Tensor Networks: Algorithms & Applications

— Raghav G Jha

rjha1@perimeterinstitute.ca

June 10 & 11, 2020

RPI Cyber Training School

[Online via Cisco Webex

INSTRUCTIONS FOR READING THESE NOTES:

- PLACES MARKED AS @ WERE ACCOMPANIED WITH ONLINE CODING
- IF YOU WANT THOSE CODES (PYTHON 3), PLEASE EMAIL ME.
- THESE NOTES ARE GOOD FOR UNDERGRADUATE LEVEL EXCEPT FEW COMMENTS.
- THESE NOTES WERE PREPARED FOR TWO 90 MINUTES LECTURE

TENSOR NETWORK APPROACH TO QUANTUM MANY-BODY SYSTEMS & QFT'S

TN

- Exact diagonalization (Ising model) for small N.
- Basics of Tensor networks
- 2d Classical Ising model & 2d Classical XY model → compare to exact result
- + others depending on time left
  - → Entanglement Entropy
  - → 2d Ising with $h \neq 0$

ADDITIONAL REMARKS

These will be Hands-on lectures and the flow will be determined accordingly.

You will participate and do the coding exercises on your laptop/PC.

Only need Internet & Google Account or your method of coding in Python 3

Tensor Network Approach To Quantum Many-Body Systems & QFT's

Single spin-$\frac{1}{2}$ particle has Hilbert space $\mathcal{H} = \mathbb{C}^2$ of dimension 2. 'N' spins have $\mathcal{H}^{(N)} = 2^N$. Consider a spin chain of $N = 30$. The dimension is $= 2^{30} = (2^{10})^3 \approx 10^9$!

$2^{10} \approx 10^3$

It is impossible to fit this in many modern day computer!

BUT NATURE IS KIND

MAIN MESSAGE: WE DON'T NEED TO WORK WITH ENTIRE HILBERT SPACE SINCE FOR MOST CASES - NATURE LURKS IN SMALL SMALL CORNER OF THE FULL SPACE.

WE'LL SEE THIS IN DETAIL LITTLE LATER.

NOTION OF GAPPED & GAPLESS

$$\Delta E = \underset{|\psi\rangle \in \mathcal{H} \setminus G}{\text{infimum}} \langle \psi | H | \psi \rangle - E_0$$

in the thermodynamic limit.

If $\Delta E = 0 \longrightarrow$ Gapless / Critical

$\Delta E \neq 0 \longrightarrow$ Gapped

Gapless models are captured in field theory language by special class of QFT's known as CFT's. → Conformal field theory..

WE'LL RESTRICT TO GAPPED SYSTEMS FOR NOW.

Q: HOW DO WE IDENTIFY WHICH CORNER OF HILBERT SPACE NATURE PREFERS FOR GAPPED SYSTEMS ?

A: LOOK AT ENTANGLEMENT ENTROPY OF THE QUANTUM STATE.

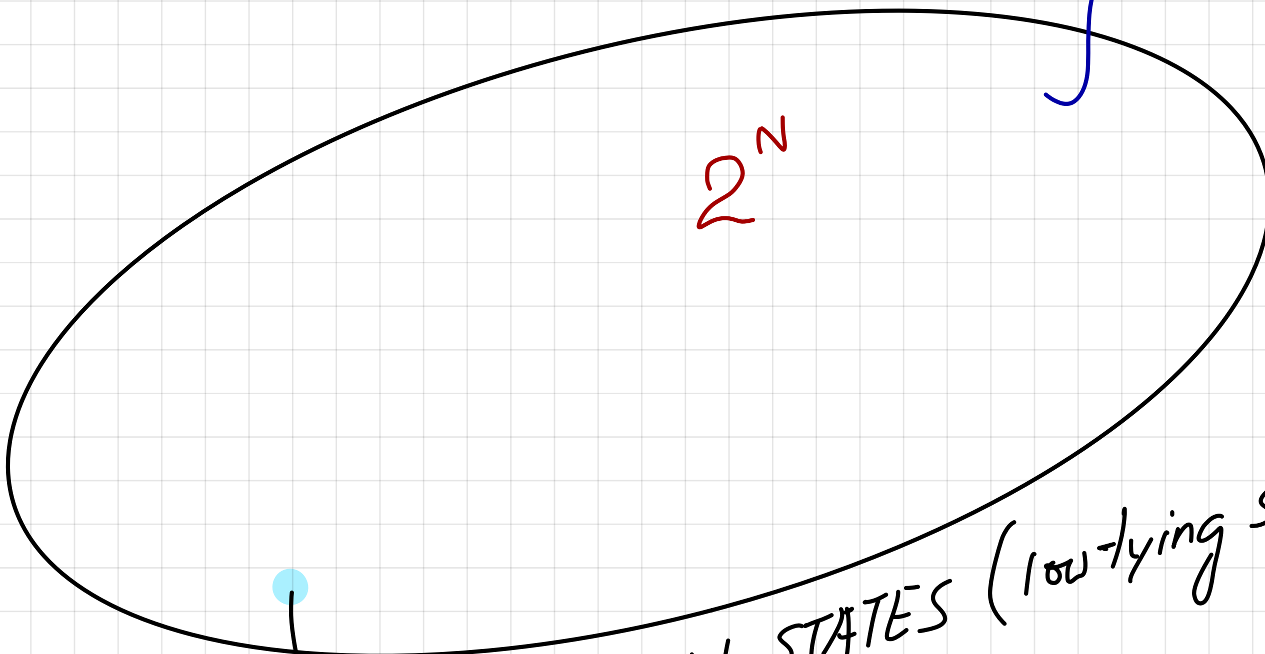
HILBERT SPACE CARTOON

(for local gapped Hamiltonian systems with short-range interactions)

$\mathcal{H}_{complete}$

VOLUME STATES

$S \propto V_{region}$



AREA-LAW STATES (low-lying states)

$S \propto A_{region}$ $\longrightarrow$ Area law of entanglement.

