Carving out OPE space and precise $O(2)$ model critical exponents

Shai M. Chester,[a] **Walter Landry,**[1,b] **Junyu Liu,**[b,c] **David Poland,**[b,d]
David Simmons-Duffin,[b] **Ning Su**[e] **and Alessandro Vichi**[e,f]

[a] *Department of Particle Physics and Astrophysics, Weizmann Institute of Science,*
Rehovot, Israel

[b] *Walter Burke Institute for Theoretical Physics, Caltech,*
Pasadena, CA 91125, U.S.A.

[c] *Institute for Quantum Information and Matter, Caltech,*
Pasadena, CA 91125, U.S.A.

[d] *Department of Physics, Yale University,*
New Haven, CT 06520, U.S.A.

[e] *Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL),*
CH-1015 Lausanne, Switzerland

[f] *Department of Physics, University of Pisa,*
I-56127 Pisa, Italy

E-mail: iahs81@gmail.com, wlandry@caltech.edu, jliu2@caltech.edu,
david.poland@yale.edu, dsd@caltech.edu, suning1985@gmail.com,
alessandro.vichi@gmail.com

Abstract: We develop new tools for isolating CFTs using the numerical bootstrap. A "cutting surface" algorithm for scanning OPE coefficients makes it possible to find islands in high-dimensional spaces. Together with recent progress in large-scale semidefinite programming, this enables bootstrap studies of much larger systems of correlation functions than was previously practical. We apply these methods to correlation functions of charge-0, 1, and 2 scalars in the 3d $O(2)$ model, computing new precise values for scaling dimensions and OPE coefficients in this theory. Our new determinations of scaling dimensions are consistent with and improve upon existing Monte Carlo simulations, sharpening the existing decades-old 8σ discrepancy between theory and experiment.

---

[1]Simons Collaboration on the Nonperturbative Bootstrap.

Contents

1 Introduction

1.1 Large-scale bootstrap problems

Numerical bootstrap methods [1, 2] (see [3, 4] for recent reviews) can help achieve two important goals: (1) make general statements about the space of all CFTs, and (2) isolate specific theories and compute their observables to high precision. In this work, we introduce new tools for isolating theories, and apply them to the 3d critical O(2) model.

To isolate a theory with the numerical bootstrap, one must choose a set of crossing symmetry equations and make reasonable assumptions about the spectrum of the theory. By analyzing the crossing equations using convex optimization, one obtains exclusion plots in the space of CFT data. In favorable circumstances, such exclusion plots contain small islands around the theory of interest — we then say that we have "isolated" the theory [5–9]. It is unknown in general which crossing equations and assumptions are needed to isolate a given theory. However, it is clearly important to incorporate as much information about the target theory as possible. In practice, this means we would like to study large systems of correlation functions involving multiple scalars [10–15], fermions [16–18], currents [19, 20], stress tensors [21], various global symmetry representations [22–42], etc.. There are many indications that such large-scale bootstrap problems could help isolate myriad interesting theories.[1]

Until recently, our ability to study large systems of correlation functions has been limited. One tool that will facilitate going beyond previous studies is a new version of the semidefinite program solver SDPB [82], which can now run on hundreds of cores across multiple machines [83].

Besides solving big semidefinite programs, another issue that arises in large-scale bootstrap studies is the difficulty of searching high-dimensional spaces. More crossing equations are parametrized by more input data, including scaling dimensions and OPE coefficients. If some input data is unknown, then we must scan over it to make an exclusion plot. For example, to study correlation functions of the scalars σ and ϵ in the 3d Ising model, we must scan over their scaling dimensions Δ_σ and Δ_ϵ. It was shown in [7] that it is also beneficial to scan over the OPE coefficient ratio $\lambda_{\sigma\sigma\epsilon}/\lambda_{\epsilon\epsilon\epsilon}$. Specifically, the island in the space of scaling dimensions and OPE coefficient ratios is smaller than the island in the space of scaling dimensions alone. To study an even larger system of correlation functions, one must scan over an even larger set of scaling dimensions and OPE coefficients.

One of the main contributions in this work is an efficient "cutting surface" algorithm for scanning over OPE coefficients. Because OPE coefficients enter quadratically in the crossing equations, our algorithm can scan a region of volume V in OPE coefficient space in time $\log V$. We also explain how to use our algorithm in conjunction with hot-starting [84], and introduce efficient methods for scanning over scaling dimensions.

We apply our methods to study correlation functions of the lowest-dimension charge-0, charge-1, and charge-2 scalars in the three-dimensional critical O(2) model. The 3d O(2)

---

[1]See for instance [43–51] or [52–74] for supersymmetric studies. Other analysis can also be found in [75–81].

CFT data	method	value	ref.
Δ_s	EXP	1.50946(22)	[85]
	MC	1.51122(15)	[86]
	CB	1.51136(**22**)	
Δ_ϕ	MC	0.519050(40)	[86]
	CB	0.519088(**22**)	
Δ_t	MC	1.2361(11)	[87]
	CB	1.23629(**11**)	
$\lambda_{\phi\phi s}$	CB	0.687126(27*)	
λ_{sss}	CB	0.830914(32*)	
λ_{tts}	CB	1.25213(14*)	
$\lambda_{\phi\phi t}$	CB	1.213408(65*)	
C_J/C_J^{free}	CB	0.904395(28*)	
C_T/C_T^{free}	CB	0.944056(15*)	

Table 1. Comparison of conformal bootstrap (CB) results with previous determinations from Monte Carlo (MC) or experiment (EXP). We denote the leading charge 0, 1, and 2 scalars by s, ϕ, t, respectively. Bold uncertainties correspond to rigorous intervals from bootstrap bounds. Uncertainties marked with a $*$ indicate that the value is estimated non-rigorously by sampling points, see sections 4.2 and 4.3.

model is one of the most studied renormalization group (RG) fixed points, both theoretically and experimentally. It describes phase transitions in numerous physical systems, including ferromagnets and antiferromagnets with easy-plane anisotropy, from which it also inherits the name of the XY universality class. Unfortunately, experimental results and Monte Carlo results for the critical exponents of the O(2) model have been in 8σ tension for two decades. We have computed the critical exponents to high precision (with rigorous error bars). We find excellent agreement with Monte Carlo results, and a clear discrepancy with experiment. In addition, we compute numerous other scaling dimensions and OPE coefficients in the O(2) model. Our results, together with comparisons to other methods, are summarized in table 1.

1.2 Experimental and theoretical approaches to the 3d O(2) model

In the remainder of this introduction, we provide an account of past approaches to the 3d O(2) model, including a history of the discrepancy between experiment and Monte Carlo. We also describe past bootstrap studies of the O(2) model and motivate the calculation in this work.

The simplest continuum field theory in the O(2) universality class is the theory of a scalar field $\vec{\phi}$ transforming in the fundamental representation of O(2), with Lagrangian

$$\mathcal{L} = \frac{1}{2}|\partial\vec{\phi}|^2 + \frac{1}{2}m^2|\vec{\phi}|^2 + \frac{g}{4!}|\vec{\phi}|^4. \tag{1.1}$$
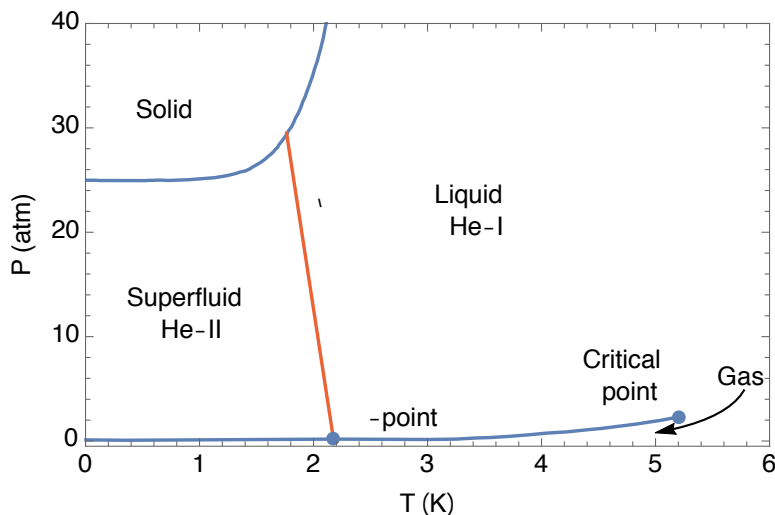
Figure 1. Schematic representation of the ^{4}He phase diagram. Figure taken from [88].

A large negative mass-squared for the scalar induces spontaneous symmetry breaking, leading to the ordered phase, while a large positive mass-squared leads to the disordered phase. The critical point is achieved by tuning the UV mass so that the IR correlation length diverges. Critical exponents are linked to operator dimensions at the fixed point by the simple relations

$$\Delta_\phi = \frac{1+\eta}{2}\,, \qquad \Delta_s = 3 - \frac{1}{\nu}\,. \tag{1.2}$$

Here, $s \sim |\vec{\phi}|^2$ denotes the lowest-dimension charge-0 scalar.

1.2.1 The λ-point experiment

Perhaps the most intriguing experimental representative of the O(2) universality class is the superfluid transition in ^{4}He along the so-called λ-line, see figure 1. Several features make this system ideal for experimental tests of critical phenomena. Firstly, the transition is second-order along the entire λ-line. This should be compared, for instance, with the liquid-vapor transition in water[2] where the critical point occurs at a single point on the temperature-pressure plane.[3] Secondly, the steep slope of the λ-line makes the critical temperature weakly dependent on the pressure. Thirdly, that compressibility is weakly divergent at the critical point and one side of the phase transition is a superfluid state (thus free of temperature gradients) renders the system less subject to gravitational effects, which still represent the major limitation for Earth-bound experiments.

The Earth's gravitational field creates a challenge for precise measurements of critical points in fluids. Gravity has two main effects [89]: (1) it induces a density gradient in the

---

[2]Which however belongs to the Ising universality class.

[3]The reason that the critical regime of liquid ^{4}He has codimension-1 on the temperature-pressure plane is that O(2) symmetry is present microscopically. It arises from phase rotations of the collective wavefunction of the superfluid condensate, which is an exact symmetry. This symmetry protects against deformations by the ϕ operator, and allows only a single relevant deformation: the lowest-dimension charge-0 scalar s.

fluid, making the system inhomogeneous; (2) more dramatically, it prevents fluctuations from growing indefinitely, making the correlation length effectively finite (gravitational rounding). Because of gravitational effects, most Earth-bound critical systems can only be tuned to $|t| \gtrsim 10^{-4}$, where t is the reduced temperature $t = 1 - T/T_c$. Due to its favorable properties, the superfluid transition of ^{4}He can instead reach $|t| \simeq 10^{-7}$. To get even closer to the critical regime, the λ-point experiment was conducted on the Space Shuttle Columbia in 1992 [90]. The micro-gravity environment allowed the experiment to reach $|t| \simeq 5 \times 10^{-9}$. In [85, 91], by fitting measurements from the λ-point experiment, the following value of the critical exponent ν was obtained:

$$\nu^{\text{EXP}} = 0.6709(1) \,. \tag{1.3}$$

1.2.2 Monte Carlo results

Over the past few decades, Monte Carlo (MC) simulations of lattice models in the O(2) universality class have provided the most precise theoretical predictions for critical exponents in the O(2) model. The most recent determination using purely MC techniques [86] gives

$$\nu^{\text{MC}} = 0.67169(7) \,. \tag{1.4}$$

We refer to [92] and references therein for older results.[4] The above value is fully in agreement with the second most precise theoretical determination of ν: in [96] MC simulations were combined with (uncontrolled) high temperature (HT) expansion[5] computations to obtain

$$\nu^{\text{MC+HT}} = 0.6717(1) \,. \tag{1.5}$$

Unfortunately, comparison of the MC results (1.4) and (1.5) with the experimental determination (1.3) reveals a large discrepancy of approximatively 8σ. The obvious questions is: which one is correct?

1.2.3 The conformal bootstrap

The numerical conformal bootstrap offers a rigorous and independent method to resolve this controversy. Three dimensional O(N)-models were first studied with bootstrap methods

---

[4]A determination that post-dates the review [92] is $\nu^{\text{MC}} = 0.6717(3)$ in [93]. A more recent computation using pseudo-ϵ expansion methods was performed in [94] giving $\nu^{\text{p}\epsilon} = 0.6706(12)$, which is closer to the experimental result. Another determination was recently obtained in [95] using only MC techniques, $\nu^{\text{MC}} = 0.67183(18)$. The latter determination and the value in (1.4) do not entirely overlap.

[5]In the HT expansion the generating functional

$$Z(J) = \sum_{\langle ij \rangle} e^{-\beta H + \vec{J}_i \cdot \vec{S}_i}$$

is expanded in powers of the inverse temperature β. Each term in the expansion can then be interpreted as a graphical sum. Each graph consists of vertices (lattice sites) connected by bonds, each of which is associated with a factor β. The graphs enumeration becomes a combinatoric problem and can be automatized (see [92] for a list of available HT series). Once the series is known to a sufficiently large order, it can be Borel resummed and extended down to the critical temperature. In [96] they used a 22nd order expansion.

in [25] by considering the correlation function $\langle \phi_i \phi_j \phi_k \phi_l \rangle$, where ϕ_i is the lowest-dimension scalar transforming in the vector representation of O(N). That work showed that the O(N)-models occupy special places in the space of three dimensional CFTs: they saturate bounds on scalar operator dimensions, and their presence is signaled by a "kink" in those bounds: a change of slope along an otherwise smooth boundary.

A rigorous determination of the critical exponents ν and η was later obtained in [6] by studying all nontrivial four-point functions containing ϕ_i and the lowest-dimension singlet scalar s, furthermore imposing that ϕ_i and s are the only relevant scalars with their respective O(N) representations. The resulting bounds on (Δ_ϕ, Δ_s) carve out an isolated island where the O(N) model lives, together with a detached region where all other O(N)-symmetric CFTs satisfying these relevancy assumptions must live.

The computation of [6] was further improved for the cases $N = 1, 2, 3$ in [7]. The latter work used essentially the same setup of [6], but additionally explored the power of scanning over OPE coefficients. Specifically, the authors asked the following question: in the space $(\Delta_\phi, \Delta_s, \theta)$, where θ parametrizes the ratio between two three point functions coefficients $\tan(\theta) \simeq \lambda_{sss}/\lambda_{\phi\phi s}$, what is the region consistent with crossing symmetry? It turned out that this apparently simple upgrade has a huge effect, but still not enough to make a conclusive statement about the MC/experiments discrepancy.

A complementary approach for the case $N = 2$ was initiated in [20], which studied the system of correlators involving the field ϕ_i and the conserved current associated to the global O(2) symmetry. Although the determination of critical exponents was not competitive with previous bootstrap analysis, this framework gives access to new CFT-data, in particular quantities related to transport properties near the quantum critical point.[6]

In this work, we study a larger system of correlation functions using numerical bootstrap techniques: in addition to ϕ_i and s, we incorporate the lowest-dimension charge-2 scalar $t_{ij} \sim \phi_{(i}\phi_{j)}$. A motivation for this choice is the idea that there exist strong constraints among the low-twist data of a CFT. For example, in [97, 98], it was shown using the lightcone bootstrap that crossing symmetry for the operators σ, ϵ in the 3d Ising model can be approximately recast as a set of constraints for a small amount of low-twist data, namely $\Delta_\sigma, \Delta_\epsilon, \lambda_{\sigma\sigma\epsilon}, \lambda_{\epsilon\epsilon\epsilon}$, and c_T. This immediately points to a deficiency in previous bootstrap studies of the O(2) model. The operator t_{ij} is expected to have lower dimension than s ($\Delta_t \approx 1.2$, while $\Delta_s \approx 1.5$). Thus, it makes sense to include it in the set of crossing equations we study.

As mentioned in section 1.1, studying the larger set of crossing equations involving $\{\phi, s, t\}$ requires searching over more input data: the operator dimensions $\{\Delta_\phi, \Delta_s, \Delta_t\}$, and the OPE coefficients $\{\lambda_{sss}, \lambda_{\phi\phi s}, \lambda_{tts}, \lambda_{\phi\phi t}\}$ (more precisely their ratios). Our new search algorithms are crucial for scanning this space efficiently. In figure 2, we show the resulting island in the space of scaling dimensions $\Delta_\phi, \Delta_s, \Delta_t$, and compare to Monte Carlo and experimental determinations. Our determination is consistent with Monte Carlo simulations and inconsistent with the results of the λ-point experiment.

---

[6]As a future direction it would be very interesting to combine this analysis with the techniques developed in this work to study the mixed system of a conserved current and multiple scalars.
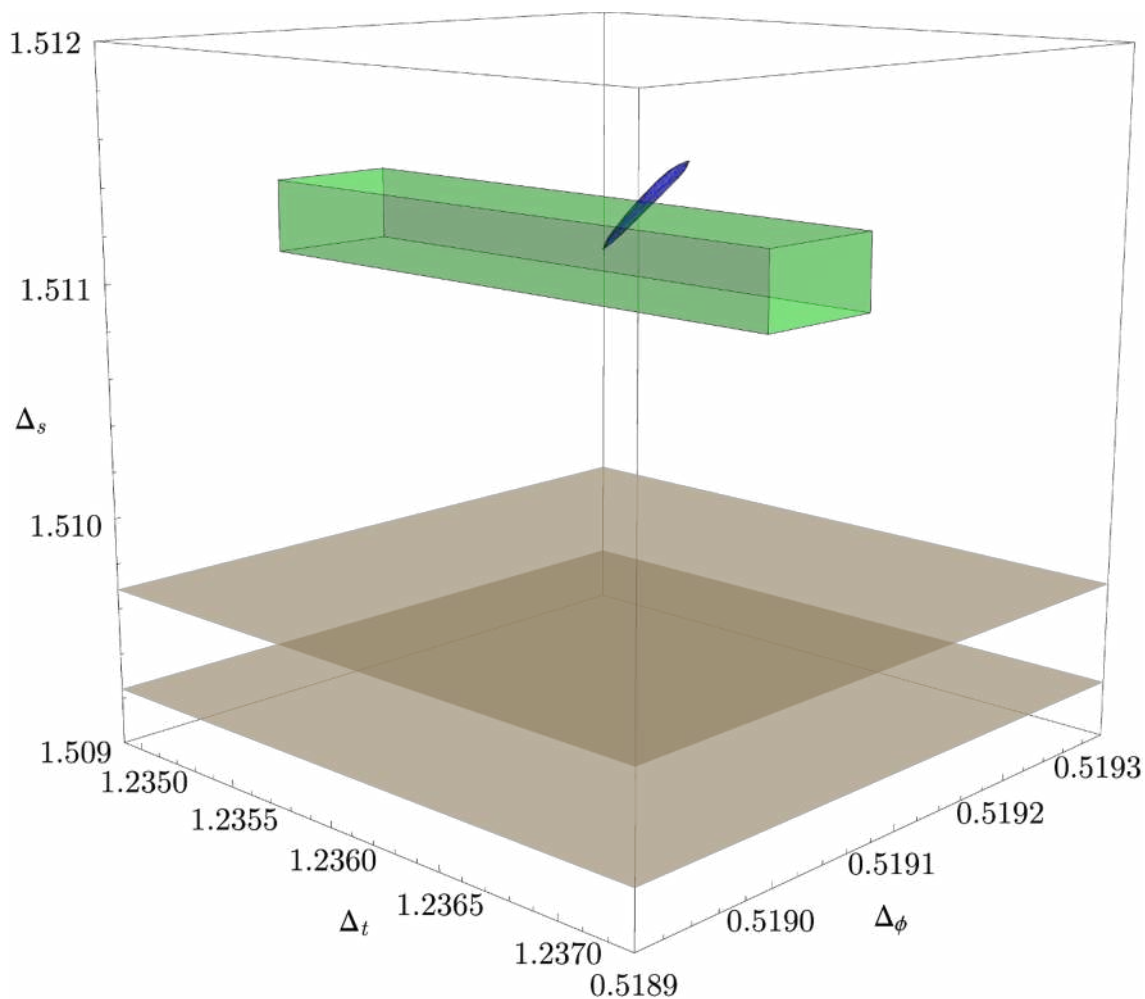
Figure 2. 3d region corresponding to our new O(2) island using the $\{\phi_i, s, t_{ij}\}$ system and OPE scans at $\Lambda = 43$ (blue). The result is compared with the best fit values of Δ_s to ^{4}He data [85] (brown planes) and the region for $\{\Delta_\phi, \Delta_s, \Delta_t\}$ reported by the Monte Carlo studies [86, 87] (green box).

1.3 Structure of this work

This work is structured as follows. In section 2, we describe the system of correlation functions in the O(2) model that we study, together with previously known information about its spectrum. In section 3, we introduce new search methods: a "cutting surface" algorithm for scanning over OPE coefficients, tricks for hot-starting, and Delaunay-triangulation methods for searching in dimension space. In section 4, we present results for scaling dimensions and OPE coefficients in the O(2) model. Appendix A provides links to the code used in this work and appendix B contains technical details of our software and hardware setup. Other appendices provide details about the crossing equations of the O(2) model and specific points that we have tested.

2 The O(2) model

2.1 Crossing equations

We begin by describing the representation theory of O(2) $\cong$ U(1) $\ltimes \mathbb{Z}_2$. The irreducible representations of O(2) are:

- The trivial representation $\mathbf{0}^+$.

- The sign representation $\mathbf{0}^-$, in which U(1) acts trivially and the nontrivial element of $\mathbb{Z}_2$ acts by -1.

- For each $q \in \mathbb{Z}_{>0}$, a unique two-dimensional irreducible representation $\mathbf{q}$. The states of $\mathbf{q}$ have U(1) charges $\pm q$ and are exchanged by $\mathbb{Z}_2$.

