed point.

In future bootstrap applications one might want to study more complicated systems of correlators and this inevitably means the introduction of new parameters. If we wish to locate the extremal point of an island in such a higher-dimensional space then any triangulation algorithm based on a sequence of feasible and infeasible points will scale extremely poorly. A constrained optimization algorithm based on a navigator function is much less sensitive to the dimensionality of the parameter space and will perform much better. We therefore expect that the use of a navigator function is essential for the high-precision determination of critical exponents in the future.

In the next section we present a simple algorithm inspired by these general ideas. We will then maximize Δ_σ in the Ising island as an illustration.

1.6.1 A constrained optimization algorithm

Suppose we want to locate an extremal point of the allowed region in the direction specified by a vector n. The problem is then:

$$\text{maximize } n^T x \qquad \text{over all } x \text{ such that } \mathcal{N}(x) \leqslant 0\,. \tag{1.62}$$

We will use optimality conditions

$$\mathcal{N}(x) = 0 \qquad \text{and} \qquad \left(I - \frac{nn^T}{n^T n}\right)\nabla\mathcal{N}(x) = 0\,, \tag{1.63}$$

where the latter equation sets to zero all components of the gradient orthogonal to n. We propose to work towards a solution of these equations in a manner inspired by the quasi-Newton method from Section 1.5. We will now explain the full algorithm (see Algorithm 1.2 below for a summary).

As in Section 1.5 we will use a quadratic model around a point x_k:

$$\mathcal{N}(x) \approx \mathcal{N}^{(2)}(x) := \mathcal{N}(x_k) + \nabla\mathcal{N}(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k)\,. \tag{1.64}$$

The function and the gradient at x_k are assumed known, while B_k can be either the exact Hessian at x_k (computed as explained in Appendix A.3), or an approximation like the one obtained from the BFGS method. In the following we will assume that $B_k \succ 0$.

Substituting the quadratic model in (1.63) we find a simple system involving one quadratic and many linear equations, which can be solved exactly, yielding two solutions.[31] These are real if x_k in the allowed region, so that $\mathcal{N}(x_k) < 0$, and by continuity also in some domain outside the feasible region. In this case the surface $\mathcal{N}^{(2)}(x) = 0$ is an ellipsoid, and the second condition in (1.63) picks out the extremal points of this ellipsoid along the n direction. Some distance away from the allowed region the ellipsoid shrinks to zero size and the solutions become complex-conjugate. We denote by $x_\#$ the real solution which has the largest value of $n^T x$, when the solutions are real. When the solutions are complex conjugate, we let $x_\#$ denote their real part (and then $x_\#$ turns out to simply correspond to the minimum of the model function).

Denote $p_k = x_\# - x_k$; this is our search direction. The next point x_{k+1} is then found using a line search along p_k starting from x_k. We use the initial step length $\alpha = 1$, however the rest of the line search algorithm is not the Moré-Thuente algorithm used in BFGS. This should not be surprising since we are now solving a different problem. Instead of minimizing $\mathcal{N}(x)$ we would now like to maximize $n^T x$ while moving along a trajectory remaining close to the boundary of the allowed region (but not exactly along the boundary). One could think that a safe choice would be to remain always inside the allowed region (a sort of interior point algorithm). We have found however that a much faster algorithm results if we allow the algorithm to choose points on both sides of the boundary. To make sure that the algorithm does not veer off too much away from the boundary, we impose

$$\mathcal{N}(x_{k+1}) \leqslant \lambda_{\rm rel} |\mathcal{N}(x_k)| \tag{1.65}$$

with a parameter $\lambda_{\rm rel} > 0$. Clearly $\lambda_{\rm rel} < 1$ would be safer but might slow down the algorithm in the later stages. We found it advantageous to use $\lambda_{\rm rel}$ somewhat above 1, e.g. $\lambda_{\rm rel} = 2$ works well.

So (1.65) is our line search termination condition. In practice, this condition with $\lambda_{\rm rel} = 2$ is not very constraining and the initial step $\alpha = 1$ is almost always accepted if we start with a good initial Hessian. (E.g. in the run shown in Section 1.6.2 this happened for 100% of the steps.) In the cases that the initial step $\alpha = 1$ does not obey Eq. (1.65), we proceed as follows. We construct cubic polynomial

[31] This is where our algorithm differs significantly from conventional constrained optimization algorithms like sequential quadratic programming methods or interior point methods (see e.g. [73]). The latter solve a linear system at each step in order to be applicable very generally. Such a linearization is unnecessary here because we only have a single quadratic equation.

approximation $P(\alpha)$, fitted to match the value and gradient at the initial point and the previous line search point. If x_k is in the feasible region we choose the next α by solving $P(\alpha) = 0$, and if not then by minimizing $P(\alpha)$. Iterating this, eventually we find an α such that $x_{k+1} = x_k + \alpha p_k$ satisfies (1.65).

Once we have accepted x_{k+1}, we construct a new quadratic model around this point. In particular, if the approximate Hessian is used, then B_k is updated as in BFGS. However, the update is carried out only if the curvature condition is obeyed at x_{k+1}; as explained in footnote 24 this is sufficient to ensure that $B_{k+1} \succ 0$. If the curvature condition is not satisfied, then the Hessian is not updated.

We then repeat the process. The algorithm terminates if the conditions (1.63) are obeyed within a certain tolerance.

1.6.2 The tip of the Ising island

As an example, let's apply the above algorithm to find the maximal value of Δ_σ within the Ising allowed island $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon) \leqslant 0$ where $\mathcal{N}$ is the 2-parameter navigator for the Ising 3-correlator setup at derivative order $\Lambda = 11$.[32] The search was started from the navigator minimum reached via a BFGS run, and with the initial Hessian approximation B_0 inherited from BFGS, which is expected to be close to the true Hessian. The algorithm path is shown in Fig. 1.13. The algorithm took 17 steps to reach the tip of the island, i.e. the point with maximal Δ_σ. Termination condition $\max(|\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)|, |\partial_{\Delta_\epsilon}\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)|) \leqslant g_{\text{tol}}$ was satisfied with $g_{\text{tol}} = 10^{-27}$.

For comparison, Fig. 1.13 also show the blue allowed region obtained from the Delaunay triangulation method. We finely sampled the zoomed-in region around the very tip of the island, with a total number of sampled points being around 480 [33]. In contrast, our algorithm takes only 10 steps to locate the maximal Δ_σ point more accurately than the triangulation resolution. The line search never had to be activated, the initial try $\alpha = 1$ having been accepted in 100% of the steps.

In Fig. 1.14 we show the convergence rate towards the minimum. These plots

[32] In this test, unlike in Section 1.3, we have not imposed the OPE relation $\lambda_{\sigma\sigma\epsilon} = \lambda_{\sigma\epsilon\sigma}$, i.e. the navigator was defined imposing positivity separately on the two terms in (1.29), which is precisely the setup in [45]. There is no particular reason for this difference with Section 1.3.

[33] To test the feasibility of those points, we only require SDPB to find primal/dual jumps. In general such a run is quicker than a typical SDPB run for an optimal solution. However, in terms of total number of SDPB iterations, we find that the 480 feasibility runs correspond to around 4705 SDPB iterations, while the 17 optimal runs correspond to around 1080 SDPB iterations. We still conclude that our method has a significant advantage.

Input: A navigator function $\mathcal{N}(x)$, a vector n indicating the maximizing direction, a precision goal g_{tol} and a line search parameter λ_{rel}.
Output: The final point x_f.
begin
  Use Algorithm 1.1 to construct a feasible point x_0 and Hessian estimate B_0
  $x_{\text{lastBFGS}} = x_0$
  $B_{\text{lastBFGS}} = B_0$
  while $\| \left(I - (n^T n)^{-1} n n^T\right) \nabla \mathcal{N}(x_k)\| > g_{\text{tol}}$ *or* $|\mathcal{N}(x_k)| > g_{\text{tol}}$ **do**
    $p_k = \text{search\_direction}(x_k, n, \mathcal{N}(x_k), \nabla\mathcal{N}(x_k), B_k)$
    $\alpha = 1$
    while $\mathcal{N}(x_k + \alpha p_k) > \lambda_{\text{rel}}|\mathcal{N}(x_k)|$ **do**
      $P(\alpha)$ is interpolating polynomial obtained from $\mathcal{N}(x_k)$, $\mathcal{N}(x_k + \alpha p_k)$ and their gradients
      if $\mathcal{N}(x_k) < 0$ **then**
        find α such that $P(\alpha) = 0$
      else
        find α such that $P(\alpha)$ is minimized
      end
    end
    $x_{k+1} = x_k + \alpha p_k$
    if $(x_{k+1} - x_{\text{lastBFGS}})^T(\nabla\mathcal{N}(x_{k+1}) - \nabla\mathcal{N}(x_{\text{lastBFGS}})) > 0$ **then**
      $B_{k+1} = \text{BFGS\_update}(x_k, B_k; x_{\text{lastBFGS}}, B_{\text{lastBFGS}})$
      $x_{\text{lastBFGS}} = x_{k+1}$
      $B_{\text{lastBFGS}} = B_{k+1}$
    else
      $B_{k+1} = B_k$
    end
  end
  Return x_k.
end

Algorithm 1.2: An algorithm for finding the extremal point of an island.

demonstrate superlinear convergence towards the end of the run, as should be expected from this type of algorithm.

We would like to warn the reader about a difference in spirit between our Algorithms 1.1 and 1.2. Algorithm 1.1 for navigator minimization is backed up by decades of experience in numerical optimization, and should be widely applicable without major modifications. On the other hand, Algorithm 1.2 is our own custom-made procedure. It served well the purpose to demonstrate the point that the navigator can be used to find extremal values of allowed parameters, but it has a somewhat tentative character and is expected to evolve more in the future.

For example, the Ising island is admittedly a simple model with a convex island and a single local maximum of Δ_σ. If the island does not have such a nice shape, Algorithm 1.2 can get stuck in a local optimum instead of the global optimum. In more realistic cases it is therefore important to have a rough idea of the shape of the island, and then perhaps an admixture of triangulation-based methods and navigator methods might be the best approach.

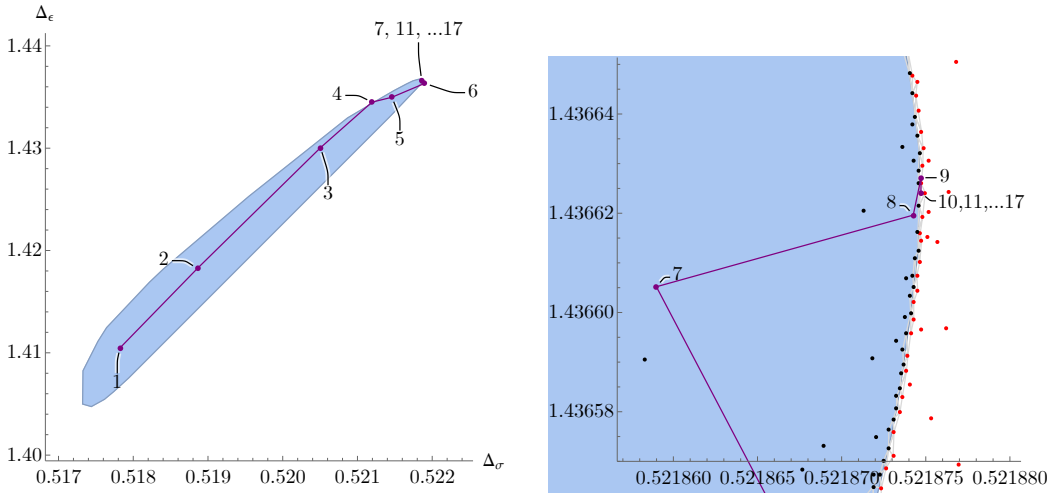


Figure 1.13: *Magenta path*: A run of Algorithm 1.2, see the text, with a zoom-in on the right. *Blue region*: the Ising island from the Delaunay method with black/red being the allowed/excluded points.

1.7 Conclusions and future directions

We have presented in this chapter a powerful alternative to the scanning-based approach employed so far in the numerical conformal bootstrap program. This came

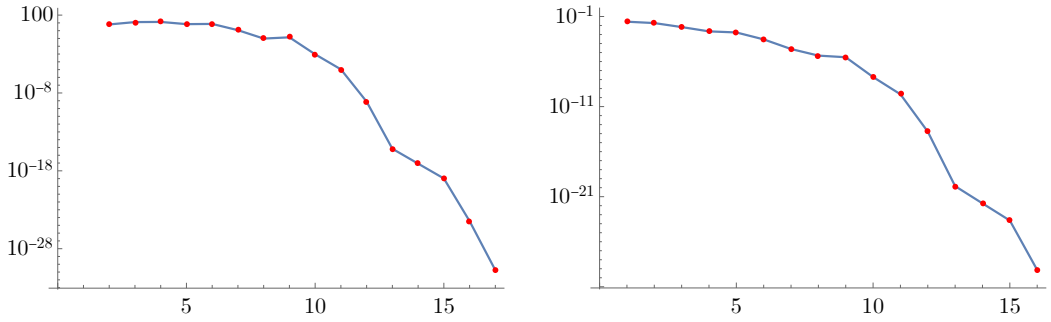


Figure 1.14: These plots refer to the run of our Constrained BFGS algorithm shown in Fig. 1.13. *Left:* Logarithmic plot of $|\partial_{\Delta_\epsilon}\mathcal{N}(\Delta_\sigma, \Delta_\epsilon)|$ at the i-th function call. *Right:* Logarithmic plot of $\|x_i - x_f\|$ at the i-th function call. Note that line search never had to be activated, as the initial step $\alpha = 1$ always satisfied condition (1.65) with $\lambda_{\text{rel}} = 2$ used in this run.

from the realization that there exist functions, for which we have coined the term "*navigator* functions," which measure how far a given point is from the boundary between allowed and disallowed regions and can thus be used to efficiently find an allowed point as well as the boundary of an allowed region. We have explicitly constructed two such navigator functions. It was shown that the computation of these navigator functions can be written as a semi-definite programming problem of the same form as an OPE maximization. Adding the generalized free field solution to the crossing equation has led us in Section 1.2.1 to the definition of the GFF navigator. The Σ-navigator was introduced in Section 1.2.1 as an another equally valid option.[34]

With the help of such functions, we have shown it is possible to quickly locate allowed regions in parameter space by ways of minimization. We have presented in Algorithm 1.1 a modified BFGS algorithm which does so quite efficiently. To prove this, we set out to study the canonical bootstrap problem of the 3d Ising model. First we showed that the navigator is (C^1) smooth and has no local minima in the disallowed region. With both a two-dimensional search space at $\Lambda = 11$, and a three-dimensional search space at $\Lambda = 19$, we have shown that it took on average

[34]While the GFF-navigator is naturally normalized, the Σ-navigator has its own set of advantages. It is actually easier to set up, since one does not have to work out the GFF OPE coefficients. In addition, there is not one but infinitely many Σ-navigators, corresponding to different choices of terms in the r.h.s. of (1.7) or (1.25), and this flexibility may prove useful in the future. At present, we see no definite reason to prefer one or the other navigator. For comparison, we performed some of the reported computations using both navigators (e.g. section 1.6.2), and they performed equally well.

a few dozen SDPB calls to find the Ising island (19.3 for the former, 50.3 for the latter), starting only from very conservative estimates of the parameters. This is competitive with previous methods for isolating islands and bounding CFT data. Moreover, these previous methods suffered from exponential scaling with the dimensionality of the search space. This constituted a major bottleneck for the kinds of problems that could be tackled: realistically only setups with a handful of free parameters could be considered. We expect that the scaling of the minimization-based navigator method with the number of parameters will greatly outperform scanning methods.

Crucially, efficient minimization of a navigator function, for example with the BFGS algorithm presented in this chapter, requires the knowledge of derivatives of the navigator function. We have derived the "SDP gradient formula," Eq. (1.48), which gives the variation of the objective function of an SDP as only a function of the variation of the SDP input parameters around the point where the derivative is requested. This means that computing derivatives does not require additional SDPB runs, making one function *and* gradient evaluation in a BFGS run just about equivalent in cost to one OPE maximization.

We also tested the efficiency of the navigator method to search for extremal parameter values allowed by the bootstrap constraints. So, we presented in Section 1.6 a way to find optimal bounds on CFT data using a custom-made constrained-optimization routine. The algorithm was able to walk in and around the allowed region and converge in 17 steps to the maximal allowed Δ_σ, determining it to an accuracy of $\sim 10^{-35}$.[35] A similar triangulation-based search only achieves an accuracy of 10^{-6} even after testing over 400 points, see Fig. 1.14. Again we expect that the increase in performance can only become greater as the dimensionality of the search space increases.

We feel that the applications shown in this chapter demonstrate only a small part of the power the navigator method, and we are hopeful that the future will show it to be a great addition to the toolbox of all bootstrap enthusiasts.

We would like to conclude by mentioning here some ideas that are worth exploring using this new tool. Some of these applications, being out of reach of traditional bootstrap techniques, were among our chief motivations to start thinking hard about the navigator function.

One class of situations where navigator is going to be useful is when we know

[35] The order of magnitude for the difference of last two points in Δ_σ is around 10^{-35}. Another estimation is that $\mathcal{N}(x)/\|\nabla\mathcal{N}(x)\|$ for the last point is around 10^{-37}.

a solution to bootstrap constraints for some value of a parameter (such as space dimension d or the symmetry group rank N) and we would like to perform a deformation in this parameter. We will actually consider such situations in the remaining chapters. We may imagine doing this by considering a navigator function depending on the dimensions of several exchanged operators, and imposing sparsity of the exchanged spectrum. Among other things, this should allow a more robust determination of critical values of parameters when bootstrap solutions disappear, than the more traditional approach of looking for kinks and trying to see when those kinks get rounded off. One long-standing problem which we will not consider in this thesis, but which was considered in [79] since this original navigator work, is determining the upper critical dimension of the 3-state Potts model. Including exchanged operator dimensions among the arguments of the navigator function could also provide a useful (and more rigorous) alternative to estimating the spectrum using the extremal functional method [63, 64, 80], a question which was considered not long after this original navigator work in [81].

The use of the navigator function to quickly find extremal allowed values (Section 1.6) will benefit all cutting-edge bootstrap computations. One problem on our to-do list is to bootstrap the system of correlators in O(3) symmetric CFTs involving lowest scalar primaries in vector (ϕ), scalar (s), rank-2 tensor (t), and rank-4 tensor (t_4) O(3) representations. This setup extends that of [27] by including t_4 as an external operator. The physics interest in doing so is that it will allow access to the OPE coefficient λ_{t_4,t_4,t_4}, and other data needed to study the RG flow leading from the O(3) fixed point to the cubic fixed point in conformal perturbation theory (see [27], Section 5). The parameter space for this problem is 13-dimensional (4 Δ's and 9 OPE coefficients), out of reach of traditional approaches, but we expect that the navigator function will put it within reach.

As a final example, the navigator functions will allow an exploration of hybrid methods where the numerical bootstrap data is complemented with analytical data at high spins obtained from the light-cone bootstrap, as suggested in Section 9.1 of [82], and eventually realized in [83]. We imagine a navigator function depending on many parameters accurately parametrizing one or more Regge trajectories. In this context a navigator function will be very useful not only to localize an allowed point, but also because the minimum of the navigator offers a natural "most feasible point" that can be used to compare different parametrizations.

Chapter 2

Navigating through the $O(N)$ archipelago*

* Published in essentially identical form as
B. Sirois, "Navigating through the $O(N)$ archipelago," SciPost Phys. 13 no. 4, (2022) 081

2.1 Introduction

In many cases, the numerical conformal bootstrap in $d > 2$ has shown that the combination of crossing symmetry and unitarity is often sufficient in isolating theories in finite regions, or "islands", inside of the space of CFT-data (see e.g. [45, 84–89]). Crucially, the shape and size, or even existence of these islands depend upon the assumptions one makes on the spectrum of the theory he/she is trying to isolate. These assumptions are usually extrapolated from a certain perturbative regime (ϵ-expansion, large-N, etc.). Their validity in the non-perturbative regime is sometimes up in the air, and one usually has to play around with these assumptions. It is therefore natural to want to solve these theories directly in the perturbative regime, where the gap assumptions can be made robustly, and flow towards the non-perturbative regime, adjusting the gap assumptions as necessary. One might also be interested in following a CFT through some external parameter space because he/she expects something interesting to happen, for example a collision of two fixed points, at a critical value p_c of the external parameter(s) p. Such an enterprise raises certain concerns:

- Performing a scan of the search space for each external parameter value is often times prohibitively expensive, even more so if there are many external parameters. One would benefit greatly if there was a way that both the flow in the external parameter space and in the search space could be made more efficient.

- For certain ranges of external parameters, like fractional spacetime dimensions, CFTs are expected to be non-unitary [90], although the bootstrap of the Ising Model was shown to be insensitive to this [91] (see [92–96] for other unitary bootstrap studies of non-unitary CFTs).

The goal of this chapter is thus twofold: firstly, we would like to study the consequences of the non-unitary nature of another prototypical CFT, the d-dimensional $O(N)$ model, due to fractional values of d [90] and N [97]. In the process, we will present estimates of scaling dimensions, both from the bootstrap and from the resummation of six loop $4-\epsilon$ expansions [46, 98], for the range $(d, N) \in [3, 4] \times [1, 3]$. We will show that above $N = 1$, the bootstrap seems insensitive, to our degree of precision, to the non-unitarity due to fractional values of d or N. $N = 1$ will represent for our analysis the absolute lower bound for N, since we will be unable to find a solution to the mixed-correlator crossing equations below it.

The main goal of this chapter will be to lay out a method that aims to address the first concern. To follow the $O(N)$ islands through the (d, N) plane, we will invoke the newly-developed navigator method of [44] (see also [81, 83, 99] for recent works using the navigator method). If x denotes the set of parameters to scan over in a certain bootstrap setup, a navigator function $\mathcal{N}(x)$ is a function which is positive oustide of the allowed regions for x, and negative inside. One can thus flow to the allowed region(s) by minimizing the navigator function. It was shown in [44] that such a navigator can be constructed in general, and that in the example considered it correctly reproduced the 3D Ising Island. The minimum x_{min} was shown to give a good estimate of the location of the true CFT. We will show that this general construction also defines a valid navigator for the case of the $O(N)$ model. Here, the navigator will also depend on some external parameters: $\mathcal{N} := \mathcal{N}(x; p)$, where $p{=}(d, N)$. To follow the islands through d and N, we will trace out the curve $x_{min}(p)$. We will see that a simple formula relates derivative information at $p = p_0$ to the location of $x_{min}(p_0 + \delta p)$, and use this to help us sail through the $O(N)$ archipelago [84].

The chapter goes as follows: the $O(N)$ model and its ϵ-expansion are introduced in Section 2.2, with the description of the procedure used to resum the ϵ-expansion postponed to Appendix B.1. Section 2.3 then describes in generality the setup that will be used to bootstrap this model, and gives a first example computation. Many details about the numerical implementation are left to Appendix B.2. We will present in Section 2.4 a simple way to move through the (d, N) plane. The $O(N)$ model is then studied for fractional values of d and N respectively in Section 2.5 and Section 2.6, with the limit $N \to 1$ being considered in Subsection 2.6.1. We conclude in Section 2.7. Finally, Appendix B.3 presents more calculations which confirm the usefulness of the navigator method supplemented with the pathfollowing prescription of Section 2.4, and Appendix B.4 gives many details on the nature of the navigator computation at $N = 1$. Those less interested in all the details of the navigator method may skip Appendices B.2 to B.4.

2.2 Theory

We will aim to study the $O(N)$ model for $3 \leqslant d \leqslant 4$, starting from the perturbative regime $\epsilon = 4 - d \ll 1$. In this regime, the $O(N)$ model may be viewed as the weakly-coupled fixed point of the lagrangian

$$\mathcal{L}(\phi) = (\partial\phi)^2 + m\phi^2 + \lambda\left(\phi^2\right)^2 \tag{2.1}$$

for the N-component field ϕ_i. The critical exponents associated to this fixed point may be given as series-expansions in the expansion parameter ϵ. These can in turn be related to the scaling dimensions of certain operators in the corresponding CFT. Some operators of interest to us will be ϕ, the two lowest-dimensional $O(N)$ singlets $s = \phi^2$ and $s' = (\phi^2)^2$, and the two lowest-dimensional two-index symmetric-traceless tensors $t = \phi_i\phi_j - \frac{1}{N}\delta_{ij}\phi^2$ and $t' = \phi^2(\phi_i\phi_j - \frac{1}{N}\delta_{ij}\phi^2)$. To the lowest non-trivial order, their dimensions are [46, 98, 100]

$$\begin{aligned}
\Delta_\phi &= \frac{d-2}{2} + \frac{\eta(\epsilon)}{2} = 1 - \frac{\epsilon}{2} + \frac{N+2}{4(N+8)^2}\epsilon^2 + \mathcal{O}(\epsilon^3) \\
\Delta_s &= d - \nu^{-1}(\epsilon) = 2 - \frac{6}{N+8}\epsilon + \mathcal{O}(\epsilon^2) \\
\Delta_t &= d - d_f(\epsilon) = 2 - \frac{N+6}{N+8}\epsilon + \mathcal{O}(\epsilon^2) \\
\Delta_{s'} &= d + \omega(\epsilon) = 4 + \mathcal{O}(\epsilon^2) \\
\Delta_{t'} &= d - y_{4,2}(\epsilon) = 4 - \frac{N}{N+8}\epsilon + \mathcal{O}(\epsilon^2) \quad .
\end{aligned} \tag{2.2}$$

6-loop expansions for the critical exponents $\eta(\epsilon)$, $\nu^{-1}(\epsilon)$ and the correction-to-scaling exponent $\omega(\epsilon)$ can be found in [46] ($\eta(\epsilon)$ is actually known to 8-loops [101], and $\nu^{-1}(\epsilon)$ to 7 [102]). The fractal dimension $d_f(\epsilon)$ is also known to 6-loops [98], while the RG dimension $y_{4,2}(\epsilon)$ is known to only 5-loops [100]. A collection of all known ϵ-expansion CFT data was recently given in [103], and we will refer to it for data not listed in Eq. (2.2).

We will want to compare bootstrap results to the epsilon expansions presented above. The epsilon expansion for critical exponents is well-known to give divergent series. A resummation procedure is necessary in order to give meaningful results at finite ϵ. We will resum these series using the algorithm of Borel-Leroy transform with conformal mapping laid out in Section V of [46]. A summary of this resummation procedure is given in Appendix B.1; for a more thorough account, see

the original full description in [46]. The algorithm is quite elaborate, and involves many parameters. We use the same values for these parameters as those cited in [46]: we verified that we always reproduce the results of [46] for the quantities the authors had computed (i.e. critical exponents at integer values of N).

Something, although much less than for scaling dimensions, is also known about OPE coefficients in the ϵ-expansion. A quantity we will need in the future is the ratio of OPE coefficients $\theta(\epsilon) = \arctan \frac{\lambda_{sss}(\epsilon)}{\lambda_{\phi\phi s}(\epsilon)}$, which is known to $\mathcal{O}(\epsilon)$ [103–105] [1]:

$$\theta(\epsilon) = \arctan 2 - \frac{2(N+2)}{5(N+8)}\epsilon + \mathcal{O}(\epsilon^2) \quad . \tag{2.3}$$

2.3 Setup and first example

We will be considering in this chapter the crossing equation arising from the 4-point functions $\langle\phi\phi\phi\phi\rangle$, $\langle\phi\phi ss\rangle$ and $\langle ssss\rangle$. This equation was first derived in [84]; it reads

$$\sum_{\mathcal{O}\in S}\begin{pmatrix}\lambda_{\phi\phi\mathcal{O}} & \lambda_{ss\mathcal{O}}\end{pmatrix}\vec{V}_{S,\Delta,\ell}\begin{pmatrix}\lambda_{\phi\phi\mathcal{O}}\\ \lambda_{ss\mathcal{O}}\end{pmatrix} + \sum_{R\in\{A,T\}}\sum_{\mathcal{O}\in R}\lambda^2_{\phi\phi\mathcal{O}}\vec{V}_{R,\Delta,\ell} + \sum_{\mathcal{O}\in V}\lambda^2_{\phi s\mathcal{O}}\vec{V}_{V,\Delta,\ell} = 0 \quad , \tag{2.4}$$

with the various crossing vectors $\vec{V}_R$ given in Appendix B.2. We will make some assumptions on the spectrum of operators present in the various exchanged representations R, and look for solutions to crossing. We will do so with the help of the navigator function. [44] describes a simple recipe to define a *GFF navigator function* $\mathcal{N}^{\text{GFF}}(x)$ (we will omit the superscript GFF in the rest of the chapter) which is positive when our assumptions, encoded in a vector x, are disallowed by crossing and negative when x is allowed. The GFF navigator function was constructed for the bootstrap of the Ising model in [44], but it easily generalizes to the $O(N)$ case. See Appendix B.2 for its construction.

For the majority of this chapter, we will take the parameters on which the navigator depends to be $x = (\Delta_\phi, \Delta_s, \Delta_t)$. We will assume the existence of a spin-1 conserved current J_μ in the 2-index antisymmetric representation with dimension $\Delta_{J^\mu} = d-1$, and the existence of a spin-2, dimension d singlet operator corresponding to the stress-energy tensor $T_{\mu\nu}$. We will assume some gaps above ϕ, s, t, J_μ and $T_{\mu\nu}$, and impose that the rest of the spectrum respects the so-called

[1] We thank Johan Henriksson for pointing us to the result for $\lambda_{sss}(\epsilon)$.

unitarity bounds. Finally, we will impose a twist gap $\tau = 10^{-10}$ above the unitarity bounds to try to forbid solutions where spurious operators would sit exactly at the unitarity bounds. This setup is summarized in Appendix B.2. It is well known by now that a bootstrap such as the computation of the navigator function can be recast in the form of a semidefinite program (SDP), which we solve with `SDPB 2.4.0` [42, 43]. The numerical parameter which governs the size of allowed regions is the *derivative order* Λ: see Appendix B.2 for its definition. Unless otherwise stated, we will use $\Lambda = 19$.

We expect that the assumptions laid out above with judicious choices of gaps $\vec{\Delta}^* = (\Delta^*_\phi, \Delta^*_s, \Delta^*_t, \Delta^*_{J_\mu}, \Delta^*_{T_{\mu\nu}})$ should lead to small isolated islands of negative navigator in the three-dimensional parameter space where x lives for every value of (d, N) we will consider. Starting from an initial guess x_0, we will sail to these isolated islands by minimizing the navigator function using the quasi-Newton BFGS algorithm laid out in Section 5.2 of [44]. As discussed in [44], the GFF navigator function is generically concave far enough away from allowed regions, and asymptotes to a constant $\mathcal{N}_{max}$. In our numerical implementation, we will, just as in [44], decide to work with the transformed navigator

$$f(x) = \frac{\mathcal{N}(x)}{\mathcal{N}_{max} - \mathcal{N}(x)} \quad . \tag{2.5}$$

We had found in [44] that this improves the efficacy of the navigator method far away from the allowed regions. As defined here, $f(x)$ is positive if and only if the actual navigator $\mathcal{N}(x)$ is positive. Many more details about the BFGS algorithm may be found in Appendix B.2 for the interested reader.

In the following example, we will use the navigator to sail into the $d = 3$ $O(2)$ island. We will use as our starting point $x_0 = (0.519, 1.5051, 1.2358)$, which corresponds to the resummed values of the six-loop expansions of the dimensions expanded to lower order in Eq. (2.2). We will set $\vec{\Delta}^* = (3, 3, 3, 2.5, 3.5)$. The first three gaps are those used in [84], and amount to stating that ϕ, s and t are respectively the only relevant vector, singlet and traceless-symmetric scalars in the theory. We use a very conservative gap of $\frac{1}{2}$ above the conserved current and stress-energy tensor. We present in Fig. 2.1 the results of this example BFGS run. It took 34 total function calls and 24 BFGS iterations (which are function calls accepted in the line searches done by BFGS) to reach the minimum of the transformed navigator. The minimum is reached at $x_{min} = (0.518899, 1.50739, 1.23446)$. Because our assumptions were very close to those of [84], it is no surprise that this

allowed value is consistent with the $3D$ allowed region of Fig. 4 of [84].

We may also know the size of the island by computing the maximal and minimal allowed values for each coordinate in x with the "Constrained BFGS" algorithm of Section 6 of [44]. This algorithm attempts to extremize x along some given direction by performing a sequence of line searches whose directions are informed by an approximate quadratic model of the navigator $\mathcal{N}(x)$ which is updated after every line search. The bounds resulting from maximizing and minimizing all three variables in x are

$$\begin{aligned} x_{\text{allowed}} \in [0.518344\ldots, 0.520557\ldots] \times [1.49996\ldots, 1.51978\ldots] \times \\ [1.23091\ldots, 1.24189\ldots] \end{aligned} \tag{2.6}$$

Comparison to Figure 4 of [84] (for which the assumptions, bar the existence of a stress-tensor and conserved current, were the same as those used here) suggests that the region of negative $O(N)$ navigator does reproduce the actual allowed region, as was the case for the Ising model [44, 81] (see also [99] for another application of the navigator method, this time to the $\mathcal{N} = 1$ super-Ising model). It is interesting to note, by comparing to Figure 3 of [84], that the addition of the assumptions of a single relevant traceless-symmetric scalar, of a stress-tensor and of a conserved current seemed only to carve out a small portion at the upper-right of the island in the (Δ_ϕ, Δ_s) plane (such a behaviour was already discussed in [30] for only the addition of the stress-tensor).

We have therefore showed in this section that the GFF navigator construction may be applied successfully, as expected, to the case of the $O(N)$ model, showing agreement between allowed regions obtained with the navigator method and allowed regions obtained with the usual "binary" allowed/disallowed bootstrap. With the help of a small trick described in the following section, we will use this construction in Sections 2.5 and 2.6 to sail through the (d, N) plane.