(will discuss later depending on time available).

SMALL EXERCISE

> Exercise 1: Consider the Hamiltonian of three spins ($N = 3$) quantum Ising model given by:
> $$H = \sigma_1^x \otimes \sigma_2^x + \sigma_2^x \otimes \sigma_3^x + \sigma_3^x \otimes \sigma_1^x + h\left(\sigma_1^z + \sigma_2^z + \sigma_3^z\right)$$
> Since $\dim(\mathcal{H}) = 8$, use exact diagonalization and compute ground state energy for various h.

WE WILL USE GOOGLE COLAB (for all coding exercises!)

FOR LARGE NO. OF SPINS SAY 2D – XY MODEL ON $2^{10} \times 2^{10}$ LATTICE, WE CAN'T DO EXACT DIAGONALIZATION & USE TN

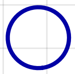
Now, Generalize to N spins

- Check that we reproduce results for $N=3$.

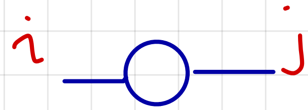
- Try it out for $N < 10$ spins which has $\mathrm{Dim}(\mathcal{H}) < 1024$.

Tensor Notation

A

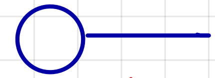


Scalar

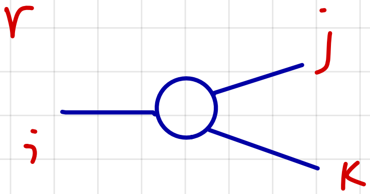


A_{ij}

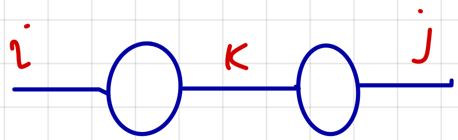
A_i



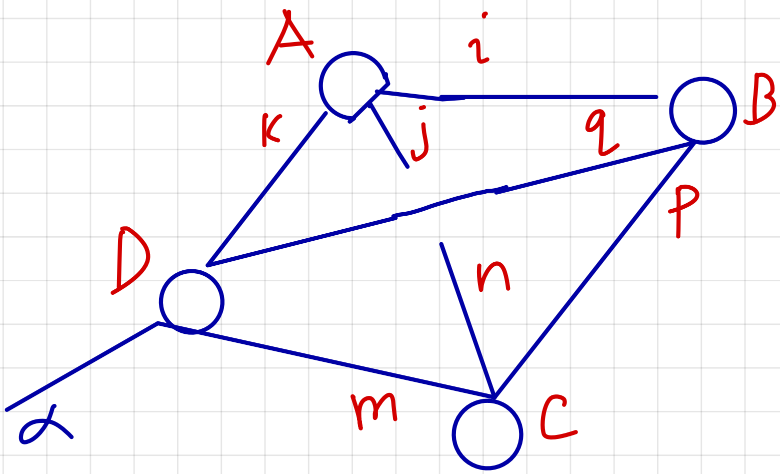
Vector



A_{ijk}



$A_{ik} B_{kj}$



ordering

$A_{ijk} B_{pqi} C_{pmn} D_{kqm\alpha}$

(starting at 12 o'clock)

IMPLEMENTATION IN PYTHON

```
import numpy as np

np.einsum('ij,jk->ik', A, B)
```

Einstein

Summation

$\hookrightarrow A_{ij} B_{jk} = C_{ik}$

Exercise 2: Calculate the trace of product of four random 3×3 matrices using einsum and check that the result agrees with that obtained from np.trace and np.dot. You can construct random matrices using: `A = np.random.rand(3,3)`

@

FASTER ALTERNATIVE USED BY RESEARCH GROUPS

$\longrightarrow$ NCON

arxiv 1402.0939

DOWNLOAD FROM :