Tensor products of these irreps are given by

$$
\begin{aligned}
\mathbf{q_1} \otimes \mathbf{q_2} &= (\mathbf{q_1} + \mathbf{q_2}) \oplus |\mathbf{q_1} - \mathbf{q_2}| \,, \\
\mathbf{q} \otimes \mathbf{q} &= (\mathbf{2q})_s \oplus \mathbf{0}_s^+ \oplus \mathbf{0}_a^- \,, \\
\mathbf{0}^{\pm} \otimes \mathbf{q} &= \mathbf{q} \,, \\
\mathbf{0}^{\pm} \otimes \mathbf{0}^{\pm} &= \mathbf{0}_s^+ \,, \\
\mathbf{0}^{\pm} \otimes \mathbf{0}^{\mp} &= \mathbf{0}^- \,,
\end{aligned}
\tag{2.1}
$$

where s/a denotes the symmetric/antisymmetric part of the tensor product, in the case of identical irreps. For any irrep $\mathcal{R}$ of O(2), we define q as the highest U(1) charge in the representation.

Operators $\mathcal{O}_{\mathbf{q}}(x)$ in irrep $\mathbf{q}$ can be written in terms of O(2) fundamental indices $i = 1, 2$ as rank-q symmetric traceless tensors $\mathcal{O}^{i_1 \cdots i_q}(x)$. It is convenient to contract these with auxiliary polarization vectors y^i that are defined to be null, $y \cdot y = 0$, so that

$$
\mathcal{O}(x, y) \equiv \mathcal{O}^{i_1 \cdots i_q}(x) y_{i_1} \cdots y_{i_q} \,.
\tag{2.2}
$$

The singlet operator $\mathcal{O}_{\mathbf{0}^+}(x)$ has no indices or y's. The $\mathbb{Z}_2$-odd operator $\mathcal{O}_{\mathbf{0}^-}(x)$ could be written with antisymmetric indices $\mathcal{O}^{[i_1 i_2]}(x)$. Alternatively, we can take into account the O(2) dependence of correlation functions that include $\mathcal{O}_{\mathbf{0}^-}(x)$ in an index-free manner by requiring that all pairs of distinct y_1, y_2 must be contracted as

$$
y_1 \cdot y_2 \equiv y_1^i y_2^j \delta_{ij} \,, \qquad y_1 \wedge y_2 \equiv \epsilon_{ij} y_1^i y_2^j \,,
\tag{2.3}
$$

where the number of $\wedge$'s must be zero/one if an even/odd number of $\mathcal{O}_{\mathbf{0}^-}(x)$'s appear.

Tensor structures for correlation functions of charged operators can be factorized into "flavor" tensor structures for the O(2) polarization vectors y_i and "kinematic" tensor structures that encode spacetime dependence. For two-point functions, we have

$$
\langle \mathcal{O}^{\mu_1 \cdots \mu_J}(x_1, y_1) \mathcal{O}_{\nu_1 \cdots \nu_J}(x_2, y_2) \rangle = c_{\mathcal{O}} (y_1 \cdot y_2)^q \frac{I_{(\nu_1}^{(\mu_1}(x_{12}) \cdots I_{\nu_J)}^{\mu_J)}(x_{12})}{x_{12}^{2\Delta}} \,,
\tag{2.4}
$$

$$
I_\nu^\mu(x) = \delta_\nu^\mu - \frac{2 x^\mu x_\nu}{x^2} \,,
$$

where Δ and J are the dimension and spin of $\mathcal{O}$, and q is the maximal U(1) charge of the O(2) representation of $\mathcal{O}$. Here, $c_{\mathcal{O}}$ is a constant that we usually set to 1.

The most general three-point function we need in this work is between two scalars and a spin-J operator. It takes the form

$$
\langle \varphi_1(x_1, y_1) \varphi_2(x_2, y_2) \mathcal{O}_3^{\mu_1 \cdots \mu_J}(x_3, y_3) \rangle
$$
$$
= \lambda_{\varphi_1 \varphi_2 \mathcal{O}_3} T_{\mathcal{R}_1 \mathcal{R}_2 \mathcal{R}_3}(y_1, y_2, y_3) \frac{Z^{(\mu_1} \cdots Z^{\mu_J)} - \text{traces}}{x_{12}^{\Delta_1 + \Delta_2 - \Delta_3} x_{23}^{\Delta_2 + \Delta_3 - \Delta_1} x_{31}^{\Delta_3 + \Delta_1 - \Delta_2}}, \tag{2.5}
$$

where

$$
Z^{\mu} = \frac{|x_{13}||x_{23}|}{|x_{12}|} \left(\frac{x_{13}^{\mu}}{x_{13}^2} - \frac{x_{23}^{\mu}}{x_{23}^2} \right). \tag{2.6}
$$

Here, $\mathcal{R}_i$ is the O(2) representation of the operator at position x_i. Our conventions for flavor three-point structures are

$$
T_{\mathbf{q_1 q_2 q_3}}(y_1, y_2, y_3) = (y_1 \cdot y_2)^{\frac{q_1 + q_2 - q_3}{2}} (y_2 \cdot y_3)^{\frac{q_2 + q_3 - q_1}{2}} (y_3 \cdot y_1)^{\frac{q_3 + q_1 - q_2}{2}}, \quad (q_1, q_2, q_3 > 0)
$$
$$
T_{\mathbf{qq0^+}}(y_1, y_2, y_3) = (y_1 \cdot y_2)^q,
$$
$$
T_{\mathbf{qq0^-}}(y_1, y_2, y_3) = (y_1 \wedge y_2)(y_1 \cdot y_2)^{q-1}, \qquad\qquad\qquad\qquad (q > 0)
$$
$$
T_{\mathbf{0^+ 0^+ 0^+}}(y_1, y_2, y_3) = 1, \tag{2.7}
$$

where we only list structures that will be needed below. In the first line, we have either $q_3 = q_1 + q_2$ or $q_3 = |q_1 - q_2|$, in accordance with the rules for tensor products.

In general, four-point functions of scalars operators $\varphi^i(x_i, y_i)$, where i here labels each operator that transforms in O(2) irrep $\mathcal{R}_i$, can be expanded in the s-channel in terms of conformal blocks as[7]

$$
\langle \varphi_{\mathcal{R}_1}^1(x_1, y_1) \varphi_{\mathcal{R}_2}^2(x_2, y_2) \varphi_{\mathcal{R}_3}^3(x_3, y_3) \varphi_{\mathcal{R}_4}^4(x_4, y_4) \rangle
$$
$$
= \frac{\left(\frac{x_{24}}{x_{14}} \right)^{\Delta_{12}} \left(\frac{x_{14}}{x_{13}} \right)^{\Delta_{34}}}{x_{12}^{\Delta_1 + \Delta_2} x_{34}^{\Delta_3 + \Delta_4}} \sum_{\mathcal{O}} (-1)^{\ell} \lambda_{\varphi_1 \varphi_2 \mathcal{O}} \lambda_{\varphi_3 \varphi_4 \mathcal{O}} T_{\mathcal{R}_1 \mathcal{R}_2 \mathcal{R}_3 \mathcal{R}_4}^{\mathcal{R}}(y_i) g_{\Delta, \ell}^{\Delta_{12}, \Delta_{34}}(u, v), \tag{2.8}
$$

where $\Delta_{ij} \equiv \Delta_i - \Delta_j$, the conformal cross ratios u, v are

$$
u \equiv \frac{x_{12}^2 x_{34}^2}{x_{13}^2 x_{24}^2}, \qquad v \equiv \frac{x_{14}^2 x_{23}^2}{x_{13}^2 x_{24}^2}, \tag{2.9}
$$

and the operators $\mathcal{O}$ that appear both OPEs $\varphi^1 \times \varphi^2$ and $\varphi^3 \times \varphi^4$ have scaling dimension Δ, spin ℓ, and transform in an irrep $\mathcal{R}$ that appears in both tensor products $\mathcal{R}_1 \otimes \mathcal{R}_2$ and $\mathcal{R}_3 \otimes \mathcal{R}_4$. For each $\mathcal{R}$, the O(2) structure $T_{\mathcal{R}_1 \mathcal{R}_2 \mathcal{R}_3 \mathcal{R}_4}^{\mathcal{R}}(y_i)$ is a polynomial in y_i for $i = 1, 2, 3, 4$, that can be derived from contracting appropriate 3-point functions as described in appendix C. If $\varphi^1 = \varphi^2$ (or $\varphi^3 = \varphi^4$), then Bose symmetry requires that $\mathcal{O}$ have only even/odd ℓ for R in the symmetric/antisymmetric product of $\mathcal{R}_1 \otimes \mathcal{R}_2$ (or $\mathcal{R}_3 \otimes \mathcal{R}_4$).

---

[7]Our conformal blocks are normalized as in the second line of table 1 in [3].

We are interested in four-point functions of the lowest dimension scalar operators transforming in the $\mathbf{0}^+$, $\mathbf{1}$, and $\mathbf{2}$ representations, which we will denote following [6, 7, 25] as s, ϕ, and t, respectively.[8] These operators are normalized via their two point functions as

$$\langle s(x_1)s(x_2)\rangle = \frac{1}{x_{12}^{2\Delta_s}}\,, \quad \langle \phi(x_1,y_1)\phi(x_2,y_2)\rangle = \frac{y_1\cdot y_2}{x_{12}^{2\Delta_\phi}}\,, \quad \langle t(x_1,y_1)t(x_2,y_2)\rangle = \frac{(y_1\cdot y_2)^2}{x_{12}^{2\Delta_t}}\,,$$

(2.10)

where $x_{12}\equiv|x_1-x_2|$. In table 2 we list the 4-point functions of s, ϕ, and t that are allowed by O(2) symmetry[9] whose s and t-channel configuration lead to independent crossing equations, along with the irreps and spins of the operators that appear in the OPE, and the number of crossing equations that they yield. These 4-point functions can be written explicitly as in (2.8), where the explicit O(2) structures $T^{\mathcal{R}}_{\mathcal{R}_1\mathcal{R}_2\mathcal{R}_3\mathcal{R}_4}(y_i)$ are computed in appendix C. Equating each of these s-channel 4-point functions with their respective t-channels yields the crossing equations

$$\sum_{\mathcal{O}_{\mathbf{0}^+},\ell^+}\left(\lambda_{ss\mathcal{O}_{\mathbf{0}^+}}\ \lambda_{\phi\phi\mathcal{O}_{\mathbf{0}^+}}\ \lambda_{tt\mathcal{O}_{\mathbf{0}^+}}\right)\vec{V}_{\mathbf{0}^+,\Delta,\ell^+}\begin{pmatrix}\lambda_{ss\mathcal{O}_{\mathbf{0}^+}}\\\lambda_{\phi\phi\mathcal{O}_{\mathbf{0}^+}}\\\lambda_{tt\mathcal{O}_{\mathbf{0}^+}}\end{pmatrix}+\sum_{\mathcal{O}_{\mathbf{0}^-},\ell^-}\left(\lambda_{\phi\phi\mathcal{O}_{\mathbf{0}^-}}\ \lambda_{tt\mathcal{O}_{\mathbf{0}^-}}\right)\vec{V}_{\mathbf{0}^-,\Delta,\ell^-}\begin{pmatrix}\lambda_{\phi\phi\mathcal{O}_{\mathbf{0}^-}}\\\lambda_{tt\mathcal{O}_{\mathbf{0}^-}}\end{pmatrix}$$

$$+\sum_{\mathcal{O}_{\mathbf{1}},\ell^\pm}\left(\lambda_{\phi s\mathcal{O}_{\mathbf{1}}}\ \lambda_{t\phi\mathcal{O}_{\mathbf{1}}}\right)\vec{V}_{\mathbf{1},\Delta,\ell^\pm}\begin{pmatrix}\lambda_{\phi s\mathcal{O}_{\mathbf{1}}}\\\lambda_{t\phi\mathcal{O}_{\mathbf{1}}}\end{pmatrix}+\sum_{\mathcal{O}_{\mathbf{2}},\ell^+}\left(\lambda_{\phi\phi\mathcal{O}_{\mathbf{2}}}\ \lambda_{ts\mathcal{O}_{\mathbf{2}}}\right)\vec{V}_{\mathbf{2},\Delta,\ell^+}\begin{pmatrix}\lambda_{\phi\phi\mathcal{O}_{\mathbf{2}}}\\\lambda_{ts\mathcal{O}_{\mathbf{2}}}\end{pmatrix}$$

$$+\sum_{\mathcal{O}_{\mathbf{2}},\ell^-}\lambda_{ts\mathcal{O}_{\mathbf{2}}}^2\vec{V}_{\mathbf{2},\Delta,\ell^-}+\sum_{\mathcal{O}_{\mathbf{3}},\ell^\pm}\lambda_{t\phi\mathcal{O}_{\mathbf{3}}}^2\vec{V}_{\mathbf{3},\Delta,\ell^\pm}+\sum_{\mathcal{O}_{\mathbf{4}},\ell^+}\lambda_{tt\mathcal{O}_{\mathbf{4}}}^2\vec{V}_{\mathbf{4},\Delta,\ell^+}=0,$$

(2.11)

where $\ell^\pm$ denotes which spins appear, and the V's are 22-dimensional vectors of matrix or scalar crossing equations that are ordered as table 2 and written in terms of

$$F^{ij,kl}_{\mp,\Delta,\ell}(u,v)=v^{\frac{\Delta_k+\Delta_j}{2}}g^{\Delta_{ij},\Delta_{kl}}_{\Delta,\ell}(u,v)\mp u^{\frac{\Delta_k+\Delta_j}{2}}g^{\Delta_{ij},\Delta_{kl}}_{\Delta,\ell}(v,u)\,.$$

(2.12)

The explicit form of the V's are given in appendix D. The same crossing equations were derived and studied independently in [84].[10]

2.2 Assumptions about the spectrum

To obtain precise results for the O(2) model, we must input some restrictions on its spectrum and OPE coefficients in order to isolate the theory. Firstly, we impose that s,ϕ,t are the only relevant scalars in their respective charge sectors. In other words, we im-

---

[8]The singlet S, traceless symmetric T, vector V and antisymmetric A irreps considered in previous O(N) bootstrap papers [6, 7, 25] correspond for O(2) to the $\mathbf{0}^+$, $\mathbf{2}$, $\mathbf{1}$, and $\mathbf{0}^-$ irreps, respectively.

[9]These 4-point functions, and the resulting crossing equations, are identical for a theory with just SO(2) symmetry. The only difference between O(2) and SO(2) is that for the latter $\mathbf{0}^+\cong\mathbf{0}^-$ and ϵ_{ij} is now an invariant tensor, so one would need to consider correlators of operators with $\mathbf{0}^-$, such as $\langle\mathcal{O}_{\mathbf{0}^+}\mathcal{O}_{\mathbf{0}^+}\mathcal{O}_{\mathbf{0}^-}\mathcal{O}_{\mathbf{0}^-}\rangle$, to distinguish between O(2) and SO(2).

[10]Furthermore, [84] includes a software package `autoboot` that can automatically derive equation (2.11).

4-pnt	s-channel	t-channel	Eqs
$\langle\phi\phi\phi\phi\rangle$	$(\ell^+, \mathbf{0}^+), (\ell^-, \mathbf{0}^-), (\ell^+, \mathbf{2})$	same	3
$\langle tttt\rangle$	$(\ell^+, \mathbf{0}^+), (\ell^-, \mathbf{0}^-), (\ell^+, \mathbf{4})$	same	3
$\langle t\phi t\phi\rangle$	$(\ell^\pm, \mathbf{1}), (\ell^\pm, \mathbf{3})$	same	2
$\langle tt\phi\phi\rangle$	$(\ell^+, \mathbf{0}^+), (\ell^-, \mathbf{0}^-)$	$(\ell^\pm, \mathbf{1}),(\ell^\pm, \mathbf{3})$	4
$\langle ssss\rangle$	$(\ell^+, \mathbf{0}^+)$	same	1
$\langle\phi s\phi s\rangle$	$(\ell^\pm, \mathbf{1})$	same	1
$\langle tsts\rangle$	$(\ell^\pm, \mathbf{2})$	same	1
$\langle ttss\rangle$	$(\ell^+, \mathbf{0}^+)$	$(\ell^\pm, \mathbf{2})$	2
$\langle\phi\phi ss\rangle$	$(\ell^+, \mathbf{0}^+)$	$(\ell^\pm, \mathbf{1})$	2
$\langle\phi s\phi t\rangle$	$(\ell^\pm, \mathbf{1})$	same	1
$\langle s\phi\phi t\rangle$	$(\ell^\pm, \mathbf{1})$	$(\ell^+, \mathbf{2})$	2

Table 2. Four-point function configurations that give independent crossing equations under equating their s- and t-channel, along with the even/odd spins that appear for each irrep in each channel, and the number of crossing equations that each configuration yields.

pose that $\Delta \geq 3$ for all charge $0, 1, 2$ scalars after these operators.[11] These assumptions are well-supported by other techniques including Monte Carlo simulations and the ϵ-expansion.

The dimension of the second charge-0 operator s' is related to the critical exponent $\omega = \Delta_{s'} - 3$, which has been determined to be irrelevant using field theory and numerical techniques [86, 99, 100] (see also [92] for a list of less precise estimates). E.g. [86] gives $\Delta_{s'} = 3.789(4)$. Indeed, irrelevance of this operator is necessary in order to have a critical point rather than a multicritical point in which multiple tunings would be required.

For the second charge-1 scalar ϕ', we are not aware of any direct determination of its scaling dimension. However, in the ϵ-expansion one can show that the naïve second charge-1 operator, schematically $(\phi_k)^2\phi_a$, becomes a descendant of ϕ_a [101]. The next charge-1 operators after this are strongly irrelevant close to 4 dimensions, and we are not aware of any evidence that continuation to $\epsilon = 1$ could change this property. Also, Monte Carlo simulations do not show any evidence of a second charge-1 relevant perturbation, which would introduce a new order parameter.

To our knowledge, the dimension of the second charge-2 operator t' has only been determined in the ϵ-expansion [102] to be $\Delta_{t'} \simeq 3.624(10)$, making it squarely irrelevant. Additionally, if this operator corresponded to a relevant perturbation it would have been readily detected in Monte Carlo studies of anisotropic perturbations of the O(2) model [87].

The lowest-dimension charge-3 scalar in the O(2) model is expected to have dimension ≈ 2.1 [87, 103].[12] This value is actually very close to the upper bound imposed by a

---

[11]We also forbid any possibility of degenerate scalar contributions at the scaling dimensions $\Delta_{s,\phi,t}$, which would require additional symmetries and by definition place the model outside of the O(2) universality class. While they wouldn't be expected, degenerate contributions at other dimensions are not forbidden by our algorithm.

[12]We find that this is consistent with estimates based on the extremal functional method [78].

bootstrap analysis [11].[13] To reflect this, we impose a much weaker bound of $\Delta \geq 1$ for charge-3 scalars.

For charge-4 scalars, there is strong evidence from the ϵ-expansion [104, 105] and MC [87, 106] that there are no relevant charge-4 scalars in the O(2) model. E.g., the recent MC study [106] gives the precise determination $\Delta_{\text{charge 4}} = 3.114(2)$. To reflect this, in most of this work we will impose $\Delta \geq 3$ for charge-4 scalars (following an initial study which imposes the weaker condition $\Delta \geq 1$).

For reasons discussed in section 3.6, it is useful to impose a small gap $\delta\tau$ in twist $\tau = \Delta - \ell$ above the unitarity bound for the non-scalar operators in the theory. (The unitarity bound for non-scalars is $\tau \geq 1$.) Of course the spectrum must include the O(2) current J^μ and the stress tensor $T^{\mu\nu}$, so we impose the twist gap only for operators with dimensions above the current and stress tensor in their respective sectors. (We impose slightly different gaps in these sectors when computing upper bounds on C_T and C_J, as discussed in section 4.3.)

The presence of a small twist gap is expected to be valid in the O(2) model. In the charge-0 sector, Nachtmann's theorem [107–109], together with the existence of double-twist operators [108, 110], implies that leading twists τ_ℓ for each even $\ell \geq 4$ satisfy

$$1 \leq \tau_4 \leq \tau_\ell \leq 2\Delta_\phi \approx 1.04 \qquad (2.13)$$

Numerous methods, including the ϵ-expansion, the lightcone bootstrap, and the extremal functional method suggest that $\tau_4 \approx 1.02$. A result from [111] shows that minimal twists in the charge-2 and charge-4 sectors are equal to or larger than the minimal twist in the charge-0 sector, for each spin. For charges 1 and 3 and odd spins in the $\mathbf{0}^-$ representation, we can appeal to the ϵ-expansion which shows there are no higher-spin operators with twist near the unitarity bound. Thus, the assumption of a twist gap $\delta_\tau < 0.02$ is well-justified. In most of this work, we choose $\delta\tau = 10^{-6}$. Overall, our assumptions about the spectrum of the O(2) model are listed in table 3.

The OPE coefficients of J^μ and $T^{\mu\nu}$ are constrained by Ward identities in terms of the two-point coefficients C_J and C_T. In our conventions, we have

$$\lambda^2_{\mathcal{O}\mathcal{O}T} = \frac{\Delta_{\mathcal{O}}^2}{2C_T/C_T^{\text{free}}}, \quad \lambda^2_{\mathcal{O}\mathcal{O}J} = \frac{q_{\mathcal{O}}^2}{2C_J/C_J^{\text{free}}}, \qquad (2.14)$$

where $C_{J,T}^{\text{free}}$ are the two-point coefficients of J and T in the free O(2) model. Thus, the contribution of these operators to the crossing equation can be parametrized purely in terms of C_T and C_J, together with the dimensions and charges of the external scalars ϕ, s, t.

---

[13]More precisely the bound requires that given a charge-1 and charge-2 operator of dimension $(\Delta_\phi, \Delta_t) = (0.51905, 1.234)$, the OPE $\phi \times t$ must contain a charge-3 operator with dimension smaller than 2.118. Strictly speaking this bound does not apply to the O(2) model since this choice of dimensions turns out to be excluded. Nevertheless, by continuity, we expect the correct bound to be very close.

charge	spin	dimensions
0	0	Δ_s or $\Delta \geq 3$
1	0	Δ_ϕ or $\Delta \geq 3$
2	0	Δ_t or $\Delta \geq 3$
3	0	$\Delta \geq 1$
4	0	$\Delta \geq 3$
0	1	$\Delta = 2$ or $\Delta \geq 2 + \delta_\tau$
0	2	$\Delta = 3$ or $\Delta \geq 3 + \delta_\tau$
$\mathcal{R}$	ℓ	$\Delta \geq \ell + 1 + \delta_\tau$

Table 3. Typical assumptions about the spectrum of the O(2) model. In the last line, $\mathcal{R}, \ell$ represent any choices of representation $\mathcal{R}$ and spin ℓ not already represented in the table. A typical choice of twist gap is $\delta_\tau = 10^{-6}$.

