

INTRODUCTION TO METHODS FOR SOLVING MATRIX MODELS

RAGHAV G. JHA^a

Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada

raghav.govind.jha@gmail.com

ABSTRACT: Matrix models play an important role in Physics ranging from nuclear physics where they were first introduced to cosmology, study of random surfaces, conformal field theories, integrable systems, knot theory, two-dimensional quantum gravity, and non-perturbative descriptions of M-theory. In these notes, we consider a wide range of matrix models and study them using analytical and numerical methods. We provide about dozens of exercise and codes in PYTHON and MATHEMATICA for convenience of the reader.

Contents

1	Introduction	2
2	Large N limit of matrix valued fields	5
2.1	QCD and large N	5
2.2	β function, counting diagrams, and factorization	5
3	Analytical solutions	8
3.1	Hermitian & Unitary matrix models	8
3.2	One-matrix model: Cubic and Quartic potentials	10
3.3	Kontsevich Model	10
3.4	Chern-Simons and ABJM matrix models	10
3.5	Ising Model on random planar graph: Two-matrix model	10
3.6	General p -matrix open chain	12
3.7	Open p -matrix chain model	16

4	Numerical solutions	16
4.1	Numerical bootstrap	17
4.2	Monte Carlo method	18
4.2.1	Leapfrog integrator and Metropolis step	19
4.2.2	Random number generator	20
4.2.3	Autocorrelation and errors	21
4.3	One-matrix model: Confirming exact results	22
4.4	Matrix model closed chain solution	22
4.5	Three-matrix Yang Mills matrix model	22
4.6	Hoppe's matrix model	23
4.7	Generalized Hoppe's model	23
4.8	Extension of Hoppe's model to D matrices	23
4.9	Dimer Model: Yang-Lee singularity	24
4.10	Matrix chain: Ends closed	24
5	Matrix models for string theory and holography	25
5.1	$c = 1$ MQM	25
5.2	BFSS matrix model	27
5.2.1	Gauge singlet sector	27
5.2.2	Including non-singlet sector	28
5.3	IKKT matrix model	28
5.4	BMN matrix model	30
5.4.1	Full sector	31
5.4.2	Bosonic sector	32
6	Summary	33
A	Character Expansion	34
B	Orthogonal polynomials	34
C	Schwinger-Dyson/loop equations	35
D	Mathematica code for solution of one-matrix model	36
E	Python code for solution of one-matrix model.	37
F	Bosonic sector of IKKT model: Python code	41

SECTION 1

Introduction

One expects that to a large extent, different phenomenon occurring in the nature can be set up as a problem that may or may not be solvable. From very early days, matrices have played an important role in Physics ranging from development in the quantum mechanics to understanding the large scale structures of the universe. On the other hand, several physical systems are known to sometimes be determined to a great extent by normally distributed elements (Gaussian distribution). The subject of random matrix theory is the study of matrices whose entries are random variables chosen from a well-defined distribution. One of the remarkable things that was first noticed by Wishart around 1928 is that one can consider a family of probability distributions which is defined over symmetric, nonnegative-definite random matrices sometimes also known as matrix-valued random variables and this is now known by ‘Wishart ensembles’. These are sometimes also ‘Wishart-Laguerre’ because the spectral properties of this distribution involve use of Laguerre polynomials.

The application of random matrices to physical problems was not until 1950s when it took the genius of Wigner to first apply the random matrix theory in understanding the energy spectrum in nuclei of heavy elements. It was experimentally shown that unlike the case when the energy levels are assumed to be uncorrelated random numbers and the variable s would be governed by the familiar Poisson distribution i.e., $P(s) = e^{-s}$, there was more to this story and the distribution was nothing like Poisson. He realized (what is now known by the name ‘Wigner’s surmise’¹) that it could be described by a distribution given by $P(s) = \pi s/2 e^{-\pi s^2/4}$. The linear growth of $P(s)$ for small s is due to quantum mechanical level repulsion which was first considered by von Neumann and Wigner in 1929. This surmise and the paper Wigner had written in 1951 [1] introduced random matrix theory to nuclear physics and then in later years to almost all of Physics.

This program was continued in the 1960s, when in their exploration of random matrices, Dyson and Mehta studied and classified three types (sometimes also called ‘the threefold way’) of matrix ensembles with different correlations. The first was ‘Gaussian orthogonal ensemble’ which was used to describe systems with time reversal invariance and integer spin

¹Why is this called a ‘surmise’? As is noted in the literature, the story goes like this: At some conference on Neutron Physics at the Oak Ridge National Laboratory in 1956, someone in the audience asked a question about the possible shape of the distribution of the spacings of energy levels in a heavy nucleus. Wigner who was in the audience, walked up to the blackboard and guessed the answer given above.

with weakest level repulsion between neighboring levels and had $\beta = 1$. The second was the Gaussian unitary ensemble with no time reversal invariance with intermediate level repulsion and $\beta = 2$. The third was the Gaussian symplectic ensemble for time reversal invariance for half integer spin with $\beta = 4$. These are now known as GOE, GUE, and GSE respectively². The general $P(s)$ is given by, $c_\beta s^\beta e^{-a_\beta s^2}$ where $\beta \in (1, 2, 4)$ depending on the symmetry in question. We mention c_β and a_β in Table (1). We show the three distributions using MATHEMATICA in Fig. 1.

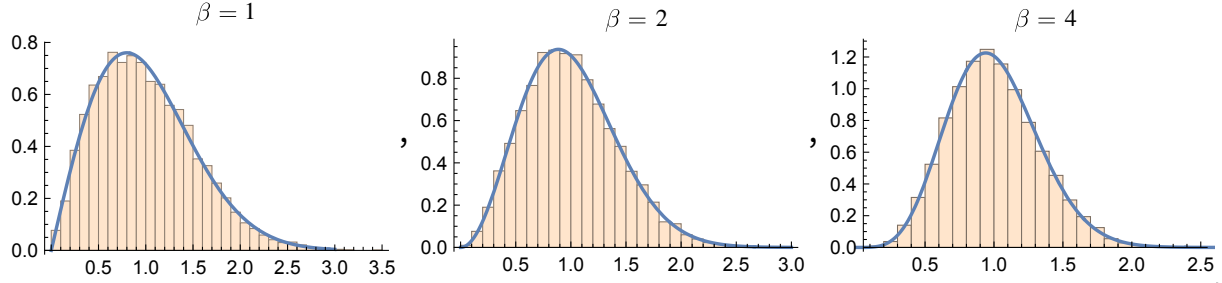


Figure 1. The distribution of three ensembles mentioned in the text.

β	c_β	a_β
1	$\pi s/2$	$\pi/4$
2	$32s^2/\pi^2$	$4/\pi$
4	$2^{18}s^4/3^6\pi^3$	$64/9\pi$

Table 1. We mention the values for c_β and a_β for different ensembles.

