The Numerical Conformal Bootstrap

David Simmons-Duffin

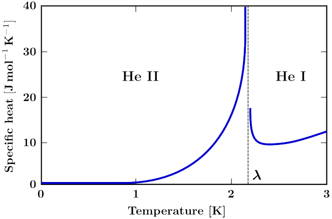
Caltech

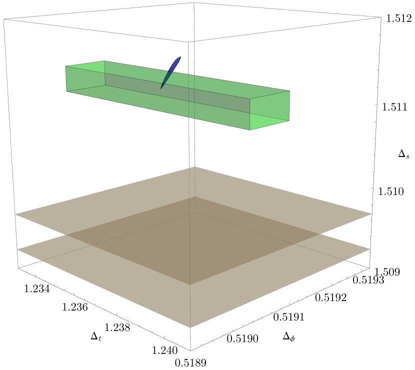
October 7, 2020

The Numerical Conformal Bootstrap

- Conformal Field Theories (CFTs) describe 2nd order phase transitions in condensed matter phys., statistical phys., particle phys., string theory
- **Bootstrap**: constrain and solve CFTs using consistency conditions (symmetry, unitarity, etc.)
- **Numerical bootstrap** [Rattazzi, Rychkov, Tonni, Vichi '08]: formulate inequalities on CFT data and solve them using convex optimization (e.g. semidefinite programming [Poland, DSD, Vichi '11])
  - CFT data = e.g. critical exponents
  - Semidefinite solver becomes an oracle that can answer **Q**: *"is this hypothetical set of CFT data consistent?"*
  - Oracle says *no*: data disallowed
  - Oracle says *yes*: data possibly allowed
  - Query oracle many times to make an exclusion plot

Numerical Bootstrap Example: $O(2)$ Model





- $O(2)$ model: superfluid transition in ^{4}He, many other systems
- Recent bootstrap computation [Chester, Landry, Liu, Poland, DSD, Su, Vichi '19]
  - $\Delta_\phi = 0.519088(22)$, $\Delta_s = 1.51136(22)$, $\Delta_t = 1.23629(11)$
  - $\lambda_{sss}/\lambda_{\phi\phi s} = 1.20932(26)$, $\lambda_{tts}/\lambda_{\phi\phi s} = 1.82228(11)$, $\lambda_{\phi\phi t}/\lambda_{\phi\phi s} = 1.765920(39)$, $\ldots$
- Results support MC, rigorously rule out λ-point experiment

Typical computation

Master

- run search algorithm (Mathematica/Haskell/Python)
- submit $O(100) - O(1000)$ jobs

Job $O(1) - O(15)$ nodes/job, ~ 32 cores/node

- compute conformal blocks (C$++$, embarrassingly $\|$)
- setup semidefinite program (Mathematica/Haskell/Python)
- run semidefinite solver SDPB (C$++$)
  - parallelized with MPI, uses all $O(1)$-$O(500)$ cores
  - dominates computation time, $\sim O(1)$-$O(100)$ hours/run

Example: $O(2)$ model computation used 1M CPU-hours on SDSC's Comet cluster, spread over $O(100)$ jobs.

SDPB

- SDPB is an open-source semidefinite program solver for the conformal bootstrap [DSD '15], [Landry, DSD '19]
- Uses a primal-dual interior point method
- Parallelized with MPI. Scales well up to hundreds of cores.
- Uses arbitrary precision arithmetic. (No GPUs.)
- Lots of embarrassingly parallel linear algebra.
- However, one global Cholesky solve.
  - For efficient scaling, a copy of the global matrix is stored on each core
  - $\Longrightarrow$ memory usage/node grows linearly with cores/node.
  - (Can optionally tradeoff memory vs. efficient scaling.)
  - Memory/core is currently a bottleneck for attacking larger problems. (Comet: 24 cores/node, 128GB RAM/node)

Larger Problems/Ideal Machine

- Can choose to study larger sets of observables (4pt correlation functions).
- More observables $\implies$ more constraints $\implies$ larger semidefinite programs (SDPs) $\implies$ stronger bounds and more CFT data.
- Can easily generate SDPs of physical interest that are larger than what can be solved today. (e.g. SDPs for conformal window of QCD)

Ideal Machine

- Cluster with queue policies allowing many medium-size jobs
- Regular (non-GPU) compute nodes with many cores, and **lots of memory/core**

Community Needs

- Currently ~ 10 groups in the US running nontrivial numerical bootstrap computations.
- Most not yet at the scale of $O(2)$ model, but growing in number, sophistication, and ambition
- Collectively using $\sim$ 10M-20M CPU-hours/year in the US [Very rough estimate based on research output, typical computation size], mostly on locally-supported clusters, but moving to larger machines

Over the next few years, these numbers will grow as we attempt larger problems, and as groups outside high-energy theory (e.g. condensed matter, statistical physics) bring numerical bootstrap methods into their toolboxes. Many of us are training students and writing software to make large-scale bootstrap computations more accessible.