3 Methods

3.1 Numerical bootstrap bounds

Given the crossing equations (2.11), we compute bounds on CFT quantities in the standard way described in [1, 5]. Suppose we would like to demonstrate that a hypothetical spectrum is inconsistent. We search for a linear functional α such that

$$\alpha(\vec{V}_{\mathbf{0^+},\Delta,\ell^+}) \succeq 0, \qquad \alpha(\vec{V}_{\mathbf{0^-},\Delta,\ell^-}) \succeq 0, \qquad \alpha(\vec{V}_{\mathbf{1},\Delta,\ell^\pm}) \succeq 0, \qquad \alpha(\vec{V}_{\mathbf{2},\Delta,\ell^+}) \succeq 0,$$
$$\alpha(\vec{V}_{\mathbf{2},\Delta,\ell^-}) \geq 0, \qquad \alpha(\vec{V}_{\mathbf{3},\Delta,\ell^\pm}) \geq 0, \qquad \alpha(\vec{V}_{\mathbf{4},\Delta,\ell^+}) \geq 0, \tag{3.1}$$

for all combinations of representations, dimensions Δ, and even or odd spins $\ell^\pm$ in some hypothetical spectrum. Here, "$M \succeq 0$" means "M is positive-semidefinite". It is conventional to normalize the contribution of the unit operator in the crossing equation to 1:

$$\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \alpha(\vec{V}_{\mathbf{0^+},0,0}) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1. \tag{3.2}$$

If a functional exists satisfying these conditions, then the hypothetical spectrum is ruled out. We search for a functional using SDPB [83].

3.2 Positivity conditions involving the external scalars s, ϕ, t

The external operators s, ϕ, t appearing in the crossing equations require special treatment when computing bootstrap bounds.[14] There are four nonvanishing OPE coefficients

---

[14] We use the term "external" to refer to operators that appear explicitly in the four-point functions being studied, as opposed to "internal" operators that appear in the conformal block expansion.

involving just s, ϕ, t. They can be grouped into a vector[15]

$$\lambda_{\text{ext}} \equiv \begin{pmatrix} \lambda_{sss} \\ \lambda_{\phi\phi s} \\ \lambda_{tts} \\ \lambda_{\phi\phi t} \end{pmatrix}. \tag{3.3}$$

We define the 4×4 symmetric matrices $\vec{V}_{\text{ext}}$ as the bilinear forms paired with λ_{ext} in the crossing equations. $\vec{V}_{\text{ext}}$ is given implicitly by

$$\lambda_{\text{ext}}^T \vec{V}_{\text{ext}} \lambda_{\text{ext}}$$
$$= \begin{pmatrix} \lambda_{sss} & \lambda_{\phi\phi s} & \lambda_{tts} \end{pmatrix} \vec{V}_{\mathbf{0}^+, \Delta_s, 0} \begin{pmatrix} \lambda_{sss} \\ \lambda_{\phi\phi s} \\ \lambda_{tts} \end{pmatrix} + \begin{pmatrix} \lambda_{\phi\phi s} & \lambda_{\phi\phi t} \end{pmatrix} \vec{V}_{\mathbf{1}, \Delta_\phi, 0} \begin{pmatrix} \lambda_{\phi\phi s} \\ \lambda_{\phi\phi t} \end{pmatrix} + \begin{pmatrix} \lambda_{\phi\phi t} & \lambda_{tts} \end{pmatrix} \vec{V}_{\mathbf{2}, \Delta_t, 0} \begin{pmatrix} \lambda_{\phi\phi t} \\ \lambda_{tts} \end{pmatrix}. \tag{3.4}$$

When computing bounds, we can treat the term $\lambda_{\text{ext}}^T \vec{V}_{\text{ext}} \lambda_{\text{ext}}$ in different ways, depending on our knowledge of λ_{ext}. If we know nothing about λ_{ext}, then we can search for a functional α such that

$$\alpha(\vec{V}_{\text{ext}}) \succeq 0, \tag{3.5}$$

where "$\succeq 0$" means "is positive semidefinite". In this way, we ensure that the contribution of external scalar OPE coefficients to the crossing equation has a definite sign after applying α, independent of the values of those coefficients. Imposing the condition (3.5), we can compute an allowed region $\mathcal{D}$ for other quantities like operator dimensions.

However, the condition (3.5) is stronger than necessary because it allows the matrix $M_{\text{ext}} \equiv \lambda_{\text{ext}} \lambda_{\text{ext}}^T$ to have rank larger than 1. Specifically, it ensures that $\text{Tr}(M_{\text{ext}} \alpha(\vec{V}_{\text{ext}})) \geq 0$ for M_{ext} of any rank. We would like a procedure that only imposes positivity when M_{ext} is a rank-1 matrix.

Such a procedure was described in [7, 112], and it results in stronger bounds. Suppose first that we know the direction of λ_{ext}. More precisely, suppose we know the equivalence class $[\lambda_{\text{ext}}] \in \mathbb{RP}^3$ of λ_{ext} under rescaling by a real number. In this case, the condition (3.5) is too strong, and it suffices to impose the weaker condition[16]

$$\lambda_{\text{ext}}^T \alpha(\vec{V}_{\text{ext}}) \lambda_{\text{ext}} \geq 0. \tag{3.6}$$

(Note that $\alpha(\vec{V}_{\text{ext}})$ is a 4×4 matrix, so that $\lambda_{\text{ext}}^T \alpha(\vec{V}_{\text{ext}}) \lambda_{\text{ext}}$ is a number.) This ensures that the contribution of external scalars to the crossing equation will be positive, independent of the magnitude or sign of λ_{ext}. If we use the weaker condition (3.6) to compute bounds on other quantities, we obtain an allowed region $\mathcal{D}_{[\lambda_{\text{ext}}]}$ that is smaller than $\mathcal{D}$, but depends on the equivalence class $[\lambda_{\text{ext}}] \in \mathbb{RP}^3$.

---

[15]Note that OPE coefficients of scalar operators are symmetric with respect to permutation $\lambda_{\phi_1\phi_2\phi_3} = \lambda_{\phi_1\phi_3\phi_2} =$ four other permutations.

[16]Here, λ_{ext} can be any representative of the equivalence class $[\lambda_{\text{ext}}]$.

If we don't know $[\lambda_{\text{ext}}]$ a-priori, we can scan over its value and compute the regions $\mathcal{D}_{[\lambda_{\text{ext}}]}$ as a function of $[\lambda_{\text{ext}}] \in \mathbb{RP}^3$. The union of the resulting allowed regions must be contained inside the original allowed region $\mathcal{D}$:

$$\mathcal{D}' \equiv \bigcup_{[\lambda_{\text{ext}}] \in \mathbb{RP}^3} \mathcal{D}_{[\lambda_{\text{ext}}]} \subseteq \mathcal{D}. \tag{3.7}$$

A key observation of [7, 112] is that this inclusion can be strict — i.e. by scanning over different directions $[\lambda_{\text{ext}}]$ in OPE space, and taking the union of the resulting allowed regions, we can obtain a smaller allowed region than if we impose the naïve condition (3.5). Scanning over OPE coefficient directions $[\lambda_{\text{ext}}]$ allows us to use that $\lambda_{\text{ext}}\lambda_{\text{ext}}^T$ is rank-1, and get better results. A disadvantage is that we must solve multiple semidefinite programs to compute the new allowed region $\mathcal{D}'$.

3.3 An algorithm for scanning over OPE coefficients

Suppose we would like to determine whether some putative scaling dimensions $(\Delta_s, \Delta_\phi, \Delta_t)$ are allowed or not. According to the previous section, we should scan over directions in OPE coefficient space $[\lambda_{\text{ext}}] \in \mathbb{RP}^3$. For each direction, we should compute whether a functional α exists satisfying (3.6) and (3.1). If α does not exist for some $[\lambda_{\text{ext}}]$, then the point $(\Delta_s, \Delta_\phi, \Delta_t)$ is allowed. If α exists for all $[\lambda_{\text{ext}}]$, then the point $(\Delta_s, \Delta_\phi, \Delta_t)$ is disallowed. In this section, we describe an algorithm that makes the scan over $[\lambda_{\text{ext}}] \in \mathbb{RP}^3$ very efficient.

Let us choose some initial direction $[\lambda_1] \in \mathbb{RP}^3$. Suppose that a functional α_1 exists obeying the condition[17]

$$\lambda_1^T \alpha_1(\vec{V}_{\text{ext}}) \lambda_1 \geq 0, \tag{3.8}$$

and additionally obeying all other necessary positivity conditions (3.1) for computing feasibility of the given point $(\Delta_s, \Delta_\phi, \Delta_t)$ in dimension space. The key observation is that $Q_1 = \alpha_1(\vec{V}_{\text{ext}})$ defines a bilinear form that is positive not only for λ_1, but also for some neighborhood $U_1 \subset \mathbb{RP}^3$ containing $\lambda_1 \in U_1$. That is, α_1 rules out an entire neighborhood $U_1 \subset \mathbb{RP}^3$. We can now focus on scanning over the complement $\mathbb{RP}^3 \setminus U_1$.

This suggests algorithm 1 for ruling out a point $(\Delta_s, \Delta_\phi, \Delta_t)$ in dimension space. Algorithm 1 is similar to so-called "cutting plane" methods. We have a region $\mathcal{A}_n$ of allowed OPE directions. We choose a point $[\lambda_{n+1}] \in \mathcal{A}_n$ and consult an "oracle" (the semidefinite program solver) to get a quadratic form Q_{n+1} that rules out that point. This quadratic form cuts away a neighborhood U_{n+1} from $\mathcal{A}_n$, giving a smaller allowed region $\mathcal{A}_{n+1} = \mathcal{A}_n \setminus U_{n+1}$.

In traditional cutting plane methods, an oracle provides *linear* forms instead of quadratic forms. If the U_i were half-spaces defined by linear forms, then the above algorithm would exhibit some nice properties. Firstly, the allowed regions $\mathcal{A}_n$ would be convex. Secondly, if we choose $[\lambda_{n+1}] \in \mathcal{A}_n$ to be the center of volume of $\mathcal{A}_n$ (in some

---

[17]If no such functional exists, then we know $(\Delta_s, \Delta_\phi, \Delta_t)$ is an allowed point in dimension space, and we can stop.

begin

    Given a list of functionals $\{\alpha_1, \ldots, \alpha_n\}$, together with quadratic forms $Q_i = \alpha_i(\vec{V}_{\text{ext}})$ and regions ruled out by those quadratic forms

$$U_i \equiv \{[\lambda] \in \mathbb{RP}^3 \text{ such that } \lambda^T Q_i \lambda \geq 0\}. \tag{3.9}$$

    The allowed region of OPE space is

$$\mathcal{A}_n \equiv \mathbb{RP}^3 \setminus (\cup_{i=1}^n U_i). \tag{3.10}$$

    if $\mathcal{A}_n$ *is empty* **then**
        All directions in OPE space are ruled out.
        return *Disallowed*
    else
        Choose some $[\lambda_{n+1}] \in \mathcal{A}_n$.
        Impose the positivity condition $\lambda_{n+1}^T \alpha(\vec{V}_{\text{ext}})\lambda_{n+1} \geq 0$, and solve the resulting semidefinite program to find a functional α_{n+1}.
        if α_{n+1} *exists* **then**
            Append α_{n+1} to the list $\{\alpha_1, \ldots, \alpha_n\}$ and go to **begin**.
        else
            We have failed to rule out all directions in OPE space.
            return *Allowed*
        end
    end
end

Algorithm 1. Cutting surface algorithm for scanning over OPE coefficients.

affine coordinates), then the neighborhood U_{n+1} would be guaranteed to cut away half of $\mathcal{A}_n$. Thus, the volume of $\mathcal{A}_n$ would decrease exponentially in the number of cuts, and the algorithm would take logarithmic time in the volume of $\mathcal{A}_n$.[18]

    Fortunately, in many examples, we have found that once the allowed region $\mathcal{A}_n$ becomes sufficiently small, the sets U_{n+1} become very close to half-spaces near the allowed region, see figure 3. Recall that U_{n+1} is defined by a quadratic inequality (3.9), and thus generically has curved edges. However, as the algorithm proceeds, the radius of curvature of these edges becomes large relative to the size of the region $\mathcal{A}_n$ (in some generic affine coordinates on $\mathbb{RP}^3$). Thus, our algorithm approximately inherits many of the nice properties of traditional cutting plane methods. We call our method a "cutting surface" algorithm.

3.4 Finding a point $[\lambda_{n+1}]$

The most difficult step in the cutting surface algorithm is determining whether $\mathcal{A}_n$ is non-empty and, if it is non-empty, choosing a point $[\lambda_{n+1}] \in \mathcal{A}_n$. For this step, we are given a list of quadratic forms $Q_1, \ldots, Q_n \in \mathbb{R}^{m \times m}$, and we wish to find $x = \lambda_{n+1} \in \mathbb{R}^m$ that is

---

[18]For example, to search a unit cube in D dimensions, it takes time proportional to D. The precise running time depends on how the algorithm terminates. We comment more on this below.
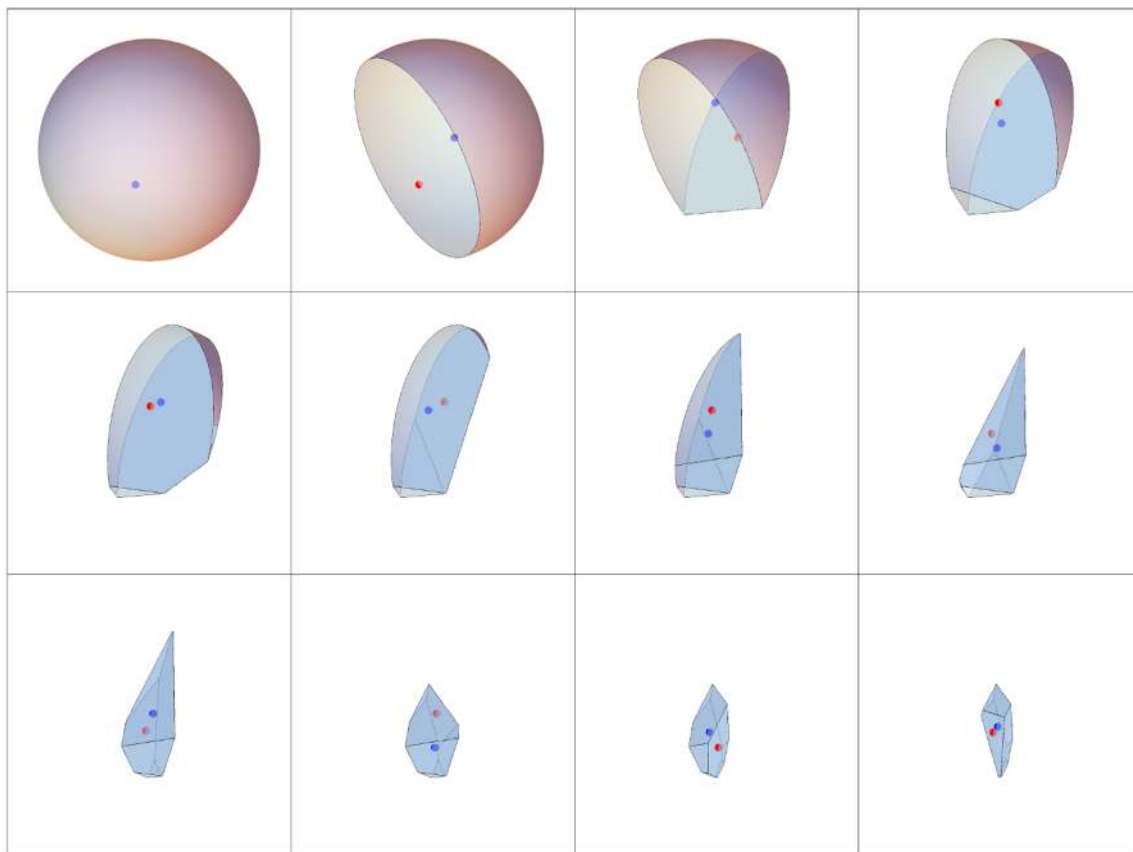
Figure 3. Example allowed regions $\mathcal{A}_1, \ldots, \mathcal{A}_{12}$ of OPE space during the cutting surface algorithm for scanning over OPE coefficients. This example is drawn from our calculation of the O(2) model island with derivative order $\Lambda = 43$. We plot OPE space after applying the affine transformation described in figure 4, which turns the initial bounding ellipsoid into the unit sphere. For each allowed region $\mathcal{A}_n$, we show the point $[\lambda_n]$ most recently ruled out by SDPB in red. This point is typically very close to the boundary of the allowed region. We also show the next point to be tested $[\lambda_{n+1}]$ in blue. We choose the blue point close to the center of $\mathcal{A}_n$. In the final frame, SDPB gives *primal feasible* for the blue point.

negative with respect to those quadratic forms. (For the computations in this work, $m = 4$.) This type of problem is called a *quadratically constrained quadratic program* (QCQP), see e.g. [113].

Unfortunately, QCQPs are NP-hard in general.[19] However, we have found several heuristic approaches that work well for the case at hand. Furthermore, these heuristics can be stacked: if one method fails to find a solution, we can try another method. For this work, we applied multiple heuristics, using one to verify the results of another when possible. In the next few subsections, we describe these heuristics.

---

[19]A notable exception is $m = 2$. In this case, the quadratic forms become quadratic functions of a single variable in an affine patch of $\mathbb{RP}^1$, and the positive and negative regions can be solved for analytically. This case is relevant, for example, in the 3d Ising model problem studied in [7], which involves two OPE coefficients $\lambda_{\sigma\sigma\epsilon}$ and $\lambda_{\epsilon\epsilon\epsilon}$.

Because we solve the QCQP using heuristics, our implementation of the cutting surface algorithm is non-rigorous (except when $m = 2$). It would be interesting to investigate whether there exists a deterministic algorithm for QCQPs in low dimensions that could be useful in bootstrap calculations.

3.4.1 Implementation in Mathematica

For low-dimensional cases, where $Q_i \in \mathbb{R}^{m \times m}$ with $m = 3, 4$, we have implemented the cutting surface algorithm in Mathematica using standard functions. For example, in order to plot the region $\mathcal{A}_n$ we pass the inequalities $\lambda^T Q_1 \lambda < 0, \ldots, \lambda^T Q_n \lambda < 0$ to the functions `RegionPlot` or `RegionPlot3D`.[20] We then use the `DiscretizeGraphics` function to convert the resulting plot into a `MeshRegion` corresponding to the allowed region.

If the resulting `MeshRegion` returns as `EmptyRegion[m-1]` then the algorithm terminates. If it is instead nonempty, then there are various approaches one can use to select a point in its interior. One simple and fast option is to take the `RegionCentroid`. This approach works most of the time, but occasionally fails when the allowed region is nonconvex.

Another simple approach is to select the point which `NMaximize`s the `RegionDistance` to the `RegionBoundary`, subject to the constraint of being inside the allowed region. We found that this approach leads to a working algorithm a majority of the time, but is often slow and sometimes picks suboptimal points. In the next subsection we describe a more robust procedure that we have developed for selecting an optimal point in the interior.

Another important point is that as the allowed region gets smaller, it is helpful to apply an `AffineTransform` at each iteration of the algorithm to make the allowed region roughly spherical. This for example helps to avoid the problem of missing a very small allowed region. We do this by computing a `BoundingRegion` of the allowed `MeshRegion` (we had good success with the form "FastOrientedCuboid"), and then constructing an `AffineTransform` which maps it to the unit cube. This transformation then gets applied to all coordinates before iterating.

3.4.2 Minimizing Q_n

We now describe some heuristics that do not depend on specialized Mathematica features and can in principle be used in general dimensions m. One important heuristic takes advantage of allowed regions $\mathcal{A}_i$ typically becoming close to convex as the cutting surface algorithm proceeds. Recall that $\mathcal{A}_{n-1}$ is the region on which all quadratic forms $Q_1, \ldots, Q_{n-1}$ are negative. Suppose this region is nonempty. Now let us add an additional quadratic form Q_n. We would like to know whether Q_n is positive on $\mathcal{A}_{n-1}$ (in which case $\mathcal{A}_n$ is empty). If it is not positive, we would like to find a point $\lambda_{n+1} \in \mathcal{A}_{n-1}$ such that Q_n is negative on λ_{n+1}.

---

[20]In cases where higher resolution is needed, we could specify a larger set of sample points using the `PlotPoints` option, or we could define a more powerful function `contourRegionPlot3D` which implements an automatic (but sometimes slow) refinement of the boundary, see https://mathematica.stackexchange.com/questions/48486/high-quality-regionplot3d-for-logical-combinations-of-predicates/.

To do so, consider the function $f(x) = x^T Q_n x / x^T x$, where $x \in \mathbb{R}^m$. Because f is homogeneous of degree zero, f defines a function on $\mathbb{RP}^{m-1}$. We would like to minimize f over $\mathcal{A}_{n-1}$. If the minimum is negative, then the solution $[x]$ gives a point in $\mathcal{A}_n$.

One possible minimization procedure is gradient descent starting from a point in $\mathcal{A}_{n-1}$. To ensure that we stay inside $\mathcal{A}_{n-1}$, we introduce a "barrier" function

$$B_{\mathcal{A}_{n-1}}(x) = -\sum_{i=1}^{n-1} \log \frac{x^T Q_i x}{x^T x}, \tag{3.11}$$

and minimize the combination

$$f(x) + \gamma B_{\mathcal{A}_{n-1}}(x), \tag{3.12}$$

where $\gamma > 0$ is a parameter that we choose. The barrier function is defined so that it is finite inside $\mathcal{A}_{n-1}$ and diverges to $+\infty$ as one approaches the boundaries of $\mathcal{A}_{n-1}$ from the interior. In the limit $\gamma \to 0$, the minimum of (3.12) converges to the minimum of $f(x)$ over $\mathcal{A}_{n-1}$.