It is appropriate here to quote a statement by Mark Kac³ - “That we are led here to the normal law (distribution), usually associated with random phenomena, is perhaps an indication that the deterministic and probabilistic points of view are not as irreconcilable as they may appear at first sight”. It was later concluded that much to Wigner’s own surprise, his guess was fairly accurate as shown and improved by Mehta [2] and Gaudin [3]. In fact, the field of random matrix theory is intimately related to number theory (especially the pair correlation of the zeros of the Riemann-zeta function) and this was observed by Montgomery and Dyson at a tea break at the Institute for Advanced Study in the 1970s. Some still believe that the proof of Riemann hypothesis lies in the deepest secrets of random matrix theory.

²For example, GUE represents a statistical distribution over complex Hermitian matrices have probability densities proportional to $\exp(-\text{Tr}(A^2/2\sigma^2))$ and where matrix elements i.e., a_{ij} are an independent collection of complex variates whose real and imaginary parts are from normal distribution with zero mean and unit variance. In MATHEMATICA, we can use: `GaussianUnitaryMatrixDistribution[σ ,N]` to get a $N \times N$ such matrix

³Kac was a Polish American mathematician. His main interest was probability theory. He is also known apart from other things for his thought provoking question - “Can one hear the shape of a drum?”

This belief is also the theme of the idea proposed independently by Hilbert and Pólya who suggested that the zeros of the zeta function might be the eigenvalues of some unknown Hermitian ‘operator’. It is another thing that no one yet knows such an operator! We refer the reader to the excellent books [4, 5] for thorough introduction.

Another major development with the matrix degrees of freedom and matrix models came with the work of ‘t Hooft in 1974. By then, it was becoming clear that the correct theory of strong interactions is QCD where there are matrix degrees of freedom which is based on $SU(3)$ gauge group. ‘t Hooft proposed to consider a general $SU(N)$ symmetry since he showed that in such limit only planar diagrams survive and calculations becomes more tractable. In fact, this large N limit was later understood to be closely related to some description of string theory. This work brought together the idea of random matrix models in the asymptotic limit (large matrices) to the mainstream Physics and was fruitful till date. This also enabled us to study quantum gravity from a field theoretic point of view via the famous AdS/CFT conjecture.

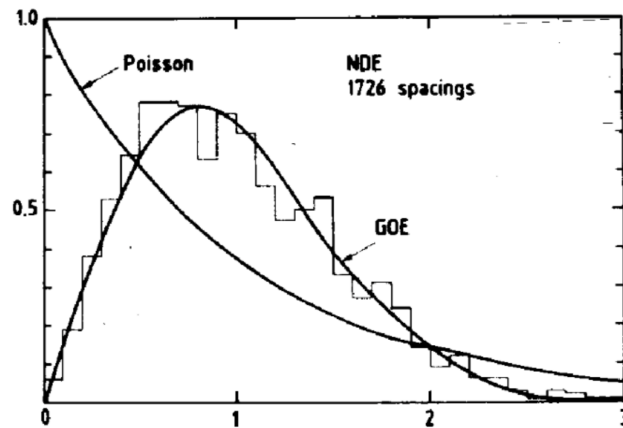


Figure 2. The nearest neighbour spacing distribution (i.e., $P(s)$) for nuclear data. The GOE (Gaussian Orthogonal Ensemble) and Poisson is given by solid curves. This figure is taken from - “Fluctuation Properties of Nuclear Energy Levels and Widths : Comparison of Theory with Experiment” by O. Bohigas, R. U. Haq, and A. Pandey

In fact, it was improved further compared to what is shown in Fig. 2 in [6]

The plan of these notes is as follows. In There are some excellent reviews about random matrix theory, matrix integrals, and their formal aspects which are beyond the scope of this article. We refer the reader to two excellent reviews written in two different centuries [7, 8]

Large N limit of matrix valued fields

One of the most important ideas which has led to numerous other interesting works is the idea of large N limit of field theories. For an extended review of many seminal papers written on this since the 't Hooft's 1974 original article [9], we refer the reader to [10].

2.1 QCD and large N

As we have already pointed out, QCD has a $SU(3)$ gauge symmetry. In order to understand QCD in form of some expansion, 't Hooft introduced the idea of the large N limit of gauge theories by considering a $SU(N)$ gauge group and studying observables in $1/N$ expansion. We often associate more variables with greater complexity, but, this is not always true. There are many systems which simplify in the large N limit. This occurs because the fields are related by a certain symmetry which ensures that the collective behaviour of the fields becomes more constraining as their number increases. The resulting coupled degrees of freedom typically look different from the Lagrangian of the initial theory. It was soon realized that the large N limit of gauge theories is related to string theory as in the large N limit one sums over surfaces of different genus and in string theory one sums over different world-sheet topologies. There has been numerous developments following this work which include factorization equations, large N phase transitions, Eguchi-Kawai (EK) volume reductions, holography, and AdS/CFT correspondence. In fact, EK is a way where there is natural connection of matrix integrals to QCD. At first glance in QCD, one might think that the expansion parameter is g_{YM} , but this is not true in the light of the renormalization group equations. QCD has no obvious free parameter and this creates difficulty for perturbation theory. It is because of this problem that 't Hooft suggested taking N , the number of colors as a parameter. The idea of large N limit is not restricted just to matrix valued fields but exists also for vector models. Pure QCD (i.e., without fermions) is an example of a matrix model because the gauge fields are $N \times N$ matrices with N^2 components. The vector models, on the other hand, only have N components. We will not discuss the large N limit of vector models in this review because we want to only focus on matrices.

2.2 β function, counting diagrams, and factorization

One way to think of the large N limit of QCD in four dimensions is to look at the QCD β -function. One can then check whether this limit captures the defining property of QCD - a negative β function. The perturbative two-loop β -function given by,

$$\mu \frac{dg}{d\mu} = -\frac{1}{16\pi^2} \left(\frac{11N - 2N_f}{3} \right) g^3 - \frac{1}{256\pi^4} \left(\frac{34N^3 - 13N^2N_f + 3N_f}{3N} \right) g^5 + \mathcal{O}(g^7) \quad (2.1)$$

This has clearly no sensible large N limit. But we can make an interesting observation. The left hand side of (2.1) goes as $\sim g$, whereas the leading term on right for small g goes as $\sim g^3 N$. This convinced 't Hooft that the correct limit to consider is the limit where $N \rightarrow \infty$ and $g^2 \rightarrow 0$ while $\lambda = g^2 N$ remains fixed. We often refer to λ as the 't Hooft coupling. In this limit, we get,

$$\mu \frac{d\lambda}{d\mu} = -\frac{11}{24\pi^2} \lambda^2 - \frac{17}{192\pi^4} \lambda^3 + \mathcal{O}(\lambda^4) \quad (2.2)$$

As can be readily seen, perturbation theory predicts that the 't Hooft limit of QCD is an asymptotically free theory which is a good first step. It is also natural to assume that Λ_{QCD} , the scale parameter of strong interactions is held fixed as $N \rightarrow \infty$. One important feature of (2.2) is that it is independent of number of flavors, N_f . The number of quark degrees of freedom is $\mathcal{O}(N)$ in the 't Hooft limit, and hence sub-leading with respect to the number of gluon degrees of freedom, which is $\mathcal{O}(N^2)$.