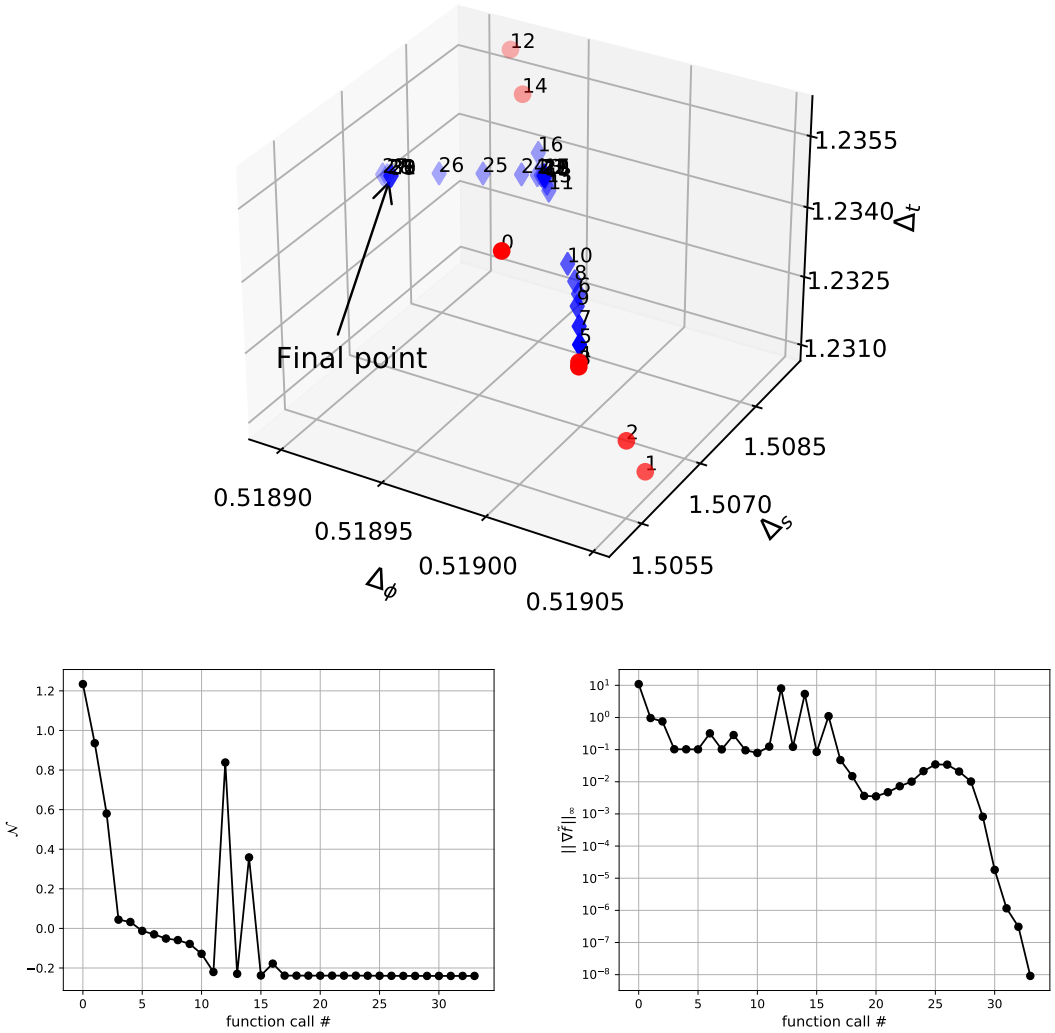


Figure 2.1: Top: BFGS run for the example described in Section 2.3 at derivative order $\Lambda = 19$. Points are numbered, starting from 0, by their function call number (and not the BFGS step number). Allowed points are marked as blue diamonds, and disallowed as red dots. Bottom left: Value of the navigator function at every function call. We can see it takes 5 function calls to reach the island. Bottom right: Maximum norm of the gradient, in the coordinates of Eq. (B.12), of the transformed function Eq. (2.5). BFGS terminates once the norm goes below gtol $= 10^{-8}$.

2.4 Simple pathfollowing

We will, in the following sections, attempt to follow the $O(N)$ models through the space of our external parameters $p = (d, N)$. As argued in Section 2.3, we will do so by following the minimum $x_{min}(p)$ of the GFF navigator function. To move as fast as possible in the external parameter space, we need to establish a way to use the knowledge of a solution at a certain p to sail to a solution at a nearby parameter $p + \delta p$. Because each $x_{min}(p)$ is the solution to a minimization problem, the problem we are faced with is referred to in the optimization literature as "parameteric optimization", and what we are trying to do, as "pathfollowing" (see [106, 107] for pedagogical references on this subject). We will implement the following simple pathfollowing method. Say we know the minimum x_0 for some $p = p_0$. Then, under certain conditions on the function being minimized, x_{min} is given in some neighbourhood of x_0 as a function of p: $x_{min} = \tilde{x}\,(p)$ with $\tilde{x}(p_0) = x_0$, and the first order variation of the position of the minimum is given (in Cartesian coordinates) at x_0 by (see [106], Theorem 4.1)

$$\frac{\partial \tilde{x}_i}{\partial p^n}\,(p_0) = -\left(\mathrm{B}_f^{-1}\,(x_0; p_0)\right)_{ij} \frac{\partial^2 f}{\partial \tilde{x}^j \partial p^n}(x_0; p_0) \quad , \tag{2.7}$$

with f the function from Eq. (2.5) which we minimize, and the Hessian given by $(\mathrm{B}_f(x,p))_{ij} = \frac{\partial^2 f}{\partial \tilde{x}^i \partial \tilde{x}^j}(x;p)$. This of course suggests the following: somehow sail to the first desired minimum (e.g. we will see in Section 2.5 that this can be done quite efficiently in a perturbative regime starting from known field-theory results). At this minimum, the Hessian and the mixed second derivative may be computed either by finite differences or by using the quadratic variation formula of Appendix C of [44] (we have found that the approximate Hessian obtained at the end of the previous BFGS run is not precise enough, hence why we advise to compute it independently). Then take steps $\delta p_1\,, \delta p_2\,, \ldots$ in the external parameter space small enough so that the first order variation (2.7) repeatedly gives good estimates of the location of the minimum of the rescaled navigator at the new parameter values $p + \delta p_1\,, p + \delta p_1 + \delta p_2\,, \ldots$ If these steps are indeed taken small enough, we should hope that the BFGS runs for the second, third, etc. parameter values would be much shorter than the first. Because we desire to compare specific external parameter values to results obtained from other methods, we will decide to choose the step sizes by hand. If one wanted to trace out the minimum curve as efficiently as possible, the choice of step size could be optimized: see [108], Chap.

6.

Let us see how this works in practice. We will start from the solution $x_0 = x_{min}(d = 3, N = 2)$ obtained in the example of Section 2.3, and attempt to reach solutions in $d = 3$ at nearby values of N. Fig. 2.2 shows that the gradient of the position of the minimum is indeed reproduced by Eq. (2.7). Using the minima predicted by Eq. (2.7) leads to an appreciable speedup of subsequent BFGS runs. Starting the $N = 1.9$ run at the extrapolated minimum, using the Hessian at the $N = 2$ minimum as the initial guess for the Hessian for the $N = 1.9$ run, it only takes 8 function calls to reach the true minimum. For comparison, it took 45 function calls to reach the same minimum starting from the $N = 2$ minimum. This amounts to a more than fivefold speedup. As stated in the previous paragraph, there should always exist some "optimal" step size if one wants to draw the solution curve as efficiently as possible. We have not explored ways of choosing this step automatically, but we can comment that in the example considered here, $\delta N = 0.1$ is a reasonable value as smaller step sizes don't seem to lead to an increase in efficiency that offsets the need for more steps. Indeed, this can be observed in Fig. 2.3, where we present the amount of function calls taken to reach the minimum for the different steps taken in Fig. 2.2 (for clarity, these steps are $\delta N \in \{\pm 0.01, \pm 0.025, \pm 0.05, \pm 0.1\}$). Now armed with this pathfollowing prescription, we should be in a good position to use the navigator to sail through the (d, N) plane.

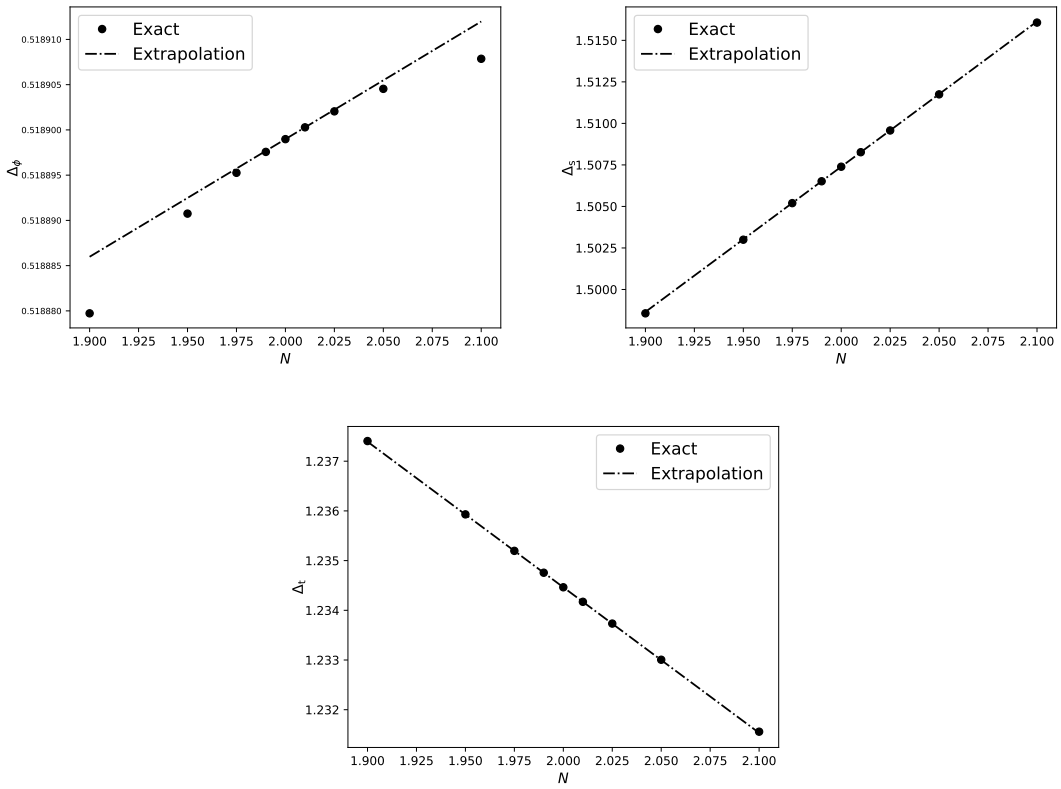


Figure 2.2: Comparison of the first-order prediction (2.7) for the location of the minima nearby the $(d = 3, N = 2)$ minimum to their actual value. Upper left: Comparison for Δ_ϕ. Upper right: Comparison for Δ_s. Lower middle: Comparison for Δ_t.

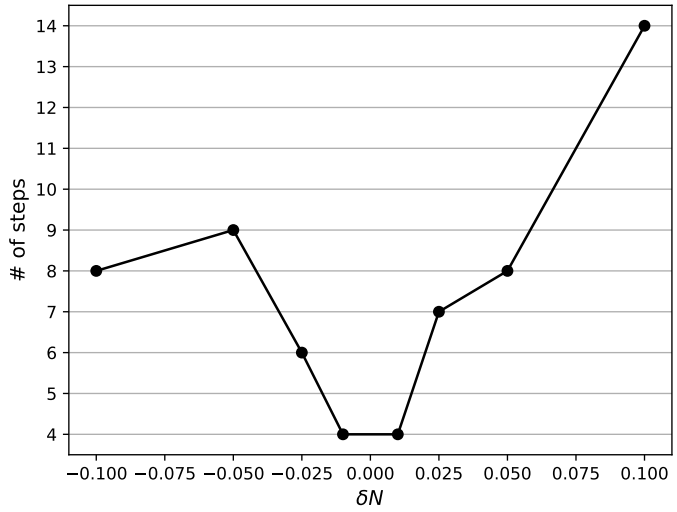


Figure 2.3: Number of function calls until BFGS terminates, for the step sizes considered in Fig. 2.2 with initial point given by Eq. (2.7).

2.5 Sailing through d: from the perturbative to the non-perturbative regime

We would now like to use the navigator construction laid out in Sections 2.3 and 2.4 to follow the $O(N)$ models above $d = 3$, and in particular to look for signs of non-unitarity in the fractional-d bootstrap solutions. Our goal will be to start from rough estimates in the perturbative regime $\epsilon = 4 - d \ll 1$, and iteratively decrease d, following the islands from $d = 4$ to $d = 3$ using the path-following prescription of Section 2.4 (here with the varying external parameter p being the dimension d). The case of $N = 1$ for $2 < d < 4$ was already considered in [91] (see also [109]). It was found that the family of kinks in the $\mathbb{Z}_2$-symmetric single-correlator $(\Delta_\sigma, \Delta_\epsilon)$-bound predicted by the $4 - \epsilon$ expansion survived all the way to $d = 2$, and that the scaling dimensions extracted from this family of kinks were in accord with those computed with the ϵ-expansion in [110]. This established that the effects of the non-unitarity of the Ising model in fractional dimensions [90] were small enough to be inconsequential to the numerical bootstrap (we will come back to this point in Section 2.6). In this section, we would like to elaborate on this result, and show that the numerical bootstrap appears insensitive, to our degree of precision, to the non-unitary nature of the fractional-d $O(N)$ models for $N = 2, 3$. Critical exponents for the fractional-d $O(N)$ models have previously been estimated using the functional RG [111] and field-theory [112–114]. Because most of the field-theory results are quite old, as stated in Section 2.2, we will when needed resum the ϵ-expansions of [46, 98] with the algorithm of [46] to use as a basis to compare bootstrap results. We use throughout this chapter as gap assumptions $\vec{\Delta}^* = (d, d, d, d - 0.5, d + 0.5)$, again looking for solutions with only one relevant scalar $O(N)$ vector, singlet and traceless-symmetric operator. We of course have to make sure that these assumptions are never violated, which we will do at the end of this section.

The outcome of the full path-following for $N = 2, 3$ is presented in Table 2.1. For greater clarity, we also plot these results for each of the 3 parameters in Fig. 2.4. We observe clear agreement between the bootstrap and ϵ-expansion results. The scaling dimensions corresponding to the minimum of the transformed navigator function follow the epsilon-expansion curves especially well, confirming the hypothesis of [44] that it provides a better estimate of the true scaling dimensions than a generic allowed point. Of course the error bars on the bootstrap results

should be taken with a grain of salt, since there is no systematic way to account for the non-unitarity of the fractional-d solution to crossing. We can only comment that, just like in the case of the Ising model in fractional dimensions [91], these non-unitary effects seem insignificant (at our level of precision), since the agreement with the ϵ-expansion stays excellent for all values of d.

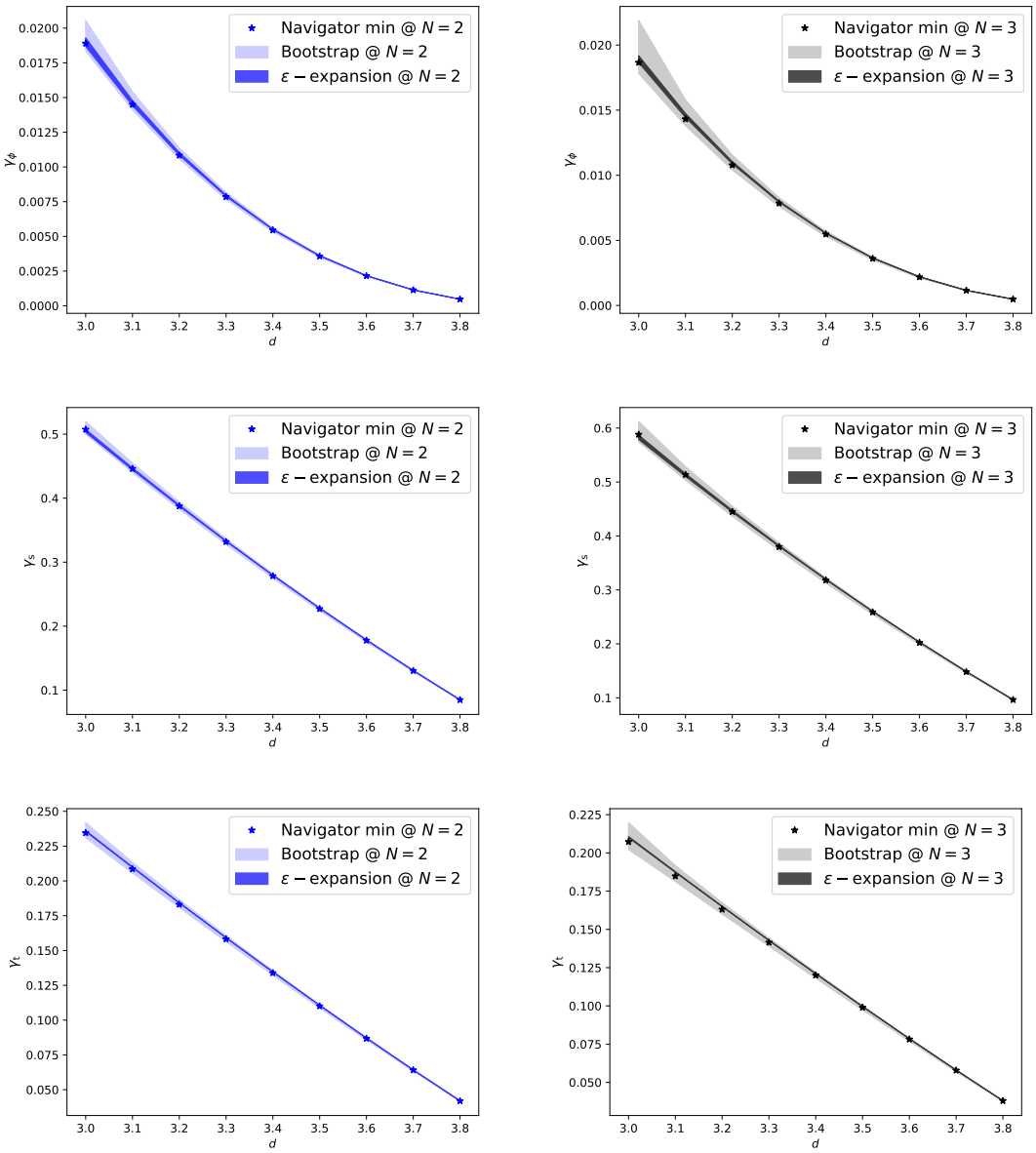


Figure 2.4: γ_ϕ, γ_s and γ_t as functions of d for $N = 2, 3$, as determined by both the conformal bootstrap with the help of the navigator function, and from the resummation of 6-loop ϵ-expansions.

	Bootstrap			Resummed $4-\epsilon$ expansion		
d	γ_ϕ	γ_s	γ_t	γ_ϕ	γ_s	γ_t
			$N=2$			
3.8	$0.000471\ {}^{+0.000003}_{-0.000010}$	$0.0848\ {}^{+0.0003}_{-0.0010}$	$0.04196\ {}^{+0.00013}_{-0.00047}$	$0.00047209\ \pm 0.00000007$	$0.084937\ \pm 0.000002$	$0.0420064\ \pm 0.0000005$
3.7	$0.001135\ {}^{+0.000015}_{-0.000028}$	$0.1304\ {}^{+0.0007}_{-0.0017}$	$0.0641\ {}^{+0.0003}_{-0.0008}$	$0.0011403\ \pm 0.0000007$	$0.13067\ \pm 0.00002$	$0.064231\ \pm 0.000003$
3.6	$0.00215\ {}^{+0.00004}_{-0.00005}$	$0.1777\ {}^{+0.0014}_{-0.0022}$	$0.0868\ {}^{+0.0007}_{-0.0011}$	$0.002165\ \pm 0.000003$	$0.17839\ \pm 0.00006$	$0.087120\ \pm 0.000009$
3.5	$0.00357\ {}^{+0.00009}_{-0.00008}$	$0.227\ {}^{+0.002}_{-0.003}$	$0.1101\ {}^{+0.0012}_{-0.0013}$	$0.003601\ \pm 0.000009$	$0.22802\ \pm 0.00014$	$0.11061\ \pm 0.00002$
3.4	$0.00545\ {}^{+0.00018}_{-0.00013}$	$0.278\ {}^{+0.003}_{-0.003}$	$0.1339\ {}^{+0.0017}_{-0.0016}$	$0.00550\ \pm 0.00002$	$0.2796\ \pm 0.0003$	$0.13465\ \pm 0.00005$
3.3	$0.00785\ {}^{+0.00031}_{-0.00018}$	$0.332\ {}^{+0.005}_{-0.004}$	$0.158\ {}^{+0.002}_{-0.002}$	$0.00793\ \pm 0.00005$	$0.3330\ \pm 0.0006$	$0.15921\ \pm 0.00009$
3.2	$0.0108\ {}^{+0.0005}_{-0.0003}$	$0.388\ {}^{+0.007}_{-0.005}$	$0.183\ {}^{+0.003}_{-0.002}$	$0.01094\ \pm 0.00010$	$0.3884\ \pm 0.0010$	$0.18427\ \pm 0.00014$
3.1	$0.0145\ {}^{+0.0009}_{-0.0004}$	$0.446\ {}^{+0.009}_{-0.006}$	$0.209\ {}^{+0.005}_{-0.003}$	$0.01460\ \pm 0.00018$	$0.4457\ \pm 0.0015$	$0.2098\ \pm 0.0002$
3	$0.0189\ {}^{+0.0017}_{-0.0006}$	$0.507\ {}^{+0.012}_{-0.007}$	$0.234\ {}^{+0.007}_{-0.004}$	$0.0190\ \pm 0.0003$ [46]	$0.505\ \pm 0.002$ [46]	$0.2358\ \pm 0.0003$ [98]
			$N=3$			
3.8	$0.000482\ {}^{+0.000003}_{-0.000013}$	$0.0964\ {}^{+0.0003}_{-0.0013}$	$0.03798\ {}^{+0.00012}_{-0.00051}$	$0.00048301\ \pm 0.00000006$	$0.096536\ \pm 0.000004$	$0.0380293\ \pm 0.0000004$
3.7	$0.001155\ {}^{+0.000017}_{-0.000034}$	$0.1482\ {}^{+0.0009}_{-0.0023}$	$0.0579\ {}^{+0.0003}_{-0.0009}$	$0.0011621\ \pm 0.0000006$	$0.14865\ \pm 0.00002$	$0.058046\ \pm 0.000002$
3.6	$0.00218\ {}^{+0.00005}_{-0.00006}$	$0.202\ {}^{+0.002}_{-0.003}$	$0.0782\ {}^{+0.0008}_{-0.0012}$	$0.002199\ \pm 0.000003$	$0.20318\ \pm 0.00009$	$0.078591\ \pm 0.000008$
3.5	$0.00360\ {}^{+0.00011}_{-0.00011}$	$0.259\ {}^{+0.003}_{-0.004}$	$0.0990\ {}^{+0.0013}_{-0.0016}$	$0.003644\ \pm 0.000008$	$0.2601\ \pm 0.0002$	$0.09960\ \pm 0.00002$
3.4	$0.00547\ {}^{+0.00023}_{-0.00017}$	$0.318\ {}^{+0.005}_{-0.005}$	$0.120\ {}^{+0.002}_{-0.002}$	$0.005549\ \pm 0.000019$	$0.3195\ \pm 0.0005$	$0.12103\ \pm 0.00005$
3.3	$0.0079\ {}^{+0.0004}_{-0.0003}$	$0.380\ {}^{+0.007}_{-0.007}$	$0.141\ {}^{+0.003}_{-0.002}$	$0.00797\ \pm 0.00004$	$0.3814\ \pm 0.0009$	$0.14284\ \pm 0.00009$
3.2	$0.0108\ {}^{+0.0008}_{-0.0004}$	$0.445\ {}^{+0.010}_{-0.008}$	$0.163\ {}^{+0.004}_{-0.003}$	$0.01096\ \pm 0.00008$	$0.4459\ \pm 0.0016$	$0.16500\ \pm 0.00015$
3.1	$0.0143\ {}^{+0.0015}_{-0.0005}$	$0.514\ {}^{+0.015}_{-0.010}$	$0.185\ {}^{+0.007}_{-0.004}$	$0.01459\ \pm 0.00015$	$0.513\ \pm 0.003$	$0.1875\ \pm 0.0002$
3	$0.0187\ {}^{+0.0032}_{-0.0008}$	$0.588\ {}^{+0.024}_{-0.014}$	$0.207\ {}^{+0.013}_{-0.005}$	$0.0189\ \pm 0.0003$ [46]	$0.583\ \pm 0.004$ [46]	$0.2103\ \pm 0.0003$

Table 2.1: Comparison of bootstrap and ϵ-expansion results for the anomalous dimensions of the first scalar $O(N)$ vector, scalar singlet and scalar traceless-symmetric 2-index tensor, for $N = 2, 3$ and $d \in [3, 4]$. Bootstrap results are given as the values at the transformed navigator minimum, with uncertainties given by the distance to the maximal and minimal values determined by the Constrained BFGS algorithm. Cited values are taken from Fig. 2 of [98] or deduced from the ancillary file "resummation.pdf" of [46].

A crucial step in following a bootstrap solution is to verify that there are no low-lying operators that dangerously approach the gaps we impose (if there were, modifications to our gap assumptions would have to be made). We show in Figs. 2.5 and 2.6 the dimensions of the subleading operators ϕ', t', J'_μ and $T'_{\mu\nu}$ as determined by the Extremal Functional Method (EFM) [80] at each navigator minimum [2]. Most of the operators stay well above the gap assumptions made, as expected from the ϵ-expansion (here we show the unresummed expansions given in the recent review article [103] [3]). Only t' near $d = 4$ has a dimension close to the gap assumed, but from the top right of Fig. 2.5 it is clear that the gap assumption is never violated.

Something strange happens in the spin-0 S sector. We observe that when the spectrum is extracted at the navigator minimum, the expected operator s' appears lower than predicted by the ϵ-expansion for most points, with s' and s'' then respectively just below and just above the prediction of the ϵ-expansion. We show this behaviour for the $N = 2$ case in Fig. 2.7. As presented in the bottom plot of Fig. 2.7, this behaviour happens when we are close enough to the GFF minimum for the test point $(d, N) = (3.4, 2)$, with allowed points further away from the GFF minimum being in much better agreement with the ϵ-expansion. It would be interesting to see if this effect disappears by increasing the derivative order Λ enough, and if this is not the case, to understand if and how this splitting is related to the GFF solution, but a high precision analysis of this question is beyond the scope of this chapter. Notwithstanding this strange behaviour in s', we've firmly established that the gap assumptions made in this section were always consistent with the actual spectrum of the $O(2)$ and $O(3)$ models.

[2] Throughout this chapter, unless otherwise stated, we maximize the stress-tensor OPE (after eliminating one of $(\lambda_{\phi\phi T_{\mu\nu}}, \lambda_{ssT_{\mu\nu}})$ using Ward identities) in order to extract the spectrum. Even though Ward identities were not imposed in our setup, the navigator minima were deep enough in the allowed region that they were still allowed after imposing Ward identities.

[3] $T'_{\mu\nu}$ is compared to the second subleading operator of [103] (in their notation, `Op[S,2,3]`); the EFM in this channel missed the true subleading operator `Op[S,2,2]` at both $N = 2$ and $N = 3$, most likely because it is close in dimension to the second subleading operator (both are of the form $\partial_\mu\partial_\nu\phi_S^4$ in the ϵ-expansion) and its OPE coefficients are small in comparison. From the ancillary file of [103], we have for example the ratios of their two OPE coefficients at $(d = 4, N = 2)$ given by $\frac{\lambda_{\phi\phi\texttt{Op[S,2,2]}}}{\lambda_{\phi\phi\texttt{Op[S,2,3]}}} \xrightarrow{d\to 4} 0.12$ and $\frac{\lambda_{ss\texttt{Op[S,2,2]}}}{\lambda_{ss\texttt{Op[S,2,3]}}} \approx 0.41$. We thank J. Henriksson for pointing this out.

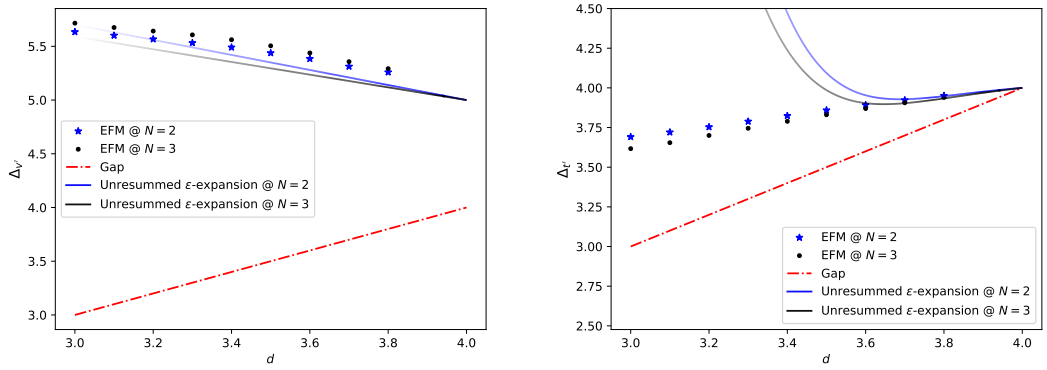


Figure 2.5: Subleading operators in 4 of the 5 channels where gap assumptions were made (continues in Fig. 2.6). Left: Spin-0 V channel. The larger discrepancy with the ϵ-expansion (the operators even appear in the opposite order as they do in the ϵ-expansion) is not surprising for operators with large twist in a setup with relatively low Λ. Right: Spin-0 T channel.

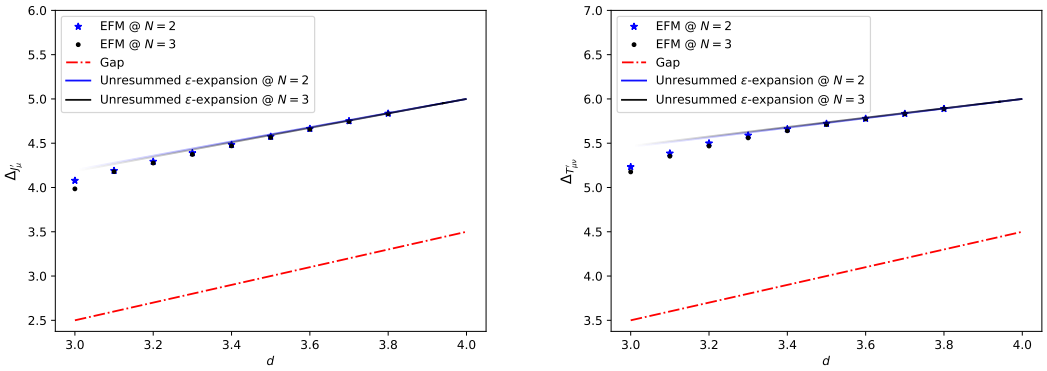


Figure 2.6: Subleading operators in 4 of the 5 channels where gap assumptions were made (continued from Fig. 2.5). Top: Spin-1 A channel. Bottom: Spin-2 S channel.

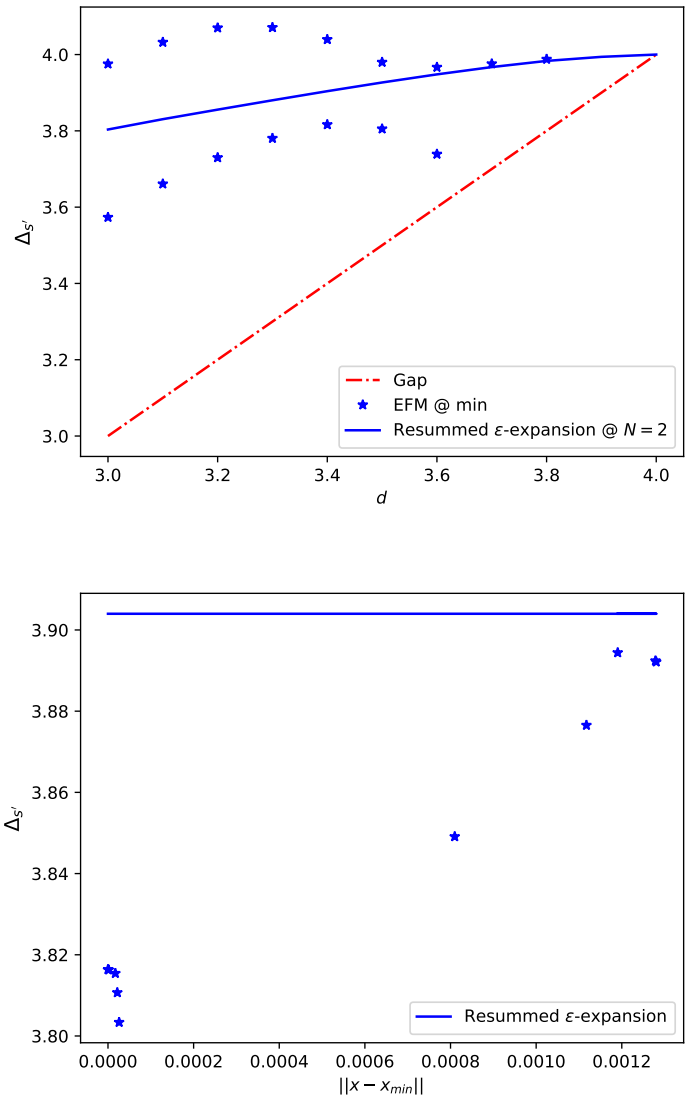


Figure 2.7: Top: Jump in subleading operator s' in the $\ell = 0$ S channel found by the EFM at the navigator minimum for $N = 2$. Bottom: Evolution of s' extracted from different points along the BFGS run for $(d = 3.4, N = 2)$, as functions of their distance to the transformed navigator minimum.

2.6 Sailing through N

We found in the previous section that non-unitarity due to fractional values of d did not influence the low-lying part of the spectrum of the $O(N)$ models determined by the unitary bootstrap. This was expected: [90] explains that the sources of the non-unitarity in fractional d are "evanescent" operators, present in fractional dimensions but which disappear in integer dimensions. Using as an example the Ising model in $4-\epsilon$ dimensions, [90] showed that a consequence of the violation of unitarity was the presence of operators with complex scaling dimensions, which the unitary conformal bootstrap should be sensitive to. However, the authors showed in their example that complex scaling dimensions should be expected only very high in the spectrum, with the first operators with complex scaling dimensions appearing only at $\Delta \approx 23$. Resolving such high-lying operators would require a much higher-precision setup than the one we used here. The fractional-N $O(N)$ models are also expected to be non-unitary, with the operators at the source of the non-unitarity also having generically large scaling dimensions for large values of N [97]. Still, a previous single-correlator analysis was able to match the results of the unitary bootstrap with RG and Monte Carlo methods right down to the limit $N \to 0$ [115]. It is natural to wonder if we might see signs of non-unitarity for fractional values of N when extending to a mixed-correlator setup. Therefore, using again the setup (B.11) with gaps $\vec{\Delta}^* = (3, 3, 3, 2.5, 3.5)$, we repeat an analysis similar to Section 2.5, this time in $d = 3$ and exploring the range $N \in [1, 3]$. The results are presented in Table 2.2 and Fig. 2.8. We see that the bootstrap results are always consistent with the ϵ-expansion. The bootstrap results have larger error bars, as getting high precision results was never the goal of this chapter, and so the bootstrap setup used here is weaker than the state-of-the-art [4]. However, the agreement is just as good in the limit $N \to 1$, where the $O(N)$ islands decrease significantly in size. The location of the transformed navigator minimum, just like it was the case in Section 2.5, follows the ϵ-expansion quite closely for all three scaling dimensions. We are therefore confident in asserting that the bootstrap at this derivative order is insensitive to non-unitarity in this range of N. We will see shortly that contrary to the results above, non-unitarity will drastically influence the fate of the $O(N)$ islands below this range.

We would like to make one comment before proceeding: we have presented a

[4] The full set of correlation functions involving ϕ, s and t for the case $N = 2$ was already investigated in [29] in 2019, with Λ pushed to 43 compared to our $\Lambda = 19$.

way to substantially cut down on the cost of following a bootstrap solution through an external parameter space with the navigator by using Eq. (2.7). The simple trick of using Eq. (2.7) is just one example of how one may use the additional information (as compared to the binary information provided by the usual scanning-based methods) encoded in the continuous navigator function to speed-up conformal bootstrap calculations. To demonstrate the efficacy of this method and to obtain the results we were after in this chapter, it was enough for us to consider the crossing equation at a relatively low derivative order $\Lambda = 19$. But of course, one could have also been interested in getting to the navigator minimum at a much higher Λ if he/she wanted to obtain very precise estimates of a certain amount of CFT data, and he/she would have benefited from Eq. (2.7) just the same. Furthermore, in that case, using a trick presented in Section 5.4 of [44], he/she could decrease his/her computational time by minimizing the navigator at a relatively low Λ first, and then using the navigator minimum and final Hessian at this low Λ as the starting point for a higher Λ calculation. By iteratively going up in Λ like this, he/she would cut down on the number of more expensive function calls at the higher Λ's. For example, going to the $\Lambda = 27$ minimum at $(d, N) = (3, 2)$ using the $\Lambda = 19$ minimum and final Hessian as the initial input only required 13 function calls, confirming the efficiency of the trick. Of course, in using the navigator method as a substitute for the usual scanning-based methods, he/she would have to deal with the fact that each navigator evaluation is more expensive than running `SDPB` on feasibility mode [5], with the very large-Λ navigator evaluations possibly becoming quite expensive.