Following standard practice in interior point optimization, we combine gradient descent with decreasing the parameter γ. In each iteration, we compute a search direction using Newton's method for the combined function (3.12). We then move along this direction and simultaneously decrease γ by a constant factor.

If the region $\mathcal{A}_{n-1}$ were convex and the function f were convex, then the above algorithm would be guaranteed to find the minimum of f. We have found that in practice, convexity holds approximately for both the region $\mathcal{A}_{n-1}$ and the function $f(x)$. Thus, typically this algorithm finds a suitable minimum after a single run. To increase its likelihood of success, we attempt the descent algorithm from many different randomly chosen starting points inside $\mathcal{A}_{n-1}$. We sample random starting points using the hessian line search method detailed in section 3.4.4.

We can make some shortcuts to the standard interior point method. First we observe that Q_n is usually very small for λ_n. This means λ_n is in fact already quite close to the $Q_n = 0$ surface. One shortcut is that we can draw a line starting from λ_n along the gradient of the function defined by Q_n, then test whether there is a feasible point on this line. Another shortcut is that we can simply sample some random points around λ_n. Both shortcuts have a very good chance to succeed, and are very cheap compared to the interior point method described above. Therefore we perform the shortcuts before the standard interior point method.

For the computations in this work, the simple method of minimizing Q_n over $\mathcal{A}_{n-1}$ works most of the time. It will be interesting to explore its applicability to higher-dimensional spaces of OPE coefficients and other bootstrap problems.

3.4.3 Semidefinite relaxation and rank minimization

Another heuristic uses the method of semidefinite relaxation, which is standard in the literature on QCQPs [113]. Recall that we would like to solve the QCQP: find x such that

$x^T Q_i x \leq 0$ for all $i = 1, \ldots, n$ (which is equivalent to $[x] \in \mathcal{A}_n$). This can also be written as:

$$\text{Find } X \succeq 0 \text{ such that } \text{Tr}(XQ_i) \leq 0 \text{ for all } i = 1, \ldots, n, \text{ and } \text{rank}(X) = 1. \qquad (3.13)$$

Here, X is an $m \times m$ matrix and "$\succeq$" means "is positive semidefinite". If such an X exists, then it can be written $X = xx^T$, and x provides the required solution to the QCQP.

Equation (3.13) *almost* defines a semidefinite program. The only difference is the condition $\text{rank}(X) = 1$. Removing the rank-1 condition, we obtain the *semidefinite relaxation* of the original QCQP. Solving the semidefinite relaxation gives two possible outcomes:

- The semidefinite relaxation is infeasible (i.e. X does not exist satisfying the conditions $\text{Tr}(XQ_i) \leq 0$ and $X \succeq 0$). In this case, the original QCQP is necessarily infeasible. Thus, we can rigorously conclude that $\mathcal{A}_n$ is empty.

- The semidefinite relaxation is feasible. Typically, the resulting matrix X is not particularly close to rank 1, so we must perform some additional work to find whether a solution of the QCQP exists.

In the case where the semidefinite relaxation is feasible, we use the method described in [114] for finding low-rank solutions of semidefinite programs. This method involves solving a sequence of semidefinite programs with objective functions designed to successively decrease the $m - 1$ smallest eigenvalues of X. We solve the semidefinite relaxation and the subsequent rank-minimization SDPs using SDPB.

If rank minimization succeeds, we are left with a positive semidefinite matrix X with one large eigenvalue and several small eigenvalues. To find a rank-1 solution xx^T, we apply the random sampling method described in [113]. We take random samples $x \in \mathbb{R}^m$ with covariance matrix $X = \langle xx^T \rangle$. By construction, each inequality in the QCQP is true in expectation:

$$\langle x^T Q_i x \rangle = \text{Tr}(Q_i \langle xx^T \rangle) = \text{Tr}(Q_i X) \leq 0. \qquad (3.14)$$

Thus, there is a reasonable probability of finding a sample x for which all inequalities in the QCQP are true. If such a sample exists, we have solved the QCQP. If we do not find such a sample, then we cannot conclude anything about the QCQP.

An implementation of the algorithm described in this section is available online.[21] In our testing, it worked consistently in cases where OPE space is relatively low-dimensional $m \leq 4$. Indeed, this algorithm is capable of finding solutions to the QCQP in cases where the Q_n-minimization of section 3.4.2 fails (for example because $\mathcal{A}_{n-1}$ has a complicated or elongated shape). Although it takes only a few minutes to run, SDP relaxation methods are more computationally intensive than the Q_n-minimization. Thus, we use them as a final heuristic, which we run only when other heuristics have failed to solve the QCQP.

---

[21]https://gitlab.com/davidsd/quadratic-net/.

3.4.4 Choosing $[\lambda_{n+1}]$

When $\mathcal{A}_n$ is non-empty, the heuristics in sections 3.4.1, 3.4.2, and 3.4.3 will usually find a point $[x] \in \mathcal{A}_n$. However, to make the cutting surface algorithm as efficient as possible, we would like to choose $[\lambda_{n+1}]$ roughly in the "center" of $\mathcal{A}_n$. In the approach using standard Mathematica functions, one possibility is to choose $[\lambda_{n+1}]$ to be the `RegionCentroid` of the allowed OPE region. However, for the other approaches it is important to have methods that don't require detailed knowledge of the shape of $\mathcal{A}_n$ (which can be expensive to compute).

One simple approach is to minimize the barrier function $B_{\mathcal{A}_n}(x)$ over $\mathcal{A}_n$ (using $[x]$ as an initial point). However, for very elongated regions $\mathcal{A}_n$, the minimum of the barrier function is sometimes not particularly close to the center of volume.

Note that in the case $m = 2$, where OPE space $\mathbb{RP}^{m-1}$ is 1-dimensional, it is trivial to find a suitable $[\lambda_{n+1}]$. The allowed region is a union of line segments that we can solve for analytically. We can then choose the midpoint of the longest line segment (in some affine coordinates).

We can use this observation in higher dimensions. Let us start with a point $[x_0] \in \mathbb{RP}^{m-1}$ and choose a random line $\ell_0 \subset \mathbb{RP}^{m-1}$ containing $[x_0]$. The intersection of the line ℓ_0 with the region $\mathcal{A}_n$ is a union of line segments (typically a single segment), and we can choose $[x_1]$ to be the midpoint of one of these segments. Repeating in this way, we obtain a sequence of points $[x_k]$ that are at the midpoints of random lines intersecting $\mathcal{A}_n$. This sequence does not typically converge to a single point. However, later points in the sequence are good candidates for $[\lambda_{n+1}]$.[22] To randomly sample the line ℓ_i, we choose coordinates around $[x_i]$ in which the Hessian of the barrier function $B_{\mathcal{A}_n}(x)$ at $[x_i]$ becomes a diagonal matrix with entries ± 1. In these coordinates, the region $\mathcal{A}_n$ typically looks roughly spherical around $[x_i]$. We then use a uniform distribution on an infinitesimal sphere around x_i in these coordinates. We call this method a "hessian line search".

The hessian line search can be modified to randomly sample points inside $\mathcal{A}_n$, with applications to the Q_n-minimization method of section 3.4.2. Instead of choosing x_{i+1} to be the midpoint of a line segment in $\ell_i \cap \mathcal{A}_n$, we can choose it randomly along a segment.

3.4.5 Bounding ellipsoids

The cutting surface method becomes most efficient when the radius of curvature of the surface defined by the quadratic form Q_n is small compared to the size of the region $\mathcal{A}_{n-1}$. If we start with the allowed region $\mathcal{A}_0 = \mathbb{RP}^{m-1}$, then it might take several iterations of the algorithm before this happens. Indeed, in our testing, the cutting surface algorithm often spent significant time cutting away parts of $\mathbb{RP}^{m-1}$ that are known to be far from the correct values of OPE coefficients. To avoid this problem, it is useful to impose a "bounding box" in OPE space. An efficient way to do this is to pick a bounding ellipsoid, and choose Q_1 to be the quadratic form that rules out the exterior of the ellipsoid.

Imposing a bounding ellipsoid is a non-rigorous optimization and should be done with care. As we worked our way up in the number of derivatives of the crossing equations, we

---

[22]In practice, we take the last 10 points in a long sequence and average them in some affine coordinates.

used the following strategy. At an initial derivative order Λ, we keep track of all values of OPE coefficients of allowed points. We choose an ellipsoid $\mathcal{E}$ that contains these values and is also enlarged by an O(1) factor. We then increase $\Lambda \to \Lambda'$ and use $\mathcal{E}$ as a bounding ellipsoid for the cutting surface algorithm. As a check on this method, we can inspect the set of allowed OPE coefficients found at derivative order Λ' and see if any of them are close to the boundary of $\mathcal{E}$. In practice, they never are, see figure 4. (In fact, they are almost never outside the cloud of points computed at derivative order Λ, so the enlargement by an O(1) factor is unnecessary.) We can now find a new ellipsoid $\mathcal{E}'$ and continue.

3.5 Hot-starting

The cutting surface algorithm requires solving multiple SDPs to rule out a single point $(\Delta_\phi, \Delta_s, \Delta_t)$ in dimension space. For example, for the computation described in section 4.2, each point in dimension space required solving an average of ~ 35 SDPs (not including the tiny SDPs encountered in the semidefinite relaxation method of section 3.4.3). Fortunately, many of these SDPs can be solved extremely quickly using *hot-starting* [84]: we reuse the final state of the semidefinite program solver from a previous calculation as the initial state in a new calculation. In practice, hot-starting means passing an old checkpoint file as an argument to SDPB.

Hot-starting is particularly advantageous in the cutting surface algorithm because SDPs only change by a small amount with each new run. Specifically, the only difference between subsequent SDPs is the replacement of the positivity condition $\lambda_n^T \alpha(V_{\text{ext}})\lambda_n \geq 0$ by the new condition $\lambda_{n+1}^T \alpha(V_{\text{ext}})\lambda_{n+1} \geq 0$. Thus, the previous checkpoint contains a functional that already satisfies all other positivity conditions in the semidefinite program. In practice, the new condition $\lambda_{n+1}^T \alpha(V_{\text{ext}})\lambda_{n+1} \geq 0$ is satisfied after a small number of iterations of SDPB. Furthermore, the number of iterations typically decreases over the course of the cutting surface algorithm, see figure 5.

Hot-starting is useful also for different points in dimension space. In practice, we keep a list of checkpoint files from all runs of SDPB over the course of a computation. For each new point in dimension space, we find the newest checkpoint file corresponding to the closest point in dimension space, and use it to initiate the cutting surface algorithm.

To demonstrate the effectiveness of hot-starting in dimension space, we study the 3d Ising model σ, ϵ mixed correlator bootstrap described in [5]. We choose a fixed point P_0 in dimension space and hot-start P_0 with several checkpoints from nearby points P_i, see figure 6(a). We observed that in general when P_i is close to P_0, the number of iterations is smaller. In figure 6(b), we show the effectiveness of hot-starting in a transformed space, where the Ising island is roughly a spherical shape. We see that the concept of "nearest" is better behaved in this transformed space.

Let us mention one additional practical optimization. In each step of the cutting surface algorithm for scanning OPE coefficients, we must solve semidefinite programs that are nearly identical: they differ only in the positivity conditions associated to the external scalars ϕ, s, t. Consequently, we can avoid re-generating the entire SDP and only re-generate the conditions for the external scalars.
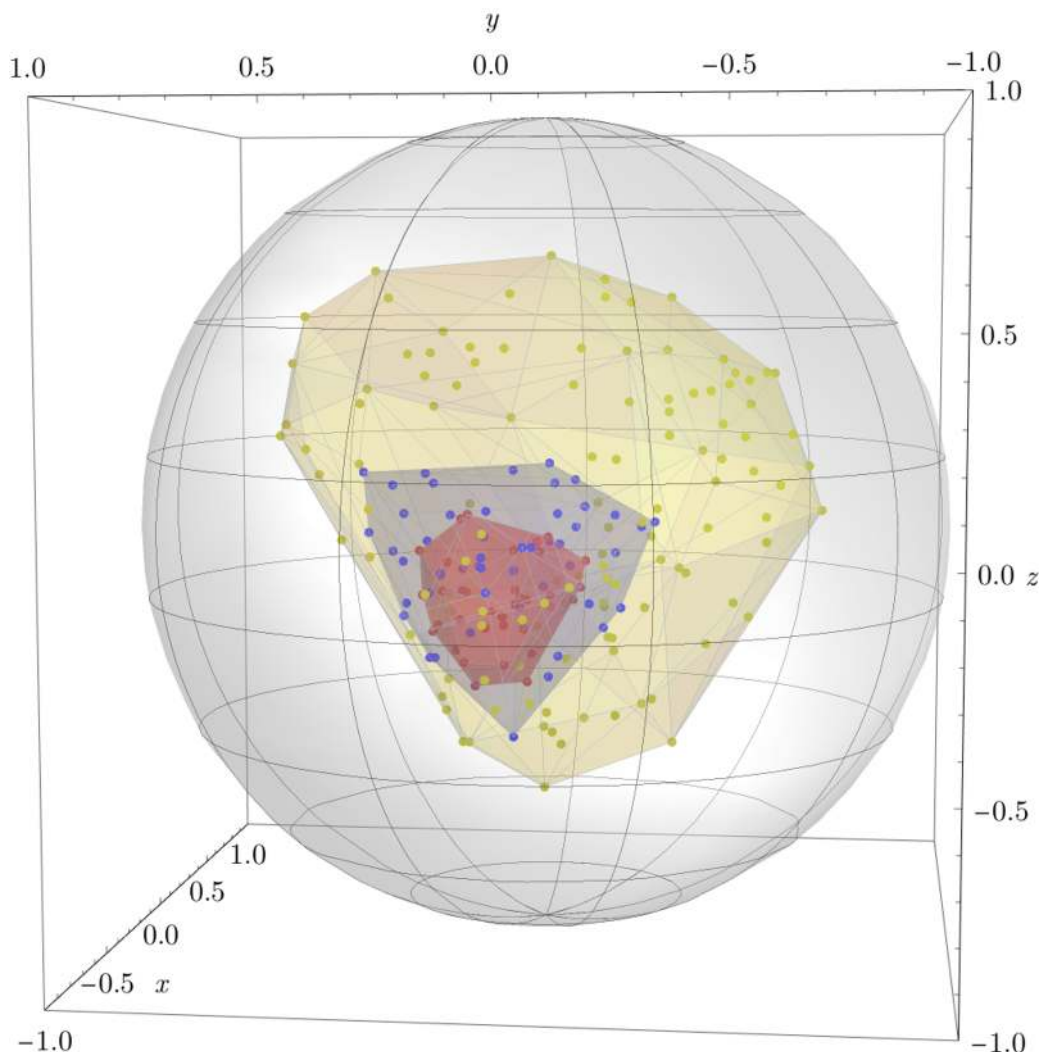
Figure 4. Allowed points in external scalar OPE coefficient space, found while computing the allowed island in dimension space, for $\Lambda = 27$ (yellow), $\Lambda = 35$ (blue), and $\Lambda = 43$ (red), together with a choice of bounding ellipsoid (gray). For each set of points, we also show their convex hull in the same color. To plot the points, we applied an affine transformation to make the $\Lambda = 27$ region roughly spherical. The relationship between the displayed coordinates x, y, z and the OPE coefficients is $\lambda_{\text{ext}} = (751.0591846177696 - 362.65959721052656x - 131.334377405401y - 41.46952958591952z, 1, 3383.753238900843 + 695.8131625006117x - 1729.4094085965235y - 607.9744222068027z, -12562.290081255807 + 123.88628689820867x - 3799.4579787849975y + 10949.506824631871z)$. After finding the $\Lambda = 27$ points, we chose the gray sphere as a bounding ellipsoid for the computation with $\Lambda = 35$. No $\Lambda = 35$ (blue) points are near the edge of the bounding ellipsoid, which justifies this choice. We used the same bounding ellipsoid for the computation with $\Lambda = 43$. Again, no $\Lambda = 43$ (red) points are near the edge of the bounding ellipsoid.

3.6 Primal/dual jumps

When testing feasibility of an SDP (as opposed to optimizing an objective function), `SDPB` includes some features that allow the solver to terminate more quickly. Internally, `SDPB`
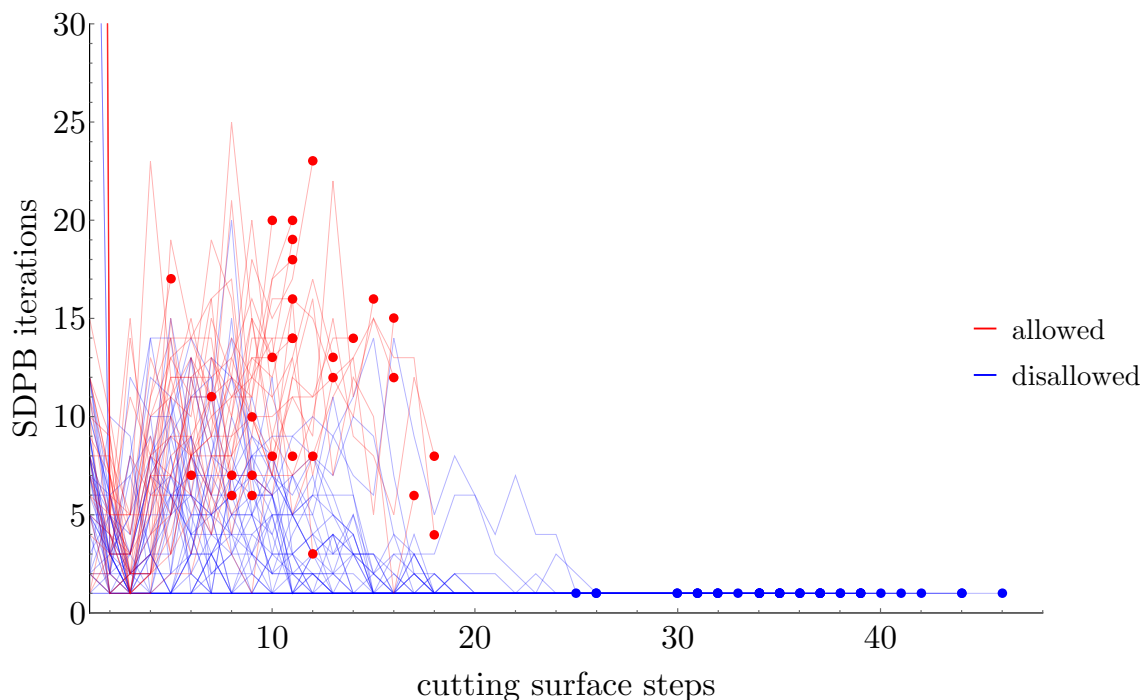
Figure 5. Number of iterations of SDPB in each step of the cutting surface algorithm, for our computation of the O(2) model island with $\Lambda = 43$. Hot-starting drastically reduces the number of iterations throughout the computation. The blue paths represent OPE scans that eventually terminate by ruling out a point in dimension space. The red paths represent scans that eventually terminate by finding an allowed (primal) point. We mark the end of each path with a dot. At the beginning of the computation, a small number of points require ~ 200 SDPB iterations during the first step of the cutting surface algorithm. Once the checkpoints from those SDPB runs have been generated, hot-starting ensures that most subsequent runs take $\lesssim 20$ SDPB iterations. The first 10–20 steps of the cutting surface algorithm typically require 1–15 SDPB iterations each. If the point is allowed, the algorithm typically finds it within 20 steps. If the point is disallowed, subsequent steps of the cutting surface algorithm take fewer iterations, with the last several steps requiring 1 iteration each.

uses a modified Newton's method to simultaneously solve three types of equations: primal feasibility equations, dual feasibility equations, and an equation relating the two. For our purposes, the dual feasibility equations are the most important. A functional α exists if and only if the dual feasibility equations are satisfied. If SDPB detects that it is possible to solve either the primal or dual feasibility equations during an iteration, then it does so immediately. We call such events primal/dual jumps.

When testing feasibility, a dual jump means that a functional α has been found (and the solver can terminate). In practice, a primal jump means a functional will not be found (so the solver can terminate in this case as well). The observation that we can stop after a primal jump was made in [82]. As far as we are aware, it has not been rigorously established. However, this does not affect the validity of the resulting bootstrap bounds, which depends only on the existence of functionals.
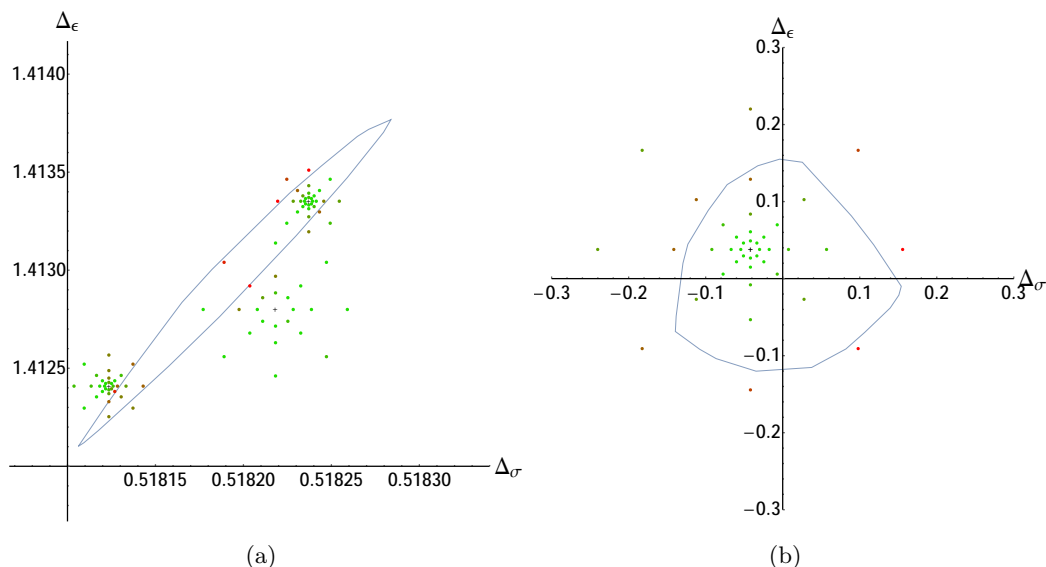
Figure 6. Hot-starting effectiveness for different nearby checkpoints. The blue line is the boundary of the 3d Ising from σ, ϵ mixed correlator bootstrap. The setup is the same as that of the dark blue region of figure 3 in [5] except $n_{\max} = 10$ (i.e. $\Lambda = 19$). We fix P_0 (indicated by a cross) to be various points and hot-start the P_0 computation with checkpoints taken from nearby points P_i around P_0. The color of P_i indicates how many SDPB iterations is needed for the hot-started computation. Red corresponds to 8 iterations, while green corresponds to 1 iteration. Without hot-starting, the typical number of iterations is about 80. On the left: we fix P_0 to be $(0.518123, 1.412409)$, $(0.518237, 1.413352)$, $(0.518218, 1.412800)$. On the right: the $(\Delta_\sigma, \Delta_\epsilon)$ space is transformed such that the island is roughly spherical. We fix P_0 to be $(0.518217, 1.413221)$.