In order to illustrate the idea of large the N limit, consider the following Lagrangian with canonical normalization:

$$L = -\frac{1}{2} \text{Tr} \left[(\partial\Phi)^2 + g\Phi^3 + g^2\Phi^4 \right] \quad (2.3)$$

We can rescale $\tilde{\Phi} = g\Phi$ to get,

$$L = \frac{1}{g^2} \text{Tr} \left[-\frac{1}{2} (\partial\tilde{\Phi})^2 + \tilde{\Phi}^3 + \tilde{\Phi}^4 \right] \quad (2.4)$$

As we take the planar limit, $\lambda = g_{\text{YM}}^2 N$ remains fixed where λ is the 't Hooft coupling. The big field Φ which we think of as potentially including scalars ϕ , gauge fields A_μ which are $N \times N$ matrices. So, the propagator $\langle \Phi\Phi \rangle$ goes as g^2 i.e., λ/N . In a generic matrix theory there will be n -point vertices such as three point and four-point interaction vertices. Both types of vertices come with the same factor of N/λ . This is the advantage of using this particular form of the Lagrangian since all vertices have same contribution. We can write the gluon propagator as,

$$\tilde{\Phi}_{\mu b}^a(x) \tilde{\Phi}_{\nu d}^c(y) = \left(\delta_d^a \delta_b^c - \frac{1}{N} \delta_b^a \delta_d^c \right) \mathcal{D}_{\mu\nu}(x-y) = \left(\delta_d^a \delta_b^c - \frac{1}{N} \delta_b^a \delta_d^c \right) \frac{g^2}{4\pi^2(x-y)^2}, \quad (2.5)$$

In some sense, the gauge field is represented by a 'quark' with index i , and an 'antiquark' with index j . The second term in the parentheses is because we need to make the gluon field traceless for the $\text{SU}(N)$ group we are considering. It would not be present if we were working with $\text{U}(N)$. In any case, in the large N , the distinction is unimportant. Following the gluon line indices we see, the index pair at the beginning is the same as that at the end. In some

sense, a gluon propagates like quark–anti-quark pair. This observation was made by 't Hooft when he devised the *double-line* notation. Loosely speaking, we will draw as many lines in a propagator as the indices it carries. Therefore, quark propagator is denoted by a single line since it carries one index, whereas a gluon propagator is drawn using double lines. For $SO(N)$ or $USp(N)$ theories, the adjoint representation may be written as a product of two fundamental representations rather than a product of a fundamental and an anti-fundamental representation (like done for $SU(N)$ and $U(N)$). Since the fundamental representation is real, there are no arrows on the propagators.⁴ If we now introduce the standard notation where V , E , F denote the vertices, edges, and faces then a vacuum diagram has the following dependence on λ and N ,

$$\text{diagram}(V, E, F) \sim N^{V-E+F} \lambda^{E-V} \sim N^\chi \lambda^{E-V}$$

where $\chi = F + V - E$ is assured by a theorem due to Euler⁵ which we note below. We have shown an example of this counting for two graphs in Fig.(4) which contribute $N^2 \lambda^2$ (left) and $N^2 \lambda$ (right).

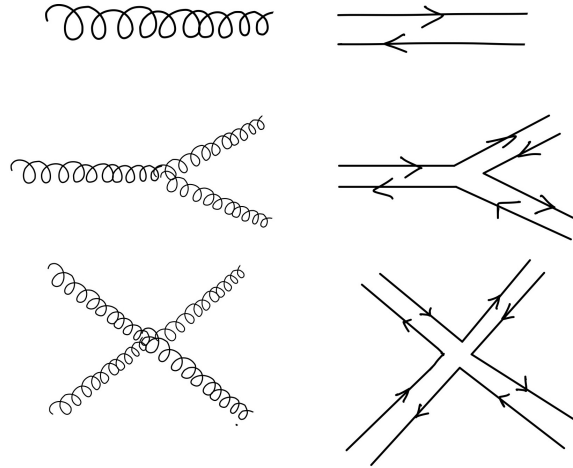


Figure 3. All diagrams (different n -point vertices) all contribute N/λ .

One of the remarkable features of the large N limit is that it implies that the measure becomes concentrated on a single orbit of the gauge group. This is sometimes also referred

⁴Recall that arrows fix the orientation of complex fields

⁵Given a surface composed of polygons with F faces, E edges and V vertices, the Euler character satisfy $\chi = F + V - E = 2 - 2h$, where h is the number of handles (also known as ‘genus’) of the surface. Since, in the large N limit, the diagrams with $h = 0$ contribute the most in the large N limit is also known as the planar limit (because $h = 0$ means no handles like spheres)

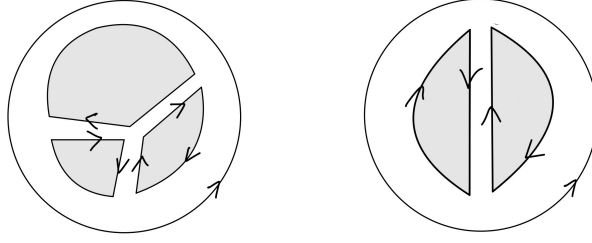


Figure 4. The diagram on the left contributes as $N^2\lambda^2$ while the one on the right is $N^2\lambda$ (right)

to as ‘master-field’ or ‘master-orbit’. It is easy to show that (left as a exercise) if we consider a gauge-invariant observable which is properly normalized the deviation of the measured value from the mean is $\mathcal{O}(1/N^2)$. Mathematically, this is written for any two gauge-invariant operators, A and B as:

$$\langle AB \rangle = \langle A \rangle \langle B \rangle + \mathcal{O}\left(\frac{1}{N^2}\right) \quad (2.6)$$

The variance of operators vanish in this limit and there are no fluctuations. This is simplest to for the case of pure gauge theory but can be easily extended to fermions. The bad news is that till date - no one has been able to calculate the master field in more than two dimensions.