2.6.1 $N \to 1$ limit and sinking of the $O(N)$ islands

Table 2.2 indicates that the mixed-correlator bootstrap correctly captures the limit $N \to 1$, and enables one to determine the dimensions of operators like Δ_t which would be invisible in the bootstrap of the Ising Model. What happens at exactly $N = 1$ is worth further consideration. As will be further discussed in Appendix B.4, the navigator function has many flat plateaux in Δ_t at $N = 1$. Nevertheless, BFGS

[5] For example, for one point tested in the $\Lambda = 19$ $O(N = 1)$ island, running `SDPB` on feasibility mode with the option `detectPrimalFeasibleJump` was about 3.8 times faster than computing the navigator with `dualityGapThreshold` $= 10^{-25}$ at that point for an otherwise identical setup, with the navigator computation taking about 20 minutes on 2 32-core nodes of the Caltech HPC cluster.

	Bootstrap			Resummed $4-\epsilon$ expansion		
N	Δ_ϕ	Δ_s	Δ_t	Δ_ϕ	Δ_s	Δ_t
3	$0.5187\,^{+0.0032}_{-0.0008}$	$1.588\,^{+0.024}_{-0.014}$	$1.207\,^{+0.013}_{-0.005}$	0.5189 ± 0.0003 [46]	1.583 ± 0.004 [46]	1.2103 ± 0.0003
2.9	$0.5187\,^{+0.0030}_{-0.0008}$	$1.581\,^{+0.023}_{-0.013}$	$1.210\,^{+0.012}_{-0.005}$	0.5189 ± 0.0003	1.576 ± 0.004	1.2127 ± 0.0003
2.8	$0.5188\,^{+0.0029}_{-0.0008}$	$1.573\,^{+0.021}_{-0.013}$	$1.212\,^{+0.011}_{-0.005}$	0.5190 ± 0.0003	1.569 ± 0.004	1.2151 ± 0.0002
2.7	$0.5188\,^{+0.0027}_{-0.0008}$	$1.565\,^{+0.020}_{-0.012}$	$1.215\,^{+0.011}_{-0.005}$	0.5190 ± 0.0003	1.561 ± 0.004	1.2176 ± 0.0003
2.6	$0.5188\,^{+0.0025}_{-0.0007}$	$1.557\,^{+0.019}_{-0.011}$	$1.218\,^{+0.010}_{-0.005}$	0.5191 ± 0.0003	1.553 ± 0.003	1.2201 ± 0.0003
2.5	$0.5189\,^{+0.0024}_{-0.0007}$	$1.549\,^{+0.018}_{-0.011}$	$1.220\,^{+0.010}_{-0.004}$	0.5191 ± 0.0003	1.546 ± 0.003	1.2226 ± 0.0003
2.4	$0.5189\,^{+0.0022}_{-0.0007}$	$1.541\,^{+0.017}_{-0.010}$	$1.223\,^{+0.009}_{-0.004}$	0.5191 ± 0.0003	1.538 ± 0.003	1.2252 ± 0.0003
2.3	$0.5189\,^{+0.0021}_{-0.0006}$	$1.533\,^{+0.016}_{-0.009}$	$1.226\,^{+0.009}_{-0.004}$	0.5191 ± 0.0003	1.530 ± 0.003	1.2278 ± 0.0003
2.2	$0.5189\,^{+0.0019}_{-0.0006}$	$1.525\,^{+0.015}_{-0.009}$	$1.229\,^{+0.008}_{-0.004}$	0.5190 ± 0.0003	1.522 ± 0.003	1.2304 ± 0.0004
2.1	$0.5189\,^{+0.0018}_{-0.0006}$	$1.516\,^{+0.013}_{-0.008}$	$1.232\,^{+0.008}_{-0.004}$	0.5190 ± 0.0003	1.513 ± 0.003	1.2331 ± 0.0003
2	$0.5189\,^{+0.0017}_{-0.0006}$	$1.507\,^{+0.012}_{-0.007}$	$1.234\,^{+0.007}_{-0.004}$	0.5190 ± 0.0003 [46]	1.505 ± 0.002 [46]	1.2358 ± 0.0003 [98]
1.9	$0.5189\,^{+0.0015}_{-0.0005}$	$1.499\,^{+0.011}_{-0.007}$	$1.237\,^{+0.007}_{-0.003}$	0.5190 ± 0.0003	1.4965 ± 0.0024	1.2385 ± 0.0004
1.8	$0.5188\,^{+0.0014}_{-0.0005}$	$1.490\,^{+0.010}_{-0.006}$	$1.240\,^{+0.006}_{-0.003}$	0.5189 ± 0.0003	1.487 ± 0.002	1.2413 ± 0.0004
1.7	$0.5188\,^{+0.0013}_{-0.0005}$	$1.480\,^{+0.009}_{-0.006}$	$1.243\,^{+0.006}_{-0.003}$	0.5188 ± 0.0003	1.479 ± 0.002	1.2441 ± 0.0005
1.6	$0.5188\,^{+0.0011}_{-0.0004}$	$1.471\,^{+0.008}_{-0.005}$	$1.246\,^{+0.005}_{-0.003}$	0.5188 ± 0.0003	1.470 ± 0.002	1.2469 ± 0.0003
1.5	$0.5187\,^{+0.0010}_{-0.0004}$	$1.462\,^{+0.007}_{-0.005}$	$1.250\,^{+0.005}_{-0.003}$	0.5187 ± 0.0003	1.460 ± 0.002	1.2499 ± 0.0003
1.4	$0.5186\,^{+0.0009}_{-0.0003}$	$1.452\,^{+0.006}_{-0.004}$	$1.253\,^{+0.004}_{-0.003}$	0.5186 ± 0.0003	1.451 ± 0.002	1.2527 ± 0.0006
1.3	$0.5185\,^{+0.0007}_{-0.0003}$	$1.443\,^{+0.005}_{-0.004}$	$1.256\,^{+0.004}_{-0.002}$	0.5185 ± 0.0003	1.4408 ± 0.0015	1.2557 ± 0.0006
1.2	$0.5184\,^{+0.0006}_{-0.0003}$	$1.433\,^{+0.004}_{-0.003}$	$1.259\,^{+0.003}_{-0.002}$	0.5184 ± 0.0003	1.4311 ± 0.0014	1.2587 ± 0.0006
1.1	$0.5183\,^{+0.0004}_{-0.0002}$	$1.423\,^{+0.003}_{-0.003}$	$1.263\,^{+0.002}_{-0.002}$	0.5182 ± 0.0003	1.4209 ± 0.0012	1.2617 ± 0.0005
1	0.518180^*	1.41296^*	1.26595^*	0.5181 ± 0.0003 [46]	1.4108 ± 0.0011 [46]	1.2648 ± 0.0006 [98]

Table 2.2: Same as Table 2.1, for the scaling dimensions instead of the anomalous dimensions, for $d = 3$ and $N \in [1, 3]$. Bootstrap results are given as the values at the transformed navigator minimum, with uncertainties given by the distance to the maximal and minimal values determined by the Constrained BFGS algorithm (the values at the navigator minimum are given with asterisks in cases where uncertainties were not calculated). Cited values are taken from Fig. 2 of [98] or deduced from the ancillary file "resummation.pdf" of [46].

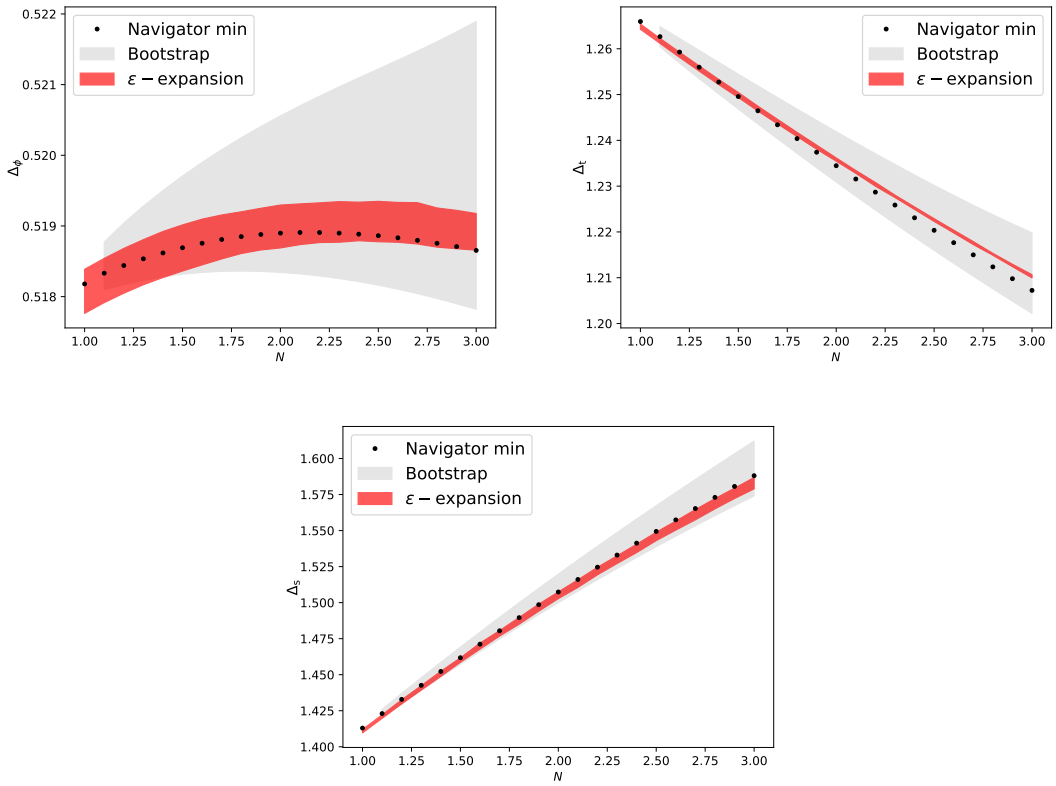


Figure 2.8: Δ_ϕ, Δ_s and Δ_t as functions of N, as determined by both the conformal bootstrap with the help of the navigator function, and from the resummation of 6-loop ϵ-expansions.

does in the end manage to converge to the following point:

$$x^{\star}_{\text{min}} = (0.518180\ldots, 1.41296\ldots, 1.26595\ldots) \quad . \tag{2.8}$$

We see that the location in Δ_t of the minimum found by BFGS matches quite well with $\Delta_t = 1.2648(6)$ predicted by the epsilon expansion [98].

There is one thing in the navigator computation at $N = 1$ that is worrying: the actual minimal value of the transformed navigator $f(x)$. We plot in Fig. 2.9 the value of $f(x)$ at its minimum near $N = 1$. Remember that $f(x)$ is positive iff the actual navigator function $\mathcal{N}(x)$ is positive. At $N = 1.5$, $f(x_{\text{min}}) = -0.072910$, and $f(x)$ increases as $N \to 1^+$, reaching -0.000064 at $N = 1$. Thus the "most allowed" point according to the navigator construction gets dangerously close to becoming disallowed as $N \to 1^+$. This offers a clear sign that the $O(N)$ islands may disappear somewhere just below $N = 1$. Although we did not precisely determine the location of the disappearance, we can say that the islands seem to disappear for good when going well below $N = 1$. Indeed, for two values we tested well below $N = 1$, BFGS was not able to find an allowed point. For $N = 0.9$ and $N = 0.5$, BFGS found minima that both lie above $\mathcal{N}(x) = 0$ (see Fig. 2.9). We hypothesize that the islands do disappear at some critical N_c close to $N = 1$, and that $N_c \xrightarrow{\Lambda\to\infty} 1$. We will see in the rest of this section that the $O(N)$ model must actually become very non-unitary below $N = 1$, which explains the observed disappearance of the islands.

So why is it that the islands seem to disappear roughly at $N = 1$? Some non-unitarity was presumably already present in the high-lying part of the spectrum of the $O(N)$ model at fractional values of N above $N = 1$ [97], but the bootstrap showed to be insensitive to this. Say we determine a unitary spectrum inside of the $\Lambda = 19$ $O(N)$ islands with the Extremal Functional Method as in the end of Section 2.5, and follow this spectrum all the way down to $N = 1$. Why could we not just perturb the unitary spectrum we find at $N = 1$ to obtain a unitary one that solves the crossing equations just below $N = 1$? We present in Figs. 2.10 and 2.11 the fate of two operators which illustrate exactly why. Fig. 2.10 shows the $\phi \times s$ OPE coefficient of the lowest-lying $\ell = 1$ V primary v_μ in the limit $N \to 1^+$. This OPE coefficient clearly goes to zero as N tends to 1. Furthermore, from a fit of all values below $N = 1.2$, we find that the OPE coefficient goes to zero roughly as a square root: $\lambda_{\phi s v_\mu} \sim (N-1)^{0.482834}$. We then show in Fig. 2.11 the fate of the subleading $\ell = 2$ V primary $v'_{\mu\nu}$. We see in the right part of

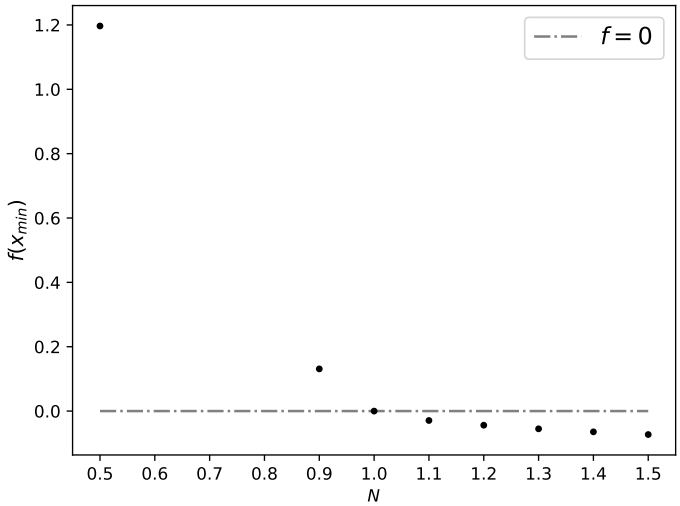


Figure 2.9: Value of the transformed navigator (2.5) at its minimum, for N between 0.5 and 1.5.

Fig. 2.11 that the lowest-lying $\ell = 2$ V primary $v_{\mu\nu}$ continues at $N = 1$ into the lowest-lying $\ell = 2$ $\mathbb{Z}_2$-odd primary $\sigma_{\mu\nu}$ of the Ising model, while the third lowest-lying $\ell = 2$ V primary $v''_{\mu\nu}$ continues approximately into $\sigma'_{\mu\nu}$ (the offset is due here to the lower derivative order used in comparison to [82]). The dimension of $v'_{\mu\nu}$ tends as N goes to 1 to a dimension that does not correspond to any primary in the Ising spectrum. As expected, when we extract the spectrum at an allowed point in the $O(N = 1)$ island, we find that this primary has disappeared[6]. We show in the left part of Fig. 2.11 that this disappearance can be seen in the $\phi \times s$ OPE coefficient of $v'_{\mu\nu}$, which tends to zero as $N \to 1^+$, again roughly as a square root (a fit of all values below $N = 1.2$ gives $\lambda_{\phi s v'_{\mu\nu}} \sim (N-1)^{0.452306}$, although the fit is much less stable than that for v_μ). We have therefore presented above two V sector primaries, with relatively small scaling dimensions, whose $\phi \times s$

[6]We have encountered some problems in extracting the full spectrum exactly at $N = 1$. If we do so by extremizing some quantity in the channels present in the Ising crossing equation, `SDPB` looks for functionals $\vec{\alpha}$ such that $\vec{\alpha} \cdot \vec{V}_{T,\Delta,\ell} = \vec{\alpha} \cdot \vec{V}_{A,\Delta,\ell} = 0$ identically. Because of this, such an extremization cannot give any information on the A and T channels. When we instead try to extremize a quantity in one of these channels, e.g. by maximizing $\lambda_{\phi\phi t}$, the solution of the *primal* problem of `SDPB` (see [42] for its definition) has a large discontinuous jump when going from $N > 1$ to $N = 1$. Using this primal solution to determine OPE coefficients [82] results in none of the OPE coefficients we gather at $N = 1$ in the A and T sectors being sensible. We do not have a definitive explanation for this behaviour.

OPE coefficients behave approximately as square roots near $N = 1$. A naive continuation to the range $N < 1$ would require their squared OPE coefficients to become negative, violating unitarity. Thus, the presence of such primaries seemingly prevents us from obtaining a unitary solution to crossing at $\Lambda = 19$ below $N = 1$ by perturbing the unitary solution to crossing we get at $N = 1$. One may wonder if the problematic primaries are related to problematic primaries in the free $O(N)$ model, or in the $O(N)$ model in $4 - \epsilon$ dimensions, where a large amount of CFT data is known analytically [103]. We are going to see shortly that the answer to this question is yes.

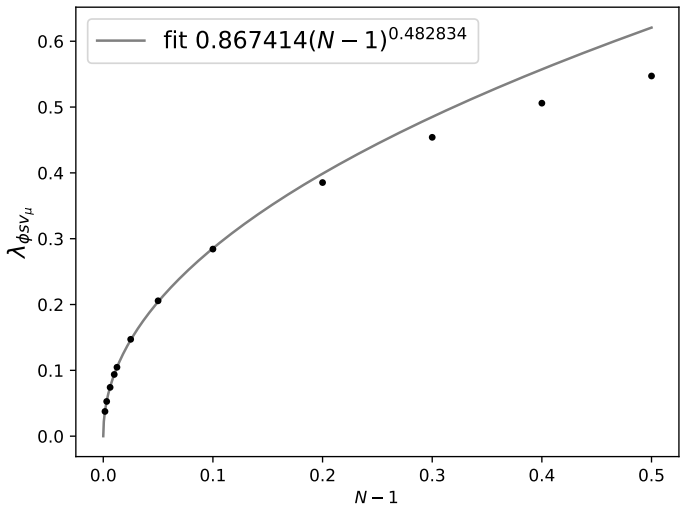


Figure 2.10: Plot of the $\phi \times s$ OPE coefficient of the lowest-lying spin-1 V primary, along with a fit of all values excluding those over $N = 1.1$.

Should we believe that the spectrum of the $O(N)$ model varies continuously from the free theory in $d = 4$ to the strongly-coupled theory in $d = 3$ (see [90] for a discussion), we might hope that the operators observed in $d = 3$ in Figs. 2.10 and 2.11 can be associated close to $d = 4$ to operators with similar behaviours in the limit $N \to 1$, and whose behaviours can be easily interpreted. From what will be presented below, it seems that this picture is valid. The lowest-lying $\ell = 1$ vector primary, which was the first operator discussed in the last paragraph, is given in the ϵ-expansion by an operator with three fields and one derivative $v_\mu = \partial_\mu \phi_V^3$ according to Table 21 of [103]. At $N = 1$, this operator should become $\phi^2 \partial_\mu \phi$, which is a descendant of ϕ^3. Therefore, the primary v_μ in $d = 3$ disappears in the

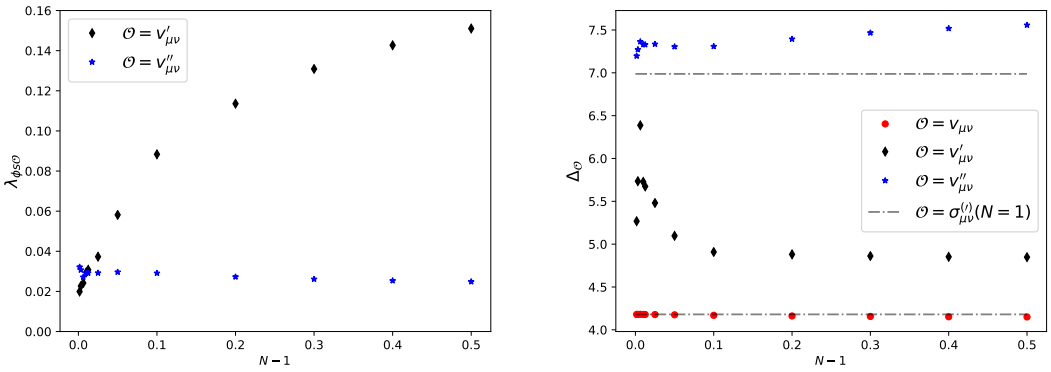


Figure 2.11: Left: $\phi \times s$ OPE coefficient of the two subleading spin-2 V primaries. Right: Scaling dimensions of the three lowest-lying spin-2 V primaries. The straight lines indicate the dimensions of the two lowest-lying spin-2 $\mathbb{Z}_2$-odd primaries in the Ising Model according to [82].

limit $N \to 1^+$ just like v_μ in $d = 4 - \epsilon$ should. Furthermore, the squared OPE coefficient $\lambda^2_{\phi\,\phi^2\,\partial_\mu\phi^3_V}$ is given to $\mathcal{O}(\epsilon)$ in the second entry of Table 4 of [116] as

$$\lambda^2_{\phi\,\phi^2\,\partial_\mu\phi^3_V} = \frac{2(N-1)}{3N}\left(1 - \frac{N+11}{3(N+8)}\epsilon\right) \quad . \tag{2.9}$$

Both its free and $\mathcal{O}(\epsilon)$ contributions come with a factor of $N-1$, in agreement with the square root behaviour observed in Fig. 2.10. Another example of a primary operator becoming a descendant at $N = 1$ is the scalar singlet $\phi^2\,(\partial\phi)\cdot(\partial\phi)$, which becomes a descendant of ϕ^4 at $N = 1$ in the free theory as derived in [117]. Unfortunately, we could not really observe the disappearance of this operator in $d = 3$ with the bootstrap, most likely because it sits too high in the spectrum to be observed at $\Lambda = 19$.

The limit $N \to 1$ also sees many sets of different primary operators with the same classical dimensions merging into a lower number of primaries at $N = 1$. This is for example the case of the two lowest-lying $\ell = 2$ V primaries $v_{\mu\nu}$ and $v'_{\mu\nu}$ (where $v'_{\mu\nu}$ was observed in Fig. 2.11 to disappear in $d = 3$). Table 21 of [103] gives these operators in the ϵ-expansion as two operators of the form $\partial_\mu\partial_\nu\phi^3_V$. They become one single primary operator $\sigma_{\mu\nu} = \partial_\mu\partial_\nu\phi^3$ at $N = 1$ (see Table 14 of [103]). Another example discussed in [103] in the ϵ-expansion is that of the two subleading $\ell = 2$ S primaries, which are of the form $\partial_\mu\partial_\nu\phi^4_S$. They become a single primary $\partial_\mu\partial_\nu\phi^4$ at $N = 1$. In this case, the $\phi \times \phi$ squared OPE coefficients for

both operators are known to $\mathcal{O}(\epsilon^2)$, and that of one of the two ([103] refer to it as $\mathcal{O}_{4,2,1}$) goes smoothly from positive to negative at $N = 1$. It is given in (3.15) of [103] as:

$$\begin{aligned}\lambda^2_{\phi\phi\mathcal{O}_{4,2,1}} &= \frac{N+2}{320N(N+8)^2}\frac{-76+N+3\sqrt{9N^2-8N+624}}{\sqrt{9N^2-8N+624}}\epsilon^2+\mathcal{O}(\epsilon^3)\\ &= \left(\frac{1}{135000}(N-1)+\ldots\right)\epsilon^2+\mathcal{O}(\epsilon^3) \quad .\end{aligned} \tag{2.10}$$

Many other primary operators disappearing at $N = 1$ because of this mechanism may be inferred from comparing Tables 8-14 to Tables 19-21 of [103]. In the cases we could find where data about OPE coefficients were known, disappearing primary operators of this sort had squared OPE coefficients that were analytic at $N = 1$, and that became negative below $N = 1$, making the $O(N)$ model in d close to 4 very non-unitary. Fig. 2.11 shows that the problematic operators in $d \approx 4$ continue into problematic operators in $d = 3$.

We have shown above two different mechanisms (primaries becoming descendants and merging of primaries) which result in severe non-unitarity in the free $O(N)$ model and the $O(N)$ model in $d = 4-\epsilon$ below $N = 1$, as many squared OPE coefficients involving low-lying primaries which were positive above $N = 1$ become negative below. We have seen in Figs. 2.10 and 2.11 that analogous behaviours could be observed in the spectra solving the $d = 3$ $O(N)$ crossing equations at $\Lambda = 19$ in the limit $N \to 1^+$. Therefore, to continue the solution to crossing below $N = 1$ would require many unitarity-violating contributions from low-lying operators, which explains the drastic difference between the results of the unitary bootstrap for $N > 1$ and $N < 1$. Finally, we would like to point out that the problematic operators we see in the limit $N \to 1$ do not seem to be related to those generically at the source of the non-unitarity of the fractional-N $O(N)$ models according to [97] (see the evanescent operator (7.82) of [97], first noticed in [118]).

2.7 Conclusion and outlook

We have shown in this chapter that the bootstrap of the $O(N)$ model was insensitive to the non-unitary nature of the model for both fractional $d > 3$ and fractional $N > 1$. In the process, we gave bootstrap and ϵ-expansion estimates of a substantial amount of CFT-data in the range $(d, N) \in [3, 4] \times [1, 3]$. We then studied in more detail the limit $N \to 1$, and obtained the clear disappearance of the $O(N)$

islands below $N = 1$ (see [119] for another case where sizeable non-unitarities lead to disagreement between bootstrap and field-theory results). We obtained these results using the newly developed navigator method, and devised a simple pathfollowing algorithm which enabled us to sail from island to island efficiently. In some cases this led to an appreciable speedup of subsequent optimization runs, and in others it was shown to be necessary in finding the next island in the first place.

The disappearance of the $O(N)$ islands at $N_c = 1$ could be observed with the navigator method as the minimum of the navigator $\mathcal{N}(x)$ went above 0 at roughly N_c. This constitutes a much clearer signature of the disappearance of the island than could be possible with the usual binary bootstrap, where one could still wonder if some small island could have evaded the scan. As was already discussed in [44], we believe that the navigator method could be helpful in studying other systems where the behavior of some family of CFTs is expected to change at some critical value(s) of the external parameters. This is the case for example for the $O(N)$ models near $(d, N) = (2, 2)$, where a critical line (d_c, N_c) (the "Cardy-Hamber" line) is hypothesized to emerge from $(d, N) = (2, 2)$ along which two fixed points collide and exchange stability [120, 121]. The bootstrap should be able to observe nonanalyticity of conformal data along this line. Should the line exist, it would be especially interesting to observe where it ends up for $N = 3$. Some other examples would include the merger and annihilation of the critical and tricritical q-state Potts model along another critical line (d_c, q_c) [79, 122, 123], and the controversial fate of the $O(N) \times O(2)$ universality class supposedly describing phase transitions in certain classes of frustrated magnets, where there should also exist a critical line (d_c, N_c) along which there is the merger and annihilation of the so-called "chiral" and "antichiral" fixed points [124] (see [85, 125] for previous bootstrap work). In cases like the $O(N)\times O(2)$ model, where the crossing equations involve lots of internal channels, we expect that the navigator will enable us to scan over many more internal exchanged operators than were considered in the past. Paired with the pathfollowing prescription laid out in this chapter which should help sail in the (d, N) plane more efficiently, this should lead to a better determination of the critical line (d_c, N_c). We will see how this story plays out in the following, and final chapter of this thesis.

Chapter 3

Bootstrapping frustrated noncollinear magnets*

* Work in progress with Marten Reehorst, Slava Rychkov and Balt van Rees. Presented at *Bootstrap 2023* at Instituto Principia, São Paulo, Brazil.

3.1 Introduction

The advent of the Renormalization Group (RG) has led to countless quantitatively accurate predictions for the behavior of systems at criticality. Although its scope is extremely expansive, it has often times proven limited, with many puzzles remaining in understanding even some of the simplest critical phenomena. One such puzzle arises in the description of the critical properties of frustrated non-collinear antiferromagnets. Although the nature of the controversy surrounding this class of magnets has been described in great detail in the literature many times [59, 85, 126, 127], it cannot hurt to quickly remind it.

It has been known for many decades that a multiscalar theory with $\mathrm{O}(N)\times\mathrm{O}(2)$ global symmetry describes the critical modes of N-component classical stacked triangular antiferromagnets and helimagnets [61, 128]. The Landau-Ginzburg-Wilson (LGW) Lagrangian for the critical modes is given by (1.2) of [58]:

$$\mathcal{L} = \frac{1}{2}(\partial_\mu \phi)^2 + \frac{1}{2} m^2 \phi^2 + u(\phi^2)^2 + v[(\vec{\phi}_1 \cdot \vec{\phi}_2)^2 - \vec{\phi}_1^{\,2}\vec{\phi}_2^{\,2}], \tag{3.1}$$

where $v > 0$, $\phi = \phi_{ai}$ ($a = 1, \ldots, N$, $i = 1, 2$) is in the bifundamental representation of $\mathrm{O}(N) \times \mathrm{O}(2)$, and we denote $\vec{\phi}_1 = \phi_{a1}$ and $\vec{\phi}_2 = \phi_{a2}$.[1] The question theory has to answer is simple: for the physical values $N = 2\,, 3$ and in three dimensions, does there exist a stable fixed point of the RG flow? If so, for materials within the basin of attraction of this fixed point, experiments should observe second order phase transitions. In the opposite case, phase transitions should be first order.

An ϵ-expansion around $d_{\mathrm{uc}} = 4$ can be performed. This expansion predicts the existence of a critical curve $N_c(4-\epsilon)$ above which a stable fixed point exists. It is the fixed point $\mathcal{C}^+$ in Fig. 3.1, also referred to in the literature as the *chiral* fixed point. This fixed point merges and annihilates with an unstable fixed point $\mathcal{C}^-$ at N_c. Below this merger, only two fixed points remain, both unstable: the Gaussian $\mathcal{G}$ and the $\mathrm{O}(2N)$ invariant "Heisenberg" fixed point $\mathcal{H}$, which has $v = 0$ in (3.1).[2] Perturbation theory enables one to know $N_c(4-\epsilon)$ as a divergent series expansion in ϵ, now known up to 6 loops [124]. Using various resummation techniques, the authors of [124] conclude that $N_c(3) = 5.96(19)$, well above the physical values:

[1] To see the symmetry, note $(\phi^2)^2 = (\phi_1^2 + \phi_2^2)^2$ and the second $\mathrm{O}(N) \times \mathrm{O}(2)$ invariant quartic combination $\phi_{ai}\phi_{aj}\phi_{bi}\phi_{bj} = (\phi_1^2)^2 + (\phi_2^2)^2 + 2(\phi_1 \cdot \phi_2)^2$. The v term is half the difference of the two.

[2] The ϵ-expansion predicts further changes to the phase diagram below $N = 2$; let us not worry about these here. See [58] for more detail.

the ϵ-expansion unequivocally predicts first-order phase transitions for $N = 2\,, 3$.

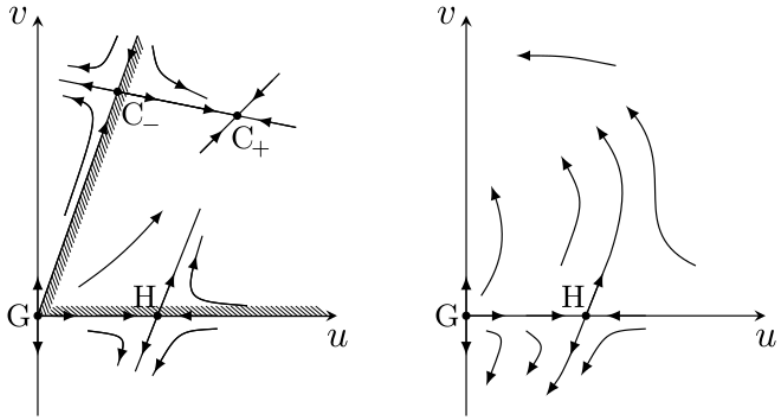


Figure 3.1: RG flow diagrams of the model (3.1) in $d = 4-\epsilon$ dimensions, for $N > N_c(d)$ (left) and $N \lesssim N_c(d)$ (right). Figure taken from [58].

The competing view is that of the perturbative RG performed directly in $d = 3$ [129–132]. For this method, β-functions are expanded in powers of the quartic couplings in the dimension of interest. They are then resummed before looking for a fixed point. This is in stark contrast to the ϵ-expansion, where fixed points and critical exponents are found in the perturbative regime as expansions in ϵ and resummation is performed only at the very last step, when extrapolating to the desired dimension. For the simpler O(N) models these two methods agree; here, they do not. The fixed-dimensional RG predicts that $N_c(d)$ is not strictly a function: although it agrees near $d = 4$ with the ϵ-expansion, it curves back on itself as d is lowered, so that for some range of d there exist *two* critical values $N_{c,1}(d) > N_{c,2}(d)$ outside of which a stable fixed point exists, and between which none does. See Fig. 3.2 for a sketch of this behavior. Furthermore, the fixed point found below $N_{c,2}(d)$ turns out to be of "focus" type, meaning that it has complex correction-to-scaling exponents. This contradicts the unitarity of the model, and basic principles of renormalization group flows such as the gradient property.[3]

Unfortunately, experimental data and Monte Carlo simulations do little to clear up the picture. Some Monte Carlo results claim to confirm the existence of

[3]The RG flow of multiscalar models is widely expected to be a gradient flow. This has been proven to three loops [133, 134]. Gradient flow property precludes focus-type fixed points. Refs. [129–132] resummed each component of the beta-function separately - a procedure which apparently did not preserve the gradient property.

focus-type fixed points [135], while others show clear signs of first order transitions [136, 137]. Experiments are plentiful, but the consensus is that there is such a large discrepancy of critical exponents in the cases where a second-order phase transition is believed to happen, that it is impossible to discern between true second-order and weak first-order behavior [59, 126]. We are left with an unfortunate situation where all traditional methods fail to give a coherent picture, and thus new perspectives from alternative methods are necessary. One such method is the so-called Functional, or Nonperturbative RG (NPRG). The results from the NPRG clearly confirm the picture of the ϵ-expansion [126, 138], producing a curve $N_c(d)$ in excellent agreement with the ϵ-expansion all the way down to three dimensions.

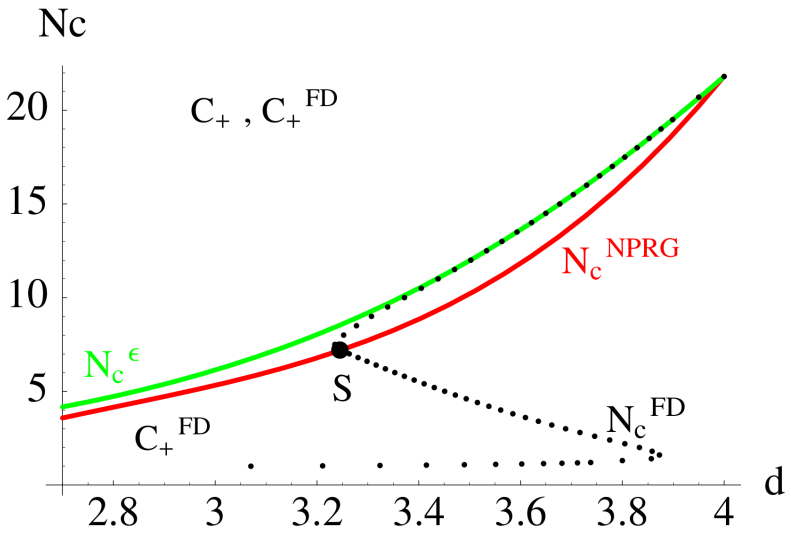


Figure 3.2: Comparison between NPRG, fixed-dimensional RG and the ϵ-expansion. Figure taken from [139].