www.github.com/rgjha/TensorCodes/blob/master/ncon.py

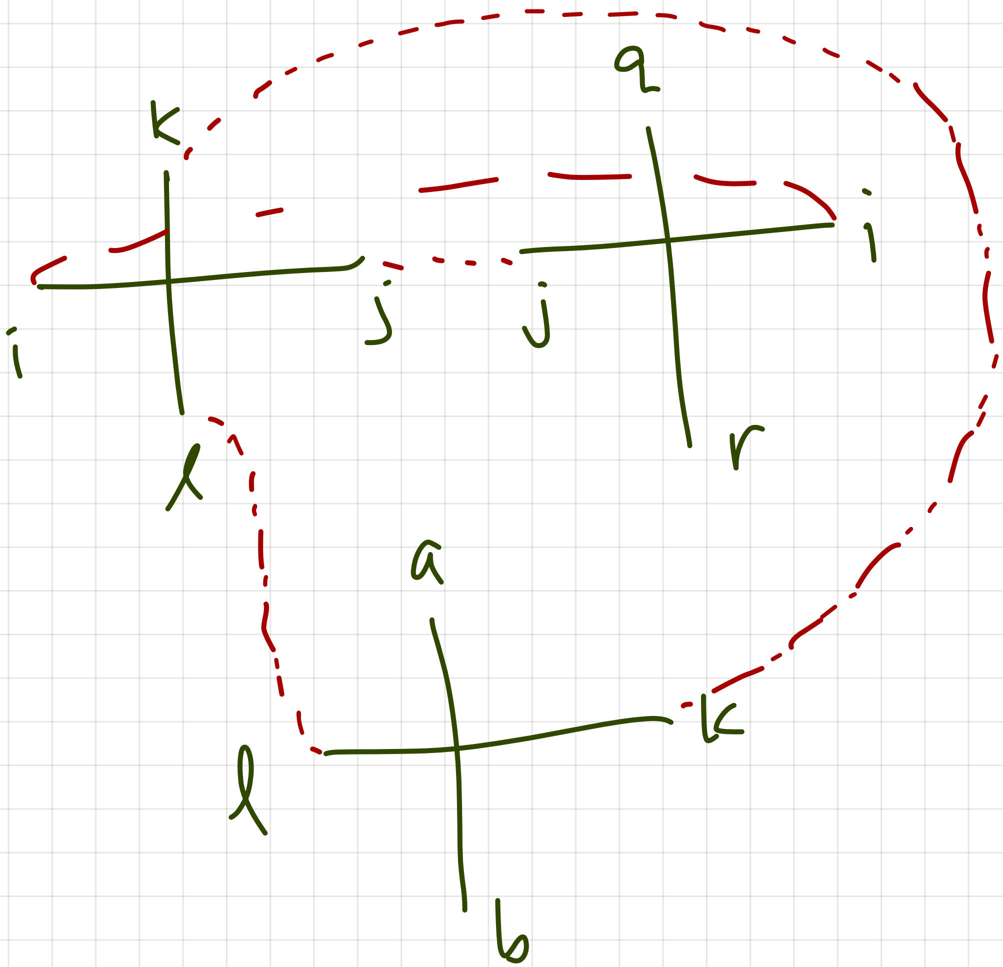
or

FROM DROPBOX FOLDER . .

For ex: $C_{ip} = A_{ijk} B_{pjk}$ is implemented as:

$$C = \text{ncon}((A, B), ([-1, 2, 1], [-2, 2, 1]))$$

Exercise 3: Compute the rank-four tensor A_{rqba} which is equal to $B_{ijkl}C_{jiqr}D_{lkab}$ using NCON where all indices run from $1 \cdots 3$. Draw a tensor diagram of this contraction. You can choose the tensors to be random like before.

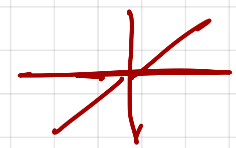


$\longrightarrow A_{rqba}$

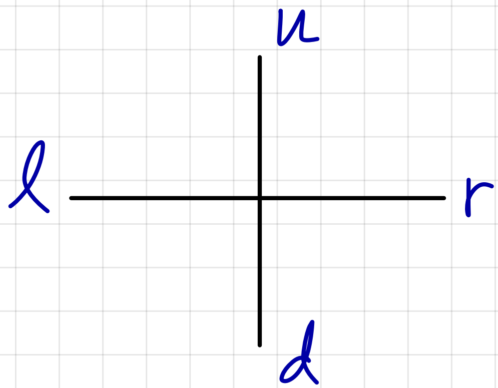
DASHED LINES denotes the contraction.

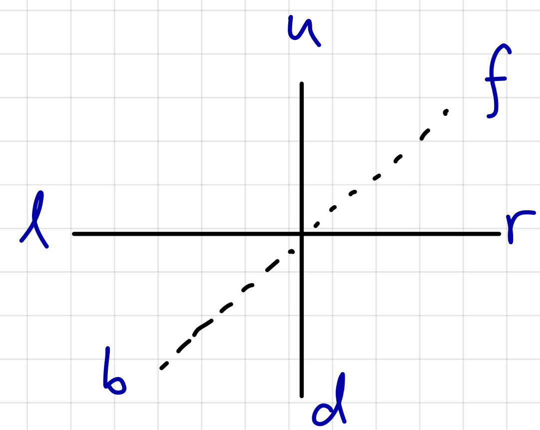
FUNDAMENTAL TENSOR

In 'd' dimensions, the fundamental tensor has total of '2d' legs or indices.