SECTION 3

Analytical solutions

3.1 Hermitian & Unitary matrix models

In these matrix models, one often needs to evaluate integrals of the form:

$$Z = \int \exp \left[-\text{Tr} \sum V(M_i) - \text{Tr} \sum c_{ij} M_i M_j \right] dM_i dM_j \quad (3.1)$$

where M^i are $N \times N$ complex Hermtian matrices.

In study of zero-dimensional gauge theories, we encounter these types of matrix models where the integral is over the Haar measure. These models often have interesting features in the large N limit. For ex., in QCD, the gauge fields are elements of $\text{SU}(3)$ and hence they are well-described in some cases by unitary matrix models. A general model is given by:

$$Z = \int \exp \left[-\text{Tr} \sum V(U_i) - \text{Tr} \sum c_{ij} U_i U_j \right] dU_i dU_j \quad (3.2)$$

where U^i are $N \times N$ unitary matrices.

One of the well-known examples is the unitary matrix model first considered by Gross, Witten, and Wadia (GWW) around 1979. This is often known as one-plaquette matrix model. This unitary matrix model has deep connections to string theory in the so called ‘double-scaling limit’ where $N \rightarrow \infty$ and $\lambda \rightarrow \lambda_c$ simultaneously. The requirement of this double scaling can be understood as follows: If we merely take $N \rightarrow \infty$ then we get genus zero surfaces in the expansion of the free energy. However, this would prohibit string interaction since they would imply change of genus which is not possible in that limit. In taking DSL, this problem is resolved and topological information is maintained. This model is given by :

$$Z = \int \exp \left[-\text{Tr}(U + U^\dagger) \right] dU \quad (3.3)$$

This model admits exact solution for all N and λ in terms of determinant of a Toeplitz matrix. However, it is not very useful and hence this model has been studied by saddle-point methods and orthogonal polynomials (OP). We will sketch the solution using OP closely following [11].

Following the discussion in Appendix, we define the polynomial:

$$P_j(x) = \sum_{k=0}^{j-1} b_k x^k + x^j \quad (3.4)$$

We have to choose polynomials which are orthonormal with respect to the measure which in this case is:

$$\rho(\theta) = \exp \left[\frac{2N}{\lambda} \cos \theta \right] \quad (3.5)$$

and this results in:

$$\int_{-\pi}^{\pi} \rho(\theta) P_m(e^{i\theta}) P_n^*(e^{i\theta}) = a_m \delta_{mn} \quad (3.6)$$

Let us denote $\kappa = 2N/\lambda$ then we have,

$$I_n(\kappa) = I_{-n}(\kappa) = \int_{-\pi}^{\pi} \rho(\theta) e^{in\theta} \quad (3.7)$$

and the polynomials defined above are given by:

$$P_n(z) = \det \begin{pmatrix} I_0 & I_1 & \cdots & I_n \\ I_1 & I_0 & \cdots & I_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z & \cdots & z^n \end{pmatrix} \frac{1}{\det[I_{i-j}]_{i,j=1 \dots N}} \quad (3.8)$$

We note that the coefficients of $P_n(z)$ are real and hence $P_n^*(e^{i\theta}) = P_n(e^{-i\theta})$. It is also straight-

forward by expanding the determinant to see that: $a_n = c_{n+1}/c_n$ where $c_N = \det[I_{i-j}]_{i,j=1\dots n}$ is the Toeplitz determinant.

- Exercise: We have derived the closed expression for the partition function of GWW model. How does this change when N is finite and we consider $SU(N)$ rather than $U(N)$ gauge group.

- Exercise: Convince yourself using MATHEMATICA that one indeed gets the Wigner semi-circle distribution when sample size is large enough for GUE ensembles.

3.2 One-matrix model: Cubic and Quartic potentials

3.3 Kontsevich Model

This model is closely related to topological gravity. In 1992, Kontsevich proved Witte's conjecture of two-dimensional quantum gravity using matrix integral. He considered decomposition of the moduli space of Riemann surfaces into cells described by ribbon graphs and identified those graphs with Feynman diagrams of matrix model in the expansion of Z

The model is given by:

$$Z = \frac{\int dX \exp \left[-N \text{Tr} \left(\frac{MX^2}{2} + \frac{iX^3}{6} \right) \right]}{\int dX \exp \left[-N \text{Tr} \left(\frac{MX^2}{2} \right) \right]} \quad (3.9)$$

$$Z = \int dX \exp \left[-N \text{Tr} \left(\frac{MX^2}{2} + \frac{iX^3}{6} \right) \right] \quad (3.10)$$

Hankel and Toeplitz (i+j) versus (i-j)?

3.4 Chern-Simons and ABJM matrix models

3.5 Ising Model on random planar graph: Two-matrix model

Ising Model on random graph was studied as two-matrix model first by Kazakov in 1986 and the partition function is given by:

$$Z = \int \mathcal{D}A \mathcal{D}B \exp N \text{Tr} \left(-A^2 - B^2 + 2cAB - g \frac{A^3}{3} - g \frac{B^3}{3} \right) \quad (3.11)$$

Note that this has \mathbb{Z}_2 symmetry because of Z being invariant under $A \mapsto B$

Crit. exponents	Ising model on random planar graph	Ising model on regular lattice
α	-1	0
β	1/2	1/8
γ	2	7/4
δ	5	15
νd	3	2
γ_{str}	-1/3	-

Table 2. Summary of critical exponents obtained for two-dimensional Ising model on different graphs

$$Z = \int \mathcal{D}A \mathcal{D}B \exp N \text{Tr} \left(-A^2 - B^2 + 2cAB - g_A e^h \frac{A^3}{3} - g_B e^{-h} \frac{B^3}{3} \right) \quad (3.12)$$

In 1986, Kazakov solved the Ising model on a random graph by **nice word** it through a two-matrix model [?]. This was later extended by Boulatov and Kazakov to admit magnetic fields [?]. We will not discuss the entire solution but will sketch the solution. They also computed the critical exponents and found different results than Onsager's case for regular square lattice. The exponents computed satisfied the usual Rushbrooke's law ($\alpha + 2\beta + \gamma = 2$) and Widom's scaling law: $\gamma/\beta = \delta - 1$. These values coincide with the exponents obtained in a three-dimensional spherical model. This is a striking correspondence between exponents of two different models in different dimensions! In fact, after few years, while discussing the Yang-Lee edge singularity (YLES) on dyamical graph, it was shown that an additional exponent $\sigma = 1/2$ also behaved accordingly. We have listed the exponents in Table (2). ggg

By turning the Z given in in terms of eigenvalues, we get:

$$Z = \int dX dY \Delta(X) \Delta(Y) \exp \left[-N \sum_i (x_i^2 + y_i^2 + 2cx_i y_i + 4ge^h x_i^4 + 4ge^{-h} y_i^4) \right] \quad (3.13)$$