To make SDPB terminate in the event of primal/dual jumps, we supply the options `--detectPrimalFeasibleJump` and `--detectDualFeasibleJump`. We have found that it is important to disallow SDPB from terminating for other reasons. For example, over the course of the cutting surface algorithm, the primal error can get quite small, and often goes below reasonable values of `primalErrorThreshold`. However, in practice only primal/dual jumps are good reasons to terminate. Thus, we recommend turning off the options `--findPrimalFeasible` and `--findDualFeasible`, and setting `primalErrorThreshold` and `dualErrorThreshold` extremely small (e.g. 10^{-200}). Our precise parameters are listed in appendix B.

The existence of dual feasible jumps is sensitive to the precise bootstrap problem being solved. In our initial bootstrap implementation for correlators of ϕ, s, t, we did not observe any dual feasible jumps. In these cases, SDPB would run for many iterations, with the `dualError` (which indicates failure of the dual feasibility equations to be satisfied) steadily decreasing but never jumping to zero. We observed that during these iterations, SDPB was working hard to find functionals that were positive when acting on operators close to the unitarity bound. We alleviate this problem by imposing a small gap in twist $\tau = \Delta - J$. Specifically, we impose

$$\tau \geq \tau_{\text{unitarity}} + \delta\tau, \tag{3.15}$$

(where $\tau_\text{unitarity}$ is the unitarity bound) in all spin/symmetry sectors not containing conserved currents. (This condition is in addition to other gaps.) The extremely conservative choice $\delta\tau = 10^{-6}$ is sufficient to restore dual feasible jumps. Imposing this small twist gap dramatically increases the efficiency of our methods.

3.7 Delaunay triangulation in dimension space

Given the above methods for determining whether a point $(\Delta_\phi, \Delta_s, \Delta_t)$ in dimension space is allowed, we would like to search for the full allowed region. For simplicity, first consider the one-dimensional case, where we have a single parameter Δ. We can map Δ-space efficiently using binary search between known points. Suppose we have a list of values $\Delta_1 < \Delta_2 < \cdots < \Delta_n$, that are known to be either allowed or disallowed. We define

$$p_i = \begin{cases} 0 & \text{if } \Delta_i \text{ is disallowed,} \\ 1 & \text{if } \Delta_i \text{ is allowed.} \end{cases} \tag{3.16}$$

For each case where $p_i \neq p_{i+1}$, we perform a binary search between Δ_i and Δ_{i+1} to find the precise threshold between allowed and disallowed.

We can reinterpret this method as follows. We can define a "probability" $p(\Delta)$ that a given point is allowed. Our eventual goal will be to make $p(\Delta)$ as close to 0 or 1 as possible for all Δ. A reasonable approximation for $p(\Delta)$ is via linear interpolation between the values $p(\Delta_i) = p_i$. To improve our knowledge of the allowed region as quickly as possible, the next test point Δ_test should have probability $p(\Delta_\text{test}) = 1/2$. If there are multiple such points, we should choose the one with the smallest slope $|p'(\Delta_\text{test})|$.[23] We then test whether Δ_test is allowed, add it to the list of known values, and repeat the algorithm.

The above method generalizes to higher dimensions. Consider a vector of dimensions $\vec{\Delta} \in \mathbb{R}^k$. Suppose that we have a list of points $\vec{\Delta}_1, \vec{\Delta}_2, \ldots, \vec{\Delta}_n \in \mathbb{R}^k$ and values p_i defined as in (3.16). To define a probability function $p(\vec{\Delta})$, we perform a Delaunay triangulation of the set of known points.[24] Within each simplex of the triangulation, we define $p(\vec{\Delta})$ via linear interpolation between its values p_i at the vertices. Within each simplex, the points satisfying $p(\vec{\Delta}) = 1/2$ are either empty or form a codimension-1 polyhedron. For every nonempty polyhedron, we define a candidate point as the mean of the vertices of the polyhedron. We choose $\vec{\Delta}_\text{test}$ as the candidate point inside the simplex with the largest "crossing distance", which is defined as the minimum distance between two vertices of the simplex with different values of p_i. After testing $\vec{\Delta}_\text{test}$, we add it to the list of known points and repeat the algorithm.

We illustrate this algorithm in 2 dimensions in figure 7.

To work properly, Delaunay triangulation search requires sufficiently good initial conditions. For example, in the 1-dimensional case (binary search), we only obtain a correct picture of the allowed region if each connected allowed component and each connected

---

[23]If we are testing points in parallel, then we can order the points in order of increasing slope and test the first few.

[24]Delaunay triangulations in 2 or 3 dimensions can be computed in Mathematica. In general, they can be computed efficiently using the software package `qhull` [115].
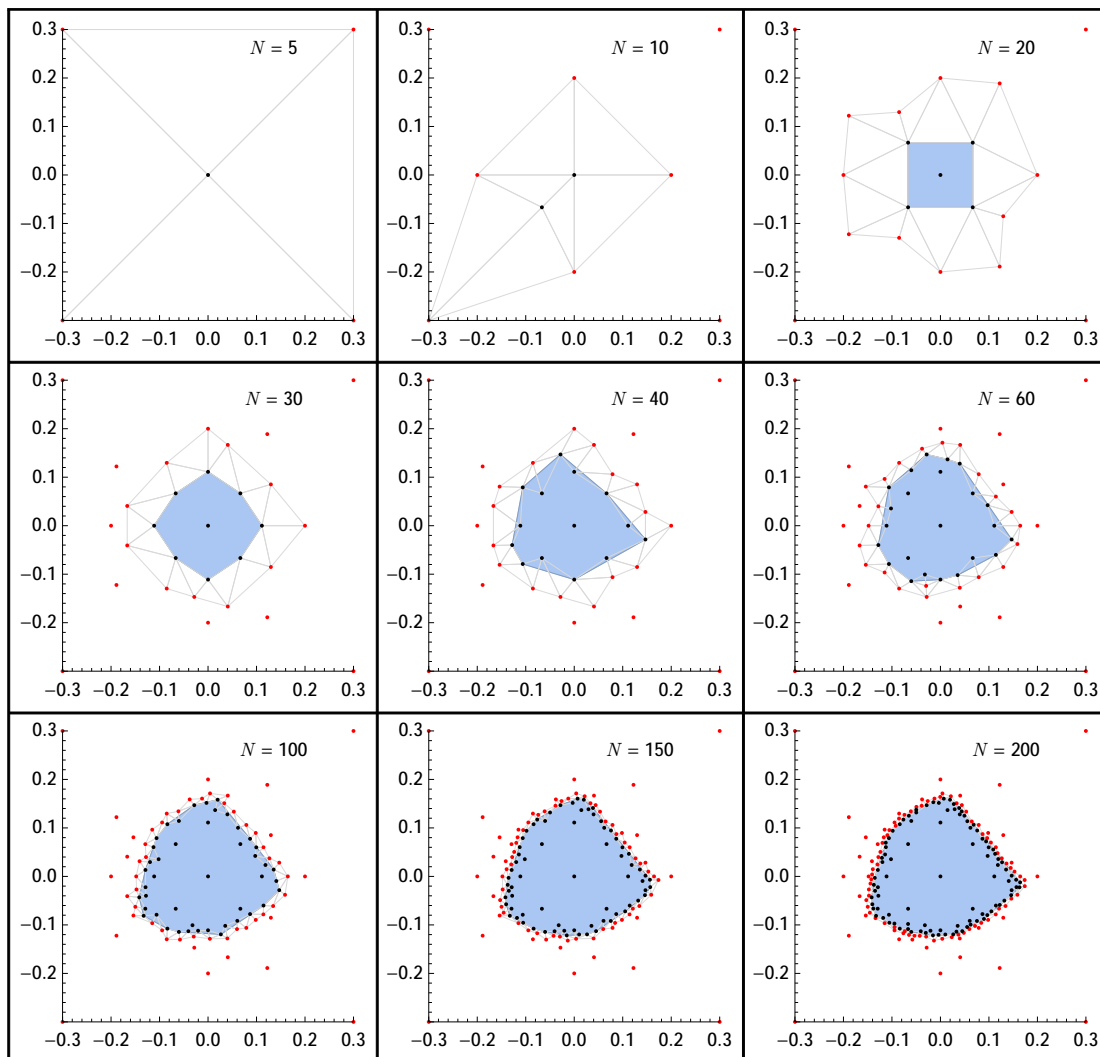
Figure 7. A series of images show intermediate states of the Delaunay triangulation algorithm for 3d Ising σ,ϵ mixed correlator bootstrap. The setup is same as the dark blue region of figure 3 in [5] except $n_{\max} = 10$ ($\Lambda = 19$). We transformed the $(\Delta_\sigma, \Delta_\epsilon)$ space such that the 3d Ising island is roughly spherical. The red points are disallowed, while black points are allowed. The blue region is the convex hull of the black points. The N in each plot is the total number of sampled points.

disallowed component contains at least one initial point. Similarly, in higher dimensions, we only find an allowed region if we start with at least one point inside that region.

For this work, we found suitable initial conditions by first studying low derivative order Λ, and then working our way up in Λ. Our typical workflow is as follows: based on computations at $\Lambda = 15, 19, 23$, we found that the allowed region is a nearly convex island, and it can be made approximately spherical by a particular affine transformation. For each subsequent computation, we applied an affine transformation determined by the previous computation before performing the Delaunay search. This increases the efficiency of the search and makes it easier to correctly resolve corners sharp corners and other features in the boundary of the island.

Because the shape of the island is so simple, Delaunay triangulation works properly given a single allowed point, together with enough disallowed points that the island does not extend outside the convex hull of the disallowed points. When increasing Λ, we can reuse all disallowed points from lower values of Λ. What remains is to determine an allowed point at the new value of Λ. We guess the allowed point in dimension space by extrapolating the way that the island shrinks with Λ, and choosing a point in the center of the extrapolated island. We test this point, and if the result is *primal feasible*, we can initiate a Delaunay search for the island. If the point is ruled out, we must make a different guess.

4 Results

4.1 Dimension bounds without OPE scans

In this section, we show bounds on the dimensions Δ_ϕ, Δ_s computed *without* the algorithm described in section 3.3 for scanning over OPE coefficients. We also explore effects of imposing a more or less conservative gap in the charge-4 scalar sector.

Figure 8 shows bounds with different gap assumptions and different values of Λ, all computed without scanning over OPE coefficients. The light orange region shows a bound with $\Lambda = 19$ and the conservative assumption that the lowest dimension charge-4 scalar operator has dimension $\Delta_4 \geq 1$. Evidence from other techniques supports the hypothesis that in fact $\Delta_4 \geq 3$. The light blue region shows the resulting bound after imposing this stronger gap assumption. Finally, the dark blue region shows the result of imposing the stronger gap assumption and increasing the derivative order to $\Lambda = 27$.

We see that the stronger gap assumption reduces the size of the island by approximately 30% in both dimensions. Furthermore, imposing the gap assumption causes the island to shrink relatively quickly with Λ. Here, we see that increasing Λ from 19 to 27 causes the island to shrink by an additional factor of 2. Because the stronger gap is well-motivated and significantly improves the results, we include it in our computations. For comparison in figure 8, we show the Monte Carlo and high temperature expansion result from [96] and more recent Monte Carlo result from [95]. Without scanning over OPE coefficients, the bootstrap results are less precise.

4.2 Dimension bounds with OPE scans

Now we show our results obtained from scanning over OPE coefficients using the cutting surface algorithm described above. The plots in this section compute the allowed values of $\{\Delta_\phi, \Delta_s, \Delta_t\}$ assuming irrelevance of the second charge 0,1,2 operators and first charge 4 operator. The stress tensor and conserved current are assumed in the spectrum with coefficients constrained by Ward identities. All other operators are allowed to exist at any scaling dimension above $\ell + 1 + \delta\tau$ with $\delta\tau = 10^{-6}$.

Figures 9 and 10 shows our determinations of the allowed regions at derivative order $\Lambda = 19, 27, 35, 43$, projected to the $\{\Delta_\phi, \Delta_s\}$ plane. Figure 11 also shows the projection to the $\{\Delta_\phi, \Delta_t\}$ plane and figure 2 in the introduction shows a view of the 3d region at $\Lambda = 43$. The improvement relative to figure 8 is readily apparent. In particular the
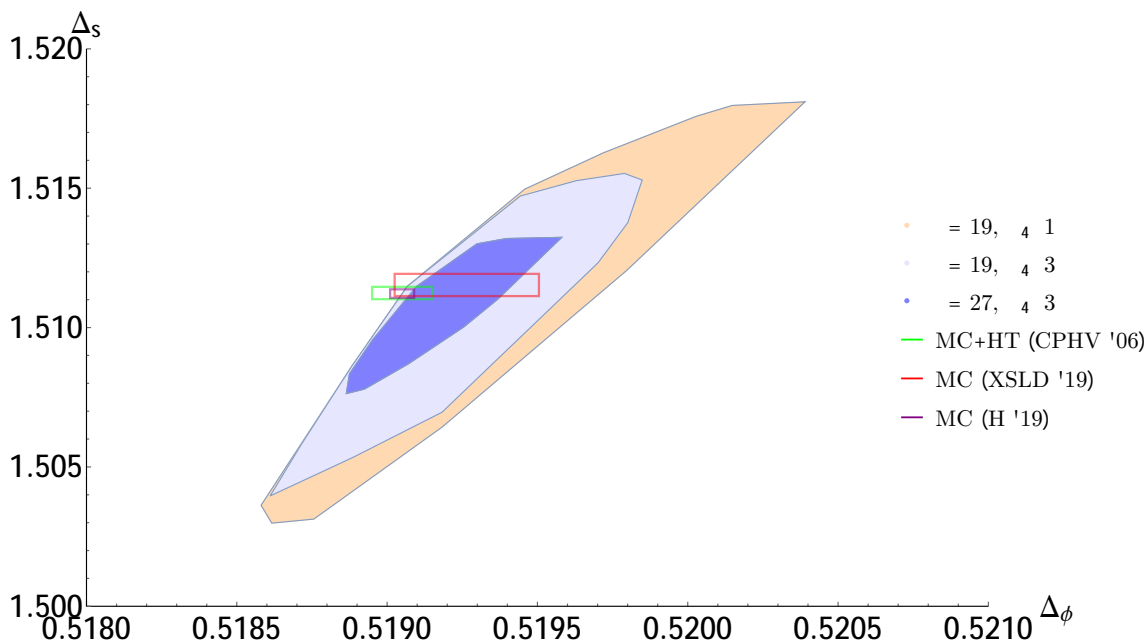
Figure 8. Bounds on the scaling dimensions Δ_ϕ, Δ_s computed *without* the cutting surface algorithm described in section 3.3. The light orange region shows the bound computed with $\Lambda = 19$ and a conservative gap assumption in the charge-4 scalar sector $\Delta_4 \geq 1$. The light blue region shows the bound at $\Lambda = 19$ with a stronger gap assumption $\Delta_4 \geq 3$. The dark blue region shows the bound at $\Lambda = 27$ with the stronger gap assumption. The results are compared with the recent Monte Carlo studies [86, 95] and an earlier study combining Monte Carlo simulations with high temperature expansion calculations in [96]. These bounds were computed at relatively low resolution, so the edges of the island show some artifacts.

conformal bootstrap results exclude the values of Δ_s extracted from ^{4}He measurements [85] and improve upon but appear compatible with both earlier [96] and recent results from Monte Carlo simulations [86, 95].

The plotted regions are obtained by constructing the Delaunay triangulation of our tested points, selecting the triangles that contain both allowed and disallowed points, and plotting the convex hull of the points in the interior of these triangles that are midway between the allowed and disallowed vertices. This represents our best determination of the allowed region at a given Λ, but has a small error associated with the distance between the boundary and the nearest disallowed point. This "best-fit" region gives the determinations

$$\Delta_\phi = 0.519088(17^*), \tag{4.1}$$

$$\Delta_s = 1.51136(18^*), \tag{4.2}$$

$$\Delta_t = 1.23629(9^*). \tag{4.3}$$

More conservatively we can consider the convex hull of the disallowed points in the Delaunay triangles straddling the boundary of the allowed region. We believe that every point outside of this more conservative region is excluded by the conformal bootstrap, giving the
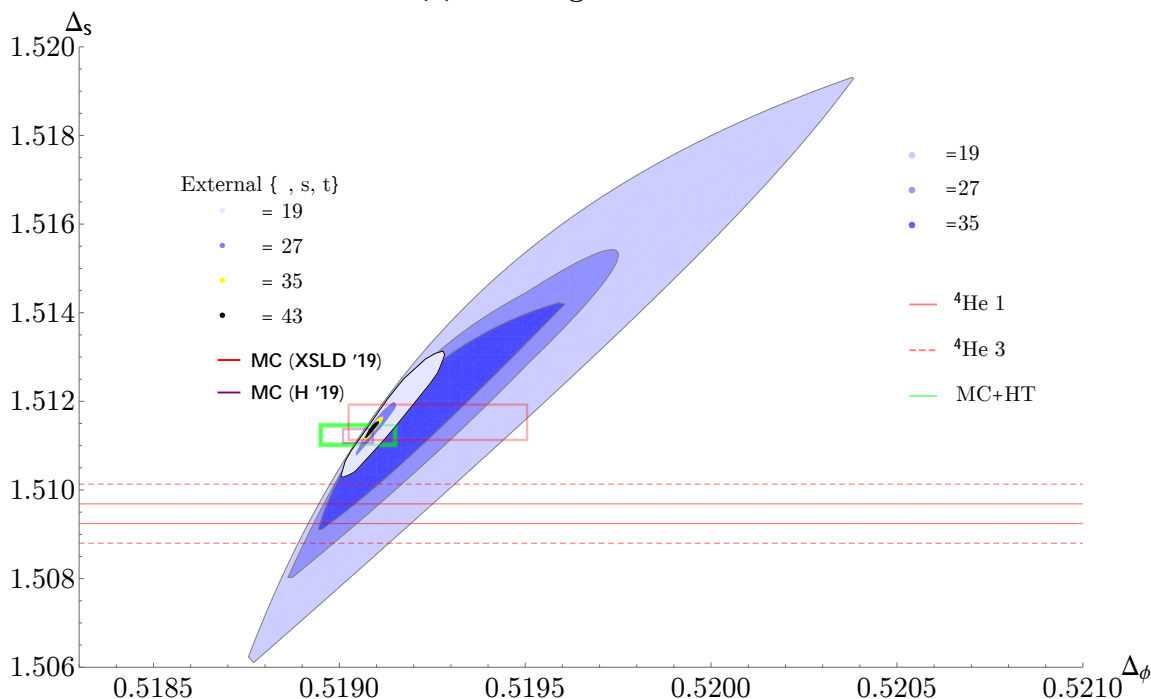
O(2): Scaling Dimensions

Figure 9. Superposition of the new O(2) islands using the $\{\phi_i, s, t_{ij}\}$ system and OPE scans over the earlier bootstrap results from [7] which used the $\{\phi_i, s\}$ system.

rigorous error bars

$$\Delta_\phi = 0.519088(\mathbf{22}), \tag{4.4}$$

$$\Delta_s = 1.51136(\mathbf{22}), \tag{4.5}$$

$$\Delta_t = 1.23629(\mathbf{11}). \tag{4.6}$$

Each allowed point in dimension space comes paired with an allowed point in the space of OPE coefficient ratios. At $\Lambda = 43$ these allowed OPE coefficient ratios live in the ranges

$$\frac{\lambda_{sss}}{\lambda_{\phi\phi s}} = 1.20926(46^*), \tag{4.7}$$

$$\frac{\lambda_{tts}}{\lambda_{\phi\phi s}} = 1.82227(19^*), \tag{4.8}$$

$$\frac{\lambda_{\phi\phi t}}{\lambda_{\phi\phi s}} = 1.765918(64^*). \tag{4.9}$$

The full allowed region in OPE coefficient space may be slightly larger.[25] The full set of computed points at $\Lambda = 43$ are shown in figure 12 and listed in appendix E.