We'll exclusively focus in these lectures on 2d, so it is represented as



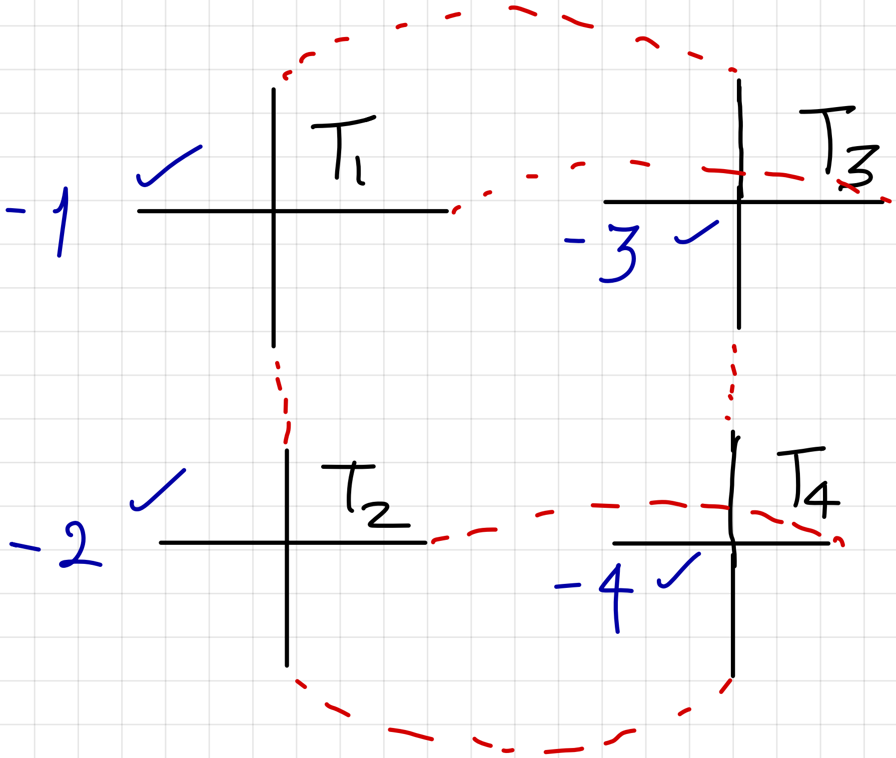


(CLOCKWISE)

T_{lrud} or T_{lurd} $\qquad$ T_{lrudfb}

We'll follow this pattern of indices.

APPLYING NCON TO A SLIGHTLY COMPLICATED NETWORK



$$(T_1)_{ijkl} \, (T_3)_{pjkq}$$

EXERCISE 4: Execute this using NCON in single line command.

Tensor Network Coarse Graining Algorithms (Two Dimensions)

Since the seminal work of Levin & Nave (TRG approach to 2d classical lattice models) dozens of algorithms have been introduced to do efficient coarse-graining. We'll use what is called HOTRG (Higher-Order TRG) based on 1201.1144 (cond-mat)

Others:

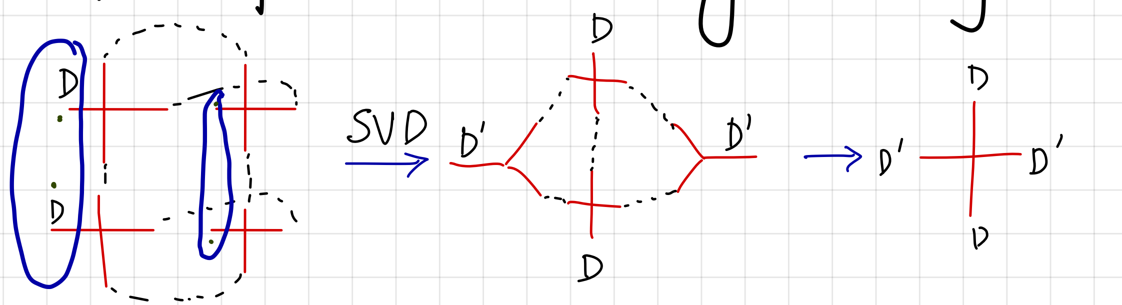
MPS (Matrix Product States), PEPS

TNR (Tensor Network Renormalization)

TEFR, Gilt-TNR, & many others..

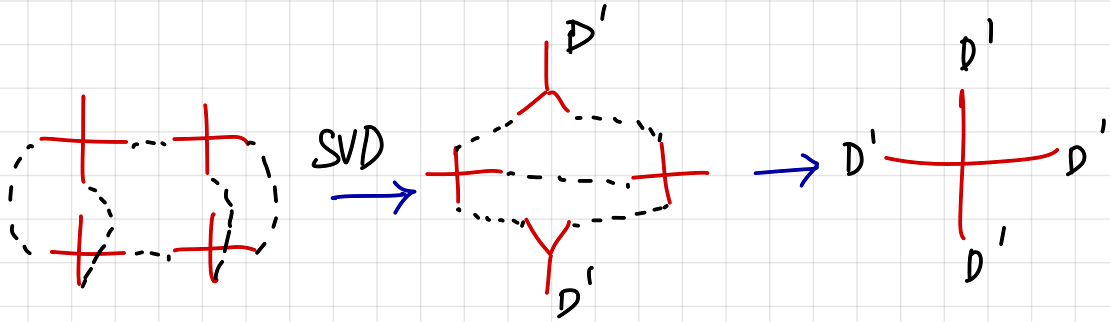
Now consider $T_1 = T_2 = T_3 = T_4 = T_f$

One step of coarse graining :-



along "x"

then,



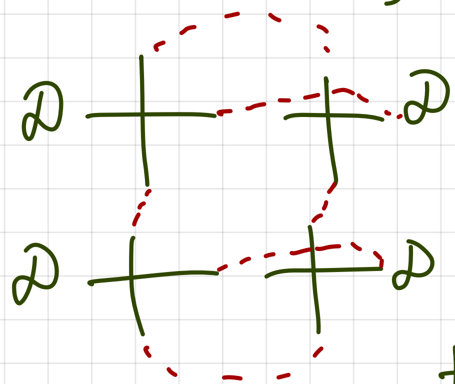
along "y"

Doing this 'N' times starting from the fundamental tensor will produce a final lattice volume 4^N !