In this chapter, we will approach the problem using another popular nonperturbative method - the numerical conformal bootstrap. Fixed points of the RG flow are generically expected to exhibit conformal invariance [21]. Thus, we can substitute the search of fixed points of the RG flow with the search for a Conformal Field Theory (CFT) with O(N) $\times$ O(2) global symmetry. The numerical bootstrap is a powerful method which attempts to pinpoint specific theories in the large landscape of all possible CFT data by imposing general conditions like unitarity, the Operator Product Expansion (OPE) and crossing-symmetry of correlation functions. A few bootstrap studies dedicated to this problem have been

performed in the past [85, 125], but it is our belief that these have suffered from the limited power of the numerical methods of their times, and have therefore been unable to fully clarify the picture. We will show in this chapter that we are able to make ample progress, in particular by harnessing the power of the newly developed navigator method [44].

The chapter is structured as follows. In Section 3.2 we describe our bootstrap setup, detailing in particular in Subsection 3.2.1 our choice of gap assumptions in important symmetry channels. We then move on to results in Section 3.3, and conclude in Section 3.4. Appendix C.1 is dedicated to details of the numerical methods used, and Appendix C.2 to the calculation of the scaling dimension at $O\left(1/N\right)$ of a certain crucial operator.

3.2 CFT setup

As stated in the introduction, we will in this chapter study the fixed points of (3.1) using the numerical conformal bootstrap in d dimensions. The stable RG fixed point $\mathcal{C}^+$ of (3.1) is a CFT in d dimensions having global $\mathrm{O}(N) \times \mathrm{O}(2)$ symmetry. We will try to establish the range of (d, N) for which such a CFT exists, using the various standard consistency conditions of the bootstrap.

Primary CFT operators are characterized by dimension Δ, spin ℓ, and by global symmetry irreps, which in our case are labelled $\mathrm{R}_1\mathrm{R}_2$, where R_1 is an irrep of $\mathrm{O}(N)$ and R_2 of $\mathrm{O}(2)$. As in [85], our study will use crossing symmetry constraints for the correlation functions

$$\langle\phi\phi\phi\phi\rangle, \langle\phi\phi ss\rangle, \langle ssss\rangle, \tag{3.2}$$

where ϕ and s are the lowest-lying scalar primaries in the bifudamental (VV) and the total singlet (SS) irreps. In model (3.1) we have $s = \phi^2$, but we will not use this notation. The operator s is relevant. All subsequent total singlet scalars have to be irrelevant - this is the RG fixed point stability condition.

Operators in the $\phi \times s$ OPE transform as VV. In model (3.1), the VV scalars after ϕ are the cubic operators $\phi_{ai}\phi^2$ and $\phi_{aj}\phi_{bi}\phi_{bj}$. One linear combination of these, the derivative of the fixed point potential with respect to ϕ_{ai}, is a descendant of ϕ.[4] The orthogonal linear combination is a primary, denoted ϕ'.[5] This primary

[4] Just like ϕ^3 in the Wilson-Fisher fixed point is a descendant of ϕ.

[5] As usual, we denote by $\mathcal{O}', \mathcal{O}'', \ldots$ the subsequent primaries having the same quantum numbers as $\mathcal{O}$.

will be relevant, while the subsequent primaries in the VV sector, made of 5 or more ϕ's, are expected to be irrelevant.

Operators in the $\phi \times \phi$ OPE have $\mathrm{R}_1, \mathrm{R}_2 = \mathrm{S}, \mathrm{T}, \mathrm{A}$, where T, A are the symmetric traceless and antisymmetric 2-tensor irreps. The important operators in this OPE include the stress tensor $T_{\mu\nu}$, which is a spin-2 SS of dimension $\Delta = d$, and the conserved currents of the $\mathrm{O}(N)$ and $\mathrm{O}(2)$ symmetries, which are spin-1 AS and SA respectively, of dimension $\Delta = d - 1$. We will make gap assumptions above these operators.

We will also be paying particular attention to the ST scalar channel in the $\phi \times \phi$ OPE. In model (3.1), the lowest operator in this channel is $\mathcal{O}_{\mathrm{ST}} = \phi_{ai}\phi_{aj} - \text{trace}$, which has dimension close to 2 in any d in the $1/N$ expansion. A recurrent theme of our study will be how to distinguish our CFT of interest $\mathcal{C}^+$ from the Heisenberg CFT $\mathcal{H}$, which has a larger $\mathrm{O}(2N)$ symmetry. This is a crucial issue since $\mathcal{H}$ is present below $N_c(d)$, and thus has the potential to pollute our bootstrap analysis. The ST channel is interesting because this leading primary has dimension 2 at large-N in $\mathcal{C}^+$, while it has dimension $d-2$ in $\mathcal{H}$. We thus expect that assumptions in this channel will help us distinguish between both theories. We will in practice assume the existence of $\mathcal{O}_{\mathrm{ST}}$ and impose a gap to the subleading primary, which also has different large-N behavior in $\mathcal{H}$ and $\mathcal{C}^+$:

$$\Delta_{\mathrm{ST}'} = \begin{cases} 4 + O(1/N) & \text{in } \mathcal{C}^+\,, \\ d + O(1/N) & \text{in } \mathcal{H}\,. \end{cases} \tag{3.3}$$

This is explained in more detail in Appendix C.2, where also the $O(1/N)$ term for $\Delta_{\mathrm{ST}'}$ in $\mathcal{C}^+$ is given for the first time. We see that, for $d < 4$ and for N sufficiently large, $\Delta_{\mathrm{ST}'}$ is expected to be larger in $\mathcal{C}^+$ than in $\mathcal{H}$. By putting a carefully chosen gap above $\mathcal{O}_{\mathrm{ST}}$, we may hope to exclude $\mathcal{H}$ and focus on $\mathcal{C}^+$. It will turn out that a smaller gap in the ST channel allows both $\mathcal{H}$ and $\mathcal{C}^+$, while isolating them into two separate islands, which will be enough for all practical purposes. The usefulness of this channel is one of the main discoveries of this chapter.

To summarize, in Table 3.1 we list operators treated in our study as isolated. ϕ and s appear as both external and internal (exchanged), the rest only as internal.

We will study the consistency of the theory by the navigator function method [44], which has already been used in several bootstrap studies since its inception [79, 81, 83, 140, 141]. Our navigator function $\mathcal{N}$ will depend on three parameters: the dimensions of $\phi\,, s$ and $\mathcal{O}_{\mathrm{ST}}$. It will also depend on the gap assumptions, as

name	ℓ	$\mathcal{R}$	Δ	Note
ϕ	0	VV		ext./int.
s	0	SS		ext./int.
$T_{\mu\nu}$	2	SS	d	int.
$J_\mu^{\mathrm{O}(N)}$	1	AS	$d-1$	int.
$J_\mu^{\mathrm{O}(2)}$	1	SA	$d-1$	int.
$\mathcal{O}_{\mathrm{ST}}$	0	ST		int.

Table 3.1: Operators treated in our study as isolated.

discussed in more detail below.

Remark 3.2.1. Another way to distinguish $\mathcal{C}^+$ from $\mathcal{H}$ would be as follows. Primaries of $\mathcal{H}$ transform in irreps of $\mathrm{O}(2N)$. Decomposing these irreps under $\mathrm{O}(N) \times \mathrm{O}(2)$, each primary of $\mathcal{H}$ gives rise to several primaries in different $\mathrm{O}(N) \times \mathrm{O}(2)$ irreps having exactly the same scaling dimension. These exact degeneracies are not expected in $\mathcal{C}^+$. For example, the conserved current of $\mathrm{O}(2N)$ would give, upon reduction under $\mathrm{O}(N) \times \mathrm{O}(2)$, conserved spin-1 operators in the AT and TA irreps. On the other hand, conserved currents in these irreps are not expected in $\mathcal{C}^+$. We could have imposed small gaps above the unitarity bound in the AT and TA spin-1 channels, with the goal of excluding $\mathcal{H}$ and keeping $\mathcal{C}^+$.

Remark 3.2.2. Here is one more idea which we will not rely on in this chapter, but which could be useful in future studies.[6] We know that $\mathcal{C}^+$ is stable and $\mathcal{H}$ is unstable, hence $\mathcal{H}$ contains one more relevant SS scalar. This operator can be written as $\mathcal{O} = h_{IJKL} W^{IJKL}$, where W is a rank-4 symmetric traceless primary of $\mathrm{O}(2N)$ and h_{IJKL} is a tensor which breaks $\mathrm{O}(2N)$ to $\mathrm{O}(N) \times \mathrm{O}(2)$. Can we use only the existence of $\mathcal{O}$ to distinguish $\mathcal{H}$ from $\mathcal{C}^+$? Unfortunately, in our setup this will not work, because the $\mathrm{O}(2N)$ selection rules of $\mathcal{H}$ preclude the appearance of $\mathcal{O}$ in the OPEs $\phi \times \phi$, $s \times s$ to which we are sensitive. To be sensitive to $\mathcal{O}$, we would have to enlarge the setup by including external fields in T representations of $\mathrm{O}(N)$ and/or $\mathrm{O}(2)$. This is left for the future.

[6] We thank Ning Su for discussions concerning this remark.

3.2.1 Gap assumptions

The gap in a channel XX of spin ℓ will be denoted gapXX_ℓ. This means that all operators in this channel, except for the isolated operators in Table 3.1, are assumed to satisfy $\Delta \geqslant \text{gapXX}_\ell$. The first group of gap assumptions is:

$$\text{gapSS}_0 = d \tag{3.4a}$$
$$\text{gapSS}_2 = d + 1 \tag{3.4b}$$
$$\text{gapAS}_1 = (d-1) + 1 \tag{3.4c}$$
$$\text{gapSA}_1 = (d-1) + 1 \tag{3.4d}$$

These are easy to motivate. The SS_0 channel gap assumption is simply the fixed point stability condition of $\mathcal{C}^+$ (of course keeping in mind Remark 3.2.2). In the SS_2, AS_1 and SA_1 channels, we have respectively $T_{\mu\nu}$, of dimension d, and conserved currents of dimension $d-1$. At large-N, the gaps above these operators are equal to 2 at leading order. The value 1 was chosen in order to be comfortably below this value.

Let us discuss next the VV_0 channel, which has two isolated, relevant operators ϕ and ϕ'. The leading order predictions for the dimension of ϕ' in the ϵ-expansion and at large-N are:

$$\Delta_{\phi'} = 3 \times \frac{d-2}{2} + O(\epsilon)\,, \tag{3.5}$$
$$\Delta_{\phi'} = \frac{d-2}{2} + 2 + O(1/N)\,. \tag{3.6}$$

We are not aware of results beyond the leading order. Note that our analysis will take place at relatively large values of N. Even so, to be safe, we will use values for gapVV_0 comfortably below the smaller $O(\epsilon^0)$ prediction:

$$\text{gapVV}_0 = d - 2\,. \tag{3.7}$$

As already mentioned, we will also need a gap assumption in the ST_0 channel. The precise value of gapST_0 will be informed by (3.3) and by Appendix C.2, and will vary depending on the circumstance. In all other channels, gapXX will be set to the unitarity bound.

3.3 Numerical analysis

As mentioned in the previous section, a key ingredient in our approach is the ability to isolate $\mathcal{C}^+$ from $\mathcal{H}$ in the three-dimensional search space consisting of Δ_ϕ, $\Delta_{\mathrm{s}} \equiv \Delta_{\mathrm{SS}}$ and Δ_{ST}. We will first show that two islands materialize for N sufficiently above N_c and for smart enough choices of gapST_0. We will refer to these two isolated allowed regions as "the $\mathcal{C}^+$ island" and "the $\mathcal{H}$ island". We will also observe a third island, which we will be able to relate to the so-called "antichiral" theory $\mathcal{C}^-$ of Fig. 3.1. We will offer a qualitative picture of the behavior of these islands as we vary the parameters they depend on. We will then use the disappearance of the $\mathcal{C}^+$ island as the criterion determining $N_c(d)$, and extract a full critical curve, starting from the perturbative regime and going all the way down to $d = 3$. We will finish this section by observing an important nonperturbative signature of merger and annihilation of fixed points.

3.3.1 Isolating $\mathcal{C}^+$ from $\mathcal{H}$

Suppose we are given an allowed point in some parameter space $\mathcal{P}$. Presumably, it lies inside a larger island of allowed points, and we often would like to delineate this island. Using the navigator function [44], it is easy to at least "box in" the island by finding the extremal allowed values in every canonical direction of $\mathcal{P}$.

For the case at hand, we want to box in islands in the $(\Delta_\phi, \Delta_{\mathrm{SS}}, \Delta_{\mathrm{ST}})$-space. We show first in Fig. 3.3 what happens in the perturbative regime by considering what happens in $d = 3.8$ and at $N = 20$, the latter being moderately above $N_c(3.8) \approx 17.4$ predicted by the ϵ-expansion. In this case, we confirm that allowed regions exist in the close vicinity of the perturbative predictions for the scaling dimensions of both $\mathcal{H}$ and $\mathcal{C}^+$. We find, for the particular choice of assumption described in the title of Fig. 3.3, that there is no allowed path linking these two regions, meaning that we have achieved our goal of isolating two islands. Unfortunately, in addition, we see a third island matching the perturbative predictions of the scaling dimensions of the $\mathcal{C}^-$ fixed point. This unstable CFT has two relevant total singlet scalars our bootstrap setup *should* be sensitive to (which was not the case for $\mathcal{H}$), and was therefore expected to be excluded by the gap assumption (3.4a). A possible explanation for the unexpected behavior is that since we are near the value of N where this operator becomes marginal, it may only be slightly relevant. Our numerics at this degree of precision may simply not be sensitive enough to the exact value of the dimension of this operator.

In Fig. 3.4, we show the results of the same exercise with $d = 3$ and $N = 8$, close to but above the expected value $N_c = 5.96(19)$ of the ϵ-expansion. We see that the structure with two disconnected islands persists all the way down into the nonperturbative regime, and while there is still good agreement with the predictions of the ϵ-expansion, the islands significantly enlarge. In this regime, we should therefore be more careful with the gap assumptions we use, as gap assumptions too conservative may lead to a single island containing $\mathcal{H}$ and $\mathcal{C}^+$.

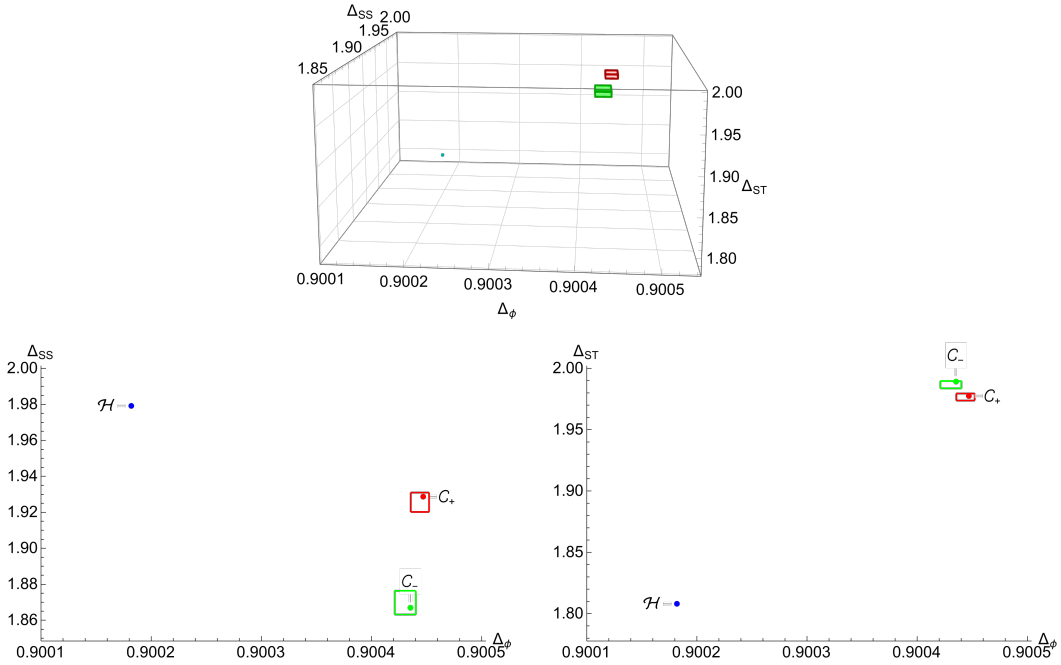


Figure 3.3: *Top:* Minimal bounding boxes containing isolated allowed regions in the space of O(20) × O(2) symmetric theories in $d = 3.8$ under the assumptions (3.4)-(3.7) and with $\text{gapST}_0 = 2.9$, well below the large-N predictions for $\Delta_{\text{ST}'}$ in $\mathcal{H}$ and $\mathcal{C}^+$. *Bottom*: Projections of these allowed regions to the $(\Delta_\phi, \Delta_{\text{SS}})$ and $(\Delta_\phi, \Delta_{\text{ST}})$-planes. For reference, we also include the perturbative estimates of the ϵ-expansion for the locations of $\mathcal{H}$, $\mathcal{C}^+$ and $\mathcal{C}^-$.

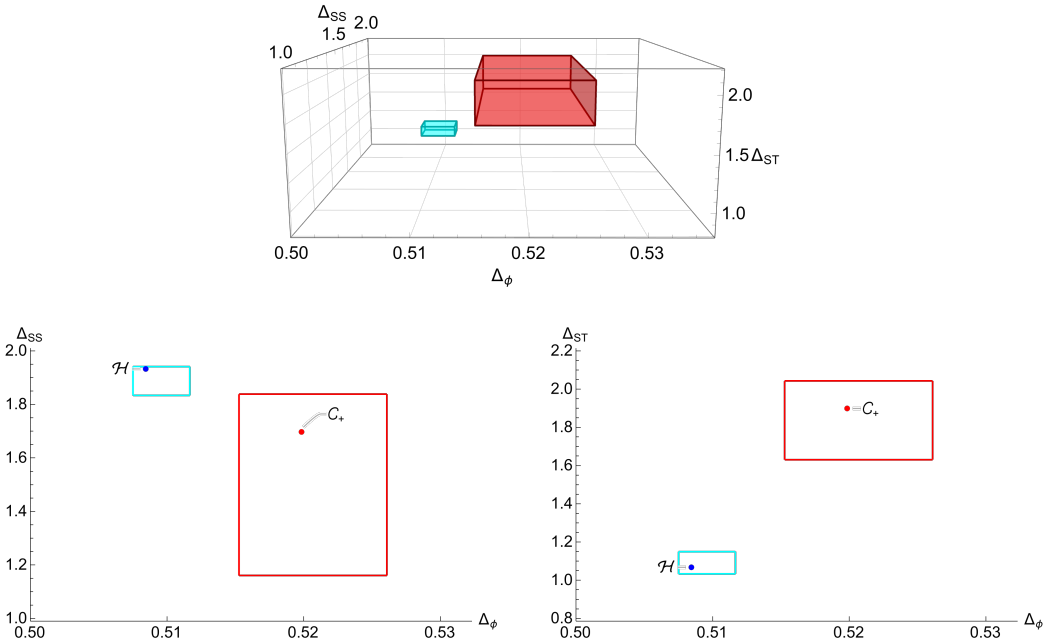


Figure 3.4: *Top:* Minimal bounding boxes containing isolated allowed regions in the space of O(8) × O(2) symmetric theories in $d = 3$ under the assumptions (3.4)-(3.7) and with $\text{gapST}_0 = 2.5$, well below the large-N predictions for $\Delta_{\text{ST}'}$ in $\mathcal{H}$ and $\mathcal{C}^+$. *Bottom*: Projections of these allowed regions to the $(\Delta_\phi, \Delta_{\text{SS}})$ and $(\Delta_\phi, \Delta_{\text{ST}})$-planes. For reference, we also include the field-theory estimates of [124, 142] for $\mathcal{C}^+$ and of the large-N expansion for $\mathcal{H}$. Here we did not find a separate isolated region associated to $\mathcal{C}^-$. This could be because we are sensitive enough to the second relevant singlet to exclude this theory, or it could be that the theory lives inside the now much larger island containing $\mathcal{C}^+$.

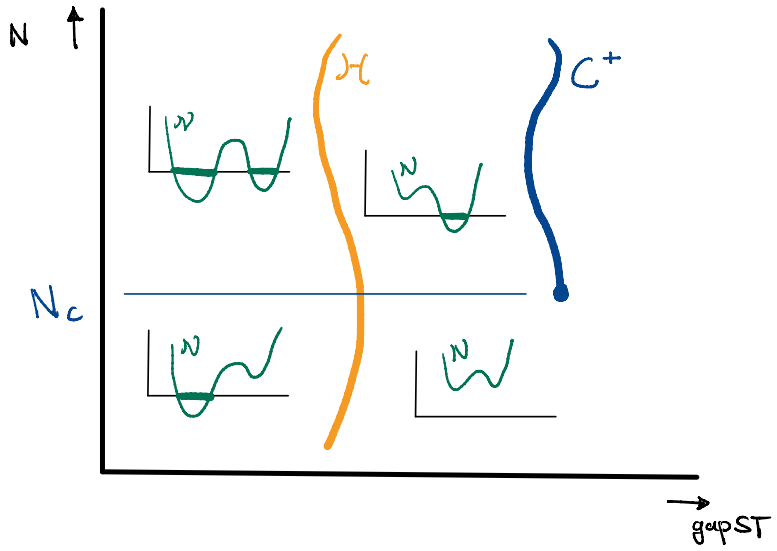


Figure 3.5: An overview of the behavior of the navigator function as N and gapST_0 are varied. The island corresponding to $\mathcal{C}^+$ disappears at N_c, whereas the island corresponding to $\mathcal{H}$ persists (if gapST_0 is low enough).

Further exploration leads us to believe that the qualitative picture is as sketched in Fig. 3.5. The orange and blue lines schematically indicate the value of $\Delta_{\text{ST}'}$ as we vary N, for $\mathcal{H}$ and $\mathcal{C}^+$ respectively. As indicated, the former fixed point exists for all N, while the latter exists only for $N > N_c$. The four insets show the behavior of the navigator function $\mathcal{N}$ as we vary another parameter (say Δ_ϕ). There are two allowed regions for sufficiently large N and sufficiently small gapST_0. Then, as we vary these parameters, one or both of the allowed regions disappear.[7] The existence of two islands is advantageous because it allows us to be conservative with our assumptions on gapST_0. Indeed, if both islands were connected as soon as they were both allowed, we could only estimate N_c accurately by putting gapST_0 somewhere between the expected value of ST$'$ in $\mathcal{H}$ and $\mathcal{C}^+$. This is tricky, since in the interesting case of $d = 3$, our only knowledge of ST$'$ comes from the large-N expansion. Close to $N_c(3)$, N is not so large anymore, and furthermore the large-N expansion is blind to the merger and annihilation of $\mathcal{C}^+$ and $\mathcal{C}^-$. It would thus be hard to justify the use of (3.3). Here, however, we can just estimate N_c by observing the vanishing of the isolated $\mathcal{C}^+$ island, while completely ignoring what

[7]For very small gapST_0, for example close to the unitarity bound with no isolated ST operator, we actually observe that there is a single allowed region in which the navigator function appears to only have a single minimum. This region is not shown in Fig. 3.5.

happens to the $\mathcal{H}$ island. We can observe this even when gapST_0 is well below its expected value in $\mathcal{H}$, so there is no fine-tuning needed! Because all other gaps were put very conservative, we can be confident in the validity of $N_c(3)$ we will determine in the following section.

Remark 3.3.1. Testing on a particular point in the search space for $d = 3.8$ and N close to N_c, we observed that the value of gapVV_0 has virtually no effect in the range $1.7 < \text{gapVV}_0 < 2.5$. However, putting it too high at $\text{gapVV}_0 = 3.2$ (where the large-N estimate is 2.9), a point with $\mathcal{N} \sim 10^{-5}$ now jumps to a large navigator value of 1.99, close to the maximal possible value of 2. It is thus fine to put this gap conservatively, as we did in (3.7), while putting it too high is dangerous. The value of the gaps above the stress-tensor and conserved currents have themselves a moderate effect on the shapes of the allowed regions. However, this effect does not seem to translate to a large variation in N_c, which changes in $d = 3.8$ by about 0.02 when lowering the gaps from 1.2 above the unitarity bounds to 0.7. This motivates the use of the more conservative values (3.4b)-(3.4d).

3.3.2 $N_c(d)$ from the disappearance of the $\mathcal{C}^+$ island

Armed with the knowledge acquired in Subsection 3.3.1, we are now ready to look for N_c. One way to determine it is to look for the value of N for which the minimal navigator point x_{min} inside of the $\mathcal{C}^+$ island gives

$$\mathcal{N}(x_{\text{min}}) = 0\,. \tag{3.8}$$

More details will be given in Appendix C.1 on how N_c is determined in practice. We use the gap assumptions laid out in Subsection 3.2.1, with the only missing ingredient being the numerical value of gapST_0. It was mentioned at the end of the last section that this gap could be put conservatively, since there was no need to actually kill the $\mathcal{H}$ island. We choose the value

$$\text{gapST}_0(d) = \frac{d}{2} + 1\,, \tag{3.9}$$

which is seen in Fig. 3.6 to lie well below the large-N prediction for the value of $\Delta_{\text{ST}'}$ in $\mathcal{C}^+$ and $\mathcal{H}$ at the expected value of N_c from the ϵ-expansion. Using these spectral assumptions, we obtain the critical curve presented in Fig. 3.7. This curve ends in $d = 3$ at

$$N_c(3) = 3.78\,, \tag{3.10}$$

well above the physically relevant values $N = 2\,, 3$. The bootstrap thus *strongly suggests the first-order behavior of the phase transition of stacked triangular antiferromagnets and helimagnets.*

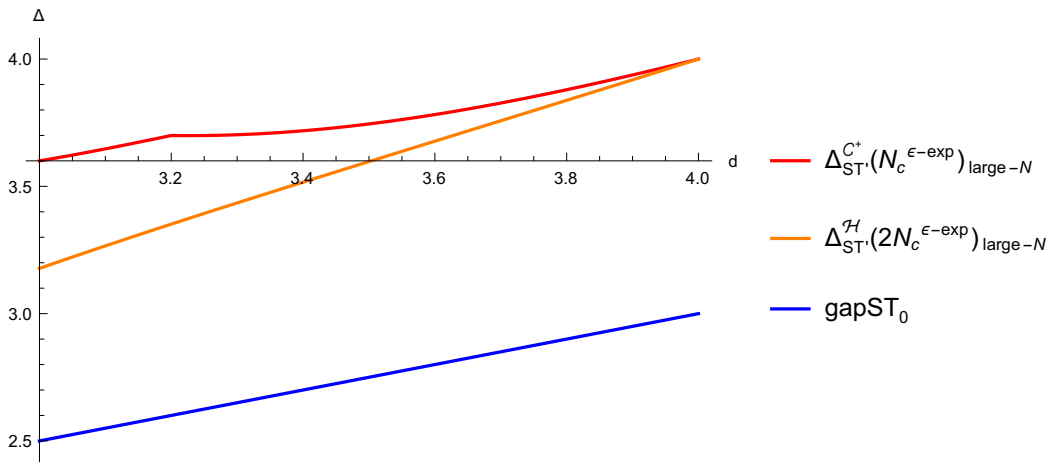


Figure 3.6: Comparison of the gap assumption $\mathrm{gapST}_0 = \frac{d}{2} + 1$ with the dimension of $\mathcal{O}_{\mathrm{ST}'}$ in $\mathcal{H}$ and $\mathcal{C}^+$ at leading order in the large-N expansion. The dimensions are quoted at the expected value of N_c from the ϵ-expansion.

We show for comparison the NPRG results of [126] as well as the ϵ-expansion prediction. All methods more or less agree in the perturbative regime. Although the bootstrap does exclude part of the NPRG curve, its power weakens as $d = 3$ is approached, so that in the end it leads to a more conservative estimate of $N_c(3)$. We opted in this chapter to make the strongest case possible not by playing the game of using more aggressive gap assumptions to maximize $N_c(3)$, but instead by using assumptions as conservative as possible so that, true to the spirit of the conformal bootstrap, the value we extract is as close as possible to a *rigorous* lower bound. In this spirit, better estimates of $N_c(3)$ should be obtained either by considering more correlators, or by increasing Λ, both beyond the scope of this work.

We feel the need now to make broader comments about the physical implications of these results, and in particular about earlier attempts to bootstrap this model [85, 125]. These studies claim to find evidence of focus-type fixed points for physical values of spin components by comparing the scaling dimensions of operators found on kinks or in islands to critical exponents observed in physical experiments and obtained from fixed-dimensional RG. However, these focus-type

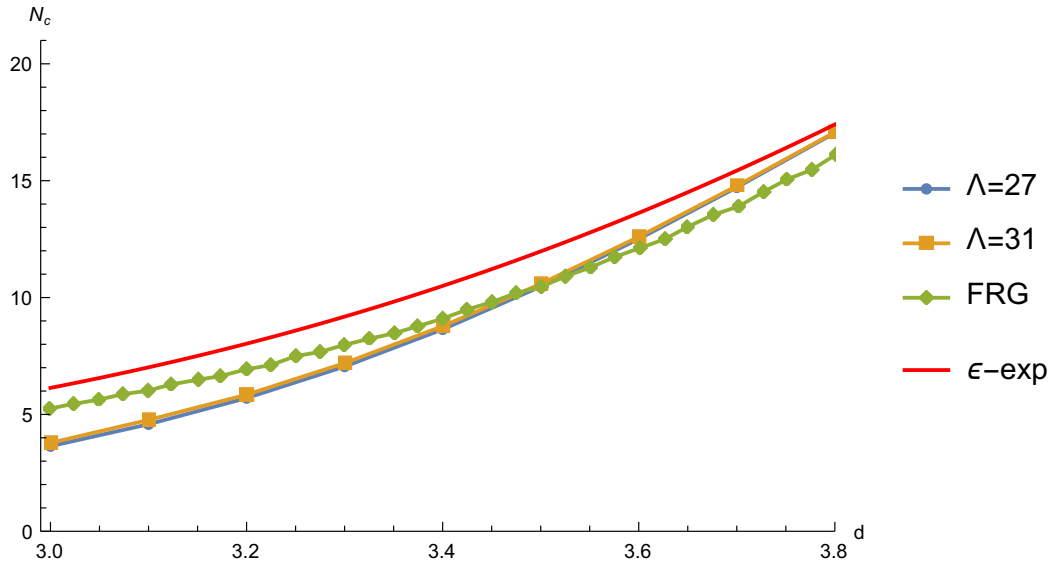


Figure 3.7: $N_c(d)$ extracted from a numerical conformal bootstrap study at derivative orders $\Lambda = 27$ (blue) and $\Lambda = 31$ (orange). We also show results obtained using NPRG [126] and the ϵ-expansion. We observe no sign of the turnaround behavior predicted by the fixed-dimensional RG.

fixed points correspond to very non-unitary CFTs, having low-dimensional operators with complex scaling dimensions. It is conceivable that "slightly" focus fixed points, with small imaginary components to the correction-to-scaling exponents, could violate unitarity mildly enough that the violations would be invisible to a unitary bootstrap analysis. However, as stated above, the fixed points in $d = 3$ for $N = 2, 3$ are expected to be very nonunitary. It is not clear how it would be possible to find signatures of these CFTs with the numerical bootstrap as implemented in these studies.

We decided to attack the problem from a somewhat different direction from [85, 125]. We set out to determine a critical curve at least in the perturbative regime, and followed it down to the nonperturbative regime. We see that this curve survives all the way down to $d = 3$, and is in qualitative agreement with the dominant theoretical scenario of the ϵ-expansion and the NPRG. We believe this is strong evidence for the validity of this scenario. We however cannot rule out a turnaround of the $N_c(d)$ curve for some $d^* < 3$, and neither can we rule out the existence of a CFT well below our $N_c(d)$ curve. We did not explore these regions of (d, N)-space since it is much trickier to motivate gap assumptions in these very nonperturbative regimes. Also, since the putative CFT living well below our $N_c(d)$

curve is expected to be strongly nonunitary, other conformal bootstrap approaches not relying on unitarity seem like the correct modes of investigation.

3.3.3 Square root behavior

A telltale signature of merger and annihilation is the square root behavior of the CFT data of the merging theories [143, 144]. This behavior is nonperturbative in nature: indeed, the ϵ-expansion for the chiral theories gives $\Delta_i \sim \sqrt{N - N_c(4)}\,\tilde{\Delta}(\epsilon)$+analyt. close to N_c,[8] in essence only making manifest the square root behavior exactly in $d = 4$. Observing it in $d < 4$ would give yet another example of the nonperturbative power of the bootstrap, and quite an important one since it has proven difficult to observe in bootstrap studies of other models [79]. Inspection of the ϵ-expansion for several low-lying operators shows that scaling dimensions exhibit more (or less) square root behavior depending on the relative importance of their analytic part near N_c. Δ_{SS} comes out of this analysis as a great candidate, and explicit computations confirm this, as can be seen in Fig. 3.8.

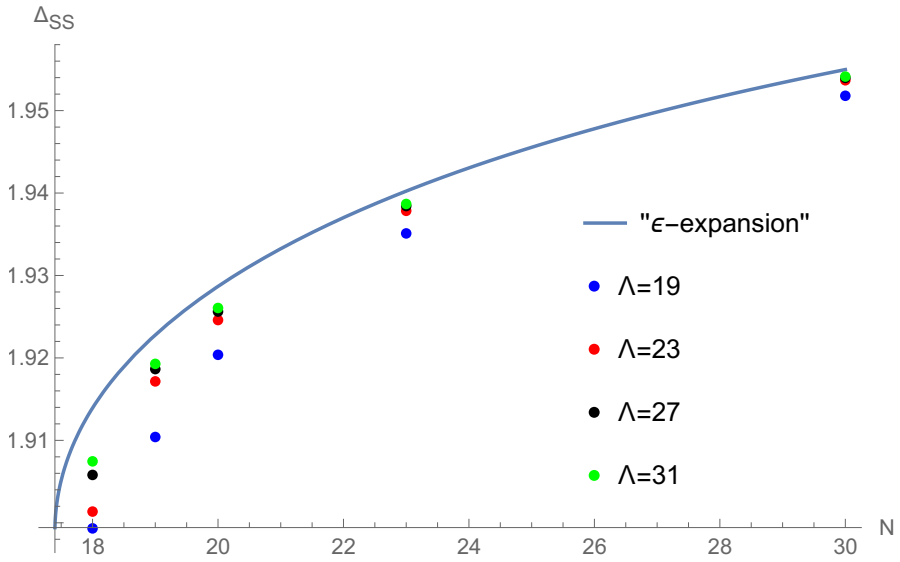


Figure 3.8: Δ_{SS} at the minimal navigator point as a function of N, for $d = 3.8$ and for various values of the derivative order Λ. This is compared to the ϵ-expansion supplied with the re-expansion trick of [145]. This trick moves the square root singularity in N from $N_c(4)$ to $N_c(d)$.

[8] See e.g. the data listed in the ancillary file of [85].

3.4 Conclusion

We concerned ourselves in this last chapter with the nature of the phase transition of classical frustrated noncollinear magnets by performing a thorough conformal bootstrap analysis of CFTs with $\mathrm{O}(N)\times\mathrm{O}(2)$ global symmetry. This analysis led to the determination of a critical curve $N_c(d)$, below which the transition of this class of magnets is expected to be first order, in agreement with orthogonal theoretical approaches like the ϵ-expansion and the NPRG. The approximate lower bound $N_c(3) = 3.78$ suggests that the scaling behavior observed in numerical simulations and experiments is a manifestation of weak first-order, and not true second-order behavior for the physical values of spin components $N = 2\,, 3$ in three dimensions, as suggested in [126]. We find no evidence of the turnaround behavior sketched in Fig. 3.2, which casts strong doubts on the scenario of the fixed-dimensional RG.