---

[25]Using the scaling dimension region as a guide we would estimate that the range of allowed values may increase in size by $\sim 20\%$ when going from the computed allowed points at $\Lambda = 43$ to the "best-fit" allowed region.
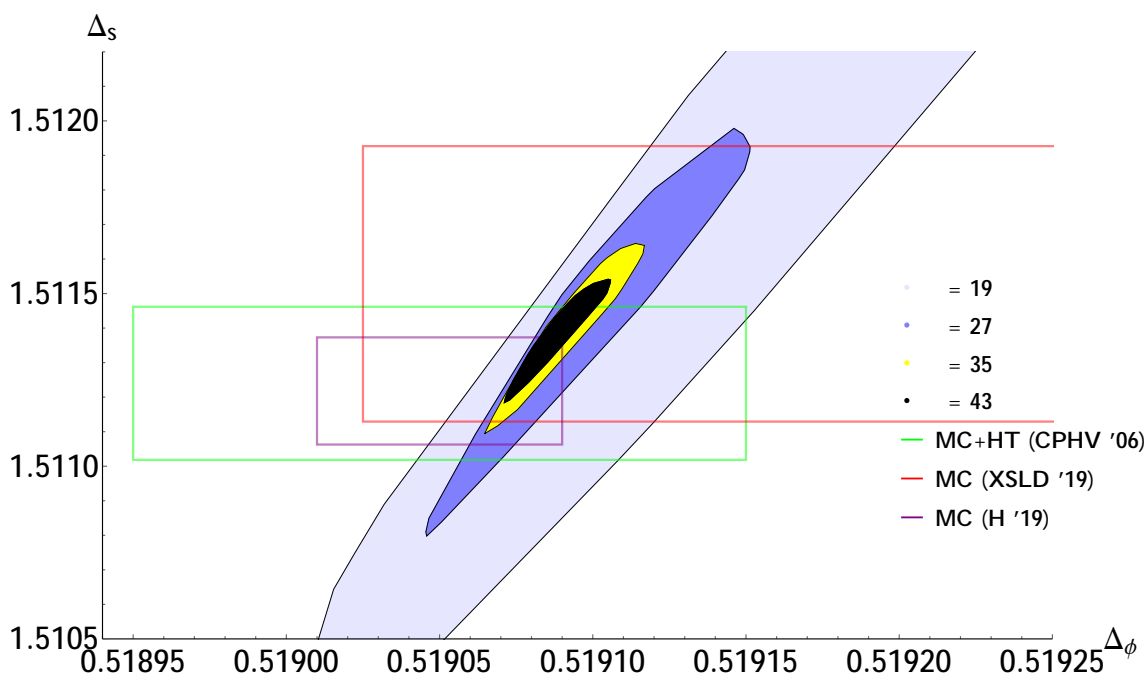
Figure 10. New O(2) islands using the $\{\phi_i, s, t_{ij}\}$ system and OPE scans at $\Lambda = 19, 27, 35, 43$. This plot shows the projection to the $\{\Delta_\phi, \Delta_s\}$ plane. The results are compared with the recent Monte Carlo studies [86, 95] and an earlier study combining Monte Carlo simulations with high temperature expansion calculations [96].

4.3 Central charges and $\lambda_{\phi\phi s}$

As stated in section 2.2, the two-point coefficient C_T for stress tensors and the two-point coefficient C_J for the O(2) current appear in the crossing equations. These coefficients are interesting for several reasons. For example they are related to transport in quantum critical systems, giving the leading term in the high frequency expansion at finite temperature [117, 118]. In particular, the zero temperature conductivity of the O(2) model is given by [117]

$$2\pi\sigma_\infty = \frac{2\pi C_J}{16 C_J^{\text{free}}}. \tag{4.10}$$

It should be possible to produce an island in the combined space of scaling dimensions $\Delta_\phi, \Delta_s, \Delta_t$, OPE coefficient $\lambda_{\phi\phi s}$, and coefficients C_T, C_J. In particular, this would give a determination of C_T and C_J with rigorous error bars. Due to limits on computational resources, we have not yet attempted this computation. Instead, we will content ourselves with non-rigorous estimates of C_T, C_J and $\lambda_{\phi\phi s}$. We chose 7 allowed points (shown in table 9 of appendix E) in our island computed with $\Lambda = 43$ derivatives. For each point, we computed upper and lower bounds on C_J, C_T, and the OPE coefficient $\lambda_{\phi\phi s}$ with $\Lambda = 35$ derivatives. The largest upper bound and smallest lower bound give an estimate for these quantities.

In order to compute upper bounds on C_J (C_T), we must assume a gap between the conserved current (stress tensor) and other operators in the same spin and global symmetry
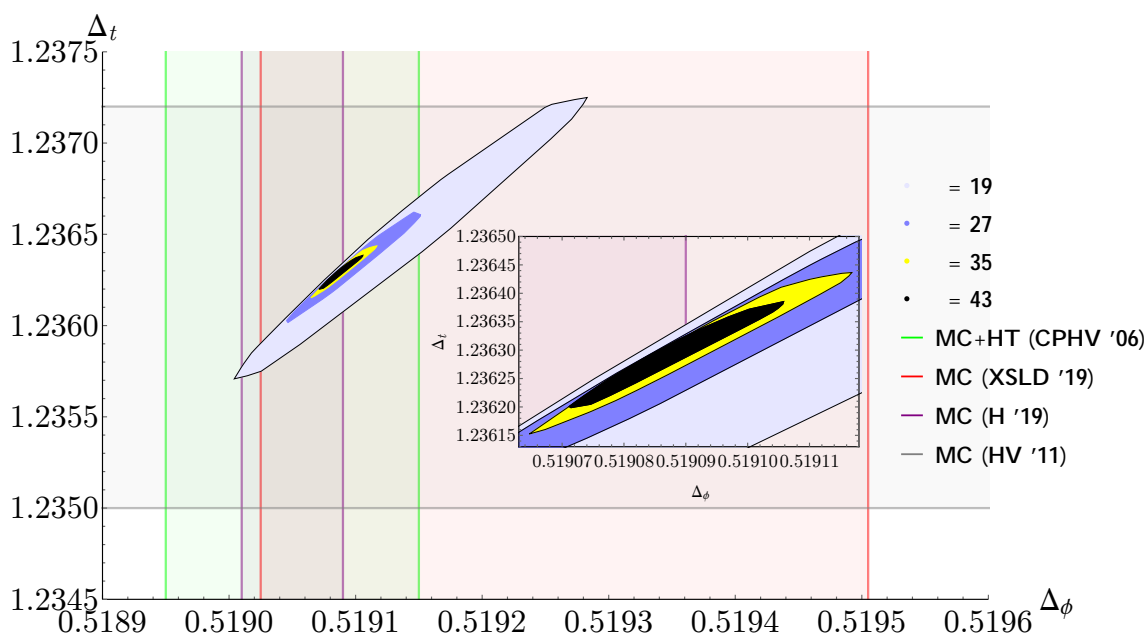
Figure 11. New O(2) islands using the $\{\phi_i, s, t_{ij}\}$ system and OPE scans at $\Lambda = 19, 27, 35, 43$. This plot shows the projection to the $\{\Delta_\phi, \Delta_t\}$ plane. The results for Δ_ϕ are compared with the recent Monte Carlo studies [86, 95] and an earlier study combining Monte Carlo simulations with high temperature expansion calculations [96], while the results for Δ_t are compared with the Monte Carlo study [87]. The latter is also compatible with the earlier pseudo-ϵ expansion estimate $\Delta_t = 1.237(8)$ [116].

sector. When computing upper bounds on C_J, we assume all other spin-1 charge-0 operators have dimension $\Delta \geq 3$. When computing upper bounds on C_T, we assume all other spin-2 charge-0 operators have dimension $\Delta \geq 4$. These assumptions are well-supported by estimates from the ϵ-expansion and from the extremal functional method.

We find

$$C_J/C_J^{\text{free}} = 0.904395(28^*), \tag{4.11}$$

$$C_T/C_T^{\text{free}} = 0.944056(15^*), \tag{4.12}$$

where in both cases the error bars are non-rigorous. Our result for C_J gives a new determination of the zero-temperature conductivity

$$2\pi\sigma_\infty = 0.355155(11^*). \tag{4.13}$$

We also find

$$\lambda_{\phi\phi s} = 0.687126(27^*). \tag{4.14}$$

Combining this result with the OPE ratios (4.7) and adding errors in quadrature leads to the values quoted in table 1.
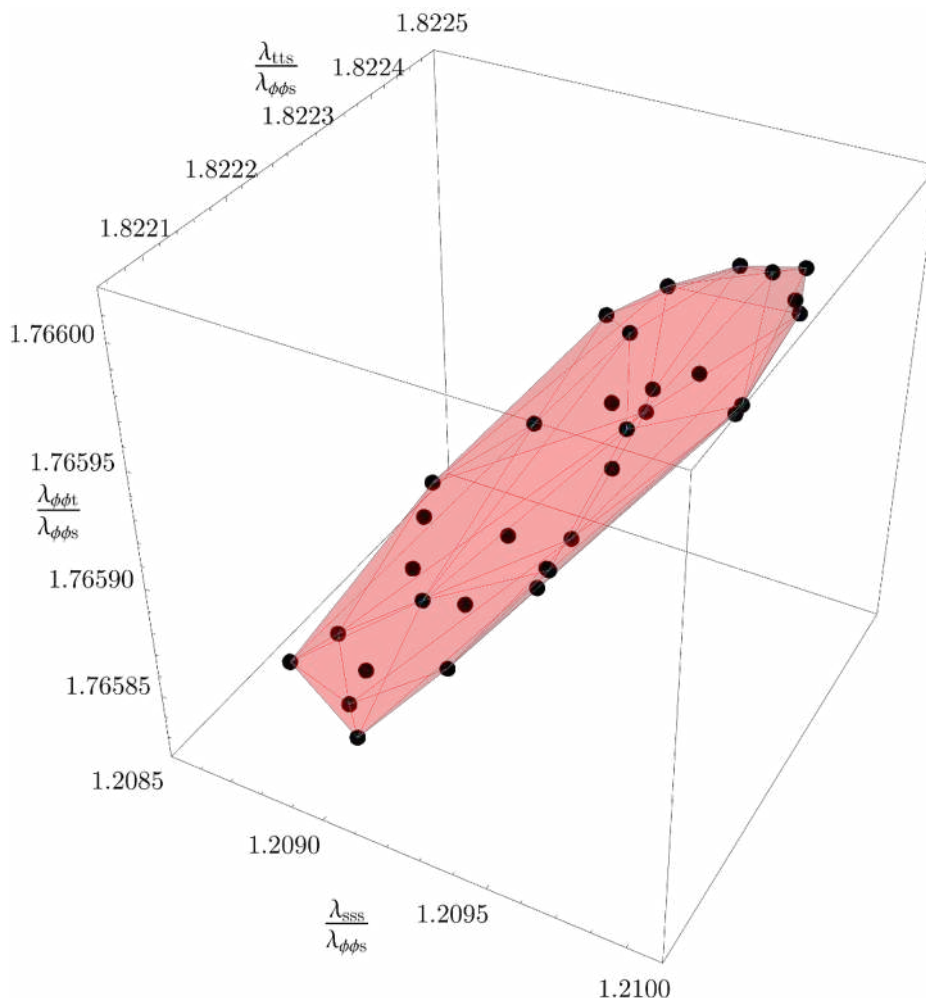
Figure 12. Allowed points in the space of OPE coefficient ratios computed using the $\{\phi_i, s, t_{ij}\}$ system at $\Lambda = 43$. The convex hull of these points (red) gives an estimate for the allowed values of these coefficients. The projection of the full 6d allowed region will be slightly larger so the shown region is non-rigorous.

4.4 Estimates from the extremal functional method

The extremal functional method [52, 78] is a non-rigorous method for estimating a large amount of CFT data from a small number of computations. We hope to present a more detailed analysis of our extremal functionals for the O(2) model in future work. For now, we give estimates of the dimensions of a few important low-lying scalars in table 4. To obtain extremal functionals, we chose 20 allowed points in the $\Lambda = 43$ island and computed lower and upper bounds on the norm of the external OPE vector $|\lambda_{\text{ext}}|$ with derivative order $\Lambda = 27$ (shown in table 10 of appendix E). Comparing the zeros of the resulting functionals, we identified *stable* zeros whose positions did not vary significantly as we changed the point in the island [97]. Thus, for 20 points, we have 40 different values of $\Delta_{s'}$, $\Delta_{t'}$, $\Delta_{\text{charge 3}}$, and $\Delta_{\text{charge 4}}$ (half of them are from the lower bound computations, while another half are

Dim	Method	value	ref.
$\Delta_{s'}$	MC	3.789(4)	[86]
	CB	3.794(8*)	
$\Delta_{t'}$	FT	3.624(10)	[102]
	CB	3.650(2*)	
$\Delta_{\text{charge 3}}$	MC	2.1085(20)	[87]
	CB	2.1086(3*)	
$\Delta_{\text{charge 4}}$	MC	3.114(2)	[106]
	CB	3.14(2*)	

Table 4. Comparison of conformal bootstrap (CB) estimates using the extremal functional method with previous Monte Carlo (MC) and ϵ-expansion (FT) determinations of operator dimensions. The values for the extremal functional determinations are means across the 40 different extremal spectra, and the errors are the standard deviations. We mark the errors with a * to emphasize that they are non-rigorous. Here, "charge-3" and "charge-4" refer to the lowest-dimension scalars with the given charges, which in field theory language are $\phi_{(i}\phi_j\phi_{k)}$ and $\phi_{(i}\phi_j\phi_k\phi_{l)}$.

from the upper bound computations). The gaps we impose are the same as in the OPE scan discussed before, except that we set the twist gap δ_τ to 10^{-4}.

Acknowledgments

A   Code availability

All code used in this work is available online. This includes

- The semidefinite program solver SDPB:
  https://github.com/davidsd/sdpb

- Code for generating tables of scalar conformal blocks:
  https://gitlab.com/bootstrapcollaboration/scalar_blocks

- A Mathematica framework for bootstrap calculations, including implementations of
  the cutting surface and Delaunay triangulation algorithms described in section 3:
  https://gitlab.com/bootstrapcollaboration/simpleboot.

- A Haskell framework for concurrent computations on an HPC cluster:
  https://github.com/davidsd/hyperion

- Haskell libraries for bootstrap computations, including implementations of the cutting
  surface and Delaunay triangulation algorithms described in section 3:
  https://gitlab.com/davidsd/sdpb-haskell
  https://gitlab.com/davidsd/hyperion-bootstrap
  https://gitlab.com/davidsd/hyperion-projects

- A Haskell library and standalone executable for solving quadratically constrained
  problems by a combination of semidefinite relaxation and other heuristics
  https://gitlab.com/davidsd/quadratic-net/

B   Software setup and parameters

The computations of the $O(2)$ model islands described in section 4.2 with $\Lambda = 19, 27$ were
performed on the Caltech HPC Cluster and the Yale Grace Cluster. For the computations with $\Lambda = 35$ and $\Lambda = 43$, we tested possible primal points using the Caltech and
Yale clusters. In each case, after finding a few initial primal points, the main Delaunay
triangulation search was performed on the XSEDE [119] Comet Cluster at the San Diego
Supercomputing Center through allocation PHY190023. The computation of the $\Lambda = 35$
island took 192K core-hours and was completed in 4 days. The computation of the $\Lambda = 43$
island took 1.03M core-hours and was completed in 2 weeks.

In table 5, we list the SDPB and scalar_blocks parameters for the $\Lambda = 35, 43$ island
computations. (Parameters for other values of Λ are available upon request.) In table 6, we
list the parameters for the extremal functional computations with $\Lambda = 27$ section 4.4. Note
that for the island computation, the parameters findPrimalFeasible, findDualFeasible,
detectPrimalFeasibleJump, and detectDualFeasibleJump are set in accordance with
the discussion in section 3.6.

Λ	35	43
keptPoleOrder	32	40
order	80	90
spins	S_{35}	S_{43}
precision	960	1024
dualityGapThreshold	10^{-30}	10^{-75}
primalErrorThreshold	10^{-200}	10^{-200}
dualErrorThreshold	10^{-200}	10^{-200}
findPrimalFeasible	false	false
findDualFeasible	false	false
detectPrimalFeasibleJump	true	true
detectDualFeasibleJump	true	true
initialMatrixScalePrimal	10^{50}	10^{60}
initialMatrixScaleDual	10^{50}	10^{60}
feasibleCenteringParameter	0.1	0.1
infeasibleCenteringParameter	0.3	0.3
stepLengthReduction	0.7	0.7
maxComplementarity	10^{160}	10^{200}

Table 5. Parameters for the computations in section 4.2. The sets $S_{35,43}$ are defined in (B.1).

Λ	27
keptPoleOrder	12
order	60
spins	S_{27}
precision	900
dualityGapThreshold	10^{-80}
primalErrorThreshold	10^{-200}
dualErrorThreshold	10^{-100}
initialMatrixScalePrimal	10^{20}
initialMatrixScaleDual	10^{20}
feasibleCenteringParameter	0.1
infeasibleCenteringParameter	0.3
stepLengthReduction	0.7
maxComplementarity	10^{200}

Table 6. Parameters for the computations in section 4.4. The set S_{27} is defined in (B.1).

The sets of spins used for each value of Λ were

$$
\begin{aligned}
S_{27} &= \{0, \dots, 31\} \cup \{49, 50\}, \\
S_{35} &= \{0, \dots, 44\} \cup \{47, 48, 51, 52, 55, 56, 59, 60, 63, 64, 67, 68\}, \\
S_{43} &= \{0, \dots, 64\} \cup \{67, 68, 71, 72, 75, 76, 79, 80, 83, 84, 87, 88\}.
\end{aligned} \tag{B.1}
$$

C Tensor structures

In this appendix we compute all the O(2) structures $T^R_{\mathcal{R}_1 \mathcal{R}_2 \mathcal{R}_3 \mathcal{R}_4}(y_i)$ that appear in the block expansion (2.8) for the 4-point functions we consider as listed in table 2. The block expansion in the s-channel is derived by inserting a complete set of states

$$
\sum_{\alpha = \mathcal{O}, P\mathcal{O}, PP\mathcal{O}} \frac{\langle \varphi^1_{\mathcal{R}_1}(x_1, y_1) \varphi^2_{\mathcal{R}_2}(x_2, y_2) | \alpha \rangle \langle \alpha | \varphi^3_{\mathcal{R}_3}(x_3, y_3) \varphi^4_{\mathcal{R}_4}(x_4, y_4) \rangle}{\langle \alpha | \alpha \rangle}, \tag{C.1}
$$

where α runs over an orthogonal basis of operators $\mathcal{O}$ (and descendents) in irrep R that appear in the OPEs $\varphi^1 \times \varphi^2$ and $\varphi^3 \times \varphi^4$. The 4-point structure $T^R_{\mathcal{R}_1 \mathcal{R}_2 \mathcal{R}_3 \mathcal{R}_4}(y_i)$ can then be written in terms of the O(2) structures $T^R_{\mathcal{R}_i \mathcal{R}_j}(y_i, y_j, y)$ of each of the pair of 3-point functions as

$$
T^R_{\mathcal{R}_1 \mathcal{R}_2 \mathcal{R}_3 \mathcal{R}_4}(y_i) = (T_{\mathcal{R}_1 \mathcal{R}_2 R}(y_1, y_2, y), T_{\mathcal{R}_3 \mathcal{R}_4 R}(y_3, y_4, y)), \tag{C.2}
$$

where $(f(y), g(y))$ denotes the contraction over y in index free notation. When R is $\mathbf{0}^{\pm}$, the contraction is just multiplication of the three-point structures. When R has nonzero charge n, this contraction can be derived by expanding each rank n O(2) tensor in the basis

$$
e = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}, \qquad e = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}, \tag{C.3}
$$

as

$$
f(y) = f(e)(y \cdot \bar{e})^n + f(\bar{e})(y \cdot e)^n, \tag{C.4}
$$

and similarly for $g(y)$. This basis has the convenient properties $e \cdot e = \bar{e} \cdot \bar{e} = 0$ and $e \cdot \bar{e} = 1$, so that the contraction of the tensors in index free notation is

$$
(f(y), g(y)) = f(e)g(\bar{e}) + f(\bar{e})g(e). \tag{C.5}
$$

The result of these contractions can then be written in terms of the quantities

$$w_i \equiv y_i \cdot e \,, \qquad \bar{w}_i \equiv y_i \cdot \bar{e} \,, \tag{C.6}$$

which have the properties

$$y_i \cdot y_i = w_i \bar{w}_i = 0 \,, \qquad y_i \cdot y_j = w_i \bar{w}_j + \bar{w}_i w_j \,, \qquad y_i \wedge y_j = i(w_i \bar{w}_j - \bar{w}_i w_j) \,, \tag{C.7}$$

which imply that $w_i = 0$ or $\bar{w}_i = 0$ since $y_i^2 = 0$ by definition.