END OF LECTURE 1

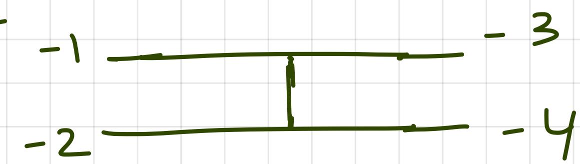
STEP 1: Take $T_{\text{initial}} = T_{(n)}$

STEP 2: Combine four of them like



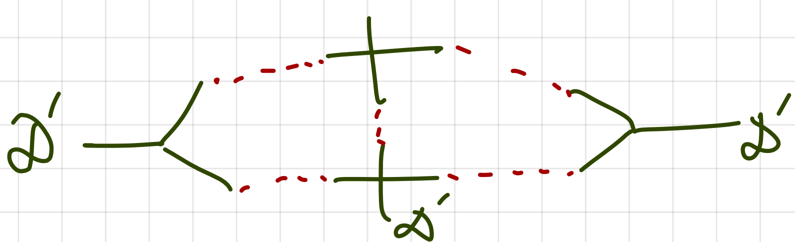
or $\frac{\pi}{2}$ rotation of this

to get

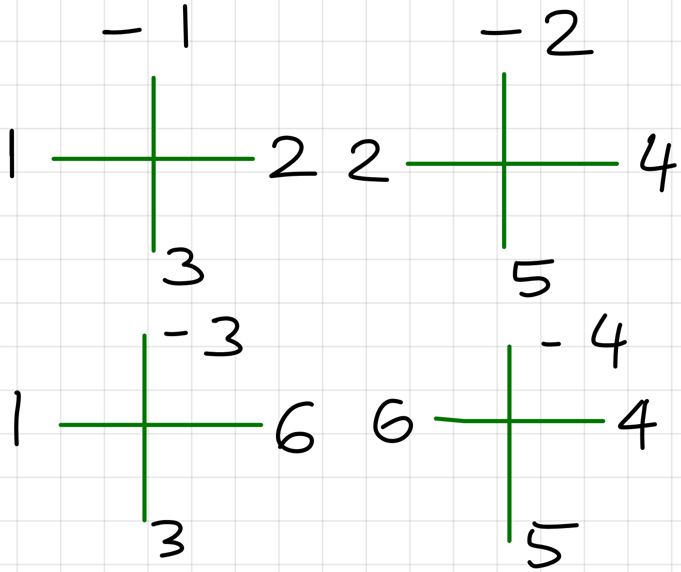


STEP 3: Combine $\underline{-1 \; \& \; -2}$ and $\underline{-3 \; \& \; -4}$ to construct a matrix. $(-1 \otimes -2) \text{---}\bullet\text{---} (-3 \otimes -4)$ and take SVD of this matrix.

STEP 4: Take two T_{initial} tensors and do



to get $T_{(n+1)}$



→ HOMEWORK.