It is now clear that we would need two polynomials $P_k(x)$ and $Q_j(y)$ for this case such that their determinant matches $\Delta(X)$ and $\Delta(Y)$ respectively. These polynomials satisfy the following orthonormal condition:

$$\int dx dy e^{-NV(x,y)} P_k(x) Q_j(y) = h_k \delta_{kj} \quad (3.14)$$

They also satisfy several recursion relations for which the interested reader can refer to [?]:

$$Z = \int dX dY \det[P_r(x_k)] \det[Q_r(y_k)] \exp \left[-N \sum V(X, Y) \right] \quad (3.15)$$

where we have denoted $\sum_i (x_i^2 + y_i^2 + 2cx_i y_i + 4ge^h x_i^4 + 4ge^{-h} y_i^4)$ by $V(X, Y)$. Transforming to the eigenvalue basis of both matrices X and Y and using the expansion of the determinant we get:

$$\begin{aligned} Z &= \epsilon^{i_1 \dots i_N} \epsilon^{j_1 \dots j_N} \int dx_1 \dots dx_N dy_1 \dots dy_N P_{i_1}(x_1) \dots P_{i_N}(x_N) Q_{j_1}(y_1) \dots Q_{j_N}(y_N) e^{-N \sum V(x_i, y_i)} \\ &= \epsilon^{i_1 \dots i_N} \epsilon^{j_1 \dots j_N} \prod_{r=1}^N \int dx_r dy_r e^{-NV(x_r, y_r)} P_{i_r}(x_r) Q_{j_r}(y_r) \\ &= N! \prod_{i=0}^{N-1} h_i \end{aligned} \quad (3.16)$$

We can define $f_k := h_k/h_{k-1}$ and hence (3.16) implies:

$$\log Z_N(c, g, h) = \log N! + N \log h_0 + \sum_{k=1}^{N-1} (N-k) \log f_k \quad (3.17)$$

One is usually interested in computing the quantity (the subscript ‘pc’ denotes planar/continuum limit i.e., $N \rightarrow \infty$):

$$F_{pc} = \frac{1}{N^2} \log \left(\frac{Z(c, g, h)}{Z(c, 0, 0)} \right) = \frac{1}{N} \sum_{k=1}^{N-1} \left(1 - \frac{k}{N} \log \left(\frac{f_k}{f_{k,0}} \right) \right) \quad (3.18)$$

3.6 General p -matrix open chain

This model can easily be solved using the method of resolvent as explained in detail in [7, 12] and first solved in the famous BIPZ [13] paper. Consider a potential $V(y)$, and assume that the eigenvalues have single cut in the interval $C = [-a, a]$ and a resolvent,

$$G(x) = \int_{-a}^a \frac{1}{2\pi i} \frac{\sqrt{x^2 - a^2}}{\sqrt{y^2 - a^2}} NV'(y) \frac{1}{x - y} \quad (3.19)$$

Note that if the cut was instead $C = [b, a]$ then we would have,

$$G(x) = \oint \frac{dy}{2\pi i} \sqrt{\frac{(x-a)(x-b)}{(y-a)(y-b)}} NV'(y) \frac{1}{x - y} \quad (3.20)$$

$$Z = \int dM e^{-\frac{N}{g} \text{Tr} V(M)} \quad (3.21)$$

$$= \int \prod d\lambda_i \Delta^2(\lambda) e^{-\frac{N}{g} \sum V(\lambda_i)} \quad (3.22)$$

where $\Delta(\lambda) = \prod_{i>j} (\lambda_i - \lambda_j) = \exp[\sum_{i>j} \log |\lambda_i - \lambda_j|]$ is the Vandermonde determinant. If we now vary above for one of the eigenvalues gives the saddle point equation ,

$$\frac{2}{N} \sum_{j \neq i} \frac{1}{\lambda_i - \lambda_j} = \frac{1}{g} V'(\lambda_i) \quad (3.23)$$

It is useful to introduce the density of eigenvalues as:

$$\rho(\lambda) = \frac{1}{N} \sum_{i=1}^N \delta(\lambda - \lambda_i) \quad (3.24)$$

In the limit of large N , we can write (3.23) as:

$$V'(\lambda) = 2g \oint_a^b d\mu \frac{\rho(\mu)}{\lambda - \mu} \quad (3.25)$$

where by \oint we meant the Cauchy principal value of the integral. We can write resolvent as:
(resolvent is the Stieljes transform of the eigenvalue density)

$$G(z) = 2 \oint_{-a}^a d\mu \frac{\rho(\mu)}{z - \mu} \quad (3.26)$$

then we can write above as (using Sokhotski-Plemelj theorem

$$G(z \pm i\epsilon) = \oint_{-a}^a d\mu \frac{\rho(\mu)}{z - \mu} \mp i\pi\rho(z) \quad (3.27)$$

which is just:

$$G(z \pm i\epsilon) = \frac{1}{2} V'(z) \mp i\pi\rho(z) \quad (3.28)$$

Once we find the resolvent, we are done and we know $\rho(z)$.

- Find the expected values for $\text{Tr}(X^2)$ for one matrix model (1MM) with $g = 1.0$ in the planar limit using the well-known ‘saddle-point’ method (also sometimes ‘stationary-phase’). Set up this problem for a fixed $N = 100$ and carry out the Monte Carlo procedure. Compare the results. The PYTHON code can be found in the **Appendix**. You can

refer to the MATHEMATICA script given in the Appendix for carrying out the analytical computation.

- Exercise: Derive the loop equations (aka Schwinger-Dyson equations) given below:

$$\langle \text{Tr} M^k V'(M) \rangle = \sum_{l=0}^{k-1} \langle \text{Tr} M^l \rangle \langle \text{Tr} M^{k-l-1} \rangle$$

We can also consider one-matrix model with cubic interaction instead of quartic as above. There is well-known instability associated with cubic potential. The model is given by:

$$V(M) = \frac{M^2}{2} + \frac{gM^3}{3} \quad (3.29)$$

It is only well-defined if $g^2 \leq 1/12 = 0.04811\sqrt{3} \implies g_c \leq 0.21935$. This is the radius of convergence of the planar perturbation series.

Harish-Chandra-Itzykson-Zuber (HCIZ) integral:

$$I = \int \mathcal{D}U \exp \left[\beta \text{Tr}(U^\dagger M_1 U M_2) \right] \quad (3.30)$$

$$= \left(\prod_{p=1}^{N-1} p! \right) \frac{\det \left(\exp \left[\beta \lambda_i \nu_j \right] \right)}{\Delta(\lambda) \Delta(\nu) \beta^{(N^2-N)/2}} \quad (3.31)$$

with β being a non-zero parameter and U is a unitary $N \times N$ matrix.

We now give a simple proof of HCIZ integral formula.