The utility of this derivation is that each 3-point structure establishes a convention for the OPE coefficient $\lambda_{\varphi_i \varphi_j \mathcal{O}}$ of the associated 3-point function, so computing the 4-point structures in terms of these 3-point structures ensures that the coefficients $\lambda_{\varphi_1 \varphi_2 \mathcal{O}} \lambda_{\varphi_3 \varphi_4 \mathcal{O}}$ that appear in (2.8) can be consistently identified with these OPE coefficients. For each s- and t-channel configuration in table 2 with an independent O(2) structure, the resulting four-point structures are:

$$
\begin{aligned}
\langle \phi\phi\phi\phi \rangle : \quad & T^{\mathbf{0}^+}_{\mathbf{1}_i \mathbf{1}_j \mathbf{1}_k \mathbf{1}_l} = (w_i \bar{w}_j + \bar{w}_i w_j)(w_k \bar{w}_l + \bar{w}_k w_l) \,, \\
& T^{\mathbf{0}^-}_{\mathbf{1}_i \mathbf{1}_j \mathbf{1}_k \mathbf{1}_l} = -(w_i \bar{w}_j - \bar{w}_i w_j)(w_k \bar{w}_l - \bar{w}_k w_l) \,, \\
& T^{\mathbf{2}}_{\mathbf{1}_i \mathbf{1}_j \mathbf{1}_k \mathbf{1}_l} = w_i w_j \bar{w}_k \bar{w}_l + \bar{w}_i \bar{w}_j w_k w_l \,, \\
\langle tttt \rangle : \quad & T^{\mathbf{0}^+}_{\mathbf{2}_i \mathbf{2}_j \mathbf{2}_k \mathbf{2}_l} = (w_i \bar{w}_j + \bar{w}_i w_j)^2 (w_k \bar{w}_l + \bar{w}_k w_l)^2 \,, \\
& T^{\mathbf{0}^-}_{\mathbf{2}_i \mathbf{2}_j \mathbf{2}_k \mathbf{2}_l} = -(w_i^2 \bar{w}_j^2 - \bar{w}_i^2 w_j^2)(w_k^2 \bar{w}_l^2 - \bar{w}_k^2 w_l^2) \,, \\
& T^{\mathbf{4}}_{\mathbf{2}_i \mathbf{2}_j \mathbf{2}_k \mathbf{2}_l} = (w_i w_j \bar{w}_k \bar{w}_l + \bar{w}_i \bar{w}_j w_k w_l)^2 \,, \\
\langle t\phi t\phi \rangle \,, \langle \phi t t \phi \rangle : \quad & T^{\mathbf{1}}_{\mathbf{2}_i \mathbf{1}_j \mathbf{2}_k \mathbf{1}_l} = (w_i \bar{w}_j + \bar{w}_i w_j)(w_k \bar{w}_l + \bar{w}_k w_l)(w_i \bar{w}_k + \bar{w}_i w_k) \,, \\
& T^{\mathbf{3}}_{\mathbf{2}_i \mathbf{1}_j \mathbf{2}_k \mathbf{1}_l} = w_i^2 w_j \bar{w}_k^2 \bar{w}_l + \bar{w}_i^2 \bar{w}_j w_k^2 w_l \,, \\
\langle tt\phi\phi \rangle : \quad & T^{\mathbf{0}^+}_{\mathbf{2}_i \mathbf{2}_j \mathbf{1}_k \mathbf{1}_l} = (w_i \bar{w}_j + \bar{w}_i w_j)^2 (w_k \bar{w}_l + \bar{w}_k w_l) \,, \\
& T^{\mathbf{0}^-}_{\mathbf{2}_i \mathbf{1}_j \mathbf{2}_k \mathbf{1}_l} = -(w_i^2 \bar{w}_j^2 - \bar{w}_i^2 w_j^2)(w_k \bar{w}_l - \bar{w}_k w_l) \,, \\
\langle ssss \rangle : \quad & T^{\mathbf{0}^+}_{\mathbf{0}^+ \mathbf{0}^+ \mathbf{0}^+ \mathbf{0}^+} = 1 \,, \\
\langle \phi s\phi s \rangle \,, \langle s\phi\phi s \rangle : \quad & T^{\mathbf{1}}_{\mathbf{1}_i \mathbf{0}^+ \mathbf{1}_k \mathbf{0}^+} = w_i \bar{w}_k + \bar{w}_i w_k \,, \\
\langle tsts \rangle \,, \langle stts \rangle : \quad & T^{\mathbf{2}}_{\mathbf{2}_i \mathbf{0}^+ \mathbf{2}_k \mathbf{0}^+} = w_i^2 \bar{w}_k^2 + \bar{w}_i^2 w_k^2 \,, \\
\langle ttss \rangle : \quad & T^{\mathbf{0}^+}_{\mathbf{2}_i \mathbf{2}_j \mathbf{0}^+ \mathbf{0}^+} = (w_i \bar{w}_j + \bar{w}_i w_j)^2 \,, \\
\langle \phi\phi ss \rangle : \quad & T^{\mathbf{0}^+}_{\mathbf{1}_i \mathbf{1}_j \mathbf{0}^+ \mathbf{0}^+} = w_i \bar{w}_j + \bar{w}_i w_j \,, \\
\langle \phi s\phi t \rangle \,, \langle s\phi\phi t \rangle : \quad & T^{\mathbf{1}}_{\mathbf{1}_i \mathbf{0}^+ \mathbf{1}_k \mathbf{2}_l} = (w_k \bar{w}_l + \bar{w}_k w_l)(w_i \bar{w}_l + \bar{w}_i w_l) \,, \\
\langle \phi\phi st \rangle : \quad & T^{\mathbf{2}}_{\mathbf{1}_i \mathbf{1}_j \mathbf{0}^+ \mathbf{2}_l} = w_l^2 \bar{w}_j \bar{w}_i + \bar{w}_l^2 w_j w_i \,.
\end{aligned}
\tag{C.8}
$$

D Crossing vectors

Here we write the explicit vectors of crossing equations. In the following, an entry of 0 will denote either a scalar or matrix of scalars depending on if the crossing equation is a scalar

or a matrix.

$$
\vec{V}_{\mathbf{O}^+,\Delta,\ell^+} = \begin{pmatrix}
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 2F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
0 \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & -2F^{\phi\phi,\phi\phi}_{+,\Delta,\ell} & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2F^{tt,tt}_{-,\Delta,\ell} \end{pmatrix} \\
0 \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2F^{tt,tt}_{+,\Delta,\ell} \end{pmatrix} \\
0 \\
0 \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & F^{tt,\phi\phi}_{-,\Delta,\ell} \\ 0 & F^{tt,\phi\phi}_{-,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & F^{tt,\phi\phi}_{-,\Delta,\ell} \\ 0 & F^{tt,\phi\phi}_{-,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -F^{tt,\phi\phi}_{+,\Delta,\ell} \\ 0 & -F^{tt,\phi\phi}_{+,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -F^{tt,\phi\phi}_{+,\Delta,\ell} \\ 0 & -F^{tt,\phi\phi}_{+,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 2F^{ss,ss}_{-,\Delta,\ell} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
0 \\
0 \\
\begin{pmatrix} 0 & 0 & F^{tt,ss}_{-,\Delta,\ell} \\ 0 & 0 & 0 \\ F^{tt,ss}_{-,\Delta,\ell} & 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & F^{tt,ss}_{+,\Delta,\ell} \\ 0 & 0 & 0 \\ F^{tt,ss}_{+,\Delta,\ell} & 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & F^{\phi\phi,ss}_{-,\Delta,\ell} & 0 \\ F^{\phi\phi,ss}_{-,\Delta,\ell} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & F^{\phi\phi,ss}_{+,\Delta,\ell} & 0 \\ F^{\phi\phi,ss}_{+,\Delta,\ell} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
0 \\
0 \\
0
\end{pmatrix},
\qquad
\vec{V}_{\mathbf{O}^-,\Delta,\ell^-} = \begin{pmatrix}
\begin{pmatrix} 2F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} & 0 \\ 0 & 0 \end{pmatrix} \\
\begin{pmatrix} -4F^{\phi\phi,\phi\phi}_{-,\Delta,\ell} & 0 \\ 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 2F^{\phi\phi,\phi\phi}_{+,\Delta,\ell} & 0 \\ 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 \\ 0 & 2F^{tt,tt}_{-,\Delta,\ell} \end{pmatrix} \\
\begin{pmatrix} 0 & 0 \\ 0 & -4F^{tt,tt}_{-,\Delta,\ell} \end{pmatrix} \\
\begin{pmatrix} 0 & 0 \\ 0 & 2F^{tt,tt}_{+,\Delta,\ell} \end{pmatrix} \\
0 \\
0 \\
\begin{pmatrix} 0 & -F^{tt,\phi\phi}_{-,\Delta,\ell} \\ -F^{tt,\phi\phi}_{-,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & F^{tt,\phi\phi}_{-,\Delta,\ell} \\ F^{tt,\phi\phi}_{-,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & F^{tt,\phi\phi}_{+,\Delta,\ell} \\ F^{tt,\phi\phi}_{+,\Delta,\ell} & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & -F^{tt,\phi\phi}_{+,\Delta,\ell} \\ -F^{tt,\phi\phi}_{+,\Delta,\ell} & 0 \end{pmatrix} \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix},
\qquad \text{(D.1)}
$$

$$\vec{V}_{\mathbf{1},\Delta,\ell\pm} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \begin{pmatrix} 0 & 0 \\ 0 & 2(-1)^{\ell}F_{-,\Delta,\ell}^{t\phi,t\phi} \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 2(-1)^{\ell}F_{+,\Delta,\ell}^{t\phi,t\phi} \end{pmatrix} \\ 0 \\ \begin{pmatrix} 0 & 0 \\ 0 & 2F_{-,\Delta,\ell}^{\phi t,t\phi} \end{pmatrix} \\ 0 \\ \begin{pmatrix} 0 & 0 \\ 0 & 2F_{+,\Delta,\ell}^{\phi t,t\phi} \end{pmatrix} \\ 0 \\ \begin{pmatrix} 2(-1)^{\ell}F_{-,\Delta,\ell}^{\phi s,\phi s} & 0 \\ 0 & 0 \end{pmatrix} \\ 0 \\ 0 \\ 0 \\ \begin{pmatrix} 2F_{-,\Delta,\ell}^{s\phi,\phi s} & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} -2F_{+,\Delta,\ell}^{s\phi,\phi s} & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & F_{-,\Delta,\ell}^{\phi s,\phi t} \\ F_{-,\Delta,\ell}^{\phi s,\phi t} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & (-1)^{\ell}F_{-,\Delta,\ell}^{s\phi,\phi t} \\ (-1)^{\ell}F_{-,\Delta,\ell}^{s\phi,\phi t} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & (-1)^{\ell}F_{+,\Delta,\ell}^{s\phi,\phi t} \\ (-1)^{\ell}F_{+,\Delta,\ell}^{s\phi,\phi t} & 0 \end{pmatrix} \end{pmatrix}, \qquad \vec{V}_{\mathbf{2},\Delta,\ell+} = \begin{pmatrix} 0 \\ \begin{pmatrix} 2F_{-,\Delta,\ell}^{\phi\phi,\phi\phi} & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 2F_{+,\Delta,J\ell}^{\phi\phi,\phi\phi} & 0 \\ 0 & 0 \end{pmatrix} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \begin{pmatrix} 0 & 0 \\ 0 & 2F_{-,\Delta,\ell}^{ts,ts} \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 2F_{-,\Delta,\ell}^{st,ts} \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & -2F_{+,\Delta,\ell}^{st,ts} \end{pmatrix} \\ 0 \\ 0 \\ 0 \\ \begin{pmatrix} 0 & F_{-,\Delta,\ell}^{\phi\phi,st} \\ F_{-,\Delta,\ell}^{\phi\phi,st} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & -F_{+,\Delta,\ell}^{\phi\phi,st} \\ -F_{+,\Delta,\ell}^{\phi\phi,st} & 0 \end{pmatrix} \end{pmatrix}, \qquad \text{(D.2)}$$

$$\vec{V}_{\mathbf{2},\Delta,\ell^-} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2F_{-,\Delta,\ell}^{ts,ts} \\ 2F_{-,\Delta,\ell}^{st,ts} \\ -2F_{+,\Delta,\ell}^{st,ts} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} , \quad \vec{V}_{\mathbf{3},\Delta,\ell^\pm} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2(-1)^\ell F_{-,\Delta,\ell}^{t\phi,t\phi} \\ -2(-1)^\ell F_{+,\Delta,\ell}^{t\phi,t\phi} \\ 2F_{-,\Delta,\ell}^{\phi t,t\phi} \\ 0 \\ 2F_{+,\Delta,\ell}^{\phi t,t\phi} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} , \quad \vec{V}_{\mathbf{4},\Delta,\ell^+} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2F_{-,\Delta,\ell}^{tt,tt} \\ 2F_{+,\Delta,\ell}^{tt,tt} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} . \tag{D.3}$$

E   Computed points

Δ_ϕ	Δ_s	Δ_t	$\frac{\lambda_{sss}}{\lambda_{\phi\phi s}}$	$\frac{\lambda_{tts}}{\lambda_{\phi\phi s}}$	$\frac{\lambda_{\phi\phi t}}{\lambda_{\phi\phi s}}$
0.519091478	1.51141697	1.23631316	1.23631316	1.23631316	1.23631316
0.519088325	1.51139275	1.23629816	1.23629816	1.23629816	1.23629816
0.519085258	1.51131148	1.23626768	1.23626768	1.23626768	1.23626768
0.519083027	1.51130787	1.23626810	1.23626810	1.23626810	1.23626810
0.519084900	1.51132513	1.23626125	1.23626125	1.23626125	1.23626125
0.519101167	1.51147622	1.23635261	1.23635261	1.23635261	1.23635261
0.519079494	1.51130889	1.23625139	1.23625139	1.23625139	1.23625139
0.519088780	1.51141601	1.23631255	1.23631255	1.23631255	1.23631255
0.519099104	1.51149674	1.23636042	1.23636042	1.23636042	1.23636042
0.519074036	1.51122813	1.23622003	1.23622003	1.23622003	1.23622003
0.519075834	1.51124069	1.23621228	1.23621228	1.23621228	1.23621228
0.519086133	1.51140646	1.23629626	1.23629626	1.23629626	1.23629626
0.519091591	1.51144378	1.23631466	1.23631466	1.23631466	1.23631466
0.519101492	1.51147122	1.23635764	1.23635764	1.23635764	1.23635764
0.519095922	1.51143997	1.23632426	1.23632426	1.23632426	1.23632426
0.519089922	1.51145388	1.23632418	1.23632418	1.23632418	1.23632418
0.519096569	1.51145694	1.23634757	1.23634757	1.23634757	1.23634757
0.519078927	1.51129693	1.23625413	1.23625413	1.23625413	1.23625413
0.519085163	1.51135762	1.23627100	1.23627100	1.23627100	1.23627100
0.519095326	1.51148380	1.23634189	1.23634189	1.23634189	1.23634189
0.519081546	1.51129674	1.23625401	1.23625401	1.23625401	1.23625401
0.519078552	1.51131491	1.23624987	1.23624987	1.23624987	1.23624987
0.519104279	1.51152063	1.23637609	1.23637609	1.23637609	1.23637609
0.519077715	1.51124447	1.23623187	1.23623187	1.23623187	1.23623187
0.519074849	1.51125858	1.23622291	1.23622291	1.23622291	1.23622291
0.519081236	1.51134317	1.23627346	1.23627346	1.23627346	1.23627346
0.519087675	1.51137648	1.23630209	1.23630209	1.23630209	1.23630209
0.519092708	1.51139697	1.23631672	1.23631672	1.23631672	1.23631672
0.519080005	1.51131897	1.23624739	1.23624739	1.23624739	1.23624739
0.519096168	1.51149661	1.23635270	1.23635270	1.23635270	1.23635270
0.519073619	1.51122331	1.23621261	1.23621261	1.23621261	1.23621261
0.519085778	1.51132384	1.23628076	1.23628076	1.23628076	1.23628076
0.519075030	1.51121405	1.23622082	1.23622082	1.23622082	1.23622082

Table 7. Allowed points in the $\Lambda = 43$ island.

Δ_ϕ	Δ_s	Δ_t
0.519102918	1.51155239	1.23637912
0.519108668	1.51153259	1.23638777
0.519084234	1.51130123	1.23627522
0.519086029	1.51135180	1.23626619
0.519093006	1.51136316	1.23630104
0.519074320	1.51118490	1.23620389
0.519102521	1.51148465	1.23635216
0.519109629	1.51158377	1.23640397
0.519077293	1.51123239	1.23621599
0.519086333	1.51131583	1.23626238
0.519088681	1.51132415	1.23628443
0.519103625	1.51156649	1.23639258
0.519097531	1.51152436	1.23635506
0.519104829	1.51155133	1.23639016
0.519106540	1.51151848	1.23638657
0.519099641	1.51149324	1.23634449
0.519091345	1.51145618	1.23631607
0.519099918	1.51143846	1.23634527
0.519081611	1.51130758	1.23626964
0.519093364	1.51150044	1.23634838
0.519090250	1.51142367	1.23630035
0.519095800	1.51148226	1.23635217
0.519079641	1.51134752	1.23626571
0.519066632	1.51113867	1.23617714
0.519089008	1.51142764	1.23631975
0.519081958	1.51136634	1.23626970
0.519073136	1.51120099	1.23619139
0.519079477	1.51125698	1.23623547
0.519092469	1.51139541	1.23629967
0.519091772	1.51141270	1.23632592
0.519092673	1.51139406	1.23632062
0.519069909	1.51118566	1.23618371
0.519101366	1.51149948	1.23637079
0.519090130	1.51135761	1.23628884
0.519082450	1.51133857	1.23628007
0.519107673	1.51153927	1.23639093
0.519096449	1.51141705	1.23632503
0.519074457	1.51126505	1.23622784
0.519089400	1.51140721	1.23631629
0.519080527	1.51132179	1.23624620
0.519075418	1.51126954	1.23622125
0.519071183	1.51118343	1.23619298

Table 8. Disallowed points computed at $\Lambda = 43$.

Δ_ϕ	Δ_s	Δ_t	$\frac{\lambda_{sss}}{\lambda_{\phi\phi s}}$	$\frac{\lambda_{tts}}{\lambda_{\phi\phi s}}$	$\frac{\lambda_{\phi\phi t}}{\lambda_{\phi\phi s}}$
0.519101167	1.51147622	1.23635261	1.20936871	1.82235941	1.76596240
0.519079494	1.51130889	1.23625139	1.20934084	1.82223619	1.76589343
0.519089922	1.51145388	1.23632418	1.20972662	1.82245009	1.76596250
0.519075834	1.51124069	1.23621228	1.20906418	1.82207926	1.76585335
0.519075030	1.51121405	1.23622082	1.20879917	1.82210575	1.76586410
0.519091591	1.51144378	1.23631466	1.20970116	1.82235729	1.76594661
0.519086715	1.51136546	1.23628759	1.20932228	1.82228021	1.76592047

Table 9. Allowed points in the $\Lambda = 43$ island used for computing upper and lower bounds on C_T, C_J, and $\lambda_{\phi\phi s}$.

Δ_ϕ	Δ_s	Δ_t	$\frac{\lambda_{sss}}{\lambda_{\phi\phi s}}$	$\frac{\lambda_{tts}}{\lambda_{\phi\phi s}}$	$\frac{\lambda_{\phi\phi t}}{\lambda_{\phi\phi s}}$
0.519130434	1.51173444	1.23648971	1.20977354	1.82254374	1.76606470
0.519135171	1.51172427	1.23649356	1.20947477	1.82245370	1.76605159
0.519076518	1.51110487	1.23620503	1.20766586	1.82191247	1.76584197
0.519115548	1.51167580	1.23642873	1.21014420	1.82257643	1.76603227
0.519113909	1.51170936	1.23646025	1.21013097	1.82272756	1.76607582
0.519096732	1.51147972	1.23636344	1.20944426	1.82251617	1.76600087
0.519128801	1.51168098	1.23648846	1.20929738	1.82252856	1.76605495
0.519119255	1.51170685	1.23646324	1.21007964	1.82275976	1.76606055
0.519109342	1.51150256	1.23640031	1.20891847	1.82236481	1.76600112
0.519087647	1.51141667	1.23630721	1.20963450	1.82247476	1.76594440
0.519105802	1.51141826	1.23635621	1.20856734	1.82219520	1.76595563
0.519125142	1.51173460	1.23646472	1.21012577	1.82250871	1.76605236
0.519107610	1.51164424	1.23640715	1.21022297	1.82258938	1.76603036
0.519115226	1.51174173	1.23647414	1.21033291	1.82281805	1.76609054
0.519084390	1.51137895	1.23628833	1.20979136	1.82229748	1.76593252
0.519096529	1.51153244	1.23635748	1.20995866	1.82250999	1.76599060
0.519122718	1.51168123	1.23647847	1.20940108	1.82261368	1.76607344
0.519138689	1.51177044	1.23653770	1.20947377	1.82262309	1.76609008
0.519057668	1.51097950	1.23611240	1.20794762	1.82181966	1.76576836
0.519074424	1.51116298	1.23616082	1.20864157	1.82181577	1.76579563

Table 10. Allowed points in the $\Lambda = 35$ island used for obtaining low-lying scalar operator dimensions via the extremal functional method.

References

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

[2] V.S. Rychkov and A. Vichi, *Universal Constraints on Conformal Operator Dimensions*, *Phys. Rev.* **D 80** (2009) 045006 [arXiv:0905.2211] [ɪɴSPIRE].

[3] D. Poland, S. Rychkov and A. Vichi, *The Conformal Bootstrap: Theory, Numerical Techniques and Applications*, *Rev. Mod. Phys.* **91** (2019) 015002 [arXiv:1805.04405] [ɪɴSPIRE].

[4] S.M. Chester, *Weizmann Lectures on the Numerical Conformal Bootstrap*, arXiv:1907.05147 [ɪɴSPIRE].

[5] F. Kos, D. Poland and D. Simmons-Duffin, *Bootstrapping Mixed Correlators in the* 3D *Ising Model*, *JHEP* **11** (2014) 109 [arXiv:1406.4858] [ɪɴSPIRE].

[6] F. Kos, D. Poland, D. Simmons-Duffin and A. Vichi, *Bootstrapping the* O(N) *Archipelago*, *JHEP* **11** (2015) 106 [arXiv:1504.07997] [ɪɴSPIRE].

[7] 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] [ɪɴSPIRE].

[8] J. Rong and N. Su, *Bootstrapping minimal* $\mathcal{N} = 1$ *superconformal field theory in three dimensions*, arXiv:1807.04434 [ɪɴSPIRE].

[9] N.B. Agmon, S.M. Chester and S.S. Pufu, *The M-theory Archipelago*, *JHEP* **02** (2020) 010 [arXiv:1907.13222] [ɪɴSPIRE].

[10] Z. Li and N. Su, *Bootstrapping Mixed Correlators in the Five Dimensional Critical* O(N) *Models*, *JHEP* **04** (2017) 098 [arXiv:1607.07077] [ɪɴSPIRE].

[11] Y. Nakayama and T. Ohtsuki, *Conformal Bootstrap Dashing Hopes of Emergent Symmetry*, *Phys. Rev. Lett.* **117** (2016) 131601 [arXiv:1602.07295] [ɪɴSPIRE].

[12] D. Li, D. Meltzer and A. Stergiou, *Bootstrapping mixed correlators in 4D* $\mathcal{N} = 1$ *SCFTs*, *JHEP* **07** (2017) 029 [arXiv:1702.00404] [ɪɴSPIRE].

[13] C. Behan, *Bootstrapping the long-range Ising model in three dimensions*, *J. Phys.* **A 52** (2019) 075401 [arXiv:1810.07199] [ɪɴSPIRE].

[14] S.R. Kousvos and A. Stergiou, *Bootstrapping Mixed Correlators in Three-Dimensional Cubic Theories*, *SciPost Phys.* **6** (2019) 035 [arXiv:1810.10015] [ɪɴSPIRE].

[15] S.R. Kousvos and A. Stergiou, *Bootstrapping Mixed Correlators in Three-Dimensional Cubic Theories II*, arXiv:1911.00522 [ɪɴSPIRE].