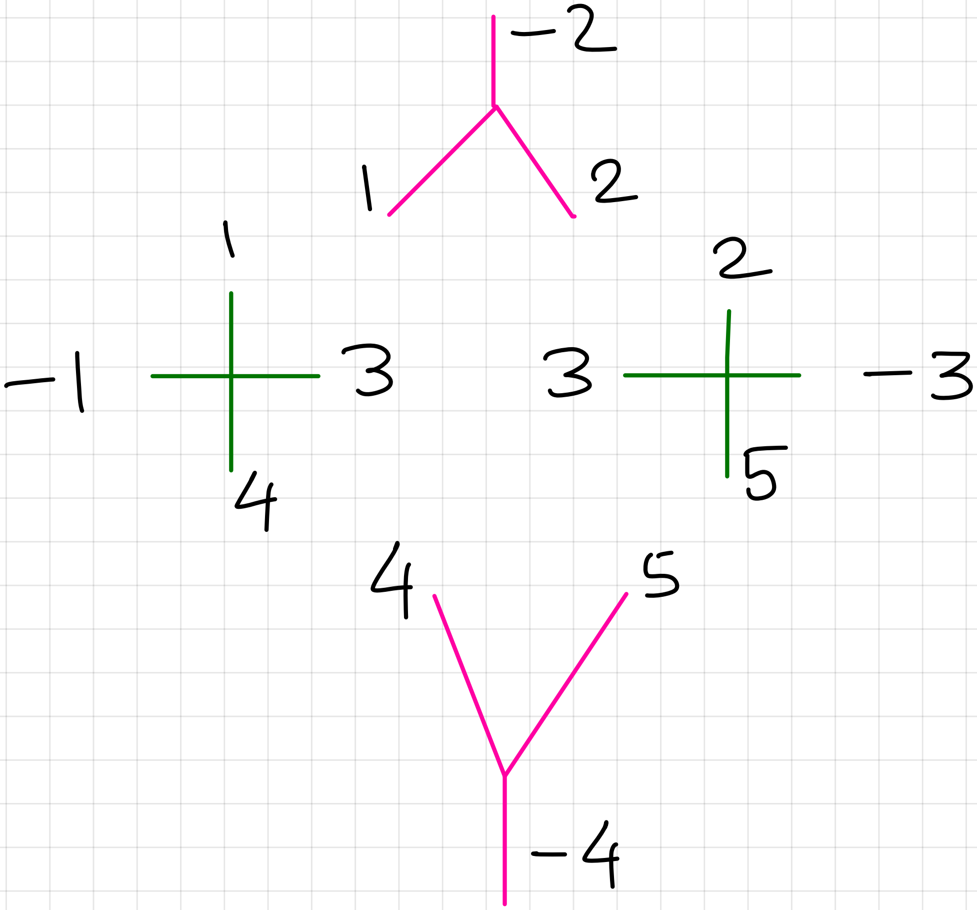
EXERCISE

←

$\pi/2$
rotation

?

(See last page)



←

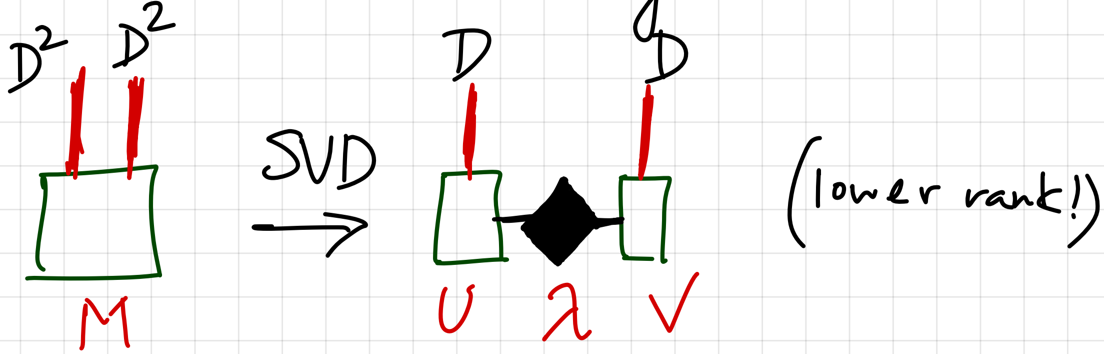
?

Few Comments:

SVD is the most expensive part of the computation. Going to large values of D' depends on available memory.

After every coarse-graining step, we normalize the tensor by its norm or biggest element. (see what happens if you don't)

There is error introduced at each step due to truncation over the singular values in SVD.



2d Classical Ising Model

(without proof for now!)

It can be shown that the fundamental tensor (T) for this case can be written as:

$$T_{ijkl} = W_{pi} W_{pj} W_{p\cancel{j}k} W_{pl}$$

where,

$$W = \begin{bmatrix} \sqrt{\cosh \beta} & \sqrt{\sinh \beta} \\ \sqrt{\cosh \beta} & -\sqrt{\sinh \beta} \end{bmatrix}$$

$$Z = \mathrm{tr}\left(\mathbb{T}^{N_\tau}\right)$$

Transfer matrix.

and, $Z = \mathrm{tTr} \prod_n \left(T_{ijkl}\right)_n$ $\quad n \rightarrow$ sites of lattice

ONSAGER'S SOLUTION (Square lattice)

for $k_B = J = 1$, we have

$$\beta_c \simeq \frac{1}{2.269185} \cong 0.4407$$

Exact free energy is known from Onsager [1944] solution."

Compute "f"

We'll reproduce β_c using tensor networks

WRITE A FUNCTION TO DO ONE-STEP OF COARSE-GRAINING

f : input $\rightarrow T$
output $\rightarrow T'$ and norm

ANOTHER FUNCTION FOR CONSTRUCTING $T_{initial}$ taking 'β' as INPUT.

Free Energy density vs. β plot

Since we normalize the tensor at each step, we cannot compute

$$f = -T \ln Z_{\text{end of coarse graining}}$$

(we need to use "norm" at each step).

Additional details: arXiv 2004.06314 Appendix.

This is already computed in the code

OTHER OBSERVABLES

$$E = -\frac{\partial \ln Z}{\partial \beta} \quad \text{———} \quad (1)$$

$$S = \text{Sp. Heat} \cong \frac{\partial^2 \ln Z}{\partial \beta^2} \quad \text{———} \quad (2)$$

PLAN: Compute 'f' for range of β and then do numerical integration one by one to get (1) & (2)

USE: numpy's gradient function i.e

$E =$ np.grad(f, db) ; where $db = \beta_{i+1} - \beta_i$

Plot & Compare

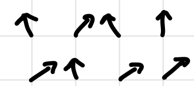
Plot f, E, S vs. β and determine β_c from the peak of S vs. β plot. Compare it to exact value!

Refer arxiv 1903.09650 and see Figure (3). if interested.

2d CLASSICAL X-Y MODEL

arXiv 2004.06314

Consider a square lattice as this page with a spin on each lattice site free to rotate in the X-Y plane. The next-neighbor Hamiltonian is given by:



$$\mathcal{H} = -J\cos(\theta_i - \theta_j) \longrightarrow O(2) \text{ symmetry}$$

We can also consider applying external magnetic field 'h' such as

breaks O(2)

$$\mathcal{H} = -J\cos(\theta_i - \theta_j) + h\cos\theta_i$$

Why Tensor Networks ??