We can do a character expansion as follows:

$$\exp \left[\beta \text{Tr}(M_1 U M_2 U^\dagger) \right] = \sum_r \alpha_r(\beta) \chi_r(M_1 U M_2 U^\dagger) \quad (3.32)$$

where we have assumed that both M_1 and M_2 belong to the group algebra. We can use two orthogonality conditions given by:

$$\int dU \chi_r(U) \chi_{r'}(U) = \delta_{rr'} \quad (3.33)$$

and,

$$\int dU U_{\sigma\beta}^r U_{\gamma\delta}^{*r'} = \frac{1}{d_r} \delta_{rr'} \delta_{\sigma\gamma} \delta_{\beta\delta} \quad (3.34)$$

Using (3.34) in (3.32), we get:

$$\exp \left[\beta \text{Tr}(M_1 U M_2 U^\dagger) \right] = \sum_r \frac{\alpha_r}{d_r} \chi_r(M_1) \chi_r(M_2) \quad (3.35)$$

Now we have to use the expansion for α given by:

$$\alpha_{\{n_1, \dots, n_N\}}(\beta) = \beta^{n_1 + \dots + n_N} \left[\prod_{i=1}^N \frac{(N-i)!}{(N+n_i-i)!} \right] d_{\{n_1, \dots, n_N\}}(\beta) \quad (3.36)$$

where $d_{\{n_1, \dots, n_N\}}$ is the dimension corresponding to the partition (n_1, n_2, \dots, n_N)

Weyl formula:

Let us consider representations of the $U(N)$ group which are labeled by a partition into N parts as (n_1, n_2, \dots, n_N) with $n_1 \geq n_2 \geq \dots \geq n_N$. The character ⁶ of the irreducible representation corresponding to the partition (n_1, n_2, \dots, n_N) of non-negative integers can be obtained using the Weyl's formula:

$$\chi_{(n_1, n_2, \dots, n_N)}(U) = \frac{\det(\lambda_i^{n_j + N - j})}{\Delta(\lambda_1, \dots, \lambda_N)} \quad (3.37)$$

where λ_i are the eigenvalues of the U matrix in the fundamental representation and $\Delta(\lambda_1, \dots, \lambda_N)$ is the well-known Vandermonde determinant.

- Exercise: Show that $\det(V) = \prod_{i < j} (\lambda_i - \lambda_j)$ where V is given by:

$$V = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & \dots & \lambda_N^{N-1} \end{pmatrix} = \lambda_i^{j-1}$$

It is now easy to show that we get the HCIZ formula. The interested reader can refer [14] for more details.

Darboux's theorem is a theorem in differential forms and a major theorem in symplectic geometry. Suppose that ω is a differential 1-form on an $2l$ dimensional manifold, such that $d\omega$ has constant rank p . Then if $\omega \wedge (d\omega)^p = 0$ everywhere, we can find a local coordinates so that

$$\omega = \sum_{i=1}^l dp_i \wedge dq_i \quad (3.38)$$

⁶it is defined as the trace of the representation matrix

- Exercise: Show that by using saddle point approximation, following holds:

$$I(\alpha) = \lim_{\alpha \rightarrow 0} \int_{-\infty}^{\infty} e^{-\frac{1}{\alpha} f(x)} dx = \sqrt{\frac{2\pi\alpha}{f''(x_0)}} e^{-f(x_0)/\alpha}$$

Find the terms which are of $\mathcal{O}(\alpha)$ by doing little more work and show that:

$$I(\alpha) = \int_{-\infty}^{\infty} e^{-\frac{1}{\alpha} f(x)} dx = \sqrt{\frac{2\pi\alpha}{f''(x_0)}} e^{-f(x_0)/\alpha} \left[1 + \left[\frac{5}{24} \frac{(f''')^2}{(f'')^3} - \frac{3}{24} \frac{f''''}{(f'')^2} \right] \alpha + \mathcal{O}(\alpha^2) \right]$$

3.7 Open p -matrix chain model

The general p -matrix model chain was first considered in [15]

$$Z_p(g, c, \kappa) = \int \mathcal{D}M_1 \cdots \mathcal{D}M_p \exp \text{Tr} \left(\sum_{i=1}^p -M_i^2 - gM_i^4 + c \sum_{i=1}^{p-1} M_i M_{i+1} + \kappa M_p M_1 \right) \quad (3.39)$$

Now we can set $p = 3$ to get the action:

$$S(g, c, \kappa) = N \text{Tr} \left(\sum_{i=1}^3 X_i^2 + \sum_{i=1}^3 g X_i^4 - c \sum_{i=1}^2 X_i X_{i+1} - \kappa X_3 X_1 \right) \quad (3.40)$$

- Exercise: Set up the action for this three-matrix model chain problem and implement the HMC algorithm. Explore the limit when the chain is open i.e., $\kappa = 0$ and when $c = \kappa$. Comment on the observation you make.

SECTION 4

Numerical solutions

There is only a selected list of analytically solvable matrix models in the planar limit. This inevitably brings the thought of attempting numerical solutions. Unfortunately, even here, there are not many methods which one can use. In fact, there only comes two methods to mind with the second method barely few years old. This clearly signals the fact that there still remains lot of work to be done in devising new numerical methods to solve matrix models. The most frequently used method is Monte Carlo (MC) but that has its own problems. The newly introduced numerical bootstrap method has already been used to solve several matrix models [16–19] but the extension to matrix models with more than two matrices i.e., 3-matrix commutator type models or IKKT models seems difficult at the moment. This is related to

the fact that with more matrices, the loop equations which become highly non-linear are difficult to handle. In these notes, we will mostly focus on MC methods since as of this writing bootstrap methods have not been able to access or constrain matrix models with more than two matrices. However, we want to explain the bootstrap solution for the case of one matrix model as first shown by [17] for the interested reader. For detailed exposition, we refer the reader to the references above. We hope that this ‘yet’ small section on bootstrap methods would be extended in later versions of these notes as more results accumulate in coming years.

4.1 Numerical bootstrap

The basic idea of bootstrapping matrix models depends on the positivity (positive-definiteness) of the bootstrap matrix which we refer in what follows by \mathcal{M} . For the case of one matrix model with a potential $V(X)$ given by:

$$V(X) = \frac{1}{2}X^2 + \frac{g}{4}X^4, \quad (4.1)$$

it is straightforward to show that the odd moments of X i.e., $t_n = \text{Tr} X^n = 0$ for odd n while the even moments (of order greater than two) are all related to $t_2 = \text{Tr} X^2/N$. This renders the model simple to bootstrap since there is no growth of words (combination of matrix or matrices!) since all non-zero t_n can be related to t_2 .