As an immediate improvement on the results of this study, it could be interesting to verify the sensitivity of $N_c(d)$ to the assumption in the ST channel. We chose to use a very conservative value for the gap in this channel, well below the expected value from the large-N expansion, which resulted in the highest Λ determination of $N_c(3)$ being relatively far from that of the NPRG and the ϵ-expansion. Perhaps one can get better agreement by fine-tuning this gap. More generally, collision of fixed points like the one observed here are quite ubiquitous in nature [79, 120, 146, 147]. We hope that this study demonstrated in such situations the power of combining perturbation theory and numerical bootstrap *by actually following solutions from the perturbative to the nonperturbative regime.* It may prove difficult to locate a collision by jumping straight to the nonperturbative regime, and one loses the ability to self-correct assumptions that could otherwise be seen to fail somewhere between the perturbative and nonperturbative regimes. Whereas such an exercise could have felt like a daunting task a number of years ago, the advent of the navigator method now enables one to seamlessly move in an external parameter space by way of methods like those described in Appendix C.1.

Conclusion

A few key results should stand out from this thesis. We first introduced a novel tool for the numerical conformal bootstrap, the *navigator function*. In fact, we saw that there is not only one navigator function, but that there exists many, each corresponding to a different term to be added to the crossing equation in order for it to always have a solution. Each navigator function is constructed so as to give a measure of "how" allowed or disallowed a point is, and can thus be used as a compass to orient the exploration of the space of CFT data. We explicitly constructed two navigator functions, the *GFF-navigator* and the Σ-*navigator*. They turn out both to have different strengths (known behavior in certain limits vs. generality and ease of use), making them more or less appealing in any given situation. Importantly, both are shown to be computable with existing state-of-the-art numerical bootstrap methods.

We then showcased the power of the navigator function in three test cases, ordered in terms of increasing complexity. We began with the 3d critical Ising Model, reproducing for the well-known case of the mixed correlator bootstrap of the lowest-lying operators σ and ϵ the Ising island as the region of negative navigator function. We proposed simple numerical algorithms to be used in order to converge to allowed regions and move around in them. We then moved on to the critical $O(N)$ model. This model displays ample interesting behaviors when its two defining parameters, the spacetime dimension d and N, are varied continuously. We showed how the navigator function helped us move efficiently in this external parameter space. In doing so, we were able to confirm that the numerical bootstrap is insensitive (at low enough precision) to the nonunitarities caused by fractional values of d and N. We saw, however, that the nonunitarities arising for $N < 1$ were strong enough to kill the unitary solution to crossing. The final problem we investigated was that of frustrated magnets with noncollinear order. The controversy regarding the order of their phase transitions can be translated

into the question of the existence of a certain stable fixed point with $O(N) \times O(2)$ global symmetry. Using the navigator function, we found in the perturbative regime $d = 4 - \epsilon$ a critical N where this fixed point collides with an unstable fixed point, and then flowed down in d along the critical curve $N_c(d)$ to extract a value in the physical dimension $d = 3$. We were able to observe nonperturbatively a key signature of the merger and annihilation of fixed points in the square root behavior of scaling dimensions as the collision is approached. We found results for $N_c(d)$ which align with most other theoretical and numerical approaches, and cast strong doubts on the validity of an alternative picture stemming from the so-called fixed-dimensional RG.

Since the original publication of [44], many groups have started both using the navigator method to tackle interesting problems [79, 81, 141, 148], but also working on improving the method itself. The recently-developed "skydiving" way to optimize the navigator function [149] has rendered the navigator method more efficient for virtually any application than the previous state-of-the-art technology. Because of this, we believe the navigator method is here to stay, and it therefore might be worth it to ponder about the following questions:

- Can we cook up a "nicer" navigator function than those proposed in [44]? In certain examples, the GFF-navigator was seen to be non-smooth along extended regions, which has the potential to cause all sorts of issues when attempting to use optimization algorithms on it. Can we find a truly smooth navigator function?

- Does the structure of the navigator function encode interesting information beyond the positive-allowed, negative-disallowed association, or is it simply a numerical tool? Is there any meaning to the navigator function having a unique minimum vs. multiple local minima inside of an allowed region? Is the fact that the navigator minimum sits close to the true theory inside of an allowed region a meaningful observation?

- In the past decade, the bootstrap has morphed into a way to approach problems in physics that has expanded much beyond conformal field theory. Could we define navigator functions for the S-matrix bootstrap, the quantum mechanical bootstrap, or the bootstrap of lattice models? Would they be useful?

Appendix A

Appendix to Chapter 1

A.1 Tweaks of the GFF-navigator

As mentioned in Section 1.2.1 and footnote 12, the GFF-navigator definition has to be tweaked in presence of additional GFF operators violating gap assumptions. These modifications will be discussed here. In addition we will explain how to deal with the case where the navigator function depends on the magnitude of a squared OPE coefficient.

A relevant example in the single-correlator setup of Section 1.2.1 is to assume a gap in the scalar spectrum above Δ_*. E.g. suppose that all further scalars above the one at Δ_* are required to be above Δ_{gap}. This corresponds to changing the constraint $\Delta \geqslant \Delta_*$ for $\ell = 0$ in (1.2) to "$\Delta = \Delta_*$ or $\Delta \geqslant \Delta_{\text{gap}}$." We can still define the navigator by the same Eq. (1.5). In this case we don't in general expect the navigator to be monotonic in the Δ_* direction. For large Δ_{gap}, definition (1.6) of M_{GFF} will have to be modified, including all scalar GFF conformal block contributions below Δ_{gap}:

$$M_{\text{GFF}}(u,v) = \sum_{n\geqslant 0\,:\, 2\Delta_\phi+2n\leqslant\Delta_{\text{gap}}} c_n F_{2\Delta_\phi+2n,0}(u,v), \tag{A.1}$$

where c_n are explicitly known coefficients ($c_0 = 2$). These are contributions of GFF operators of schematic form $\phi\Box^n\phi$.

For the 3-correlator setup, let us discuss how the GFF-navigator definition (1.23) should be modified in the case of gap assumptions in the spectrum of $\ell \geqslant 1$ operators. As a concrete example, let us define the navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, c_T)$ where c_T is the 2pt function coefficient of the canonically normalized stress-tensor. The

c_T parametrizes the OPE coefficients of the corresponding unit-normalized $\Delta = 3$, $\ell = 2$ primary $\mathcal{O}^+$ as:

$$\lambda_{\sigma\sigma\mathcal{O}} = K_3 \frac{\Delta_\sigma}{\sqrt{c_T}}, \qquad \lambda_{\epsilon\epsilon\mathcal{O}} = K_3 \frac{\Delta_\epsilon}{\sqrt{c_T}} \tag{A.2}$$

where K_d is a known d-dependent constant. To isolate the stress tensor, we need to impose a gap assumption on the higher-dimension $\ell = 2$ $\mathcal{O}^+$ operators. We will assume that all of them have $\Delta \geqslant \Delta_{\text{gap}}$ where $\Delta_{\text{gap}} > 3$ is some fixed parameter. E.g. let us choose $\Delta_{\text{gap}} = 5$, which allows the 3d Ising CFT.[1]

For this problem, the analogue of Eq. (1.21) will be

$$\begin{aligned}\vec{V}_{0,0} + \lambda \vec{M} + \text{Tr}\left[P_{\Delta_\epsilon,0}\left(\vec{V}_{+,\Delta_\epsilon,0} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \vec{V}_{-,\Delta_\sigma,0}\right)\right] \\ + \frac{(K_3)^2}{c_T}\,\text{Tr}\left[\begin{pmatrix} \Delta_\sigma^2 & \Delta_\sigma\Delta_\epsilon \\ \Delta_\sigma\Delta_\epsilon & \Delta_\epsilon^2 \end{pmatrix} \vec{V}_{+,3,2}\right] \\ + \sum_{(\Delta,\ell)\in S_+} \text{Tr}\left[P_{\Delta,\ell}\vec{V}_{+,\Delta,\ell}\right] + \sum_{(\Delta,\ell)\in S_-} p_{\Delta,\ell}\vec{V}_{-,\Delta,\ell} = 0\,,\end{aligned} \tag{A.3}$$

where the stress tensor contribution is now isolated, and S_+ compared to (1.18) implements the stronger requirement that $\Delta \geqslant \Delta_{\text{gap}}$ for $\ell = 2$.

To define the GFF navigator, we will proceed analogously to (A.1) and include in $\vec{M}_{\text{GFF}}$ additional terms corresponding to all GFF primaries violating the gap assumptions. In the case at hand, we have to check the spin-2 GFF operators of the schematic form $\sigma\partial\partial\Box^n\sigma$ and $\epsilon\partial\partial\Box^n\epsilon$. For $\Delta_{\text{gap}} = 5$ and $\Delta_\sigma, \Delta_\epsilon$ around the 3d

[1] Recall that the 3d Ising CFT has $\Delta_{T'} = 5.50915(44)$ [82].

Ising island, only the $n = 0$ operators of this form are below the gap. So we take

$$\begin{aligned}\vec{M}_{\text{GFF}} = \text{Tr}\left[\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}\vec{V}_{+,2\Delta_\sigma,0}\right] + \text{Tr}\left[\begin{pmatrix} c(\Delta_\sigma) & 0 \\ 0 & 0 \end{pmatrix}\vec{V}_{+,2\Delta_\sigma+2,2}\right] \\ + \text{Tr}\left[\begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}\vec{V}_{+,2\Delta_\epsilon,0}\right] + \text{Tr}\left[\begin{pmatrix} 0 & 0 \\ 0 & c(\Delta_\epsilon) \end{pmatrix}\vec{V}_{+,2\Delta_\epsilon+2,2}\right] + \vec{V}_{-,\Delta_\sigma+\Delta_\epsilon,0} \\ - \frac{(K_3)^2}{c_T}\,\text{Tr}\left[\begin{pmatrix} \Delta_\sigma^2 & \Delta_\sigma\Delta_\epsilon \\ \Delta_\sigma\Delta_\epsilon & \Delta_\epsilon^2 \end{pmatrix}\vec{V}_{+,3,2}\right], \quad \text{(A.4)}\end{aligned}$$

where $c(\Delta_\phi)$ is the (explicitly known) coefficient of the $\Delta = 2\Delta_\phi + 2$, $\ell = 2$ conformal block in the decomposition of the GFF 4pt function $\langle\phi\phi\phi\phi\rangle$.

The first two lines in (A.4) are the analogue of (A.1). The last line is an additional small modification needed due to the presence of OPE coefficient parameter c_T among navigator function variables. It is the negative of the stress tensor contribution in (A.3). Including this piece into $\vec{M}_{\text{GFF}}$ is needed to guarantee that problem (A.3) has a solution with $\lambda = 1$ for any fixed value of c_T. This in turn guarantees that the navigator $\mathcal{N}(\Delta_\sigma, \Delta_\epsilon, c_T)$ is bounded from above by 1 for any value of its arguments.

A.2 Feasibility as optimization

In this appendix we discuss the problem of finding a navigator function from the more abstract semidefinite programming perspective. We will assume the reader is familiar with the semidefinite programming terminology of Section 1.4.1 in the main text. As we review there, a general numerical bootstrap problem of the opimization type can be formulated as the dual problem given in Eq. (1.33) on p.42. For a feasibility problem, on the other hand, the question is merely whether there exist any y and Y that obey the constraints. In that case the standard approach is to set $b = 0$ in (1.33) and run SDPB until one of two termination conditions are met:

- If a dual feasible point (y, Y) is found, terminate with 'success';
- If a primal feasible point x is found and $c^T x < 0$, then terminate with 'failure'.

The last termination condition is explained by the duality gap: if $b = 0$ then $D(x, y) = c^T x$, which can only be negative (for a primal feasible x) if no dual feasible point exists.[2]

The above two termination conditions correspond to the binary oracle output discussed in Section 1.1: "success" means that the point is excluded (CFT does not exist), while "failure" means that the point is allowed (CFT may exist).

To pass from this to a navigator function, we need to reformulate the feasibility search as an optimization problem. The commonly adopted approach to do so is to use *slack variables* that relax the constraints. As discussed in the main text, in the context of the conformal bootstrap one can add an additional term to the crossing equations, in such a way that these equations can always be obeyed if the coefficient of this extra term is positive. The minimization of the coefficient of this term is then a potential navigator function: if it is positive we are in the 'success' region and if negative we are in the 'failure' region.

We will now describe an alternative navigator function construction, which does not rely on the physical intuition of the crossing symmetry equations. Instead, we will start with a general feasibility semidefinite program of the type (1.33) with $b = 0$, and transform it into an optimization SDP.

As a first attempt, consider replacing the condition $Y \succeq 0$ in (1.33) with a maximization problem:

$$Y \succeq 0 \qquad \Longrightarrow \qquad \text{maximize } \nu \in \mathbb{R} \text{ such that } Y - \nu I \succeq 0 \tag{A.5}$$

with I the identity matrix. With this transformation the 'success' and 'failure' cases mentioned above respectively correspond to $\nu > 0$ and $\nu < 0$ at optimality, and (in the conventions of the main text) we can take ν at optimality as a candidate navigator function.

Unfortunately the modification (A.5) is not guaranteed to give a finite navigator in the "success" region. E.g. suppose there exists a $Y' \succ 0$ such that $\mathrm{Tr}(A_* Y') = 0$. In the 'success' region one can add this Y' to any feasible solution Y with arbitrarily large coefficient. This would then imply that $\nu \to +\infty$ at optimality. We therefore

[2] With $b = 0$ the primal problem is completely homogeneous in the sense that the constraints are invariant under rescalings $x \to \lambda x$ with non-negative λ. In particular, there is an obviously primal feasible point $x = 0$. Since this point teaches us nothing about dual feasibility, the inequality in the second termination condition has to be strict. Furthermore, if we were to ignore the above termination conditions and run the program to optimality then we would either find $x \to 0$ (in the 'success' case) or x diverges such that $c^T x \to -\infty$ (in the 'failure' case). We thank Petr Kravchuk for a discussion of these issues.

cannot exclude a divergence in this candidate navigator function unless we know that the program does not allow such Y'.

To guarantee boundedness in the 'success' region, we apply the same idea, but on the primal side, that its, by modifying the primal problem (1.34). For simplicity, let us first assume that there always exists an x such that

$$B^T x = 0, \qquad c^T x < 0\,, \tag{A.6}$$

meaning we only need to introduce a slack variable for the positive semidefiniteness condition. In that case the right problem to solve is:

$$\begin{aligned} &\text{minimize } \nu \quad \text{over} \quad x \in \mathbb{R}^P, \nu \in \mathbb{R} \\ &\text{such that } X(x) := x^T A_* + \nu I \succeq 0 \\ &\qquad\qquad B^T x = 0 \\ &\qquad\qquad c^T x = -1 \end{aligned} \tag{A.7}$$

This is a standard primal semidefinite programming problem, and we can run it to optimality without special termination conditions. The value of ν at optimality is the navigator function. In the 'success' region it is guaranteed to be positive (and finite) and then it is likely to be as good a navigator function as the ones used in the main text.

The dual version of the program in (A.7) is:

$$\begin{aligned} &\text{maximize } -\xi \quad \text{over} \quad y \in \mathbb{R}^n, Y \in \mathcal{S}^K, \xi \in \mathbb{R} \\ &\text{such that } Y \succeq 0 \\ &\qquad\qquad -c\,\xi = By + \mathrm{Tr}(A_* Y) \\ &\qquad\qquad \mathrm{Tr}(Y) = 1 \end{aligned} \tag{A.8}$$

As usual, the introduction of free variables on one side yields additional constraints on the other side. In this case the trace condition on Y guarantees the boundedness of the problem, and the parameter ξ allows for the re-scaling of a feasible (y, Y) such that this constraint can be met.

Let us also discuss boundedness (from below) in the 'failure' region of (A.7).

We do not have a first-principles argument for boundedness everywhere:[3] for the same reasons as above, the navigator function of (A.7) diverges in the 'failure' region if there exists a x' which obeys

$$(x')^T A_* \succ 0, \qquad B^T x' = 0, \qquad c^T x' = 0. \tag{A.9}$$

Fortunately, in conformal bootstrap applications this is unlikely. To see this, recall that the formulation (1.33) with c and b arises only after eliminating one component of y from a normalization condition $n^T y = 1$ for some normalization vector n, which is typically the identity operator. Reinstating this normalization condition as a separate constraint to (1.33) one finds that unboundedness of the modification (A.8) can really only occur if there is a solution to the crossing symmetry equations (with positive coefficients) without an identity operator. Although this is known to be the case for problems in $d = 2$ and $d = 1$, it is an unlikely possibility in most numerical bootstrap problems and then (A.8) is also bounded in the 'failure' region. The corresponding navigator therefore obeys the same manifest properties as those used in the main text.

Finally let us consider the case where the equality constraints in the primal problem cannot obviously be met. In that case not all is lost: one can simply replace them with

$$B^T x = b + \nu \mathbf{1} - \lambda, \qquad \lambda > 0, \tag{A.10}$$

with $\mathbf{1} = (1, 1, \ldots 1)$ a constant vector, and proceed by minimizing $\nu + \sum_i \lambda_i$. As before, a positive value at optimality means that no feasible point exists and so we still have a good candidate for a navigator function in the 'success' region.

The navigator functions introduced in this appendix are more general since they work for any feasibility problem of the type described in Eq. (1.33) with $b = 0$. On the other hand, for numerical conformal bootstrap applications they offer little upside compared to the GFF and Σ-navigators discussed in the main text. Furthermore they also suffer from a practical disadvantage. To see this, note that the GFF and Σ-navigators are readily implemented with the usual conformal bootstrap software: programs like `sdp2input` or `pvm2sdp` can be used to translate the problems into a format acceptable by SDPB, which e.g. involves setting up matrices B and A_*, and SDPB then does the rest of the computation. Unfortu-

[3] Of course the problem becomes trivially bounded if we impose that $\nu > -1$ in the primal problem. This is however all but guaranteed to result in a non-smooth (and locally constant) navigator function in the primal feasible region, which is of limited use for our purposes.

nately this workflow does not quite work for the navigator function described in Eq. (A.8). The main problem is that SDPB is meant to solve problems where the matrices A_p have rank one and the constraint $\mathrm{Tr}(Y) = 1$ is not of this form.[4]

A.3 Comments on variations of the objective

In section 1.4, we found a simple formula (1.48) for the linear-order variation in the objective function under changing the SDP. In this appendix, we give a formula for the quadratic-order variation as well, and explain how it can be computed easily using machinery already present in SDPB. We also present numerical checks of both the linear and quadratic variations, determining how their errors scale with the duality gap.[5]

A.3.1 A formula for the quadratic variation

Consider changing an SDP by $(b, c, B, A) \to (b, c, B, A) + (db, dc, dB, dA)$. For simplicity, we assume $dA = 0$. (In practice, we can ensure this by keeping constant the "bilinear basis" and "sample scalings" discussed in [42].) The linear-order change in the objective at optimality is

$$dL = db^T y + dc^T x - x^T dB y, \tag{A.11}$$

where L is the Lagrange function (1.49).

As explained in section 1.4.3, dL is independent of (dx, dy, dX, dY) because the variation of the Lagrange function with respect to (x, y, X, Y) vanishes at optimality. The same reasoning implies that the quadratic variation in the objective should be linear in (dx, dy, dX, dY). To compute it, we will work at finite μ. Afterwards, we consider the $\mu \to 0$ limit of the resulting expression and assess the size of finite-μ corrections.

[4] One can probably impose the trace constraint in an SDPB compatible way, by extending y with spurious variables $\hat{y}$. One then needs to set these equal to the diagonal components of Y in the sense that $\hat{y}_1 = Y_{11}$, $\hat{y}_2 = Y_{22}$, etc. This can be done by including one additional equation for each diagonal value of Y by extending b, c, B and A. Finally, by extending these quantities by one more entry we can impose the trace constraint by demanding $\sum_i \hat{y}_i = 1$. Alternatively one can use this equation to eliminate one of these extra components instead. It is unclear whether such an altered semi-definite problem still corresponds to any polynomial matrix problem.

[5] The quadratic variation of the objective could be used to compute the Hessian of the navigator function, enabling the use of Newton's method for finding allowed points and extremizing CFT data. We leave possible applications of the quadratic variation to future work.

For brevity, let us write $s = (b, c, B)$ and $z = (x, y, X, Y)$. Given a change $s \to s + ds$, the solution changes as $z \to z + dz + d^2z + \dots$, where dz and d^2z are linear and quadratic in ds, respectively, and "..." represent higher order terms in ds. The quadratic change in the Lagrange function is

$$\begin{aligned} d^2L &= \frac{\partial L}{\partial z} d^2z + \frac{1}{2}\frac{\partial^2 L}{\partial z^2} dz^2 + \frac{\partial L}{\partial s \partial z} ds\, dz + \frac{1}{2}\frac{\partial^2 L}{\partial s^2} ds^2 \\ &= \frac{1}{2}\frac{\partial^2 L}{\partial z^2} dz^2 + \frac{\partial L}{\partial s \partial z} ds\, dz. \end{aligned} \tag{A.12}$$

Here, s and z are multidimensional and we suppress indices for brevity. The first term on the first line vanishes by the optimality equations $\frac{\partial L}{\partial z} = 0$, and the last term vanishes because L is linear in s. The remaining two terms are proportional to each other. To see this, note that under changing $s \to s + ds$, the shifted optimality equations become

$$\begin{aligned} 0 &= \left.\frac{\partial L(s,z)}{\partial z}\right|_{\substack{z \to z+dz+d^2z+\dots \\ s \to s+ds}} \\ &= \frac{\partial L(s,z)}{\partial z} + \frac{\partial^2 L(s,z)}{\partial s \partial z} ds + \frac{\partial^2 L(s,z)}{\partial z^2} dz \\ &= \frac{\partial^2 L(s,z)}{\partial s \partial z} ds + \frac{\partial^2 L(s,z)}{\partial z^2} dz, \end{aligned} \tag{A.13}$$

Contracting (A.13) with dz and plugging this result into (A.12), we find

$$d^2L = \frac{1}{2}\frac{\partial L}{\partial s \partial z} ds\, dz = \frac{1}{2}(db^T dy + dc^T dx - dx^T dB\, y - x^T dB\, dy). \tag{A.14}$$

The variations dx, dy can be computed from the linearized optimality equations (A.13), which are written in more detail in (1.40). After some rearrangement, we find

$$\begin{pmatrix} S & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -dc + dBy \\ db - dB^T x \end{pmatrix}, \tag{A.15}$$

where $S_{pq} = \mathrm{Tr}(A_p X^{-1} A_q Y)$ is the so-called Schur complement matrix. This is precisely the equation solved by SDPB in its main optimization algorithm, with a modified right-hand side. Consequently, it is straightforward to adapt SDPB to determine dx, dy and compute dL and d^2L. We have implemented this computation

in a program `approx_objective` packaged with SDPB as of version 2.5.[6]

A.3.2 Possible sources of error

We note two possible sources of error in the results for dL and d^2L — one conceptual and one practical:

(E_1) **Finite-μ effects.** The formulas for dL and d^2L were derived assuming finite μ (so that the optimization problem is well-posed). Is the $\mu \to 0$ limit of these expressions well-behaved? How big are the finite-μ corrections?

As with the objective function itself, we expect errors in dL and d^2L to be of order $O(\mu \log \mu)$, provided the SDP is generic. This expectation comes from thinking about L as a function to be optimized $b^T y + c^T x - x^T B y + \mathrm{Tr}\Big((X - x^T A_*)Y\Big)$, plus a barrier function $-\mu \log\det X$ that imposes that X is positive semidefinite. Near a smooth point on the boundary of the positive-semidefinite cone, the barrier function effectively moves the boundary of the cone by a smoothly-varying amount proportional to μ.

As we vary the parameters (b, c, B, A), the optimal solution with $\mu = 0$ moves along the boundary of the positive semidefinite cone. Similarly, the optimal solution with finite μ moves along the "effective" boundary a distance μ away. As long as the boundary is smooth, derivatives of the finite-μ objective will differ from derivatives of the $\mu = 0$ objective by $O(\mu \log \mu)$ (the size of the barrier function).

(E_2) **Errors from $XY \neq \mu I$.** One of the optimality equations (1.50) is $XY = \mu I$. Under normal operation, SDPB does not attempt to solve this equation with high precision. Instead, it performs repeated Newton steps toward solutions of $XY = \mu^{(i)} I$ with values $\mu^{(i)}$ that *change* with each iteration. This turns the equation $XY = \mu I$ into a kind of moving target. Solutions computed by SDPB will generally have nonzero (but small) $XY - \mu I$.

It is not a-priori obvious how large errors resulting from nonzero $XY - \mu I$ will be. (We show a numerical example in figure A.2.) However, they can be mitigated with a simple strategy: After SDPB terminates with a primal-dual optimal solution, we can perform a few extra iterations toward a solution of $XY = \mu I$. In practice, this can be done by running SDPB from the most

[6] We thank Walter Landry for collaboration on `approx_objective`.

option	explanation
`-maxIterations=`n	Control the number of iterations. We take $n = 10$ below.
`-stepLengthReduction=1`	Take full Newton steps instead of decreasing the step size.
`-infeasibleCenteringParameter=1`	Ensure that μ stays (nearly) constant instead of changing $\mu \to \mu$ with each iteration. This option is only effective if SDPB has both a primal- and dual-infeasible internal state.
`-dualityGapThreshold=0`	Ensure a dual-infeasible internal state.
`-primalErrorThreshold=0`	Ensure a primal-infeasible internal state.
`-dualErrorThreshold=0`	Ensure SDPB doesn't terminate early.

Table A.1: SDPB options for performing centering iterations.

recent checkpoint with the options listed in table A.1 (in addition to whatever other options were used in the optimization). Because the locus $XY = \mu I$ is called the "central path," we call these extra iterations "centering iterations."

A.3.3 Numerical checks

To describe our numerical checks of the expressions for dL and d^2L, we need some quick definitions. Given an SDP s, let $f(s)$ be the optimal value of its objective. We also define

$$\begin{aligned} g(s, ds) &\equiv f(s) + dL + d^2L \\ &= f(s) + \frac{\partial L(s,z)}{\partial s} ds + \frac{1}{2} \frac{\partial L(s,z)}{\partial s \partial z} ds\, dz, \end{aligned} \tag{A.16}$$

where z is the optimum of s, and dz (which is linear in ds) is the solution to equation (A.13). Note that g is arbitrarily nonlinear in its first argument, but quadratic in its second argument — in fact, $g(s_0, s - s_0)$ provides a quadratic

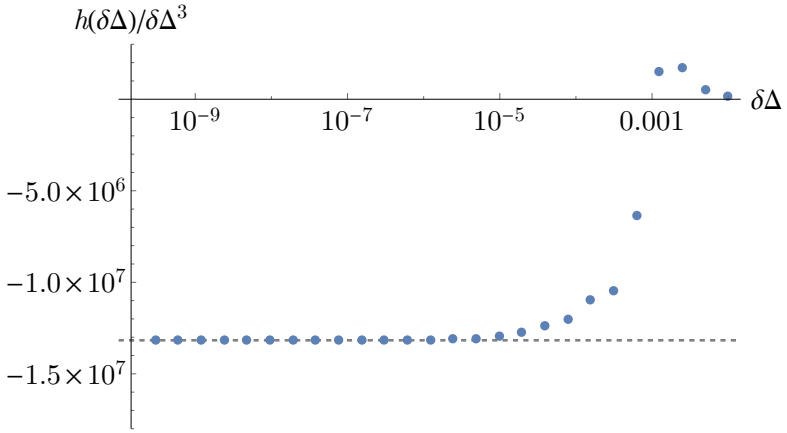


Figure A.1: The ratio $h(\delta\Delta)/\delta\Delta^3$ for a family of SDPs describing σ and ϵ correlators in the 3d Ising model. Specifically, we studied the GFF navigator function in the 2-parameter 3d Ising setup described in section 1.2.2, with fixed $\Delta_\epsilon = 1.4$ and varying $\Delta_\sigma = 0.518 + \delta\Delta$, where $\delta\Delta = 0.01 \times 2^{-n}$ and $n \in \{0, \ldots, 25\}$, and derivative order $\Lambda = 11$. We see that the difference between the true objective and its quadratic approximation is cubic in $\delta\Delta$. The optimizations for this plot were computed with a duality gap threshold of 10^{-30}, and 10 centering iterations.

approximation to $f(s)$ around a given s_0:[7]

$$f(s) = g(s_0, s - s_0) + O((s - s_0)^3). \tag{A.17}$$

Consider now a family of SDP's $s(\Delta)$ depending smoothly on a parameter Δ. Consider a sequence of values $\Delta_0 + \delta\Delta$ converging to Δ_0, and let us write $s_0 = s(\Delta_0)$. Equation (A.17) with $s = s(\Delta_0 + \delta\Delta)$ implies that

$$h(\delta\Delta) \equiv f(s(\Delta_0 + \delta\Delta)) - g(s_0, s(\Delta_0 + \delta\Delta) - s_0) \sim O(\delta\Delta^3), \tag{A.18}$$

where we used that $s(\Delta)$ depends locally smoothly on Δ. We can use this to check our expressions for dL and d^2L: we compute $h(\delta\Delta)$ for several values of $\delta\Delta$ and check whether it decreases cubically in $\delta\Delta$.

In figure (A.1), we plot the ratio $h(\delta\Delta)/\delta\Delta^3$ for a one-parameter family of SDP's describing the GFF navigator function for correlators of σ and ϵ in the 3d Ising model. For small $\delta\Delta$, the ratio $h(\delta\Delta)/\delta\Delta^3$ approaches a constant. This is a strong check of our results for dL and d^2L and our ability to compute them accurately: cubic dependence of $h(\delta\Delta)$ on $\delta\Delta$ requires delicate cancellations between

[7] In other words, g is a 2-jet of f at s_0.

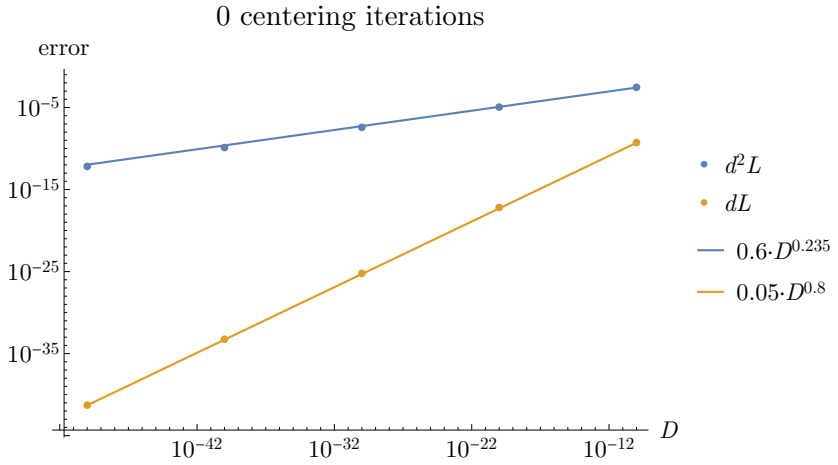


Figure A.2: The relative error in dL and d^2L, as a function of the duality gap D (which is proportional to μ), computed with no centering iterations. We use the setup described in the caption of figure A.1, with $\delta\Delta = 0.01 \times 2^{-25}$. We define relative error for a quantity x by $|x - x_{\text{ref}}|/|x_{\text{ref}}|$, where x_{ref} is a reference value. Reference values for this plot were computed with duality gap 10^{-50} and 30 centering iterations. For both dL and d^2L, we show best fits to powers of D.

the true objectives of $s(\Delta_0+\delta\Delta)$ and s_0, the linear correction dL, and the quadratic correction d^2L. The SDPB computations in figure A.1 were performed with duality gap threshold $D = 10^{-30}$, with 10 centering iterations. Evidently these choices effectively remove both sources of error (E_1) and (E_2) in this example.[8]

In figures A.2 and A.3, we show the effects of (E_1) and (E_2) on dL and d^2L. Figure A.2 was produced with no centering iterations, so it shows the effects of both (E_1) and (E_2). In that case, the relative error in dL scales approximately as $\mu^{0.8}$, and the relative error in d^2L scales as $\mu^{0.235}$. These numbers presumably are not universal: they depend on the whole history of the optimization procedure in SDPB, and are not uniquely determined by the final solution. Figure A.3 was produced with 10 centering iterations. In that case, the errors in dL and d^2L both scale linearly with μ, and are much smaller overall. This is strong evidence that centering iterations effectively mitigate (E_2), and it also supports our estimate of the size of finite-μ effects.

[8] More precisely (E_1) and (E_2) are unimportant for the values of $\delta\Delta$ shown in the plot. They will become important again at smaller values of $\delta\Delta$. To get accurate results for even smaller $\delta\Delta$, we can decrease μ by further lowering the duality gap threshold.

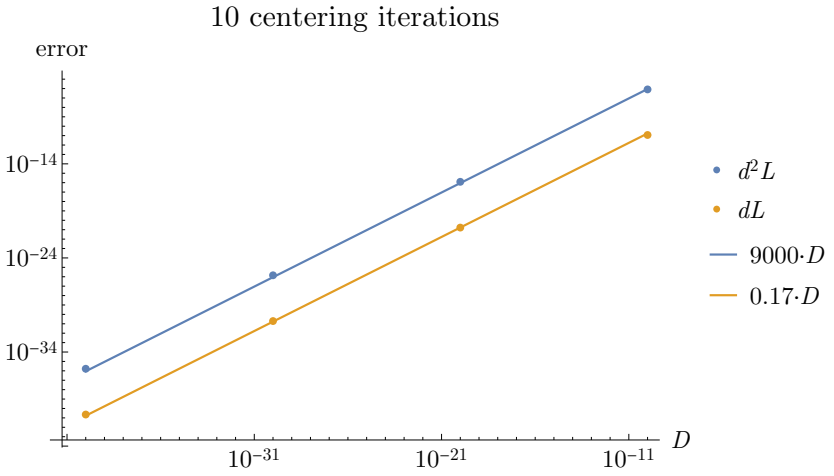


Figure A.3: Errors for dL and d^2L as a function of the duality gap D with the same setup as figure A.3, but where for each optimization we perform 10 centering iterations of SDPB. The errors now decrease linearly with D (which is proportional to μ). This is consistent with our naive estimate $\mu \log \mu$ in section A.3.2. (To detect the logarithm $\log \mu$, we would need more data and a more careful fit.)

A.4 Parameters for numerics

The computation of the navigator function can be translated to the form of a semidefinite program (SDP), to solve which we use the arbitrary precision solver `SDPB` [42, 43]. We used `simpleboot` [150], `PyCFTBoot` [151], and `sdpb-haskell`[9] to setup the SDPs. The parameters used for the computations are presented in Table A.2. We used the same conformal block normalization as [36].

For the $\Lambda = 19$ results in Section 1.5.3, we used the Python package `PyCFTBoot` [151] to setup the SDP, with parameters $(k_{max}, l_{max}, n_{max}, m_{max}) = (28, 28, 1, 9)$. The parameters (n_{max}, m_{max}) control the number of derivatives used in the (a, b) coordinates (see [151] for more details). This choice results in the same navigator value as taking $(z, \bar{z})$ derivatives up to $\Lambda = 19$.

[9] https://gitlab.com/davidsd/sdpb-haskell

[10] The computations presented in Sections 1.5.3 and 1.6 were set up using a version of `simpleboot` where the definition of `keptPoleOrder` was slightly different. Here the poles were kept without modifying the residue to better approximate the contribution of discarded poles and thus the blocks were less accurate than those used in [42].