Using efficient coarse-graining algorithms we can do Physics directly in the thermodynamic limit Say $2^{50} \times 2^{50}$ lattice on simple computers which even parallelized Monte Carlo methods usually are unable to achieve !!

Skipping lot of details, we can write down the initial tensor (T) for this model and compute free energy, magnetization etc.

✓ ✗

Only sketch not compute.

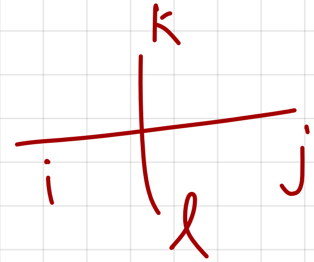
See paper or GitHub for details.

Fundamental Tensor

Skipping details.

$$T_{ijkl} = \sqrt{I_i(\beta)\, I_j(\beta)\, I_k(\beta)\, I_l(\beta)}\;\; I_{i+k-j-l}(\beta h)$$

If $h=0$, $I_{i+k-j-l}(0)$ is only non-zero when $i+k = j+l$. It becomes



$$T_{ijkl} = \sqrt{I_i(\beta)\, I_j(\beta)\, I_k(\beta)\, I_l(\beta)}\;\; \delta^{j+l}_{i+k}$$

i, j, k, l $\cdots$ $-\infty$ $\cdots$ ∞

truncate to

$-n$ $\cdots$ n

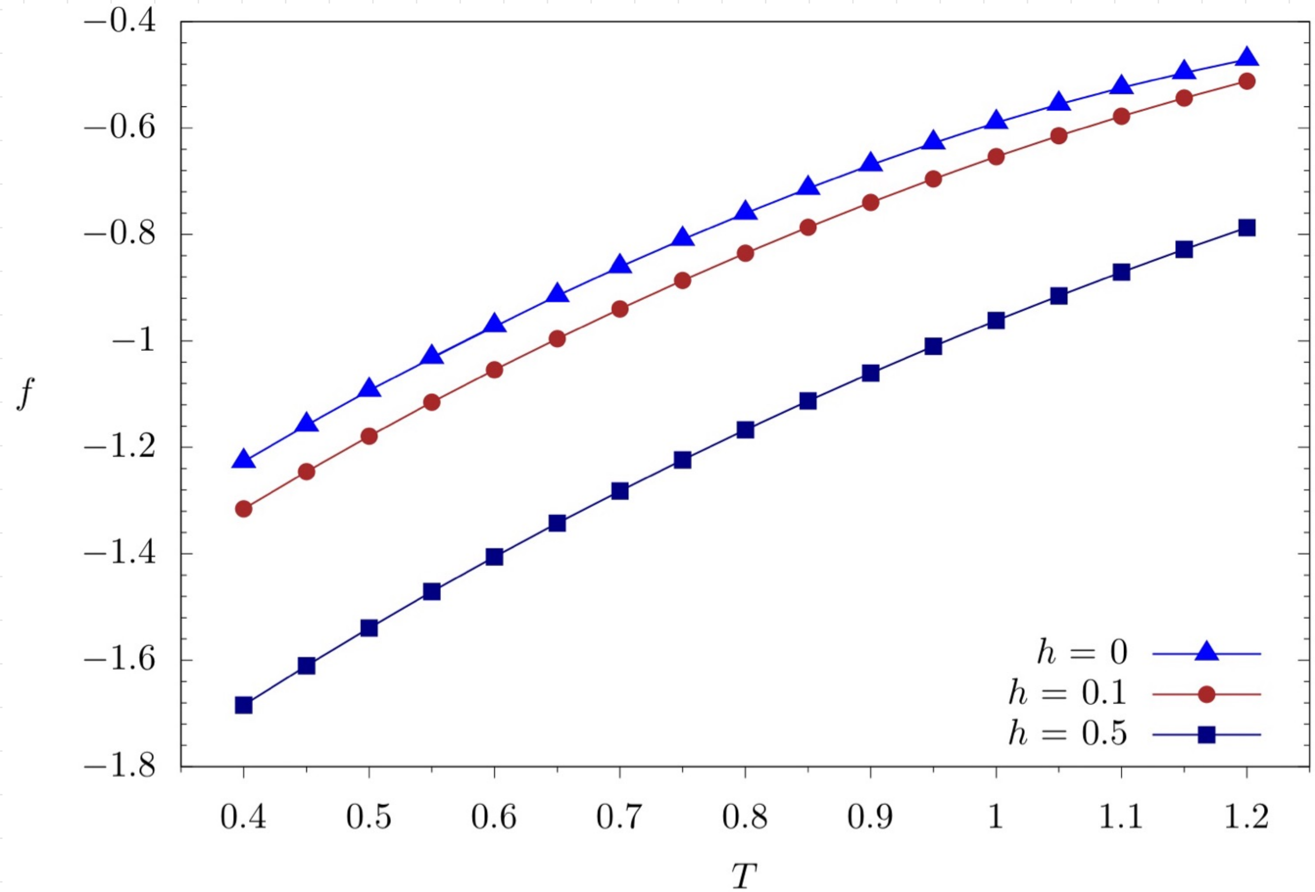
Say $n = \underline{\underline{15, 20}}$..