[16] L. Iliesiu, F. Kos, D. Poland, S.S. Pufu, D. Simmons-Duffin and R. Yacoby, *Bootstrapping* 3D *Fermions*, *JHEP* **03** (2016) 120 [arXiv:1508.00012] [ɪɴSPIRE].

[17] L. Iliesiu, F. Kos, D. Poland, S.S. Pufu and D. Simmons-Duffin, *Bootstrapping* 3D *Fermions with Global Symmetries*, *JHEP* **01** (2018) 036 [arXiv:1705.03484] [ɪɴSPIRE].

[18] D. Karateev, P. Kravchuk, M. Serone and A. Vichi, *Fermion Conformal Bootstrap in 4d*, *JHEP* **06** (2019) 088 [arXiv:1902.05969] [INSPIRE].

[19] A. Dymarsky, J. Penedones, E. Trevisani and A. Vichi, *Charting the space of 3D CFTs with a continuous global symmetry*, *JHEP* **05** (2019) 098 [arXiv:1705.04278] [INSPIRE].

[20] M. Reehorst, E. Trevisani and A. Vichi, *Mixed Scalar-Current bootstrap in three dimensions*, arXiv:1911.05747 [INSPIRE].

[21] A. Dymarsky, F. Kos, P. Kravchuk, D. Poland and D. Simmons-Duffin, *The 3d Stress-Tensor Bootstrap*, *JHEP* **02** (2018) 164 [arXiv:1708.05718] [INSPIRE].

[22] R. Rattazzi, S. Rychkov and A. Vichi, *Bounds in 4D Conformal Field Theories with Global Symmetry*, *J. Phys.* **A 44** (2011) 035402 [arXiv:1009.5985] [INSPIRE].

[23] A. Vichi, *Improved bounds for CFT's with global symmetries*, *JHEP* **01** (2012) 162 [arXiv:1106.4037] [INSPIRE].

[24] D. Poland, D. Simmons-Duffin and A. Vichi, *Carving Out the Space of 4D CFTs*, *JHEP* **05** (2012) 110 [arXiv:1109.5176] [INSPIRE].

[25] F. Kos, D. Poland and D. Simmons-Duffin, *Bootstrapping the O(N) vector models*, *JHEP* **06** (2014) 091 [arXiv:1307.6856] [INSPIRE].

[26] M. Berkooz, R. Yacoby and A. Zait, *Bounds on $\mathcal{N} = 1$ superconformal theories with global symmetries*, *JHEP* **08** (2014) 008 [*Erratum ibid.* **01** (2015) 132] [arXiv:1402.6068] [INSPIRE].

[27] Y. Nakayama and T. Ohtsuki, *Approaching the conformal window of $O(n) \times O(m)$ symmetric Landau-Ginzburg models using the conformal bootstrap*, *Phys. Rev.* **D 89** (2014) 126009 [arXiv:1404.0489] [INSPIRE].

[28] F. Caracciolo, A. Castedo Echeverri, B. von Harling and M. Serone, *Bounds on OPE Coefficients in 4D Conformal Field Theories*, *JHEP* **10** (2014) 020 [arXiv:1406.7845] [INSPIRE].

[29] Y. Nakayama and T. Ohtsuki, *Bootstrapping phase transitions in QCD and frustrated spin systems*, *Phys. Rev.* **D 91** (2015) 021901 [arXiv:1407.6195] [INSPIRE].

[30] S.M. Chester, S.S. Pufu and R. Yacoby, *Bootstrapping O(N) vector models in $4 < d < 6$*, *Phys. Rev.* **D 91** (2015) 086014 [arXiv:1412.7746] [INSPIRE].

[31] Y. Nakayama and T. Ohtsuki, *Five dimensional O(N)-symmetric CFTs from conformal bootstrap*, *Phys. Lett.* **B 734** (2014) 193 [arXiv:1404.5201] [INSPIRE].

[32] S.M. Chester, S. Giombi, L.V. Iliesiu, I.R. Klebanov, S.S. Pufu and R. Yacoby, *Accidental Symmetries and the Conformal Bootstrap*, *JHEP* **01** (2016) 110 [arXiv:1507.04424] [INSPIRE].

[33] S.M. Chester, L.V. Iliesiu, S.S. Pufu and R. Yacoby, *Bootstrapping O(N) Vector Models with Four Supercharges in $3 \le d \le 4$*, *JHEP* **05** (2016) 103 [arXiv:1511.07552] [INSPIRE].

[34] S.M. Chester and S.S. Pufu, *Towards bootstrapping QED$_3$*, *JHEP* **08** (2016) 019 [arXiv:1601.03476] [INSPIRE].

[35] Y. Nakayama, *Bootstrap bound for conformal multi-flavor QCD on lattice*, *JHEP* **07** (2016) 038 [arXiv:1605.04052] [INSPIRE].

[36] H. Iha, H. Makino and H. Suzuki, *Upper bound on the mass anomalous dimension in many-flavor gauge theories: a conformal bootstrap approach*, *PTEP* **2016** (2016) 053B03 [arXiv:1603.01995] [INSPIRE].

[37] Y. Nakayama, *Bootstrap experiments on higher dimensional CFTs*, *Int. J. Mod. Phys.* **A 33** (2018) 1850036 [arXiv:1705.02744] [INSPIRE].

[38] J. Rong and N. Su, *Scalar CFTs and Their Large N Limits*, *JHEP* **09** (2018) 103 [arXiv:1712.00985] [INSPIRE].

[39] S.M. Chester, L.V. Iliesiu, M. Mezei and S.S. Pufu, *Monopole Operators in* U(1) *Chern-Simons-Matter Theories*, *JHEP* **05** (2018) 157 [arXiv:1710.00654] [INSPIRE].

[40] A. Stergiou, *Bootstrapping hypercubic and hypertetrahedral theories in three dimensions*, *JHEP* **05** (2018) 035 [arXiv:1801.07127] [INSPIRE].

[41] Z. Li, *Solving QED$_3$ with Conformal Bootstrap*, arXiv:1812.09281 [INSPIRE].

[42] J. Rong and N. Su, *Bootstrapping the $\mathcal{N} = 1$ Wess-Zumino models in three dimensions*, arXiv:1910.08578 [INSPIRE].

[43] S. Rychkov, *Conformal Bootstrap in Three Dimensions?*, arXiv:1111.2115 [INSPIRE].

[44] S. El-Showk, M.F. Paulos, D. Poland, S. Rychkov, D. Simmons-Duffin and A. Vichi, *Solving the* 3D *Ising Model with the Conformal Bootstrap*, *Phys. Rev.* **D 86** (2012) 025022 [arXiv:1203.6064] [INSPIRE].

[45] D. Gaiotto, D. Mazac and M.F. Paulos, *Bootstrapping the* 3d *Ising twist defect*, *JHEP* **03** (2014) 100 [arXiv:1310.5078] [INSPIRE].

[46] 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] [INSPIRE].

[47] C.-M. Chang, M. Fluder, Y.-H. Lin and Y. Wang, *Spheres, Charges, Instantons and Bootstrap: A Five-Dimensional Odyssey*, *JHEP* **03** (2018) 123 [arXiv:1710.08418] [INSPIRE].

[48] Z. Li and N. Su, *3D CFT Archipelago from Single Correlator Bootstrap*, *Phys. Lett.* **B 797** (2019) 134920 [arXiv:1706.06960] [INSPIRE].

[49] C. Hasegawa and Y. Nakayama, *Three ways to solve critical ϕ^4 theory on $4 - \epsilon$ dimensional real projective space: perturbation, bootstrap and Schwinger-Dyson equation*, *Int. J. Mod. Phys.* **A 33** (2018) 1850049 [arXiv:1801.09107] [INSPIRE].

[50] C.N. Gowdigere, J. Santara and Sumedha, *Conformal Bootstrap Signatures of the Tricritical Ising Universality Class*, arXiv:1811.11442 [INSPIRE].

[51] A. Stergiou, *Bootstrapping MN and Tetragonal CFTs in Three Dimensions*, *SciPost Phys.* **7** (2019) 010 [arXiv:1904.00017] [INSPIRE].

[52] D. Poland and D. Simmons-Duffin, *Bounds on 4D Conformal and Superconformal Field Theories*, *JHEP* **05** (2011) 017 [arXiv:1009.2087] [INSPIRE].

[53] C. Beem, L. Rastelli and B.C. van Rees, *The $\mathcal{N} = 4$ Superconformal Bootstrap*, *Phys. Rev. Lett.* **111** (2013) 071601 [arXiv:1304.1803] [INSPIRE].

[54] L.F. Alday and A. Bissi, *The superconformal bootstrap for structure constants*, *JHEP* **09** (2014) 144 [arXiv:1310.3757] [INSPIRE].

[55] L.F. Alday and A. Bissi, *Generalized bootstrap equations for $\mathcal{N} = 4$ SCFT*, *JHEP* **02** (2015) 101 [arXiv:1404.5864] [INSPIRE].

[56] S.M. Chester, J. Lee, S.S. Pufu and R. Yacoby, *The $\mathcal{N} = 8$ superconformal bootstrap in three dimensions*, *JHEP* **09** (2014) 143 [arXiv:1406.4814] [INSPIRE].

[57] C. Beem, M. Lemos, P. Liendo, L. Rastelli and B.C. van Rees, *The $\mathcal{N} = 2$ superconformal bootstrap*, *JHEP* **03** (2016) 183 [arXiv:1412.7541] [INSPIRE].

[58] N. Bobev, S. El-Showk, D. Mazac and M.F. Paulos, *Bootstrapping SCFTs with Four Supercharges*, *JHEP* **08** (2015) 142 [arXiv:1503.02081] [INSPIRE].

[59] C. Beem, M. Lemos, L. Rastelli and B.C. van Rees, *The $(2, 0)$ superconformal bootstrap*, *Phys. Rev.* **D 93** (2016) 025016 [arXiv:1507.05637] [INSPIRE].

[60] D. Poland and A. Stergiou, *Exploring the Minimal 4D $\mathcal{N} = 1$ SCFT*, *JHEP* **12** (2015) 121 [arXiv:1509.06368] [INSPIRE].

[61] M. Lemos and P. Liendo, *Bootstrapping $\mathcal{N} = 2$ chiral correlators*, *JHEP* **01** (2016) 025 [arXiv:1510.03866] [INSPIRE].

[62] Y.-H. Lin, S.-H. Shao, D. Simmons-Duffin, Y. Wang and X. Yin, *$\mathcal{N} = 4$ superconformal bootstrap of the $K3$ CFT*, *JHEP* **05** (2017) 126 [arXiv:1511.04065] [INSPIRE].

[63] Y.-H. Lin, S.-H. Shao, Y. Wang and X. Yin, *$(2, 2)$ superconformal bootstrap in two dimensions*, *JHEP* **05** (2017) 112 [arXiv:1610.05371] [INSPIRE].

[64] J.-B. Bae, D. Gang and J. Lee, *3d $\mathcal{N} = 2$ minimal SCFTs from Wrapped M5-branes*, *JHEP* **08** (2017) 118 [arXiv:1610.09259] [INSPIRE].

[65] M. Lemos, P. Liendo, C. Meneghelli and V. Mitev, *Bootstrapping $\mathcal{N} = 3$ superconformal theories*, *JHEP* **04** (2017) 032 [arXiv:1612.01536] [INSPIRE].

[66] C. Beem, L. Rastelli and B.C. van Rees, *More $\mathcal{N} = 4$ superconformal bootstrap*, *Phys. Rev.* **D 96** (2017) 046014 [arXiv:1612.02363] [INSPIRE].

[67] M. Cornagliotto, M. Lemos and V. Schomerus, *Long Multiplet Bootstrap*, *JHEP* **10** (2017) 119 [arXiv:1702.05101] [INSPIRE].

[68] C.-M. Chang and Y.-H. Lin, *Carving Out the End of the World or (Superconformal Bootstrap in Six Dimensions)*, *JHEP* **08** (2017) 128 [arXiv:1705.05392] [INSPIRE].

[69] M. Cornagliotto, M. Lemos and P. Liendo, *Bootstrapping the (A_1, A_2) Argyres-Douglas theory*, *JHEP* **03** (2018) 033 [arXiv:1711.00016] [INSPIRE].

[70] N.B. Agmon, S.M. Chester and S.S. Pufu, *Solving M-theory with the Conformal Bootstrap*, *JHEP* **06** (2018) 159 [arXiv:1711.07343] [INSPIRE].

[71] M. Baggio, N. Bobev, S.M. Chester, E. Lauria and S.S. Pufu, *Decoding a Three-Dimensional Conformal Manifold*, *JHEP* **02** (2018) 062 [arXiv:1712.02698] [INSPIRE].

[72] P. Liendo, C. Meneghelli and V. Mitev, *Bootstrapping the half-BPS line defect*, *JHEP* **10** (2018) 077 [arXiv:1806.01862] [INSPIRE].

[73] A. Atanasov, A. Hillman and D. Poland, *Bootstrapping the Minimal 3D SCFT*, *JHEP* **11** (2018) 140 [arXiv:1807.05702] [INSPIRE].

[74] C.-M. Chang, M. Fluder, Y.-H. Lin, S.-H. Shao and Y. Wang, *3d $N = 4$ Bootstrap and Mirror Symmetry*, arXiv:1910.03600 [INSPIRE].

[75] R. Rattazzi, S. Rychkov and A. Vichi, *Central Charge Bounds in 4D Conformal Field Theory*, *Phys. Rev.* **D 83** (2011) 046011 [arXiv:1009.2725] [INSPIRE].

[76] 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] [INSPIRE].

[77] P. Liendo, L. Rastelli and B.C. van Rees, *The Bootstrap Program for Boundary CFT_d*, *JHEP* **07** (2013) 113 [arXiv:1210.4258] [INSPIRE].

[78] S. El-Showk and M.F. Paulos, *Bootstrapping Conformal Field Theories with the Extremal Functional Method*, *Phys. Rev. Lett.* **111** (2013) 241601 [arXiv:1211.2810] [INSPIRE].

[79] Y. Nakayama, *Bootstrapping critical Ising model on three-dimensional real projective space*, *Phys. Rev. Lett.* **116** (2016) 141602 [arXiv:1601.06851] [INSPIRE].

[80] A. Castedo Echeverri, B. von Harling and M. Serone, *The Effective Bootstrap*, *JHEP* **09** (2016) 097 [arXiv:1606.02771] [INSPIRE].

[81] A. Cappelli, L. Maffi and S. Okuda, *Critical Ising Model in Varying Dimension by Conformal Bootstrap*, *JHEP* **01** (2019) 161 [arXiv:1811.07751] [INSPIRE].

[82] D. Simmons-Duffin, *A Semidefinite Program Solver for the Conformal Bootstrap*, *JHEP* **06** (2015) 174 [arXiv:1502.02033] [INSPIRE].

[83] W. Landry and D. Simmons-Duffin, *Scaling the semidefinite program solver SDPB*, arXiv:1909.09745 [INSPIRE].

[84] M. Go and Y. Tachikawa, *autoboot: A generator of bootstrap equations with global symmetry*, *JHEP* **06** (2019) 084 [arXiv:1903.10522] [INSPIRE].

[85] J.A. Lipa, J.A. Nissen, D.A. Stricker, D.R. Swanson and T.C.P. Chui, *Specific heat of liquid helium in zero gravity very near the lambda point*, *Phys. Rev.* **B 68** (2003) 174518 [cond-mat/0310163] [INSPIRE].

[86] M. Hasenbusch, *Monte Carlo study of an improved clock model in three dimensions*, *Phys. Rev.* **B 100** (2019) 224517 [arXiv:1910.05916] [INSPIRE].

[87] M. Hasenbusch and E. Vicari, *Anisotropic perturbations in three-dimensional O(N)-symmetric vector models*, *Phys. Rev.* **B 84** (2011) 125136.

[88] D. Tilley and J. Tilley, *Superfluidity and Superconductivity*, Graduate Student Series in Physics, Taylor & Francis (1990).

[89] M.R. Moldover, J.V. Sengers, R.W. Gammon and R.J. Hocken, *Gravity effects in fluids near the gas-liquid critical point*, *Rev. Mod. Phys.* **51** (1979) 79 [INSPIRE].

[90] 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 [INSPIRE].

[91] J.A. Lipa et al., *Specific Heat of Helium Confined to a 57- mum Planar Geometry near the Lambda Point*, *Phys. Rev. Lett.* **84** (2000) 4894 [INSPIRE].

[92] A. Pelissetto and E. Vicari, *Critical phenomena and renormalization group theory*, *Phys. Rept.* **368** (2002) 549 [cond-mat/0012164] [INSPIRE].

[93] E. Burovski, J. Machta, N. Prokof'ev and B. Svistunov, *High-precision measurement of the thermal exponent for the three-dimensionalxyuniversality class*, *Phys. Rev.* **B 74** (2006) .

[94] A.I. Sokolov and M.A. Nikitina, *Critical Exponents of Superfluid Helium and Pseudo-ϵ Expansion*, *Physica* **A 444** (2016) 177 [arXiv:1402.4318] [InSPIRE].

[95] W. Xu, Y. Sun, J.-P. Lv and Y. Deng, *High-precision Monte Carlo study of several models in the three-dimensional* U(1) *universality class*, *Phys. Rev.* **B 100** (2019) 064525 [arXiv:1908.10990] [InSPIRE].

[96] M. Campostrini, M. Hasenbusch, A. Pelissetto and E. Vicari, *The Critical exponents of the superfluid transition in He-4*, *Phys. Rev.* **B 74** (2006) 144506 [cond-mat/0605083] [InSPIRE].

[97] D. Simmons-Duffin, *The Lightcone Bootstrap and the Spectrum of the 3d Ising CFT*, *JHEP* **03** (2017) 086 [arXiv:1612.08471] [InSPIRE].

[98] S. Albayrak, D. Meltzer and D. Poland, *More Analytic Bootstrap: Nonperturbative Effects and Fermions*, *JHEP* **08** (2019) 040 [arXiv:1904.00032] [InSPIRE].

[99] R. Guida and J. Zinn-Justin, *Critical exponents of the N vector model*, *J. Phys.* **A 31** (1998) 8103 [cond-mat/9803240] [InSPIRE].

[100] F. Jasch and H. Kleinert, *Fast-convergent resummation algorithm and critical exponents of ϕ^4 theory in three dimensions*, *J. Math. Phys.* **42** (2001) 52 [cond-mat/9906246] [InSPIRE].

[101] S. Rychkov and Z.M. Tan, *The ϵ-expansion from conformal field theory*, *J. Phys.* **A 48** (2015) 29FT01 [arXiv:1505.00963] [InSPIRE].

[102] P. Calabrese, A. Pelissetto and E. Vicari, *Multicritical phenomena in $O(n_1) \oplus O(n_2)$ symmetric theories*, *Phys. Rev.* **B 67** (2003) 054505 [cond-mat/0209580] [InSPIRE].

[103] M. De Prato, A. Pelissetto and E. Vicari, *Third harmonic exponent in three-dimensional N vector models*, *Phys. Rev.* **B 68** (2003) 092403 [cond-mat/0302145] [InSPIRE].

[104] M. Caselle and M. Hasenbusch, *The Stability of the* O(N) *invariant fixed point in three-dimensions*, *J. Phys.* **A 31** (1998) 4603 [cond-mat/9711080] [InSPIRE].

[105] J.M. Carmona, A. Pelissetto and E. Vicari, *The N component Ginzburg-Landau Hamiltonian with cubic anisotropy: A Six loop study*, *Phys. Rev.* **B 61** (2000) 15136 [cond-mat/9912115] [InSPIRE].

[106] H. Shao, W. Guo and A.W. Sandvik, *Monte Carlo Renormalization Flows in the Space of Relevant and Irrelevant Operators: Application to Three-Dimensional Clock Models*, *Phys. Rev. Lett.* **124** (2020) 080602 [arXiv:1905.13640] [InSPIRE].

[107] O. Nachtmann, *Positivity constraints for anomalous dimensions*, *Nucl. Phys.* **B 63** (1973) 237 [InSPIRE].

[108] Z. Komargodski and A. Zhiboedov, *Convexity and Liberation at Large Spin*, *JHEP* **11** (2013) 140 [arXiv:1212.4103] [InSPIRE].

[109] M.S. Costa, T. Hansen and J. Penedones, *Bounds for OPE coefficients on the Regge trajectory*, *JHEP* **10** (2017) 197 [arXiv:1707.07689] [InSPIRE].

[110] 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] [InSPIRE].

[111] D. Meltzer, *Higher Spin ANEC and the Space of CFTs*, *JHEP* **07** (2019) 001 [arXiv:1811.01913] [InSPIRE].

[112] S. Rychkov *unpublished work*.

[113] J.H. Park and S. Boyd, *General heuristics for nonconvex quadratically constrained quadratic programming*, `arXiv:1703.07870`.

[114] C. Sun and R. Dai, *An iterative rank penalty method for nonconvex quadratically constrained quadratic programs*, *SIAM J. Control Optim.* **57** (2019) 3749.

[115] C.B. Barber, D.P. Dobkin and H. Huhdanpaa, *The quickhull algorithm for convex hulls*, *ACM Trans. Math. Software* **22** (1996) 469.

[116] P. Calabrese and P. Parruccini, *Harmonic crossover exponents in* O(n) *models with the pseudo-ϵ-expansion approach*, *Phys. Rev.* **B 71** (2005) 064416 [`cond-mat/0411027`] [INSPIRE].

[117] E. Katz, S. Sachdev, E.S. Sørensen and W. Witczak-Krempa, *Conformal field theories at nonzero temperature: Operator product expansions, Monte Carlo and holography*, *Phys. Rev.* **B 90** (2014) 245109 [`arXiv:1409.3841`] [INSPIRE].

[118] L. Iliesiu, M. Koloğlu, R. Mahajan, E. Perlmutter and D. Simmons-Duffin, *The Conformal Bootstrap at Finite Temperature*, *JHEP* **10** (2018) 070 [`arXiv:1802.10266`] [INSPIRE].

[119] J. Towns et al., *Xsede: Accelerating scientific discovery*, *Comput. Sci. Eng.* **16** (2014) 62.