Section(s)	1.3	1.3	1.5.3, 1.6	1.5.3
Λ	11	19	11	19 (`PyCFTBoot`, see below)
`keptPoleOrder`[10]	8	14	14	
`order`	60	60	27	
`spins`	$\{0,\dots,21\}$	$\{0,\dots,26,49,50\}$	$\{0,\dots,27\}$	$\{0,\dots,28\}$
`precision`	640	768	768	660
`dualityGapThreshold`	10^{-30}	10^{-30}	10^{-20}	10^{-30}
`primalErrorThreshold`	10^{-30}	10^{-30}	10^{-60}	10^{-30}
`dualErrorThreshold`	10^{-30}	10^{-30}	10^{-60}	10^{-30}
`initialMatrixScalePrimal`	10^{20}	10^{40}	10^{20}	10^{20}
`initialMatrixScaleDual`	10^{20}	10^{40}	10^{20}	10^{20}
`feasibleCenteringParameter`	0.1	0.1	0.1	0.1
`infeasibleCenteringParameter`	0.3	0.3	0.3	0.3
`stepLengthReduction`	0.7	0.7	0.7	0.7
`maxComplementarity`	10^{100}	10^{100}	10^{100}	10^{100}

Table A.2: Parameters used to setup the SDPs, along with the `SDPB` parameters. The definition of these can be found in [42] (where `order` was 90 and `keptPoleOrder` was κ).

To numerically implement the BFGS Algorithm 1.1, we have used the BFGS algorithm `minimize(method='BFGS')` of Python's SciPy library, with the additional modifications of the rescaling of the initial Hessian and the implementation of the bounding box. All parameters used were the default ones, both for the Moré and Thuente line search SciPy implements and the actual BFGS algorithm.

A.5 Further plots

Here we collect plots like Figs. 1.7 and 1.8 for six additional runs of our modified BFGS algorithm, for both the two parameter $\Lambda = 11$ case discussed in Sec. 1.5.3, and the three parameter $\Lambda = 19$ case discussed in Sec. 1.5.3

A.5.1 2-parameter searches

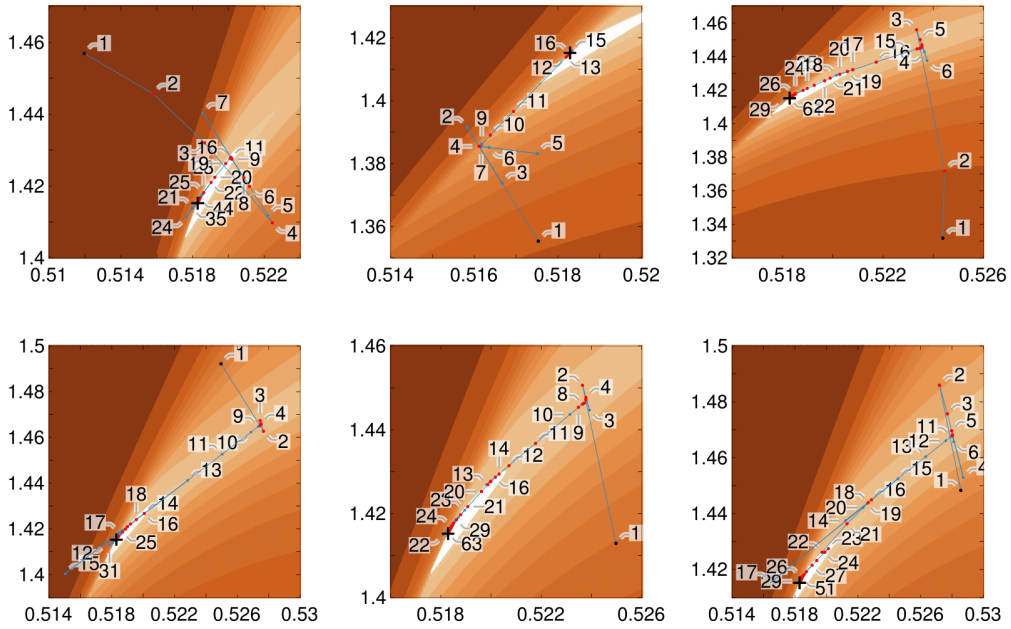


Figure A.4: Six more runs of our algorithm, see Section 1.5.3, in addition to the run shown in Fig. 1.7. Plotting conventions are the same as in that figure.

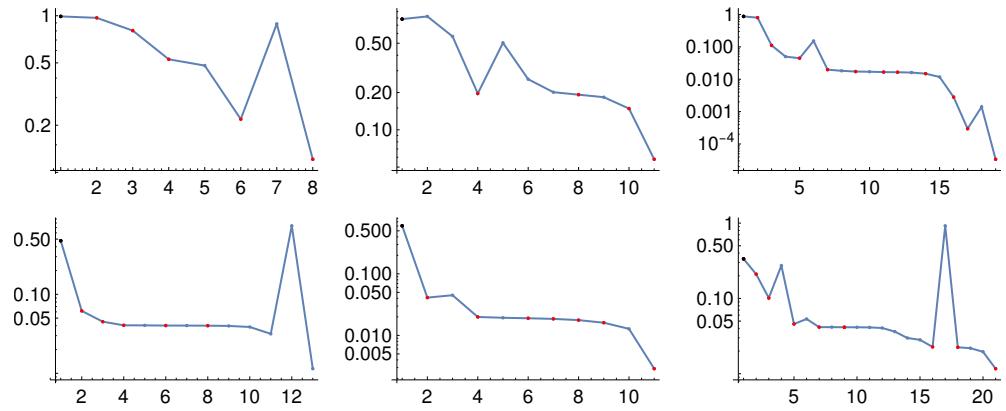


Figure A.5: This plot is analogous to Fig. 1.8(left). It shows navigator values $\mathcal{N}_i$ at the i-th function call for the 6 runs from Fig. A.4, and with the same color code for the dots.

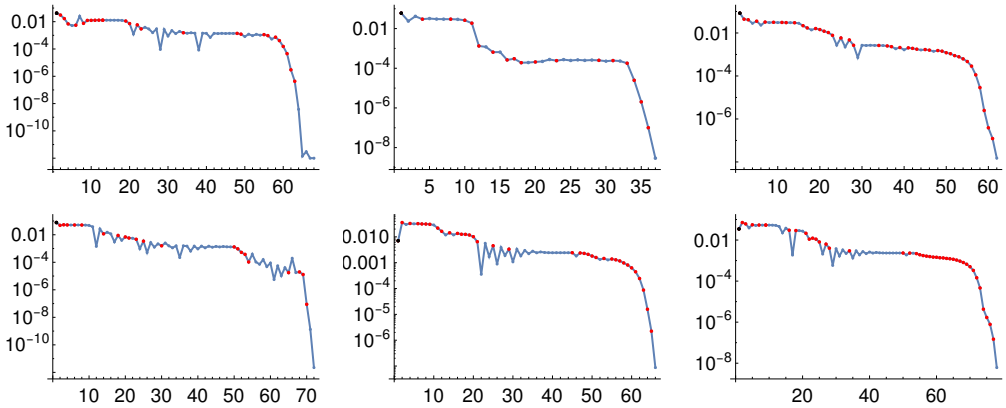


Figure A.6: This plot is analogous to Fig. 1.8(right). It shows logarithmic plots of $\|x_i - x_f\|$ at the i-th function call for the 6 runs from Fig. A.4, and with the same color code for the dots.

A.5.2 3-parameter searches

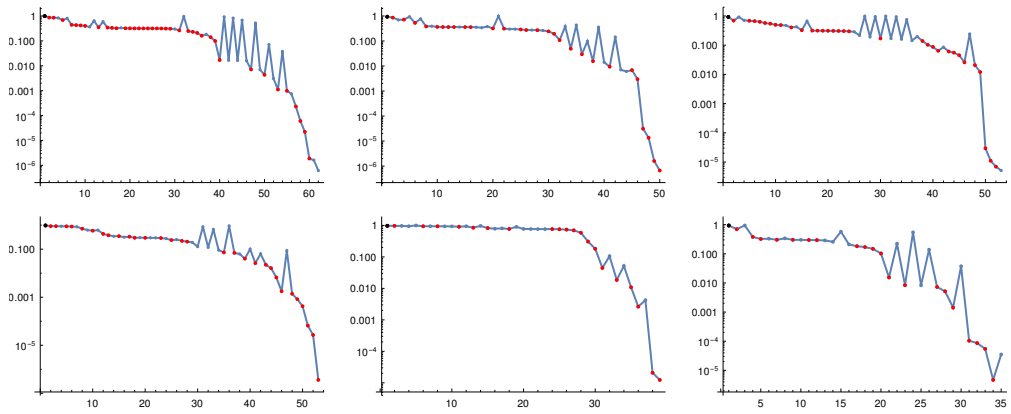


Figure A.7: Same as Fig. 1.12(left), for 6 additional runs appearing in Fig. 1.10. The figure shows navigator values $\mathcal{N}_i$ at the i-th function call for the 6 additional runs, with the same color code for the dots as in Fig. 1.12 .

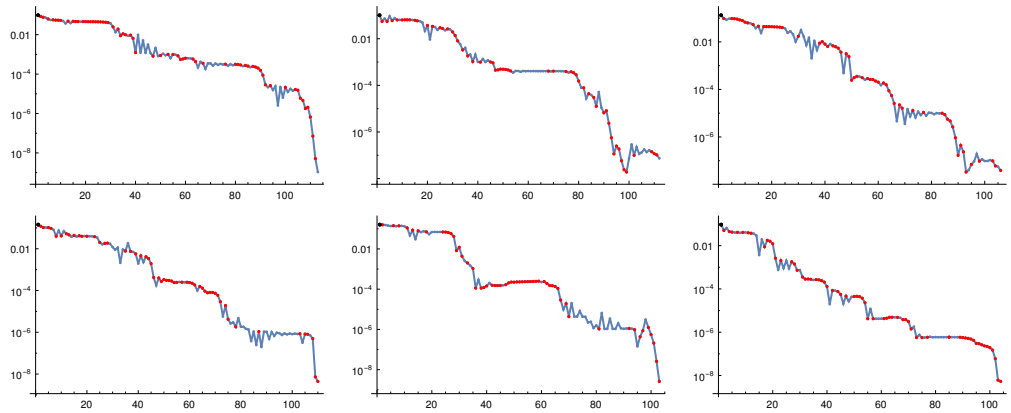


Figure A.8: Same as Fig. 1.12(right), for 6 additional runs appearing in Fig. 1.10. The figure shows logarithmic plots of $\|x_i - x_f\|$ at the i-th function call for the 6 additional runs, with the same color code for the dots as in Fig. 1.12.

Appendix B

Appendix to Chapter 2

B.1 Resummation algorithm

To determine the value of a critical exponent C^* in dimension $d = 4 - \epsilon^*$ from its ϵ-expansion $C(\epsilon) = \sum_{k=0}^{\infty} c_k \epsilon^k$, one can do the following: first, compute the Borel-Leroy transform of $C(\epsilon)$

$$C_B^b(t) = \sum_{k=0}^{\infty} \frac{c_k}{\Gamma(k+b+1)} t^k \quad , \tag{B.1}$$

The gamma function kills the standard factorial growth of the coefficients c_k, resulting in a series with finite radius of convergence. Next, find an analytic continuation $\widetilde{C_B}(t)$ of $C_B^b(t)$ defined for all positive t. [46] uses conformal mapping to get such an analytic continuation: under the change of variables

$$t(w) = \frac{4w}{a(1-w)^2} \quad , \tag{B.2}$$

with $a = \frac{3}{N+8}$ for the $O(N)$ model, the positive real line in t is mapped to the interval $(0, 1)$ in w. The analytic continuation is simply the re-expansion of the original Borel transform as a series in the new variable w, which should converge for all $|w| < 1$. Finally, C^* is given by the Borel sum of the analytic continuation:

$$C^* = \int_0^{\infty} \mathrm{d}t \, t^b e^{-t} \, \widetilde{C_B}(\epsilon^* t) \quad . \tag{B.3}$$

Since we only have access to coefficients c_k up to some k_{max}, critical exponents are computed using the *approximate* analytic continuation

$$\widetilde{\mathcal{C}}_B^{b,\lambda,q}(t) = \left(\frac{at}{w(t)}\right)^{\lambda} \sum_{k=0}^{k_{max}} B^{b,\lambda}_{(C,q),k} \left(w(t)\right)^k \quad , \tag{B.4}$$

where coefficients $B^{b,\lambda}_{f,k}$ for a series f are chosen to reproduce the known series expansion of f to order k_{max}. This approximate analytic continuation depends on 3 parameters (b, λ, q). b is the Leroy parameter of the Borel-Leroy transform Eq. (B.1), and λ governs the growth of $\widetilde{\mathcal{C}}_B^{b,\lambda,q}(t)$ at large t. The q in the subscript (C, q) indicates that before Borel transform and conformal mapping, the original series was expanded in terms of a new variable ϵ', related to ϵ by the so-called homographic transformation $\epsilon' = \frac{\epsilon}{1-q\epsilon}$ first introduced in [152].

The value of a critical exponent which is reported is that for the specific choice of (b, λ, q) which minimizes the following error, given in (36) of [46]:

$$\begin{aligned}\mathrm{Err}^{C^*}_{k_{max}}(b,\lambda,q) = \max\{\left|C^{*b,\lambda,q}_{k_{\max}} - C^{*b,\lambda,q}_{k_{\max}-1}\right|, \left|C^{*b,\lambda,q}_{k_{\max}} - C^{*b,\lambda,q}_{k_{\max}-2}\right|\}+ \\ \max\{\mathrm{Var}_{b,\Delta_b}\left(C^{*b,\lambda,q}_{k_{\max}}\right), \mathrm{Var}_{b,\Delta_b}\left(C^{*b,\lambda,q}_{k_{\max}-1}\right)\} + \mathrm{Var}_{q,\Delta_q}\left(C^{*b,\lambda,q}_{k_{\max}}\right) + \mathrm{Var}_{\lambda,\Delta_\lambda}\left(C^{*b,\lambda,q}_{k_{\max}}\right)\end{aligned} \quad . \tag{B.5}$$

This error quantifies both the stability of the result with respect to the loop order (the first term) and the stability with respect to a variation of the parameters (the last three terms). For each parameter p, this last measure of stability, denoted $\mathrm{Var}_{p,\Delta_p}$, is given by the minimum spread of C^* in an interval of size Δ_p which includes p. Finally, if we call the minimal error E_{opt}, the uncertainty reported is this error, plus two times the standard deviation of all critical exponent estimates for which the error is at most $3E_{\mathrm{opt}}$. This should result in a more realistic uncertainty in cases where many resummations are close to optimal.

B.2 Numerical Implementation

The crossing vectors in the crossing equation (2.4) are given by [84]

$$\vec{V}_{V,\Delta,\ell} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ F^{\phi s,\phi s}_{-,\Delta,\ell} \\ (-1)^\ell F^{s\phi,\phi s}_{-,\Delta,\ell} \\ -(-1)^\ell F^{s\phi,\phi s}_{+,\Delta,\ell} \end{pmatrix}, \quad \vec{V}_{T,\Delta,\ell} = \begin{pmatrix} F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} \\ \left(1-\frac{2}{N}\right) F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} \\ -\left(1+\frac{2}{N}\right) F^{\phi\phi,\phi\phi}_{+,\Delta,\ell} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{V}_{A,\Delta,\ell} = \begin{pmatrix} F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} \\ -F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} \\ F^{\phi\phi,\phi\phi}_{+,\Delta,\ell} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\vec{V}_{S,\Delta,\ell} = \begin{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} F^{\phi\phi,\phi\phi}_{+,\Delta,\ell} & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & F^{ss,ss}_{-,\Delta,\ell} \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{2}F^{\phi\phi,ss}_{-,\Delta,\ell} \\ \frac{1}{2}F^{\phi\phi,ss}_{-,\Delta,\ell} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{2}F^{\phi\phi,ss}_{+,\Delta,\ell} \\ \frac{1}{2}F^{\phi\phi,ss}_{+,\Delta,\ell} & 0 \end{pmatrix} \end{pmatrix}. \tag{B.6}$$

We have suppressed above the dependence of the convolved blocks $F^{ij,kl}_{\pm,\Delta,\ell}(u,v)$ on the cross-ratios (u,v). For their exact expression, see e.g. (2.4) of [84]. S,T,A,V

refer respectively to the singlet, two-index traceless-symmetric, two-index antisymmetric and vector representations of $O(N)$ which are present in either the $\phi_i \times \phi_j$ OPE (S, T, A), the $\phi_i \times s$ OPE (V) or the $s \times s$ OPE (S).

To construct the GFF navigator function $\mathcal{N}^{\text{GFF}}(x)$, we can add to Eq. (2.4) a contribution $\lambda \vec{M}_{GFF}$ corresponding to the operators whose dimensions are below the gaps we assumed in a solution where ϕ_i is an $O(N)$ generalized free field (GFF) and where s is another independent GFF, and minimize λ [1]. The OPEs for this $O(N)$ mixed-correlator GFF solution are

$$\begin{aligned}
\phi_i \times \phi_j = \delta_{ij} \sum_{\ell \text{ even}} \sum_{n \in \mathbb{Z}_{\geqslant 0}} \lambda^{\text{S}}_{\phi\phi(n\ell)} \text{“ } \phi_k \Box^n \partial^\ell \phi_k \text{ ”} + \sum_{\ell \text{ even}} \sum_{n \in \mathbb{Z}_{\geqslant 0}} \lambda^{\text{T}}_{\phi\phi(n\ell)} \text{“ } \phi_{(i} \Box^n \partial^\ell \phi_{j)} \text{ ”} + \\
\sum_{\ell \text{ odd}} \sum_{n \in \mathbb{Z}_{\geqslant 0}} \lambda^{\text{A}}_{\phi\phi(n\ell)} \text{“ } \phi_{[i} \Box^n \partial^\ell \phi_{j]} \text{ ”} \\
\phi_i \times s = \sum_{\ell} \sum_{n \in \mathbb{Z}_{\geqslant 0}} \lambda^{\text{V}}_{\phi s(n\ell)} \text{“ } \phi_i \Box^n \partial^\ell s \text{ ”} \\
s \times s = \sum_{\ell \text{ even}} \sum_{n \in \mathbb{Z}_{\geqslant 0}} \lambda^{\text{S}}_{ss(n\ell)} \text{“ } s \Box^n \partial^\ell s \text{ ”} \quad ,
\end{aligned} \tag{B.7}$$

where the various dimensions and OPE coefficients are given by [153, 154]

$$\Delta_{\text{“}A \Box^n \partial^\ell B\text{”}} = \Delta_A + \Delta_B + 2n + \ell \tag{B.8}$$

$$\begin{aligned}
\left(\lambda^{\text{T}}_{\phi\phi(n\ell)}\right)^2 = \left(\lambda^{\text{A}}_{\phi\phi(n\ell)}\right)^2 = c_{n,\ell}\left(\Delta_\phi, \Delta_\phi\right) \\
\left(\lambda^{\text{S}}_{\phi\phi(n\ell)}\right)^2 = \frac{2}{N} c_{n,\ell}\left(\Delta_\phi, \Delta_\phi\right) \\
\left(\lambda^{\text{S}}_{ss(n\ell)}\right)^2 = 2\, c_{n,\ell}\left(\Delta_s, \Delta_s\right) \\
\left(\lambda^{\text{V}}_{\phi s(n\ell)}\right)^2 = c_{n,\ell}\left(\Delta_\phi, \Delta_s\right) \quad ,
\end{aligned} \tag{B.9}$$

$$\begin{aligned}
c_{n,\ell}\left(\Delta_1, \Delta_2\right) = \frac{2^\ell (\Delta_1 - \frac{d-2}{2})_n (\Delta_2 - \frac{d-2}{2})_n}{\ell! n! (\frac{d-2}{2} + \ell + 1)_n (\Delta_1 + \Delta_2 + n - d + 1)_n} \times \\
\frac{(\Delta_1)_{\ell+n} (\Delta_2)_{\ell+n}}{(\Delta_1 + \Delta_2 + 2n + \ell - 1)_\ell (\Delta_1 + \Delta_2 + n + \ell - \frac{d}{2})_n}
\end{aligned} \tag{B.10}$$

[1] Another form of navigator function, coined the Σ-*navigator*, was proposed in [44], which amounts to adding a different contribution to the crossing equation that still ensures the augmented crossing equation always has a solution.

with $(\cdot)_x$ the Pochhammer symbol and the differences between Eq. (B.10) and its equivalents in [153, 154] are due to different conformal block normalizations.

Computing the navigator function with all assumptions given in Section 2.3 amounts to solving the following optimization problem:

$$\mathcal{N}(x) = \max_{\vec{\alpha}} \quad \begin{pmatrix} 1 & 1 \end{pmatrix} \vec{\alpha} \cdot \vec{V}_{S,0,0} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{such that} \tag{B.11}$$

$$\vec{\alpha} \cdot \vec{M}_{GFF} = -1$$

$$\vec{\alpha} \cdot (\vec{V}_{S,\Delta_s,0} + \vec{V}_{V,\Delta_\phi,0} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}) \succcurlyeq 0$$

$$\vec{\alpha} \cdot \vec{V}_{T,\Delta_t,0} \geqslant 0$$

$$\vec{\alpha} \cdot \vec{V}_{A,d-1,1} \geqslant 0$$

$$\vec{\alpha} \cdot \vec{V}_{S,d,2} \succcurlyeq 0$$

$$\vec{\alpha} \cdot \vec{V}_{V,\Delta,\ell} \geqslant 0 \quad \begin{cases} \Delta \geqslant \Delta^*_\phi & \ell = 0 \\ \Delta \geqslant d + \ell - 2 + \tau & \ell > 0 \end{cases}$$

$$\vec{\alpha} \cdot \vec{V}_{S,\Delta,\ell} \succcurlyeq 0 \quad \begin{cases} \Delta \geqslant \Delta^*_s & \ell = 0 \\ \Delta \geqslant \Delta^*_{T_{\mu\nu}} & \ell = 2 \\ \Delta \geqslant d + \ell - 2 + \tau & \ell > 2 \end{cases}$$

$$\vec{\alpha} \cdot \vec{V}_{T,\Delta,\ell} \geqslant 0 \quad \begin{cases} \Delta \geqslant \Delta^*_t & \ell = 0 \\ \Delta \geqslant d + \ell - 2 + \tau & \ell > 0 \end{cases}$$

$$\vec{\alpha} \cdot \vec{V}_{A,\Delta,\ell} \geqslant 0 \quad \begin{cases} \Delta \geqslant \Delta^*_{J_\mu} & \ell = 1 \\ \Delta \geqslant d + \ell - 2 + \tau & \ell > 1 \end{cases}$$

where an example of a valid $\vec{M}_{GFF}$ will be given shortly. As stated in Section 2.3, we solve this problem with `SDPB 2.4.0` [42, 43]. We compute conformal blocks with `scalar_blocks` [155], and use `simpleboot` [150] as our user-interface in setting up the bootstrap problem. The parameter Λ which was said to govern the size

of allowed regions in Section 2.3 is simply the order to which we Taylor expand the functionals in $\vec{\alpha}$.

Minimization of the navigator using the modified BFGS algorithm of [44] enables us to locate allowed regions. This algorithm uses gradient information. It therefore greatly benefits from the fact that the gradient of the solution to the SDP associated to Eq. (B.11) may be computed "for free", i.e. only knowing the solution of the SDP at the point where the gradient is demanded, and the variations of the SDP. The exact formula for this gradient is given in Section 4.2 of [44]. The BFGS algorithm requires a bounding box $\mathfrak{B} = [\Delta_\phi^{\min}, \Delta_\phi^{\max}] \times [\Delta_s^{\min}, \Delta_s^{\max}] \times [\Delta_t^{\min}, \Delta_t^{\max}]$ inside of which the search will take place. This bounding box has two purposes: it both prohibits BFGS from flowing into allowed regions which have nothing to do with the $O(N)$ models (referred in the bootstrap jargon as the "peninsulas") and provides a scale for the initial hessian B_0. Given the bounding box $\mathfrak{B}$, we perform the change of variables $x \to y(x)$ defined by

$$x_i = x_{0;i} + (\Delta_{x_i}^{\max} - \Delta_{x_i}^{\min}) \cdot r \cdot y_i \quad . \tag{B.12}$$

Then when an approximate Hessian is not otherwise known, we use

$$B_0 = \left\| \nabla \tilde{f}(0) \right\| \mathbb{1} \quad , \tag{B.13}$$

where $\tilde{f}(y) = f(x)$, as the initial Hessian. This is completely equivalent to the prescription for the initial Hessian given in Algortihm 1 of [44]. r of Eq. (B.12) gives the ratio of the bounding box that we wish to explore in each direction in the first step. It was set to 0.2 in [44]. We throughout this chapter use the smaller $r = 0.05$ as we usually stay close to the islands we're looking for, and don't need to explore too much of the parameter space. BFGS terminates once the maximum norm ($\|a\|_\infty = \max(|a_1|, |a_2|, \ldots)$) of the gradient of the transformed function $\tilde{f}$ reaches a cutoff $\mathrm{g_{tol}}$, where we chose $\mathrm{g_{tol}} = 10^{-8}$ [2]. This is the standard termination criteria of the SciPy [74] implementation of BFGS we were using. Finally, to actually compute $\mathcal{N}(x)$, we have to define a valid GFF contribution vector $\vec{M}_{GFF}$. If we defined it at every x as the contribution from all GFF vectors with dimensions below our gaps $\vec{\Delta}^*$, we would run the risk that it might change as Δ_ϕ and Δ_s vary within a BFGS run, resulting in a discontinuous change of the navigator function (remember that the dimensions of those GFF operators are

[2] The change of variables Eq. (B.12) obviously impacts when this termination criterion is reached.

given by Eq. (B.8)). To make sure that we have at least all the operators we need (and maybe some superfluous ones) throughout the BFGS run, we choose $\vec{M}_{GFF}$ to be the contribution from the minimal set of operators required for $(\Delta_\phi, \Delta_s) = (\Delta_\phi^{\min}, \Delta_s^{\min})$. For the example of Section 2.3, this means we take

$$\vec{M}_{GFF} = \frac{1}{2} c_{0,0}\left(\Delta_\phi, \Delta_\phi\right) \begin{pmatrix} 1 & 0 \end{pmatrix} \vec{V}_{S,2\Delta_\phi,0} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_{0,0}\left(\Delta_s, \Delta_s\right) \begin{pmatrix} 0 & 1 \end{pmatrix} \vec{V}_{S,2\Delta_s,0} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \frac{1}{2} c_{0,2}\left(\Delta_\phi, \Delta_\phi\right) \begin{pmatrix} 1 & 0 \end{pmatrix} \vec{V}_{S,2\Delta_\phi+2,2} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{2} \left(c_{0,0}\left(\Delta_\phi, \Delta_\phi\right) \vec{V}_{T,2\Delta_\phi,0} + c_{0,1}\left(\Delta_\phi, \Delta_\phi\right) \vec{V}_{A,2\Delta_\phi+1,1} + c_{0,0}\left(\Delta_\phi, \Delta_s\right) \vec{V}_{V,\Delta_\phi+\Delta_s,0} \right) . \tag{B.14}$$

We of course could have multiplied $\vec{M}_{GFF}$ with any positive prefactor. This choice would influence the value of $\mathcal{N}_{max}$, and with the one made here, $\mathcal{N}_{max} = 2$.

The "Constrained BFGS" algorithm of the end of Section 2.3 works in many ways similarly to the BFGS algorithm described above. However, its line searches are forced to remain close to the boundary of the island, and the algorithm terminates once $|\mathcal{N}(x)|$ or the component of the gradient of $\mathcal{N}(x)$ perpendicular to the extremization direction go below some tolerance g_{tol}. We chose for the example $g_{\text{tol}} = 10^{-10}$. The algorithm also requires a starting point inside of the island and an initial guess for the Hessian. We chose for the example in Section 2.3 the minimum x_{min} of the BFGS run in Fig. 2.1 as the starting point and the approximate Hessian supplied by BFGS at this minimum point as the initial Hessian. Fig. B.1 shows the path taken for the minimization of Δ_ϕ, one of the six extremization runs.

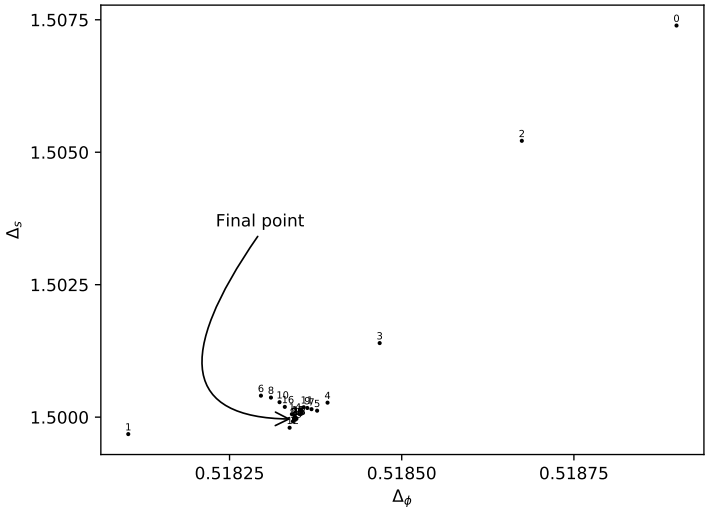


Figure B.1: (Δ_ϕ, Δ_s) projection of the "Constrained BFGS" run for the minimization of Δ_ϕ at $\Lambda = 19$. The run terminates on the tip of the arrow at $x_f = (0.518344, 1.49996, 1.23104)$. The "Constrained BFGS" algorithm allows for the probing of disallowed points in its search for extremal allowed values; e.g. point 1 here has a navigator of 1.22824.

B.3 Further tests of the navigator method

As already noticed in [151] for the case of the Ising model, bootstrap islands tend to decrease significantly in size as d gets closer to 4. One might fear that the navigator method would struggle to find such small islands, even more so because good initial guesses for CFT-data (e.g. ϵ-expansions to high loop-orders) can be rare, and especially rare for OPE coefficients. To prove that the navigator method can handle all of these concerns, let us consider a slightly more constraining setup than (B.11) : we want our navigator to now also depend on the OPE angle $\theta = \arctan \frac{\lambda_{sss}}{\lambda_{\phi\phi s}}$. This means we are changing the second condition of (B.11) to

$$\begin{pmatrix} \cos\theta & \sin\theta \end{pmatrix} \vec{\alpha} \cdot \left(\vec{V}_{S,\Delta_s,0} + \vec{V}_{V,\Delta_\phi,0} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \geqslant 0 \tag{B.15}$$

For $N = 2$ and $\epsilon = 0.2$, we minimized (2.5) over the 4-parameter search space $x = (\Delta_\phi, \Delta_s, \Delta_t, \theta)$, with the initial guess for θ its free theory value $\theta = \arctan 2$, and the lowest non-trivial order estimates of Eq. (2.2) for the dimensions. We found that even with these very rough starting points, BFGS was eventually able to reach the allowed region, taking 81 function calls to find the first allowed point $x = (0.900470379, 1.8847650, 1.8419186, 1.0753019)$. We show a 3D projection of this BFGS run in Fig. B.2, where the tiny allowed region on the right figure lies on the tip of the arrow on the left figure. For the rest of this appendix, including in Fig. B.2, we will substitute for the scaling dimensions the more natural anomalous dimensions $(\gamma_\phi, \gamma_s, \gamma_t) = (\Delta_\phi, \Delta_s, \Delta_t) - (\frac{d-2}{2}, d-2, d-2)$.

Fig. B.2 clearly demonstrates the power of the navigator method: the very tiny island could be located in a relatively small number of steps, starting from quite a rough estimate (notice especially the different scales for the θ variable in the two plots: the island in this direction is roughly two orders-of-magnitude smaller than its distance to the starting point). With the amount of information used here, a scan would have needed to be extremely fine in the θ direction to locate an allowed point. A rough estimate gives $10 \times 10 \times 10 \times 100 = 100{,}000$ points needed to be tested in order to find the island.

Let's now go back to the original setup of (B.11). At small values of ϵ, the islands are so small that using the location of the BFGS minimum at the previous d is not sufficient for the run at $d-0.1$ to converge to the allowed island. We see in Fig. B.3 that for $N = 2$, if one was to start at $d = 3.7$ from the minimum obtained

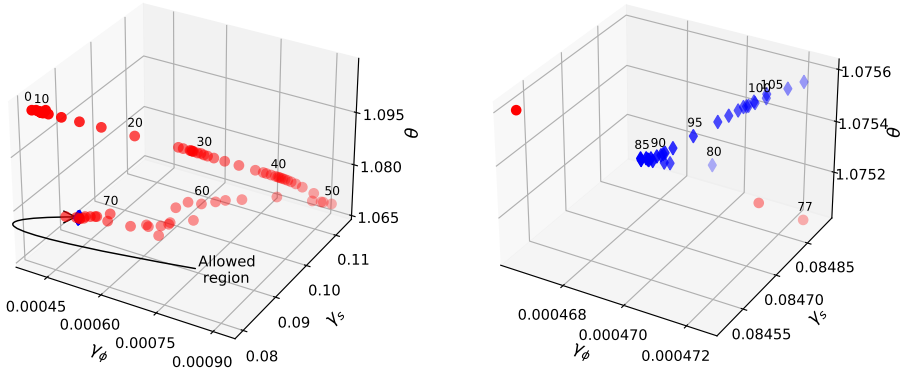


Figure B.2: 3D projection of the BFGS run at $\Lambda = 19$ for the 4-parameter setup where $x = (\Delta_\phi, \Delta_s, \Delta_t, \theta)$. Red points are disallowed, blue diamonds are allowed. Left: Full run. The allowed region is located at the tip of the arrow and is barely visible. Points are labelled by their function call number. Right: Zoom in on the allowed region.

at $d = 3.8$, BFGS would run towards the peninsula to the right of the island. With the initial guess provided by Eq. (2.7), as evidenced in Fig. B.4, the run at $d = 3.7$ reaches the first allowed point $x = (0.00110929, 0.128838, 0.0633290)$ inside the $O(2)$ island after 64 function calls. A prescription like that of Section 2.4 was therefore necessary in making the navigator method viable in this context. However, the initial point $x_0 = (0.000975890, 0.129421, 0.0637559)$ resulted in a navigator of $\mathcal{N}(x_0) = 1.99661$, indicating that this point was deep in the disallowed region. This is much different to the example given in Section 2.3, and the difference is attributable to the small of size of the small-ϵ islands. Because of this, we had to make no assumption about the initial Hessian (using the Hessian at the previous minimum would be a bad guess if the starting point is not close enough to the next minimum), which partly explains why it took a considerable amount of function calls to even reach the island.

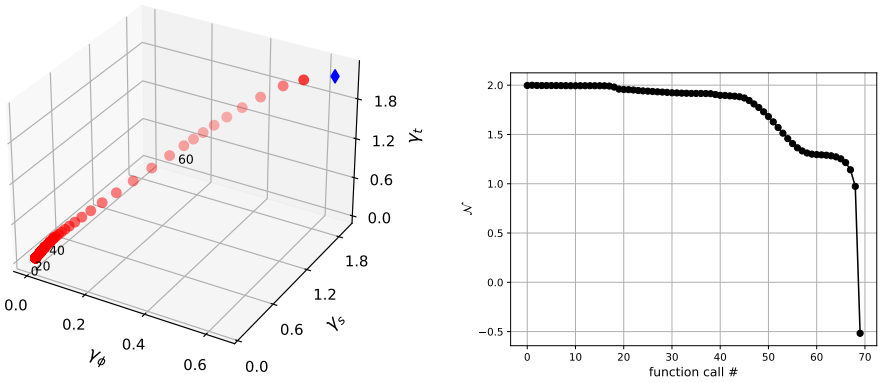


Figure B.3: BFGS run for $N = 2$, $d = 3.7$ at $\Lambda = 19$ using as the initial guess the minimum of the run at $d = 3.8$. Red points are disallowed, blue diamonds are allowed. From the monotonic increase of $\gamma_\phi, \gamma s$ and γ_t to large values, we see that this run misses the (tiny) island and runs off to the peninsula.