• Exercise: Use loop equations and show that it is possible to write t_4, t_6, t_8 in terms of t_2 . Also check this using either MATHEMATICA or PYTHON [see Appendix for codes]

If we consider positive constraints that can be derived from $\langle \text{Tr}(\Phi^\dagger \Phi) \rangle \geq 0$ where Φ is superposition of matrices which for one matrix model is $\Phi = \sum_k \alpha_k M^k$. This condition is equivalent to positive definite nature of $\mathcal{M} \succeq 0$ where $\mathcal{M}_{ij} = \langle \text{Tr} M^{i+j} \rangle$. We can only enforce subset of these constraints. For example, it was sufficient to access positive definite nature of $\mathcal{M}_{6 \times 6} \succeq 0$ and sub-matrices to get to the exact solution in the original article [17]. For example, $\mathcal{M}_{2 \times 2}$ is given by:

$$\mathcal{M}_{jk} = \begin{pmatrix} t_{2j} & t_{j+k} \\ t_{j+k} & t_{2k} \end{pmatrix} \succeq 0 \quad (4.2)$$

In this case, solving the model just means finding the bounds on t_2 since all others can then be calculated (see the exercise above).

Following this work, a two matrix quantum mechanics model (in 0+1-dimensions) was solved using similar techniques. The results obtained were shown to be consistent with MC

results. In this work, the Hamiltonian considered was given by:

$$H = \text{Tr}\left(P^2 + X^2 + \frac{g}{N}X^4\right) \quad (4.3)$$

and corresponding to the trial operators up to length $(L) = 2$, they considered \mathbb{I}, X, X^2 and P . The bootstrap matrix of size $2^L \times 2^L$ which should be positive definite is constructed as:

$$\mathcal{M} = \begin{pmatrix} \langle \text{Tr} \mathbb{I} \rangle & \langle \text{Tr} X^2 \rangle & 0 & 0 \\ \langle \text{Tr} X^2 \rangle & \langle \text{Tr} X^4 \rangle & 0 & 0 \\ 0 & 0 & \langle \text{Tr} X^2 \rangle & \langle \text{Tr} XP \rangle \\ 0 & 0 & \langle \text{Tr} PX \rangle & \langle \text{Tr} P^2 \rangle \end{pmatrix} \succeq 0 \quad (4.4)$$

They considered bootstrap matrices up to $L = 4$ and observed convergence to the expected result. For the case of two-matrix quantum mechanics, the convergence was found to be slow but consistent with results expected using Monte Carlo results and bounds from Born-Oppenheimer wavefunction.

Another two-matrix integral model given by the action (4.17) was recently solved [19]. In this work, authors used relaxation bootstrap methods for $\Lambda = 11$ (determining the length of largest words considered) to constrain the moment to five decimal places. They obtained $\text{Tr} X^2/N = t_2$ to be bounded between $0.421783612 \leq t_2 \leq 0.421784687$ while $\text{Tr} X^4/N = t_4$ between $0.333341358 \leq t_4 \leq 0.333342131$. This model in more detail and its corresponding solution using MC methods will be discussed in **SECTION** —

4.2 Monte Carlo method

The numerical method which is state-of-the-art in classical computations of matrix models and quantum field theories in general is the Monte Carlo approach. For higher-dimensional models, one starts with the lattice formulation which reduces the path-integral introduced by Feynman into many ordinary integrals. But even for a simple gauge theory like \mathbb{Z}_2 in four dimensions this is not practical to evaluate. The fact that we need to do so many integrals suggests that may be some statistical interpretation and this is where the basic idea of Monte Carlo comes in. It selects the configuration which dominates the path integral and then ignores the rest of the available phase space. In that way one constructs a chain of configurations which can lead to the required distribution. Metropolis-Hastings algorithm and Hamiltonian/Hybrid Monte Carlo (HMC) are two frequently used methods which can generate a Markov chain and lead to unique stationary distribution. We will here focus on the latter since that has now become the state-of-the-art in various numerical computations. This method was introduced in 1987 by Duane, Kennedy, Pendleton, and Roweth who put

together the ideas from Markov chain Monte Carlo (MCMC)⁷ and molecular dynamics (MD) methods. For a detailed review about HMC and its extension to rational HMC which is required for fermions, the interested readers can consult the Refs. [20]. The two basic parts of HMC are: a) Use of integrator to evolve and propose a new configuration, b) accept or reject the proposed configuration.

4.2.1 Leapfrog integrator and Metropolis step

The leapfrog method is used to numerically integrating differential equations. This is a second-order method in the sense that energy non-conservation depends on square of step-size used. This method is ‘symplectic’, i.e., it preserves the area of the phase space. We can intuitively

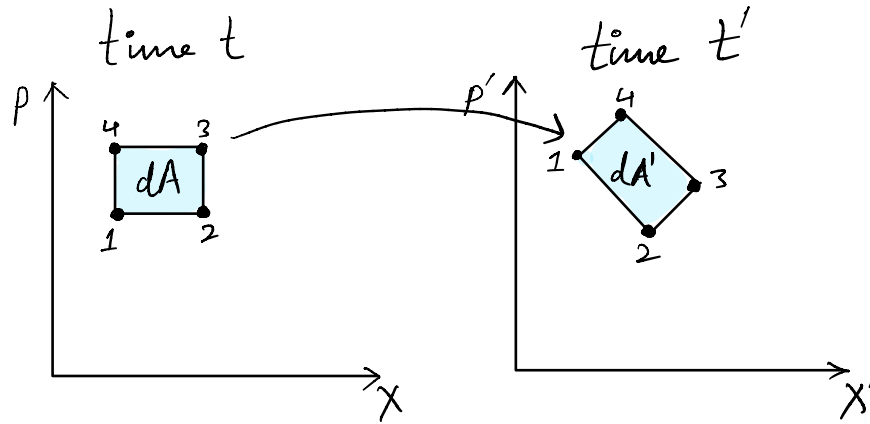


Figure 5. A representation of conservation of phase space area. This is also referred to as ‘Liouville theorem. it asserts that under evolution of the system, it may change the shape of the shaded region but not the volume since probability has to be conserved.

understand this as follows: Consider a rectangular region of area dA as shown below in Fig. (??). The four corners at time t are denoted by (x, p) , $(x + dx, p)$, $(x, p + dp)$, $(x + dx, p + dp)$. At some later time t' , this will change to form corners of some quadrilateral as shown with area dA' . The Liouville’s theorem⁸ states that the areas are equal, i.e., $dA = dA'$. Using this idea we can easily prove an important equality used to check various lattice computations employing symplectic algorithms. The basic steps of the algorithm are as follows:

- $X_i(\delta\tau/2) = X_i(0) + P_i(0)\delta\tau/2$
- Now several inner steps where $n = 1 \cdots (N - 1)$

⁷MCMC originated in the seminal paper of Metropolis et al [cite](#), where it was used to simulate the state distribution for a system of ideal molecules.