For BKT deter. we need $n = 26+$

27

WRITE TENSOR NETWORK FOR 2d XY

& COMPUTE $f, \frac{\partial f}{\partial \beta}, \frac{\partial^2 f}{\partial \beta^2}$



@

DETERMINING THE T_{BKT}

Compute magnetization by introducing external field 'h' and compute susceptibility . .

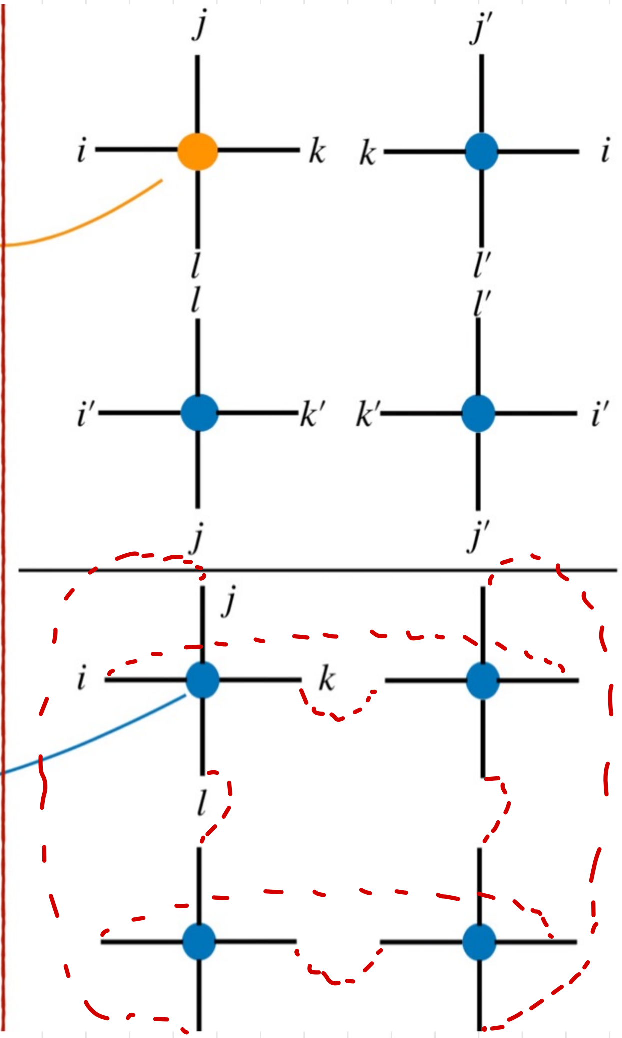
Take the $h \to 0$ limit and read off T_{BKT}.

But, we have only computed things related to "Z" up to now. How do we compute an observable say $\langle O \rangle$?

MAGNETIZATION

$$M = -\frac{\partial F}{\partial h} = \frac{1}{\beta}\frac{\partial \ln Z}{\partial h}$$

$$\tilde{T}_{ikjl} = \sqrt{I_i(\beta)I_k(\beta)I_j(\beta)I_l(\beta)}\left(\frac{I_{i+j-k-l-1}(\beta h) + I_{i+j-k-l+1}(\beta h)}{2}\right)$$



$= M$

$$T_{ikjl} = \sqrt{I_i(\beta)I_k(\beta)I_j(\beta)I_l(\beta)}\, I_{i+j-k-l}(\beta h)$$

$Z \Leftarrow$

WRITE NCON TO COMPUTE 'M'

Something like:

$$M = \frac{\mathrm{ncon}((\tilde{T}, T, T, T), [\,\cdots\cdots\,])}{\mathrm{ncon}((T, T, T, T), [\,\cdots\cdots\,])}$$

Same pattern.

Instead of inserting this impure tensor (orange) we could have computed "M" just by taking numerical derivative of "f" vs. "h", but it is both cumbersome & prone to numerical errors..

ENTANGLEMENT ENTROPY (EE)

Consider a state describing two-subsystems A & B

i.e $|\psi_{AB}\rangle = |\phi_A\rangle|\phi_B\rangle$. If this state is separable,

then the reduced density matrix $\rho_A = Tr_B |\psi_{AB}\rangle\langle\psi_{AB}|$

$$= \underbrace{|\phi_A\rangle\langle\phi_A|}_{\text{pure state.}}$$

And entropy is zero! $(S_A = 0)$

If this is at $T=0$, then $S_A = S_B$ and hence $S_B = 0$.

Non-zero entropy signals $\rightsquigarrow$ ENTANGLEMENT!

Bipartite or Von Neumann entropy

$\rightsquigarrow$ with an examp

Von Neumann Entropy (1927)

$$S = -\mathrm{Tr}\, \rho \log_2(\rho)$$

$$= -\sum_i \rho_i \log_2 \rho_i$$

In a random ensemble, each state is equally probable. So if $\dim(\mathcal{H}) = d$, then all 'ρ_i' are same i.e $\frac{1}{d}$

$$S = \underbrace{-\frac{1}{d} \log \frac{1}{d} + \dots \dots}_{d \text{ times}}$$

$$= \log_2 d = \log_2 2^p = p.$$

MAXIMALLY ENTANGLED.. SINCE "S" is saturated. i.e $S_{max} = p$.

(Page 15 of Preskill's Ph 229 notes Ch.5)

Thank You

Approx time with
online coding
$\simeq$ 3 hrs.