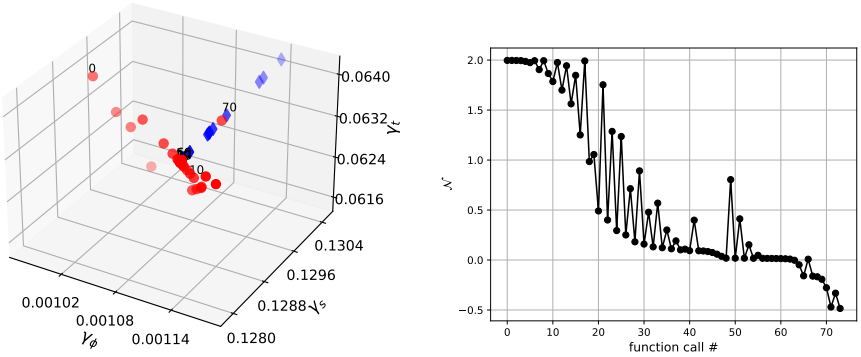


Figure B.4: BFGS run for $N = 2$, $d = 3.7$ at $\Lambda = 19$ with the initial guess supplied by Eq. (2.7). Red points are disallowed, blue diamonds are allowed. This run does converge to the island.

B.4 Further details on the $N \to 1$ limit

As already noted in [115], the full system of crossing equations at $N = 1$ contains the Ising crossing equations, where the $O(N = 1)$ V and S sectors correspond respectively to the Ising $\mathbb{Z}_2$-odd and $\mathbb{Z}_2$-even sectors. Indeed, one concludes from the crossing vectors (B.6) that the lines $\{1 + 2, 4, 5, 6, 7\}$ in the V and S crossing vectors for $N = 1$ are simply those of the Ising mixed-correlator $\mathbb{Z}_2$-odd and $\mathbb{Z}_2$-even vectors, as given e.g. in (3.12) of [45]. This combination of lines renders the contribution from the A and T sectors identically zero, completing the identification to the Ising crossing equations. The fact that the crossing vectors separate into "Ising + rest" implies a similar separation of the navigator problem. Indeed, our $O(N)$ navigator problem

$$\min_{(\Delta_\phi,\Delta_s,\Delta_t)} \lambda \quad | \quad \sum_{\mathcal{R}\in\{V,S,A,T\}} \sum_{\mathcal{O}\in\mathcal{R}} \vec{\lambda}^{\intercal}_{\mathcal{O}_\mathcal{R}} \cdot \vec{V}_{\mathcal{R},\Delta_\mathcal{O},\ell_\mathcal{O}} \cdot \vec{\lambda}_{\mathcal{O}_\mathcal{R}} = -\lambda \vec{M}_{\text{GFF}} \tag{B.16}$$

becomes at $N = 1$

$$\min_{(\Delta_\phi,\Delta_s,\Delta_t)} \lambda \quad \text{such that}$$
$$\begin{cases} \sum_{\mathcal{R}\in\{V,S\}} \sum_{\mathcal{O}\in\mathcal{R}} \vec{\lambda}^{\intercal}_{\mathcal{O}_\text{R}} \cdot \vec{V}^{\mathbb{Z}_2}_{\mathcal{R},\Delta_\mathcal{O},\ell_\mathcal{O}} \cdot \vec{\lambda}_{\mathcal{O}_\mathcal{R}} = -\lambda \vec{M}^{\mathbb{Z}_2}_{\text{GFF}} \\ \sum_{\mathcal{R}\in\{V,S,A,T\}} \sum_{\mathcal{O}\in\mathcal{R}} \vec{\lambda}^{\intercal}_{\mathcal{O}_\text{R}} \cdot \vec{\tilde{V}}_{\mathcal{R},\Delta_\mathcal{O},\ell_\mathcal{O}} \cdot \vec{\lambda}_{\mathcal{O}_\mathcal{R}} = -\lambda \vec{\tilde{M}}_{\text{GFF}} \end{cases} . \tag{B.17}$$

Crossing vectors with a $\mathbb{Z}_2$ superscript are 5-component vectors made up of lines $\{1 + 2, 4, 5, 6, 7\}$ of the full $O(N = 1)$ vectors. Crossing vectors with a tilde are two-component vectors made up of the rest of the full $O(N = 1)$ vectors (so for example, lines 1 and 3). The first condition states that the V and S sectors should solve the Ising crossing equation augmented by a GFF contribution $\lambda \vec{M}^{\mathbb{Z}_2}_{\text{GFF}}$ containing all V and S GFF operators below the gap assumptions of (B.11). If the first condition was the only condition, the navigator would be independent of Δ_t, and equal to the Ising navigator of [44] $\mathcal{N}^{\mathbb{Z}_2}(\Delta_\sigma, \Delta_\epsilon)$ under the identifications $(\Delta_\phi \leftrightarrow \Delta_\sigma,\ \Delta_s \leftrightarrow \Delta_\epsilon)$ if identical assumptions were made on the equivalent sectors. Because the tilde vectors are non-zero in the A and T sectors, we expect that the second condition of (B.17) will cause the navigator to depend on the assumptions made in these sectors. In particular, we will have some dependence of the navigator on Δ_t if Δ_t is to be constrained to a small finite region of allowed values at $N = 1$, as suggested by the limit $N \to 1$ in Fig. 2.8.

What we observe numerically is that for a given (Δ_ϕ, Δ_s), there is some range

of Δ_t where the navigator is flat. As expected, for that range, (B.17) effectively reduces only to minimization over the Ising condition. Indeed, in minimizing the transformed navigator at $N = 1$, the navigator function was computed at 45 points, some determined to be allowed and some disallowed, and actually the gradients of the navigator function for all those points were found to be zero (to our numerical precision) in the Δ_t direction[3]. We have checked for a number of those points that the values of the navigator function match exactly those obtained in the pure Ising navigator setup of [44] when equivalent assumptions were made between the V and $\mathbb{Z}_2$-odd and the S and $\mathbb{Z}_2$-even sectors. One should however not conclude that the navigator function never depends on Δ_t. For example, its dependence on Δ_t for a certain (Δ_ϕ, Δ_s) is shown in Fig. B.5. There is some small range where the navigator is constant at its Ising value, and then it increases once it leaves this region. This picture makes sense: for any given $(\Delta_\phi, \Delta_s, \Delta_t)$, we have

$$\mathcal{N}(\Delta_\phi, \Delta_s, \Delta_t) = \lambda_{\text{min}}^{O(N=1)} \geqslant \lambda_{\text{min}}^{\mathbb{Z}_2} = \mathcal{N}^{\mathbb{Z}_2}(\Delta_\phi, \Delta_s) \quad , \tag{B.18}$$

since from (B.17), every solution of the $O(N = 1)$ augmented crossing equations gives a V and S sector that solves the Ising augmented crossing equations. Furthermore, the equality in Eq. (B.18) is reached if and only if there is a solution of the second condition of (B.17) with $\lambda = \lambda_{\text{min}}^{\mathbb{Z}_2}$, where the V and S sectors solve the first condition also with $\lambda = \lambda_{\text{min}}^{\mathbb{Z}_2}$. In other words, the $O(N = 1)$ navigator is equal to the Ising navigator if and only the second condition of (B.17) can be solved with $\lambda = \lambda_{\text{min}}^{\mathbb{Z}_2}$ for a spectrum with V and S sectors that solve the pure Ising navigator problem.

[3]For example, $\nabla\mathcal{N}(0.518127\ldots, 1.41243\ldots, 1.26559\ldots) = (-16685.8, 1311.79, 1.14678 \times 10^{-39})$.

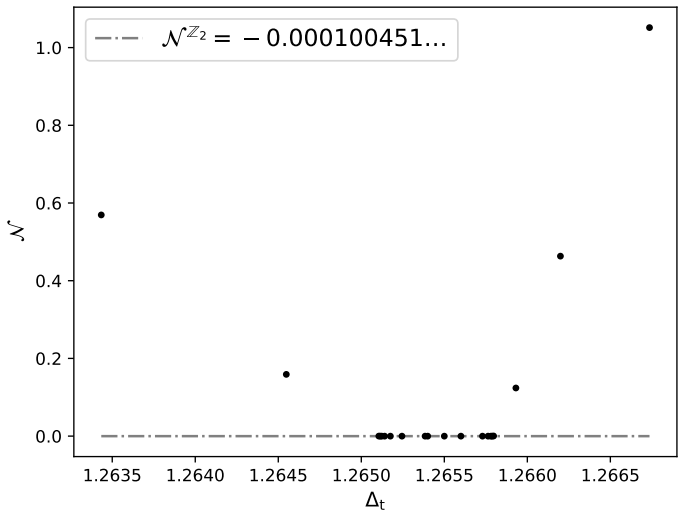


Figure B.5: Dependence of the navigator function on Δ_t at $N = 1$ for $(\Delta_\phi, \Delta_s) = (0.518139\ldots, 1.41252\ldots)$. The navigator function is constant in a small region, and equal in that region to the navigator function for the case of a mixed-correlator Ising setup where identical assumptions were made on the equivalent sectors ($V \leftrightarrow \mathbb{Z}_2-$odd , $S \leftrightarrow \mathbb{Z}_2-$even).

Appendix C

Appendix to Chapter 3

C.1 Technical details

One could expect that plotting out the whole critical curve is most easily achieved in parallel, by first choosing a grid $d_i \in [3,4]$, and launching separate searches for each $N_c(d_i)$ using field theory estimates as starting points. For the sake of computational efficiency, it is often desirable to begin with computations at low constraining power, and gradually improve it, using the previous results as starting points. It turns out that in the low-power regime, the bootstrap curve is a little too far from field theory close to $d = 3$, and so the searches there generically fail to locate the critical line. The better strategy turned out to be to locate $N_c(d)$ near $d = 4$, and then to move *along* the critical line down to $d = 3$. This is quite efficiently done with the help of the navigator function, in a spirit similar to that of [140].

Say we minimized the navigator function, defined for a set of internal parameters x, somewhere in the external parameter space $p = (d, N)$. How we find and move along $N_c(d)$ can be determined by how a variation δp can solve

$$\mathcal{N}(x_{\min}(p) + \delta x_{\min}, p + \delta p) = 0 \tag{C.1}$$

to first order in the variations. This is just Newton's method. For example, if we are at a certain d, and the minimum of the navigator function at a given N is non-zero, (C.1) predicts that we should take a step

$$\delta N = -\frac{\mathcal{N}(x_{\min}(p), p)}{\frac{\partial \mathcal{N}}{\partial N}(x_{\min}(p), p)} \tag{C.2}$$

to move towards $N_c(d)$. Then, once we are close enough to the critical curve at this d and want to take a step δd, we can take a step $\delta N = N_c'(d)\,\delta d$. (C.1) gives the derivative as

$$N_c'(d) = -\frac{\frac{\partial \mathcal{N}}{\partial d}(x_{\min}(p), p)}{\frac{\partial \mathcal{N}}{\partial N}(x_{\min}(p), p)} \,. \tag{C.3}$$

We used this Newton-type method to first plot the entire critical curve for a specific set of gap assumptions and numerical power. We were then able to use this curve as a starting point for parallelized computations of $N_c(d)$ when wanting to increase numerical power or to change gap assumptions.

We sometimes found useful to replace the Newton-type method with a more direct method. Indeed, we can simply add N as one of the internal parameters x. We then get N_c by finding the extremal point in the N direction inside the allowed region in this enlarged space, an exercise easily accomplished with the extremization algorithm described in [44]. This method proved less robust, most likely because of the shape of the tip of the allowed region in this space, but more computationally efficient when successful. We were able to perform every such extremization in close to the equivalent of a single `SDPB` run using the newly-developed `skydive` [149].

As for the specifics of the numerical computations, we used `scalar_blocks` [155] to generate conformal blocks; either `SDPB` [42, 43] to compute navigator functions and a BFGS algorithm to minimize them, or the new `skydive` to do it all at once; and `simpleboot` [150] to setup and manage the bootstrap problems. These programs depend on a number of parameters that must be set by the user, the most crucial of which being the derivative order Λ dictating the dimensionality of the space of functionals `SDPB` will search in. We welcome anyone interested in the particular values these parameters take in any given computation presented in this chapter to message the authors. For the most important computation of $N_c(3)$ at $\Lambda = 31$ using `skydive`, the parameters used are displayed in Table C.1.

C.2 $1/N$ expansion

We review here the $1/N$ expansion of model (3.1) [58, 145, 156, 157], and compute the leading $1/N$ correction to the dimension of ST′. The Lagrangian in the form

Λ	31
$\varkappa$	30
order	120
spins	$\{0,\ldots,40\} \cup \{46,47,51,52,55,56,59,60\}$
precision	768
dualityGapThreshold	10^{-20}
primalErrorThreshold	10^{-15}
dualErrorThreshold	10^{-15}
initialMatrixScalePrimal	10^{20}
initialMatrixScaleDual	10^{20}
feasibleCenteringParameter	0.3
infeasibleCenteringParameter	0.3
stepLengthReduction	0.7
maxComplementarity	10^{100}
dualityGapUpperLimit	0.2
externalParamInfinitestimal	10^{-40}
findBoundaryObjThreshold	10^{-20}
betaScanMin	0.3
betaScanMax	1.01
betaScanStep	0.1
stepMinThreshold	0.1
stepMaxThreshold	0.6
primalDualObjWeight	0.2
navigatorWithLogDetX	False
navigatorAutomaticShift	False
gradientWithLogDetX	True
betaClimbing	1.5
optimalbeta	False
stickToGCP	False

Table C.1: Parameters for the numerics.

relevant to the large-N expansion reads [157]

$$\mathcal{L} = \frac{1}{2}(\partial_\mu \phi)^2 + \frac{1}{2}\sigma\phi_{ai}\phi_{ai} + \frac{1}{2}T_{ij}\phi_{ai}\phi_{aj} - \frac{\sigma^2}{16u+4v} - \frac{1}{8v}\operatorname{Tr} T^2 \ , \tag{C.4}$$

where we introduced two auxiliary fields σ and T_{ij}. They are scalars belonging to the SS and ST irreps, and they morally replace ϕ^2 and $\phi_{ai}\phi_{aj} - \frac{1}{2}\phi^2\delta_{ij}$. They have scaling dimensions

$$\Delta_\sigma, \Delta_{T_{ij}} = 2 + O(1/N) \tag{C.5}$$

in any d, in contrast with the classical dimension $\Delta = d-2$ of the fields they replace. Scaling dimensions and OPE coefficients may be obtained as expansions in $1/N$, the coefficients of these expansions being computable by standard diagrammatic techniques.

At large N, all four fixed points on the left of Fig. 3.1 exist. They are distinguished by which of the auxiliary fields propagate: at $\mathcal{G}$, of course none does; at $\mathcal{H}$, only σ does, leading to the well known $1/N$ expansion of the $O(N)$ model; at $\mathcal{C}_-$, only T_{ij} propagates; and finally, both auxiliary fields propagate at $\mathcal{C}_+$. The difference in field content between $\mathcal{C}_+$ and $\mathcal{H}$ leads to an appreciable difference between the large-N spectrum of both theories, the most obvious differences being in the scalar ST sector. At $\mathcal{C}_+$, $\mathcal{O}_{\mathrm{ST}} = T_{ij}$, followed by two subleading fields $\mathcal{O}_{\mathrm{ST}'}$ and $\mathcal{O}_{\mathrm{ST}''}$ with $\Delta = 4 + O(1/N)$ resulting from the mixing of $(T^2)_{ij} - \frac{1}{2}\operatorname{Tr} T^2\delta_{ij}$ and σT_{ij}. In contrast, T_{ij} is absent at $\mathcal{H}$, so that $\mathcal{O}_{\mathrm{ST}} = \phi_{ai}\phi_{aj} - \frac{1}{2}\phi^2\delta_{ij}$, followed by $\mathcal{O}_{\mathrm{ST}'} = \sigma\mathcal{O}_{\mathrm{ST}}$ with $\Delta = d + O(1/N)$.

Although imposing a gap of the order of $d-2$ in the ST channel might have been enough to differentiate between both theories, we have found more constraining power by assuming the existence of $\mathcal{O}_{\mathrm{ST}}$, and imposing a gap above it. To motivate the value of this gap, it helps to know the $1/N$ correction to the dimension of $\mathcal{O}_{\mathrm{ST}'}$ in $\mathcal{H}$ and $\mathcal{C}_+$. In $\mathcal{H}$, this operator descends from the subleading symmetric traceless 2-tensor operator of O(N), which has been computed long ago in [158]:

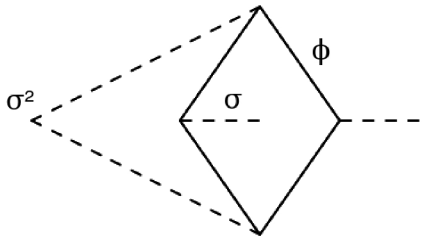
$$\Delta_{\mathrm{ST}'} = d - \frac{2\left(d^2-3d+4\right)\Gamma(d-1)}{\Gamma\left(1-\frac{d}{2}\right)\Gamma\left(\frac{d}{2}+1\right)\Gamma\left(\frac{d}{2}\right)^2}\frac{1}{N}\,. \qquad (\mathcal{H}) \tag{C.6}$$

In $\mathcal{C}_+$, this computation requires resolving the mixing problem between $\mathcal{O}_{\mathrm{ST}'}$ and $\mathcal{O}_{\mathrm{ST}''}$. Ref. [159] gives a nice and concise description of the procedure to follow to compute anomalous dimensions of composite operators at large-N. The example provided in this paper lists and explicitly computes all diagrams contributing to

the anomalous dimension of σ^2 in the $O(N)$ model. Since $\mathcal{C}^+$ differs from $\mathcal{H}$ by the addition of a new T_{ij} propagator as well as a new $T\phi\phi$ vertex [145]

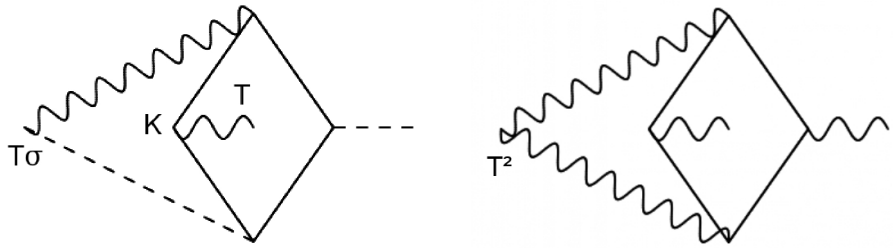
$$\begin{aligned} T_{ij}\phi_{ak}\phi_{bl} &\to \delta_{ab}\mathrm{K}_{ijkl}\,, \\ \mathrm{K}_{ijkl} &= \frac{1}{2}\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}-\delta_{ij}\delta_{kl}\right)\,; \end{aligned} \tag{C.7}$$

it should be clear that we can recycle some of the results of [159]. Indeed, the diagrams contributing to the renormalization of $(T^2)_{ij}$ – trace and σT_{ij} will have the same geometry as those contributing to σ^2. Here is an example of such a diagram arising in the computation of the anomalous dimension of σ^2:



(C.8)

One can think of this diagram as computing the renormalization of σ^2 by considering the three point function $\langle\sigma^2\sigma\sigma\rangle$. Diagram (C.8) has the following two nonzero cousins in $\mathcal{C}_+$:



(C.9)

These are determinable from Diagram (C.8) up to $O(2)$ group structure factors, coming from T_{ij} propagators and $T\phi\phi$ vertices, and differences in symmetry factors. There is luckily no need to compute any loop integrals. After all of the dust settles,

the leading order anomalous dimensions of this mixing problem are given by

$$\begin{aligned}\gamma^1 &= \frac{4(12-22d+13d^2-2d^3)\sin\left(\frac{\pi d}{2}\right)\Gamma(d-2)}{\pi d\Gamma\left(\frac{d}{2}-1\right)\Gamma\left(\frac{d}{2}\right)}\frac{1}{N}\,,\\ \gamma^2 &= \frac{2^{d-3}(24-28d+5d^2+d^3)\sin\left(\frac{\pi d}{2}\right)\Gamma\left(\frac{d-1}{2}\right)}{\pi^{3/2}\Gamma\left(\frac{d}{2}+1\right)}\frac{1}{N}\,.\end{aligned} \tag{C.10}$$

We confirmed this result by comparing it at $O(\epsilon, 1/N)$ to the anomalous dimensions of the corresponding operators in the ϵ-expansion. These were never computed before, but they are easily obtained using the OPE method described in [160]. The scaling dimensions of $\mathcal{O}_{\mathrm{ST}'}$ and $\mathcal{O}_{\mathrm{ST}''}$ at $\mathcal{C}_+$ are then of course given by

$$\Delta_{\mathrm{ST}'} = 4+\gamma^1+O(1/N^2)\,,\quad \Delta_{\mathrm{ST}''} = 4+\gamma^2+O(1/N^2)\,. \qquad (\mathcal{C}_+) \tag{C.11}$$

Appendix D

Résumé en francais

D.1 Chapitre 1: Fonction de navigation pour le bootstrap conforme

La première partie de cette thèse sert à introduire le concept de *fonction de navigation* pour le bootstrap conforme numérique. Ressuscité il y a un peu plus d'une dizaine d'années [6] pour l'étude de CFTs en $d > 2$ dimensions, le bootstrap est une manière d'extraire la valeur des données définissant une CFT (*CFT data*) en imposant seulement des conditions de cohérence générales sur ses fonctions à 4 points, comme la symétrie de croisement et la consistance avec l'expansion du produit d'opérateurs (*OPE*).

Jusqu'à présent, la machine du bootstrap numérique pouvait être apparentée à un oracle: le bootstrappeur arrivait à l'oracle (son ordinateur) avec une conjecture x sur la nature du spectre de la théorie conforme qui l'intéresse, et tout ce qu'il lui était possible de savoir en retour était si, oui ou non, cette conjecture était en accord avec les conditions de cohérence citées plus haut. Ce que nous avons été capables de faire dans ce chapitre est d'introduire des fonctions $\mathcal{N}(x)$ qui sont *positives* si nos suppositions x sur la nature du spectre de la CFT d'intérêt sont incohérentes, et donc interdites, et *négatives* lorsqu'elles sont permises. Cette fonction donne une mesure du *degré* auquel un point x est permis ou interdit, et devrait permettre de *naviguer* plus facilement dans l'espace de recherche dans lequel vit x vers les régions permises voulues.

Le chapitre débute avec la construction de telles fonctions. La logique est simple: si l'équation traduisant la symétrie de croisement des fonctions à 4 points qui

nous intéressent a la forme $\mathrm{EC}(x) = 0$, nous devons trouver un terme additionnel λM_{add} tel que l'équation augmentée $\mathrm{EC}(x) + \lambda M_{\mathrm{add}} = 0$ a toujours une solution pour un certain $\lambda = \lambda(x) > 0$. Si cela est possible, alors nous pouvons définir la fonction de navigation $\mathcal{N}(x)$ comme la valeur minimale permise de λ. Deux exemples sont donnés: $\mathcal{N}_{\mathrm{GFF}}(x)$, qui correspond à ajouter à l'équation de croisement des termes associés à la soi-disant solution libre généralisée (*Generalized Free Field*), existant pour toute valeur de x, et $\mathcal{N}_{\Sigma}(x)$, où simplement assez de termes sont ajoutés à l'équation pour garantir par indépendance linéaire une solution.

Il s'avère que la valeur de la fonction de navigation à un point donné peut être obtenue à partir de la solution optimale à un problème d'optimisation SDP d'une forme bien connue des bootstrappeurs, tâche qui peut être accomplie efficacement par des solveurs préexistants. Toutefois, pour naviguer efficacement, il est utile de connaître non seulement la fonction de navigation elle-même, mais aussi ses dérivées. Nous démontrons dans ce chapitre comment les premières, et secondes, dérivées de la fonction de navigation peuvent être calculées essentiellement sans effort. En étudiant la perturbation d'une solution optimale d'un problème SDP, nous sommes capables de déterminer que la perturbation de la fonction de navigation ne dépend pas de la solution du problème SDP perturbé, mais seulement de la perturbation du problème lui-même. Ainsi, calculer un gradient de n composantes ne requiert pas la résolution de $n+1$ problèmes SDP, mais seulement du problème SDP originel. Il en est de même pour les secondes dérivées.

Dans le reste de ce chapitre, nous explorons avec la fonction de navigation *GFF* peut-être le problème de bootstrap le plus standard, c'est-à-dire le bootstrap mixte du modèle d'Ising en trois dimensions. Nous visualisons l'allure de $\mathcal{N}_{\mathrm{GFF}}(x)$ pour $x = (\Delta_\sigma, \Delta_\epsilon)$ et $x = (\Delta_\sigma, \Delta_\epsilon, \theta)$, où Δ_σ et Δ_ϵ sont respectivement les dimensions d'échelle des premiers opérateurs scalaires impaires et paires sous la symétrie $\mathbb{Z}_2$ de la CFT d'Ising, et θ est un certain ratio de coefficients d'OPE. Nous observons que l'"île permise", observée maintes fois auparavant comme la région où doit exister la CFT d'Ising, est reproduite parfaitement par la région où la fonction de navigation est négative. Nous observons aussi que la fonction de navigation est assez lisse pour y appliquer des routines d'optimisation. Ainsi, nous sommes capables, à partir d'un point initial x_{init} à l'extérieur de l'île permise, de naviguer vers cette dernière en minimisant la fonction de navigation à l'aide d'une version quelque peu modifiée de l'algorithme quasi-Newtonien BFGS. Nous sommes aussi capables de naviguer efficacement vers des points extrêmes sur la frontière de cette île, toujours à l'aide d'un algorithme d'optimisation quasi-Newtonien utilisant les

dérivées si peu coûteuses computationnellement de la fonction de navigation.

D.2 Chapitre 2: Naviguer au travers de l'archipel $O(N)$

Nous avons appliqué dans le premier chapitre la "méthode navigatrice" à un exemple bien particulier, celui du modèle d'Ising, mais il devrait être clair de son exposition que cette méthode est complètement générale. Dans ce second chapitre, nous entreprenons de démontrer ce fait en étudiant un autre système de bootstrap prototypique, celui du bootstrap des opérateurs ϕ et s du modèle $O(N)$. Le modèle critique $O(N)$ forme en fait une famille de CFTs, une pour chaque valeur du paramètre N et pour chaque dimension $d > 2$. De par des études préalables sur cette famille, on s'attend à ce que pour un choix assez judicieux d'hypothèses sur le spectre x, une région permise isolée, ou une "île" pour rappeler la terminologie du dernier chapitre, existe pour chaque paire $p = (d, N)$. Il est donc intéressant de voir comment la fonction de navigation peut nous permettre de naviguer d'île en île dans cet espace de paramètres externes p.

Pour suivre une île en fonction de p, il s'avère plus simple de suivre un point particulier à l'intérieur de cette île, soit le point $x_{\min}(p)$ où la fonction de navigation est minimisée. Lorsque celui-ci est trouvé, et qu'on veut effectuer un changement δp de p, on trouve que la variation de la position du minimum $\delta x_{\min}(p)$ au premier ordre dépend seulement des dérivées de la fonction de navigation au minimum précédent (voir (2.7)). Ainsi, moyennant des pas δp assez petits, nous devrions facilement être capables de suivre notre "archipel" d'îles isolées. On démontre en premier lieu que cela s'avère vrai dans notre cas du modèle $O(N)$. En utilisant cette prescription de prédiction de la localisation d'îles voisines, nous sommes capables dans un exemple donné de couper de 45 à 8 le nombre d'itérations nécessaires de la routine BFGS à trouver le minimum de la fonction de navigation à $p + \delta p$.

Armés de cette méthode, nous explorons ensuite la région $3 < d < 4$ pour des valeurs physiquement importantes de N. C'est un fait bien connu qu'on s'attend génériquement à ce qu'une CFT en dimension non-entière soit non-unitaire, dû à la présence d'opérateurs "évanescents" qui découplent de la théorie en dimensions entières. Le bootstrap utilise l'unitarité comme l'une des conditions importantes à imposer, et on peut donc se demander si on trouvera bel et bien une solution aux équations de croisement pour $3 < d < 4$. Notre étude confirme que ces

opérateurs sont invisibles à notre degré de précision, puisque nous trouvons bel et bien des solutions aux équations de croisement, et de celles-ci on peut tirer des prédictions pour certaines dimensions d'opérateurs qui sont en parfait accord avec les prédictions provenant de d'autres méthodes, comme l'ϵ-*expansion*.

Il est pareillement attendu que le modèle $O(N)$ soit génériquement non-unitaire pour des valeurs non-entières de N. Nous confirmons que le bootstrap conforme ne peut déceler cette non-unitarité pour $N > 1$ en trois dimensions. Nous observons toutefois que le minimum de la fonction de navigation s'approche de 0 lorsqu'on approche de $N = 1$. On observe ultimement la disparition de l'île permise d'$O(N)$ à $N = 1$. On explique ce phénomène par la présence d'opérateurs primaires devenant descendants, et par la présence de multiples opérateurs primaires distincts devenant un seul opérateur à $N = 1$. Pour continuer la solution sous $N = 1$, tous ces opérateurs disparaissant du spectre devraient acquérir des coéfficients d'OPE imaginaires, violant fortement l'unitarité de la CFT. Nous observons des signatures claires indiquant que ce scénario est réalisé, et comparons les opérateurs spéciaux à ceux prédits par l'ϵ-*expansion*.

D.3 Chapitre 3: Bootstrap de l'aimant frustré non colinéaire

Après avoir testé suffisamment la méthode navigatrice sur des exemples déjà bien connus, nous nous attaquons dans ce dernier chapitre à une question physique plus intéressante, une question à laquelle les théoriciens n'ont pas réussi à se mettre en accord pour des décennies!

Nous commençons ce chapitre en expliquant la nature de la controverse. Il existe une certaine classe d'aimants, les aimants dits **non colinéaires**, étant en fait probablement le plus simple exemple de matériau magnétique frustré. À cause de cette frustration, on s'attend à ce que si leur transition de phase soit continue, elle devrait se trouver dans une classe d'universalité distincte de l'$O(N)$ standard des ferro aimants classiques. Elle devrait en fait se trouver dans la soi-disant classe d'universalité $O(N) \times O(2)$. Différentes approches théoriques arrivent toutefois à différentes conclusions. L'ϵ-*expansion* autour de $d = 4$ prédit que cette classe d'universalité existe pour N assez large, mais cesse d'exister à une valeur critique N_c. À cette valeur critique, le point fixe stable du groupe de renormalization (RG) entre en collision avec un autre point fixe, et ceux-ci s'annihilent. Les meilleurs

estimés de N_c le placent bien au-dessus de 3, et donc une transition de phase de premier ordre est prédite pour les matériaux réels, ayant $N = 2$ ou $N = 3$. Une autre approche, le RG en dimension fixe, prédit une situation bien différente, où un point fixe stable existerait génériquement en trois dimensions pour $N = 2$ ou $N = 3$, mais sa nature serait distincte de celle du point fixe prédit par l'*ϵ-expansion*.

Dans le reste de ce chapitre, nous tentons de déterminer laquelle de ces approches est correcte. Pour ce faire, nous avons comme objectif de tracer entièrement la courbe $N_c(d)$, en partant du régime perturbatif près de $d = 4$ et en terminant à la dimension d'intérêt $d = 3$. En utilisant des méthodes analogues à celles développées au dernier chapitre, il nous est facile, une fois N_c déterminé pour une valeur de d dans le régime pertubatif, de naviguer dans l'espace des paramètres externes $p = (d, N)$ tout en suivant la courbe critique définie comme les valeurs de N pour lesquels le minimum de la fonction de navigation est 0. Une fois cet exercice fait, nous arrivons à une courbe critique en excellent accord avec la prédiciton de l'*ϵ-expansion*, déterminant notamment $N_c(3) \approx 3.78$. Il en découle donc que la transition de phase devrait être de premier ordre, et que le caractère continu qui a été observé dans plusieurs expériences et simulations résulterait en fait d'une transition de phase *faiblement* de premier ordre.

Pour observer la disparition de la CFT définissant la classe d'universalité $O(N) \times O(2)$, il nous était bien important d'éliminer toutes autres solutions de nos équations de croisement. Habituellement, si on s'attend à ce que notre CFT d'intérêt soit associée au seul point fixe stable du RG, imposer qu'il existe seulement un scalaire totalement invariant important ($\Delta < d$) est assez pour l'isoler dans notre espace de recherche. Malheureusement, cette logique ne s'applique pas dans notre situation, où le modèle $O(2N)$ continue à polluer notre solution d'intérêt même après l'imposition de cette condition. En s'inspirant de la limite de N large, nous avons réussi à trouver une autre condition qui nous permet d'isoler la solution $O(N) \times O(2)$. En effet, le spectre du modèle $O(2N)$ est différent pour N large de celui du modèle $O(N) \times O(2)$, puisque ce dernier possède deux, et non pas un seul comme pour le modèle $O(N)$, opérateurs dits de Hubbard-Stratonovich ayant une dimension d'échelle de 2 (plus corrections en expansion $\frac{1}{N}$), peu importe la dimension d. Cette différence en engendre plusieurs autres dans le spectre complet d'opérateurs des deux théories. Puisque la courbe $N_c(d)$ se trouve toujours à N relativement large, il nous est possible d'utiliser cette intuition des théories à N large pour imposer certains gaps dans le spectre, qui nous permettent ultimement d'isoler notre modèle $O(N) \times O(2)$. Pour s'assurer que la disparition de la solution

des équations de croisement provenait bel et bien d'une collision de deux points fixes, nous avons finalement observé une signature directe de ce scénario, soit le comportement en racine carrée des données définissant la CFT en approchant la collision.

Bibliography

[1] A. M. Polyakov, "Nonhamiltonian approach to conformal quantum field theory," *Zh. Eksp. Teor. Fiz.* **66** (1974) 23–42.

[2] G. F. Chew, "Theory of strong coupling of ordinary particles," in *9th International Annual Conference on High Energy Physics*, vol. Vol.1, pp. 313–352. 1960.

[3] G. F. Chew and S. Mandelstam, "Theory of the low-energy pion-pion interaction — II," *Il Nuovo Cimento* **19** (1961) 752–776.

[4] G. F. Chew, *The Analytic S Matrix; A Basis for Nuclear Democracy.* W. A. Benjamin, New York, 1966.

[5] A. A. Belavin, A. M. Polyakov, and A. B. Zamolodchikov, "Infinite Conformal Symmetry in Two-Dimensional Quantum Field Theory," *Nucl. Phys. B* **241** (1984) 333–380.

[6] R. Rattazzi, V. S. Rychkov, E. Tonni, and A. Vichi, "Bounding scalar operator dimensions in 4D CFT," *JHEP* **12** (2008) 031, arXiv:0807.0004 [hep-th].

[7] Z. Komargodski and A. Zhiboedov, "Convexity and Liberation at Large Spin," *JHEP* **11** (2013) 140, arXiv:1212.4103 [hep-th].

[8] A. L. Fitzpatrick, J. Kaplan, D. Poland, and D. Simmons-Duffin, "The Analytic Bootstrap and AdS Superhorizon Locality," *JHEP* **12** (2013) 004, arXiv:1212.3616 [hep-th].

[9] C. Beem, M. Lemos, L. Rastelli, and B. C. van Rees, "The (2, 0) superconformal bootstrap," *Phys. Rev. D* **93** no. 2, (2016) 025016, arXiv:1507.05637 [hep-th].

[10] I. Heemskerk, J. Penedones, J. Polchinski, and J. Sully, "Holography from Conformal Field Theory," *JHEP* **10** (2009) 079, arXiv:0907.0151 [hep-th].

[11] G. F. Chew, ""Bootstrap": A Scientific Idea?," *Science* **161** (1968) 762–765.

[12] T. Hartman, D. Mazac, D. Simmons-Duffin, and A. Zhiboedov, "Snowmass White Paper: The Analytic Conformal Bootstrap," in *Snowmass 2021*. 2, 2022. arXiv:2202.11012 [hep-th].

[13] D. Poland and D. Simmons-Duffin, "Snowmass White Paper: The Numerical Conformal Bootstrap," in *Snowmass 2021*. 3, 2022. arXiv:2203.08117 [hep-th].

[14] J. L. Cardy, "Operator Content of Two-Dimensional Conformally Invariant Theories," *Nucl. Phys. B* **270** (1986) 186–204.

[15] J. L. Cardy, "Operator content and modular properties of higher dimensional conformal field theories," *Nucl. Phys. B* **366** (1991) 403–419.

[16] S. Hellerman, "A Universal Inequality for CFT and Quantum Gravity," *JHEP* **08** (2011) 130, arXiv:0902.2790 [hep-th].

[17] M. Kruczenski, J. Penedones, and B. C. van Rees, "Snowmass White Paper: S-matrix Bootstrap," arXiv:2203.02421 [hep-th].

[18] X. Han, S. A. Hartnoll, and J. Kruthoff, "Bootstrapping Matrix Quantum Mechanics," *Phys. Rev. Lett.* **125** no. 4, (2020) 041601, arXiv:2004.10212 [hep-th].

[19] P. D. Anderson and M. Kruczenski, "Loop Equations and bootstrap methods in the lattice," *Nucl. Phys. B* **921** (2017) 702–726, arXiv:1612.08140 [hep-th].

[20] M. Cho, B. Gabai, Y.-H. Lin, V. A. Rodriguez, J. Sandor, and X. Yin, "Bootstrapping the Ising Model on the Lattice," arXiv:2206.12538 [hep-th].

[21] J. Polchinski, "Scale and Conformal Invariance in Quantum Field Theory," *Nucl. Phys. B* **303** (1988) 226–236.

[22] Y. Nakayama, "Scale invariance vs conformal invariance," *Phys. Rept.* **569** (2015) 1–93, arXiv:1302.0884 [hep-th].

[23] L. P. Kadanoff, W. Gotze, D. Hamblen, R. Hecht, E. A. S. Lweis, V. V. Palciauskas, M. Rayl, J. Swift, D. Aspens, and J. Kane, "Static Phenomena Near Critical Points: Theory and Experiment," *Rev. Mod. Phys.* **39** (1967) 395–431.

[24] P. Heller, "Experimental investigations of critical phenomena," *Reports on Progress in Physics* **30** no. 2, (Jul, 1967) 731.

[25] G. Ahlers, "Critical phenomena at low temperature," *Rev. Mod. Phys.* **52** (1980) 489–503.

[26] K. G. Wilson, "The renormalization group and critical phenomena," *Rev. Mod. Phys.* **55** (1983) 583–600.

[27] S. M. Chester, W. Landry, J. Liu, D. Poland, D. Simmons-Duffin, N. Su, and A. Vichi, "Bootstrapping Heisenberg magnets and their cubic instability," *Phys. Rev. D* **104** no. 10, (2021) 105013, arXiv:2011.14647 [hep-th].

[28] J. A. Lipa, D. R. Swanson, J. A. Nissen, T. C. P. Chui, and U. E. Israelsson, "Heat Capacity and Thermal Relaxation of Bulk Helium very near the Lambda Point," *Phys. Rev. Lett.* **76** (1996) 944–947.

[29] S. M. Chester, W. Landry, J. Liu, D. Poland, D. Simmons-Duffin, N. Su, and A. Vichi, "Carving out OPE space and precise $O(2)$ model critical exponents," *JHEP* **06** (2020) 142, arXiv:1912.03324 [hep-th].

[30] F. Kos, D. Poland, D. Simmons-Duffin, and A. Vichi, "Precision Islands in the Ising and $O(N)$ Models," *JHEP* **08** (2016) 036, arXiv:1603.04436 [hep-th].

[31] R. S. Erramilli, L. V. Iliesiu, P. Kravchuk, A. Liu, D. Poland, and D. Simmons-Duffin, "The Gross-Neveu-Yukawa archipelago," *JHEP* **02** (2023) 036, arXiv:2210.02492 [hep-th].

[32] A. Pelissetto and E. Vicari, "Critical phenomena and renormalization group theory," *Phys. Rept.* **368** (2002) 549–727, arXiv:cond-mat/0012164.

[33] A. M. Polyakov, "Conformal symmetry of critical fluctuations," *JETP Lett.* **12** (1970) 381–383.

[34] A. Antunes, M. S. Costa, V. Goncalves, and J. V. Boas, "Lightcone bootstrap at higher points," *JHEP* **03** (2022) 139, arXiv:2111.05453 [hep-th].

[35] D. Poland, V. Prilepina, and P. Tadić, "The five-point bootstrap," arXiv:2305.08914 [hep-th].

[36] D. Poland, S. Rychkov, and A. Vichi, "The Conformal Bootstrap: Theory, Numerical Techniques, and Applications," *Rev. Mod. Phys.* **91** (2019) 015002, arXiv:1805.04405 [hep-th].

[37] D. Simmons-Duffin, "The Conformal Bootstrap," in *Theoretical Advanced Study Institute in Elementary Particle Physics: New Frontiers in Fields and Strings*, pp. 1–74. 2017. arXiv:1602.07982 [hep-th].

[38] F. A. Dolan and H. Osborn, "Conformal partial waves and the operator product expansion," *Nucl. Phys. B* **678** (2004) 491–507, arXiv:hep-th/0309180.

[39] S. El-Showk and M. F. Paulos, "Extremal bootstrapping: go with the flow," *JHEP* **03** (2018) 148, arXiv:1605.08087 [hep-th].

[40] M. F. Paulos, "JuliBootS: a hands-on guide to the conformal bootstrap," arXiv:1412.4127 [hep-th].

[41] F. Kos, D. Poland, and D. Simmons-Duffin, "Bootstrapping the $O(N)$ vector models," *JHEP* **06** (2014) 091, arXiv:1307.6856 [hep-th].

[42] D. Simmons-Duffin, "A Semidefinite Program Solver for the Conformal Bootstrap," *JHEP* **06** (2015) 174, arXiv:1502.02033 [hep-th].

[43] W. Landry and D. Simmons-Duffin, "Scaling the semidefinite program solver SDPB," arXiv:1909.09745 [hep-th].

[44] 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, arXiv:2104.09518 [hep-th].

[45] F. Kos, D. Poland, and D. Simmons-Duffin, "Bootstrapping Mixed Correlators in the 3D Ising Model," *JHEP* **11** (2014) 109, arXiv:1406.4858 [hep-th].

[46] M. V. Kompaniets and E. Panzer, "Minimally subtracted six loop renormalization of $O(n)$-symmetric ϕ^4 theory and critical exponents," *Phys. Rev. D* **96** no. 3, (2017) 036016, arXiv:1705.06483 [hep-th].

[47] M. Hasenbusch, "Finite size scaling study of lattice models in the three-dimensional Ising universality class," *Phys. Rev. B* **82** (2010) 174433, arXiv:1004.4486 [cond-mat.stat-mech].

[48] H. Osborn and A. Stergiou, "Seeking fixed points in multiple coupling scalar theories in the ϵ expansion," *JHEP* **05** (2018) 051, arXiv:1707.06165 [hep-th].

[49] S. Rychkov and A. Stergiou, "General Properties of Multiscalar RG Flows in $d = 4 - \varepsilon$," *SciPost Phys.* **6** no. 1, (2019) 008, arXiv:1810.10541 [hep-th].

[50] F. Gliozzi, "More constraining conformal bootstrap," *Phys. Rev. Lett.* **111** (2013) 161602, arXiv:1307.3111 [hep-th].

[51] A. Laio, U. L. Valenzuela, and M. Serone, "Monte Carlo approach to the conformal bootstrap," *Phys. Rev. D* **106** no. 2, (2022) 025019, arXiv:2206.05193 [hep-th].

[52] G. Kántor, V. Niarchos, and C. Papageorgakis, "Solving Conformal Field Theories with Artificial Intelligence," *Phys. Rev. Lett.* **128** no. 4, (2022) 041601, arXiv:2108.08859 [hep-th].

[53] P. Bak and D. Mukamel, "Physical realizations of n $\geqslant$ 4-component vector models. 3. Phase transitions in Cr, Eu, Mn S-2, Ho, Dy, and Tb," *Phys. Rev. B* **13** (1976) 5086–5094.

[54] Z. Barak and M. B. Walker, "First-order phase transitions in Tb, Dy, and Ho," *Phys. Rev. B* **25** (Feb, 1982) 1969–1972.

[55] H. Kawamura and S. Miyashita, "Phase Transition of the Two-Dimensional Heisenberg Antiferromagnet on the Triangular Lattice," *Journal of the Physical Society of Japan* **53** no. 1, (1984) 9–12.

[56] H. Kawamura, "Phase Transition of the Three-Dimensional Heisenberg Antiferromagnet on the Layered-Triangular Lattice," *Journal of the Physical Society of Japan* **54** no. 9, (1985) 3220–3223.

[57] H. Kawamura, "Phase Transition of the Three-Dimensional XY Antiferromagnet on the Layered-Triangular Lattice," *Journal of the Physical Society of Japan* **55** no. 7, (1986) 2095–2098.

[58] H. Kawamura, "Renormalization-group analysis of chiral transitions," *Phys. Rev. B* **38** (1988) 4916–4928. [Erratum: Phys.Rev.B 42, 2610–2610 (1990)].

[59] H. Kawamura, "Universality of phase transitions of frustrated antiferromagnets," *Journal of Physics: Condensed Matter* **10** no. 22, (Jun, 1998) 4707.

[60] B. Sirois, M. Reehorst, S. Rychkov, and B. van Rees, "Bootstrapping noncollinear magnets," 2023.

[61] H. Kawamura, "Renormalization-Group Approach to the Frustrated Heisenberg Antiferromagnet on the Layered-Triangular Lattice," *Journal of the Physical Society of Japan* **55** no. 7, (1986) 2157–2165.

[62] S. M. Chester, "Weizmann Lectures on the Numerical Conformal Bootstrap," arXiv:1907.05147 [hep-th].

[63] 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, arXiv:1403.4545 [hep-th].

[64] D. Poland and D. Simmons-Duffin, "Bounds on 4D Conformal and Superconformal Field Theories," *JHEP* **05** (2011) 017, arXiv:1009.2087 [hep-th].

[65] N. Afkhami-Jeddi, T. Hartman, and A. Tajdini, "Fast Conformal Bootstrap and Constraints on 3d Gravity," *JHEP* **05** (2019) 087, arXiv:1903.06272 [hep-th].

[66] Wikipedia, "Broyden–Fletcher–Goldfarb–Shanno algorithm.".

[67] F. Caracciolo and V. S. Rychkov, "Rigorous Limits on the Interaction Strength in Quantum Field Theory," *Phys. Rev. D* **81** (2010) 085037, arXiv:0912.2726 [hep-th].

[68] N. Su, "Search Methods in the Numerical Bootstrap,". talk at the workshop 'Developments in the Numerical Bootstrap', Simons Center for Geometry and Physics, Stony Brook University, November 4-6, 2019.

[69] D. Poland, D. Simmons-Duffin, and A. Vichi, "Carving Out the Space of 4D CFTs," *JHEP* **05** (2012) 110, arXiv:1109.5176 [hep-th].

[70] R. W. Freund and F. Jarre, "A sensitivity result for semidefinite programs," *Operations Research Letters* **32** no. 2, (2004) 126–132.

[71] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, "Complementarity and nondegeneracy in semidefinite programming," *Mathematical programming* **77** no. 1, (1997) 111–128.

[72] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2004.

[73] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, New York, NY, USA, 2006.

[74] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001. http://www.scipy.org/.

[75] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, 2000.

[76] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Softw.* **20** no. 3, (1994) 286–307.

[77] A. Conn, N. Gould, and P. Toint, "Convergence of quasi-Newton matrices generated by the symmetric rank one update," *Mathematical Programming* **50** no. 1-3, (1991) 177–195.

[78] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948. 1995.

[79] S. M. Chester and N. Su, "Upper critical dimension of the 3-state Potts model," arXiv:2210.09091 [hep-th].

[80] S. El-Showk and M. F. Paulos, "Bootstrapping Conformal Field Theories with the Extremal Functional Method," *Phys. Rev. Lett.* **111** no. 24, (2013) 241601, arXiv:1211.2810 [hep-th].

[81] M. Reehorst, "Rigorous bounds on irrelevant operators in the 3d Ising model CFT," *JHEP* **09** (2022) 177, arXiv:2111.12093 [hep-th].

[82] D. Simmons-Duffin, "The Lightcone Bootstrap and the Spectrum of the 3d Ising CFT," *JHEP* **03** (2017) 086, arXiv:1612.08471 [hep-th].

[83] N. Su, "The Hybrid Bootstrap," arXiv:2202.07607 [hep-th].

[84] F. Kos, D. Poland, D. Simmons-Duffin, and A. Vichi, "Bootstrapping the $O(N)$ Archipelago," *JHEP* **11** (2015) 106, arXiv:1504.07997 [hep-th].

[85] J. Henriksson, S. R. Kousvos, and A. Stergiou, "Analytic and Numerical Bootstrap of CFTs with $O(m) \times O(n)$ Global Symmetry in 3D," *SciPost Phys.* **9** no. 3, (2020) 035, arXiv:2004.14388 [hep-th].

[86] S. R. Kousvos and A. Stergiou, "Bootstrapping Mixed Correlators in Three-Dimensional Cubic Theories," *SciPost Phys.* **6** no. 3, (2019) 035, arXiv:1810.10015 [hep-th].

[87] S. R. Kousvos and A. Stergiou, "Bootstrapping Mixed Correlators in Three-Dimensional Cubic Theories II," *SciPost Phys.* **8** no. 6, (2020) 085, arXiv:1911.00522 [hep-th].

[88] S. Albayrak, R. S. Erramilli, Z. Li, D. Poland, and Y. Xin, "Bootstrapping $N_f = 4$ conformal QED_3," arXiv:2112.02106 [hep-th].

[89] Z. Li and N. Su, "3D CFT archipelago from single correlator bootstrap," *Physics Letters B* **797** (Oct, 2019) 134920.

[90] M. Hogervorst, S. Rychkov, and B. C. van Rees, "Unitarity violation at the Wilson-Fisher fixed point in 4-ϵ dimensions," *Phys. Rev. D* **93** no. 12, (2016) 125025, arXiv:1512.00013 [hep-th].

[91] S. El-Showk, M. Paulos, D. Poland, S. Rychkov, D. Simmons-Duffin, and A. Vichi, "Conformal Field Theories in Fractional Dimensions," *Phys. Rev. Lett.* **112** (2014) 141601, arXiv:1309.5089 [hep-th].

[92] C. N. Gowdigere, J. Santara, and Sumedha, "Conformal bootstrap signatures of the tricritical Ising universality class," *Phys. Rev. D* **101** no. 11, (2020) 116020, arXiv:1811.11442 [hep-th].

[93] S. M. Chester, S. S. Pufu, and R. Yacoby, "Bootstrapping $O(N)$ vector models in $4 < d < 6$," *Phys. Rev. D* **91** no. 8, (2015) 086014, arXiv:1412.7746 [hep-th].

[94] Y.-C. He, J. Rong, and N. Su, "Non-Wilson-Fisher kinks of $O(N)$ numerical bootstrap: from the deconfined phase transition to a putative new family of CFTs," *SciPost Phys.* **10** no. 5, (2021) 115, arXiv:2005.04250 [hep-th].

[95] J. Rong and J. Zhu, "On the ϕ^3 theory above six dimensions," *JHEP* **04** (2020) 151, arXiv:2001.10864 [hep-th].

[96] N. Bobev, S. El-Showk, D. Mazac, and M. F. Paulos, "Bootstrapping SCFTs with Four Supercharges," *JHEP* **08** (2015) 142, arXiv:1503.02081 [hep-th].

[97] D. J. Binder and S. Rychkov, "Deligne Categories in Lattice Models and Quantum Field Theory, or Making Sense of $O(N)$ Symmetry with Non-integer N," *JHEP* **04** (2020) 117, arXiv:1911.07895 [hep-th].

[98] M. Kompaniets and K. J. Wiese, "Fractal dimension of critical curves in the $O(n)$-symmetric ϕ^4 model and crossover exponent at 6-loop order: Loop-erased random walks, self-avoiding walks, Ising, XY , and Heisenberg models," *Phys. Rev. E* **101** no. 1, (2020) 012104, arXiv:1908.07502 [cond-mat.stat-mech].

[99] A. Atanasov, A. Hillman, D. Poland, J. Rong, and N. Su, "Precision Bootstrap for the $\mathcal{N} = 1$ Super-Ising Model," arXiv:2201.02206 [hep-th].

[100] P. Calabrese, A. Pelissetto, and E. Vicari, "Multicritical phenomena in $O(n_1) + O(n_2)$ symmetric theories," *Phys. Rev. B* **67** (2003) 054505, arXiv:cond-mat/0209580.

[101] O. Schnetz, "Eight loop gamma in ϕ^4." Emmy Noether seminar, Erlangen (Germany), April, 2021.

[102] O. Schnetz, "Numbers and Functions in Quantum Field Theory," *Phys. Rev. D* **97** no. 8, (2018) 085018, arXiv:1606.08598 [hep-th].

[103] J. Henriksson, "The critical $O(N)$ CFT: Methods and conformal data," arXiv:2201.09520 [hep-th].

[104] J. Henriksson, *Analytic bootstrap for perturbative conformal field theories*. PhD thesis, University of Oxford, 2020. arXiv:2008.12600 [hep-th].

[105] P. Dey, A. Kaviraj, and A. Sinha, "Mellin space bootstrap for global symmetry," *JHEP* **07** (2017) 019, arXiv:1612.05032 [hep-th].

[106] G. Still, "Lectures on Parametric Optimization: An Introduction." Preprint, March, 2018.

[107] J. Guddat, F. Vazquez, and H. Jongen, *Parametric Optimization: Singularities, Pathfollowing and Jumps*. Vieweg+Teubner Verlag, 1990.

[108] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*. Springer-Verlag, Berlin, Heidelberg, 1990.

[109] A. Cappelli, L. Maffi, and S. Okuda, "Critical Ising Model in Varying Dimension by Conformal Bootstrap," *JHEP* **01** (2019) 161, arXiv:1811.07751 [hep-th].

[110] J. Le Guillou and J. Zinn-Justin, "Accurate critical exponents for Ising like systems in non-integer dimensions," *Journal de Physique* **48** no. 1, (1987) 19–24. https://hal.archives-ouvertes.fr/jpa-00210418.

[111] A. Codello, N. Defenu, and G. D'Odorico, "Critical exponents of $O(N)$ models in fractional dimensions," *Phys. Rev. D* **91** no. 10, (2015) 105003, arXiv:1410.3308 [hep-th].

[112] Y. Holovatch, "Phase transition in continuous symmetry model in general dimensions — fixed dimension Renormalization Group approach," *International Journal of Modern Physics A* **08** no. 30, (1993) 5329–5351.

[113] Y. Holovatch, "Renormalization group study of the m-vector model between two and four dimensions," *Ferroelectrics* **192** no. 1, (2, 1997) 55–59.

[114] Y. Holovatch, "Critical behaviour in non-integer dimension," in *From Quantum Mechanics to Technology*, Z. Petru, J. Przystawa, and K. Rapcewicz, eds., pp. 269–281. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

[115] H. Shimada and S. Hikami, "Fractal dimensions of self-avoiding walks and Ising high-temperature graphs in 3D conformal bootstrap," *J. Statist. Phys.* **165** (2016) 1006, arXiv:1509.04039 [cond-mat.stat-mech].

[116] F. Bertucci, J. Henriksson, and B. McPeak, "Analytic bootstrap of mixed correlators in the $O(n)$ CFT," arXiv:2205.09132 [hep-th].

[117] F. A. Dolan and H. Osborn, "Conformal four point functions and the operator product expansion," *Nucl. Phys. B* **599** (2001) 459–496, arXiv:hep-th/0011040.

[118] J. Maldacena and A. Zhiboedov, "Constraining Conformal Field Theories with A Higher Spin Symmetry," *J. Phys. A* **46** (2013) 214011, arXiv:1112.1016 [hep-th].

[119] J. Golden and M. F. Paulos, "No unitary bootstrap for the fractal Ising model," *JHEP* **03** (2015) 167, arXiv:1411.7932 [hep-th].

[120] J. L. Cardy and H. W. Hamber, "The O(n) Heisenberg Model Close to $n = d = 2$," *Phys. Rev. Lett.* **45** (1980) 499. [Erratum: Phys.Rev.Lett. 45, 1217 (1980)].

[121] A. Chlebicki and P. Jakubczyk, "Analyticity of critical exponents of the $O(N)$ models from nonperturbative renormalization," *SciPost Phys.* **10** no. 6, (2021) 134, arXiv:2012.00782 [cond-mat.stat-mech].

[122] A. Aharony and E. Pytte, "First- and second-order transitions in the Potts model near four dimensions," *Phys. Rev. B* **23** (Jan, 1981) 362–367.

[123] K. E. Newman, E. K. Riedel, and S. Muto, "Q state Potts model by Wilson's exact renormalization group equation," *Phys. Rev. B* **29** (1984) 302–313.

[124] M. V. Kompaniets, A. Kudlis, and A. I. Sokolov, "Six-loop ϵ expansion study of three-dimensional $O(n) \times O(m)$ spin models," *Nucl. Phys. B* **950** (2020) 114874, arXiv:1911.01091 [cond-mat.stat-mech].

[125] Y. Nakayama and T. Ohtsuki, "Bootstrapping phase transitions in QCD and frustrated spin systems," *Phys. Rev. D* **91** no. 2, (2015) 021901, arXiv:1407.6195 [hep-th].

[126] B. Delamotte, D. Mouhanna, and M. Tissier, "Nonperturbative renormalization group approach to frustrated magnets," *Phys. Rev. B* **69** (2004) 134413, arXiv:cond-mat/0309101.

[127] B. Delamotte, M. Dudka, Y. Holovatch, and D. Mouhanna, "About the relevance of the fixed dimension perturbative approach to frustrated magnets in two and three dimensions," *Phys. Rev. B* **82** (2010) 104432, arXiv:1009.1492 [cond-mat.stat-mech].

[128] T. Garel and P. Pfeuty, "Commensurability effects on the critical behaviour of systems with helical ordering," *Journal of Physics C: Solid State Physics* **9** no. 10, (May, 1976) L245.

[129] A. Pelissetto, P. Rossi, and E. Vicari, "The Critical behavior of frustrated spin models with noncollinear order," *Phys. Rev. B* **63** (2001) 140414, arXiv:cond-mat/0007389.

[130] P. Calabrese, P. Parruccini, and A. I. Sokolov, "Chiral phase transitions: Focus driven critical behavior in systems with planar and vector ordering," *Phys. Rev. B* **66** (2002) 180403, arXiv:cond-mat/0205046.

[131] P. Calabrese, P. Parruccini, and A. I. Sokolov, "Critical thermodynamics of three-dimensional chiral model for N greater than 3," *Phys. Rev. B* **68** (2003) 094415, arXiv:cond-mat/0304154.

[132] P. Calabrese, P. Parruccini, A. Pelissetto, and E. Vicari, "Critical behavior of $O(2) \times O(N)$ symmetric models," *Phys. Rev. B* **70** (2004) 174439, arXiv:cond-mat/0405667.

[133] D. J. Wallace and R. K. P. Zia, "Gradient Flow and the Renormalization Group," *Phys. Lett. A* **48** (1974) 325.

[134] D. J. Wallace and R. K. P. Zia, "Gradient Properties of the Renormalization Group Equations in Multicomponent Systems," *Annals Phys.* **92** (1975) 142.

[135] Y. Nagano, K. Uematsu, and H. Kawamura, "Monte Carlo study of the critical properties of noncollinear Heisenberg magnets: $O(3) \times O(2)$ universality class," *Phys. Rev. B* **100** no. 22, (2019) 224430, arXiv:1910.13112 [cond-mat.str-el].

[136] V. T. Ngo and H. T. Diep, "Stacked triangular XY antiferromagnets: End of a controversial issue on the phase transition," *Journal of Applied Physics* **103** no. 7, (2008) 07C712.

[137] V. T. Ngo and H. T. Diep, "Phase transition in Heisenberg stacked triangular antiferromagnets: End of a controversy," *Phys. Rev. E* **78** (Sep, 2008) 031119.

[138] M. Tissier, B. Delamotte, and D. Mouhanna, "Heisenberg frustrated magnets: A Nonperturbative approach," *Phys. Rev. Lett.* **84** (2000) 5208–5211, arXiv:cond-mat/0001350.

[139] B. Delamotte, Y. Holovatch, D. Ivaneyko, D. Mouhanna, and M. Tissier, "Spurious fixed points in frustrated magnets," 2006. https://arxiv.org/abs/cond-mat/0609285.

[140] B. Sirois, "Navigating through the $O(N)$ archipelago," *SciPost Phys.* **13** no. 4, (2022) 081, arXiv:2203.11597 [hep-th].

[141] J. Henriksson, S. R. Kousvos, and M. Reehorst, "Spectrum continuity and level repulsion: the Ising CFT from infinitesimal to finite ε," *JHEP* **02** (2023) 218, arXiv:2207.10118 [hep-th].

[142] P. Calabrese, A. Pelissetto, and E. Vicari, "Multicritical behavior in frustrated spin systems with noncollinear order," *Nucl. Phys. B* **709** (2005) 550–577, arXiv:cond-mat/0408130.

[143] S. Gukov, "RG Flows and Bifurcations," *Nucl. Phys. B* **919** (2017) 583–638, arXiv:1608.06638 [hep-th].

[144] V. Gorbenko, S. Rychkov, and B. Zan, "Walking, Weak first-order transitions, and Complex CFTs," *JHEP* **10** (2018) 108, arXiv:1807.11512 [hep-th].

[145] A. Pelissetto, P. Rossi, and E. Vicari, "Large n critical behavior of $O(n) \times O(m)$ spin models," *Nucl. Phys. B* **607** (2001) 605–634, arXiv:hep-th/0104024.

[146] Z. Li, "Bootstrapping conformal QED_3 and deconfined quantum critical point," *JHEP* **11** (2022) 005, arXiv:1812.09281 [hep-th].

[147] S. R. Kousvos and A. Stergiou, "CFTs with $U(m) \times U(n)$ Global Symmetry in 3D and the Chiral Phase Transition of QCD," arXiv:2209.02837 [hep-th].

[148] 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, arXiv:2201.02206 [hep-th].

[149] A. Liu, D. Simmons-Duffin, N. Su, and B. van Rees, "Skydiving to Bootstrap Islands," *(to appear)* .

[150] N. Su, "simpleboot: A mathematica framework for bootstrap calculations," 2019. https://gitlab.com/bootstrapcollaboration/simpleboot.

[151] C. Behan, "PyCFTBoot: A flexible interface for the conformal bootstrap," *Commun. Comput. Phys.* **22** no. 1, (2017) 1–38, arXiv:1602.02810 [hep-th].

[152] J. C. Le Guillou and J. Zinn-Justin, "Accurate critical exponents from the epsilon expansion," *J. Physique Lett.* **46** no. 4, (1985) 137–141.

[153] F. Gliozzi, A. L. Guerrieri, A. C. Petkou, and C. Wen, "The analytic structure of conformal blocks and the generalized Wilson-Fisher fixed points," *JHEP* **04** (2017) 056, arXiv:1702.03938 [hep-th].

[154] J. Henriksson and M. Van Loon, "Critical $O(N)$ model to order ϵ^4 from analytic bootstrap," *J. Phys. A* **52** no. 2, (2019) 025401, arXiv:1801.03512 [hep-th].

[155] W. Landry, "scalar_blocks," 2018. https://gitlab.com/bootstrapcollaboration/scalar_blocks.

[156] J. A. Gracey, "Chiral exponents in $O(N) \times O(m)$ spin models at $O(1/N^2)$," *Phys. Rev. B* **66** (2002) 134402, arXiv:cond-mat/0208309.

[157] J. A. Gracey, "Critical exponent omega at $O(1/N)$ in $O(N) \times O(m)$ spin models," *Nucl. Phys. B* **644** (2002) 433–450, arXiv:hep-th/0209053.

[158] S.-k. Ma, "Scaling Variables and Dimensions," *Phys. Rev. A* **10** no. 5, (1974) 1818.

[159] A. N. Vasiliev and A. S. Stepanenko, "A Method of calculating the critical dimensions of composite operators in the massless nonlinear sigma model," *Theor. Math. Phys.* **94** (1993) 471–481.

[160] M. Hogervorst, S. Rychkov, and B. C. van Rees, "Unitarity violation at the Wilson-Fisher fixed point in 4-ϵ dimensions," *Phys. Rev. D* **93** no. 12, (2016) 125025, arXiv:1512.00013 [hep-th].

RÉSUMÉ

Pour plus d'une décennie, le bootstrap conforme numérique s'est avéré être un outil indispensable à l'étude des théories des champs conformes (CFTs). Des problèmes de plus en plus difficiles et intéressants peuvent être abordés chaque année grâce à l'amélioration constante de son arsenal de méthodes numériques. Nous proposons dans cette thèse le développement d'une nouvelle méthode numérique, que nous appliquons ensuite à l'étude de questions qui auparavant auraient été pénibles à analyser. Nous introduisons des "fonctions de navigation". Ces fonctions donnent une mesure du degré auquel un ensemble de CFT data est permis ou exclu. On propose que cette mesure devrait orienter la recherche de régions permises de CFT data, ce qui devrait mener dans la plupart, sinon tous les cas, à des améliorations drastiques en performance.

Nous utilisons pour commencer le modèle d'Ising critique en 3d comme modèle jouet. Nous montrons comment la recherche d'un point permis, ou celle d'un point aux extrêmes de la frontière de la région permise, peut être traduite en un problème d'optimisation de la fonction de navigation. Pour résoudre ce problème, nous adaptons des algorithmes existants aux besoins du bootstrap. Quelconque algorithme d'optimisation efficace requerra la détermination de la valeur de la fonction à optimiser ainsi que certaines de ses dérivées. Nous démontrons que le calcul de la fonction de navigation est équivalent à la résolution d'un programme semidéfini (SDP) typique dans le bootstrap conforme, et que les dérivées peuvent être obtenues gratuitement par l'analyse de la perturbation de solutions optimales d'un SDP. Nous notons aussi que chaque fonction de navigation définie naturellement un point "le plus permis" à l'intérieur d'une région permise. Nous naviguons jusqu'à ce point pour le bootstrap mixte des opérateurs sigma et epsilon, et nous observons qu'il se trouve bien plus près du modèle d'Ising qu'un point permis générique.

Plusieurs questions intéressantes peuvent concerner non pas une seule CFT, mais une famille de CFTs. L'exemple typique d'une telle famille est le modèle $O(N)$ critique, et nous l'utilisons dans la deuxième partie de cette thèse pour démontrer comment la fonction de navigation peut être utilisée pour rendre l'étude de familles de CFTs plus efficace. La valeur de la fonction de navigation et de ses dérivées en son minimum pour un membre de la famille peuvent être utilisées pour prédire la position du minimum d'un membre voisin de la famille. Ceci permet de suivre efficacement la position de régions permises comme fonction de N (ou de quelconque autre paramètre externe, par exemple la dimensionalité de l'espace d). En utilisant ce truc, nous observons, à notre connaissance pour la première fois dans l'histoire du bootstrap conforme, la disparition d'une île permise. Nous observons que le modèle $O(N)$ critique en 3d disparaît lorsque $N \to 1^+$, phénomène qu'on explique par la perte sévère d'unitarité dans ce régime. Cette observation est en accord avec des méthodes perturbatives comme l'epsilon-expansion.

Nous finissons cette thèse en s'attaquant à un problème non résolu depuis des décennies dans le domaine des transitions de phase classiques: quelle est la nature (de premier ou second ordre) de la transition de phase des stacked triangular antiferromagnets? Les fluctuations critiques de ce modèle devraient être décrites par un lagrangien LGW avec symétrie globale $O(N) \times O(2)$, où N réfère au nombre de composantes des spins. Proche de la dimension critique supérieure, ce modèle possède une fenêtre conforme, où une CFT non triviale existe seulement au-dessus d'une valeur critique $N_c(d)$. Avec l'aide de la fonction de navigation, nous sommes capables de suivre efficacement le point de disparation de la solution unitaire aux équations de croisement à travers d. Nous sommes ainsi capables de générer une courbe critique qui peut être comparée à celles obtenues par d'autres approches théoriques et numériques.

MOTS CLÉS

Bootstrap, Théorie conforme des champs, Optimisation numérique, Fonction de navigation, Transitions de phase, Matériaux magnétiques non-colinéaires

ABSTRACT

For more than a decade, the numerical conformal bootstrap has proved to be an invaluable tool for the study of higher-dimensional Conformal Field Theories (CFTs). The conformal bootstrap program being so young, each year more and more intricate and interesting questions are tackled thanks to the consistent improvement of its arsenal of numerical methods. We propose in this thesis the development of a novel numerical method and in tandem, we make use of it to study questions previously out of reach. We introduce so-called "navigator functions". These functions are defined so as to give a measure of how allowed or disallowed a putative subset of CFT data is. We propose that this measure should be used to orient the search for allowed regions of CFT data, leading in most, if not all cases, to drastic improvements in efficiency.

We first use the 3d critical Ising model as a testing ground. We show how the search for any allowed point, or for an extremal point along the boundary of an allowed region, can be translated to an optimization problem for the navigator function. To solve this problem efficiently, we adapt existing optimization algorithms for the needs of the bootstrap. Any efficient optimization algorithm requires the computation of the function to be optimized, as well as some of its derivatives. We show that the value of a navigator function at any point in the space of CFT data is obtained by solving a semidefinite program (SDP) of the same type as is customary in the bootstrap, and that derivatives may be obtained, for free, through a clever analysis of the perturbation of the optimal solution to an SDP. One peculiar thing is that each navigator function naturally defines a "most allowed" point for a particular bootstrap setup. We flow to this point in the case of the mixed-correlator bootstrap of sigma and epsilon, and observe that it sits much closer to the true location of the Ising model than a generic allowed point.

Many interesting questions can also be asked not about a single CFT, but about a family of CFTs. A prototypical family of CFTs is the critical $O(N)$ model, and we choose it in the second part of this thesis to demonstrate how the navigator function can be used to study families of CFTs more efficiently. One can use the navigator function and its derivatives at the minimal navigator point of a particular member of the family to predict the location of the minimal navigator point of a neighboring member of the family. This allows to efficiently follow allowed islands through N (and also any other external parameters, for example the spacetime dimension d). Using this trick, we observe, to our knowledge for the first time in the numerical bootstrap program, the clear disappearance of an allowed island. We see that the 3d $O(N)$ model disappears as $N \to 1^+$, and we are able to understand this disappearance as caused by a severe loss of unitarity below $N = 1$. We are able to match this observation with perturbative field theory methods like the epsilon-expansion.

We finish this thesis by tackling a decades-old problem in the theory of classical phase transitions: what is the nature of the phase transition (whether first or second order) of the stacked triangular antiferromagnet, perhaps the simplest of all frustrated magnets? The critical modes of this model should be described by a multiscalar LGW Lagrangian with $O(N) \times O(2)$ symmetry, where N corresponds to the number of spin components. Near its upper critical dimension, it is well known that this model exhibits a conformal window, where a nontrivial CFT with $O(N) \times O(2)$ global symmetry exists only down to some critical value $N_c(d)$. With the help of the navigator function, we are able to efficiently follow the point of disappearance of the unitary solution to the bootstrap crossing equations accross d. We are thus able to obtain a full critical curve, which we compare with those obtained from other theoretical and computational approaches.

KEYWORDS

Bootstrap, Conformal Field Theory, Numerical optimization, Navigator function, Phase transitions, Non-collinear magnets