⁸This statement of Liouville theorem is closely related to detailed balance condition which says that in equilibrium there is balance between any two pairs of states i.e., equal probability.

$$P(n\delta\tau) = P((n-1)\delta\tau) - f_i((n-\frac{1}{2})\delta\tau)\delta\tau$$

$$X((n+1/2)\delta\tau) = X((n-1/2)\delta\tau) + P_i(n\delta\tau)\delta\tau$$

- $P(N\delta\tau) = P((N-1)\delta\tau) - f_i((N-\frac{1}{2})\delta\tau)\delta\tau$
- $X(N\delta\tau) = X((N-1/2)\delta\tau) + P_i(N\delta\tau)\frac{\delta\tau}{2}$

• Exercise: Show that if a symplectic integrator is used, then $\langle e^{-\Delta H} \rangle = 1$. Check this this also holds in any given HMC simulation within errors.

In the last step of HMC, a Metropolis test is carried out to accept or reject the proposed configuration. Suppose we start from the configuration X of a one matrix model which is a $N \times N$ matrix and carry out the leapfrog part with some parameters and obtain a new configuration X^* . The test then computes: $\min.(1, e^{-\Delta H})$ and generates a random number between $r \in [0, 1]$. The new configuration is rejected if $\min.(1, e^{-\Delta H}) < r$ otherwise accepted.

4.2.2 Random number generator

We will now discuss how to generate momentum matrices at the start of the leapfrog method taken from a Gaussian distribution. Suppose we have two numbers U and V taken from uniform distribution i.e., $(0,1)$ and we want two random numbers with probability density function $p(X)$ and $p(Y)$ given by:

$$p(X) = \frac{1}{\sqrt{2\pi}} e^{-X^2/2} \quad (4.5)$$

and,

$$p(Y) = \frac{1}{\sqrt{2\pi}} e^{-Y^2/2} \quad (4.6)$$

Since X and Y are independent sets:

$$p(X, Y) = p(X)p(Y) = \frac{1}{2\pi} e^{-R^2/2} = p(R, \Theta) \quad (4.7)$$

where $R = X^2 + Y^2$. This then means that we can identify the below:

$$U = \frac{\Theta}{2\pi} \quad (4.8)$$

and,

$$V = e^{-R^2/2} \implies R = \sqrt{-2 \log(V)} \quad (4.9)$$

This then immediately means,

$$X = R \cos \Theta = \sqrt{-2 \log(V)} \cos(2\pi U) \quad (4.10)$$

$$Y = R \sin \Theta = \sqrt{-2 \log(V)} \sin(2\pi U) \quad (4.11)$$

The code to implement this is given below:

‡ Sample code of random number generator

```
def box_muller():
    # Basic form (alt: polar form) of Box-Muller random number generator for pairs of
    # independent, standard, normally distributed (mean=0, sigma=1)
    # given a source of uniformly distributed random numbers in (0,1)
    # p, q have weights exp(-p^2/2)/sqrt(2pi) and exp(-q^2/2)/sqrt(2pi) respectively.

    PI = 2.0*math.asin(1.0);
    r = random.uniform(0,1)
    s = random.uniform(0,1)
    x = np.sqrt(-2.0*np.log(r)) * math.sin(2.0*PI*s)
    y = np.sqrt(-2.0*np.log(r)) * math.cos(2.0*PI*s)
    return x,y
```

It is easy to check that it indeed generates a sGaussian distribution as shown in Fig.6.

4.2.3 Autocorrelation and errors

It is well-known that in any given Markov chain, the new states (i.e., configurations) can be highly correlated to previous ones. In order to ascertain that the measurement of an observable \mathcal{O} is not affected by correlated configurations, it is essential for proper statistical analysis to know the extent to which they are correlated. In this regard, it is important to measure the autocorrelation time τ_{auto} which measures the time it takes for two measurements to be considered independent of each other. So, if we generate L configurations, then actually only L/τ_{auto} are useful for computing averages. We define autocorrelation function of observable \mathcal{O} as:

$$C(t) = \frac{\langle \mathcal{O}(t_0) \mathcal{O}(t_0 + t) \rangle - \langle \mathcal{O}(t_0) \rangle \langle \mathcal{O}(t_0 + t) \rangle}{\langle \mathcal{O}^2(t_0) \rangle - \langle \mathcal{O}(t_0) \rangle^2} \quad (4.12)$$

The behaviour of $C(t)$ is $\sim \exp(-t/\tau_{\text{auto}})$ as $t \rightarrow \infty$. This is called exponential autocorrelation time. We can also compute something which is called ‘integrated autocorrelation time’ defined

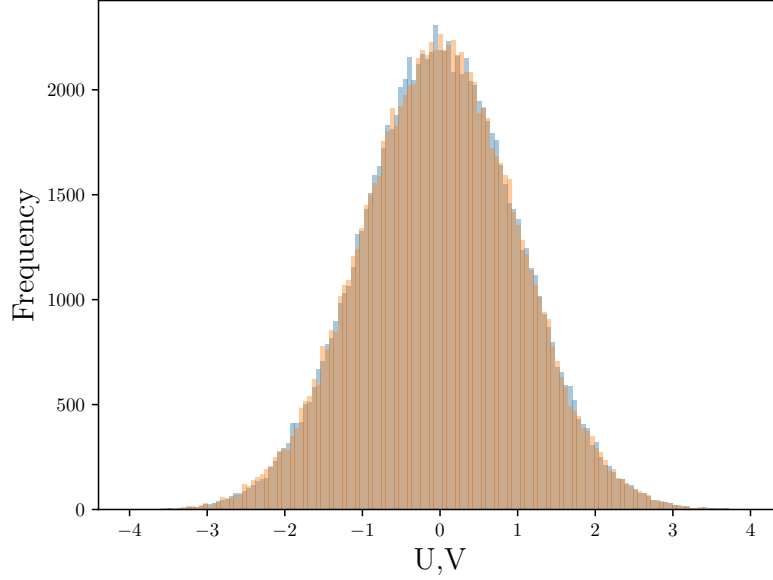


Figure 6. In the limit of large sample size, it tends to Gaussian with mean zero and $\sigma = 1$.

as:

$$\tau_{\text{auto}}^{\text{int.}} = \frac{\sum_{t=1}^{\infty} \langle \mathcal{O}(t_0) \mathcal{O}(t_0 + t) \rangle - \langle \mathcal{O} \rangle^2}{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2} \quad (4.13)$$

We can write this in terms of sum over autocorrelation function as: $\tau_{\text{auto}}^{\text{int.}} = 1 + \sum_{t=1}^N C(t)$. In general, τ_{auto} increases with system size, close to critical point. One can express the statistical error in the average of \mathcal{O} denoted by $\delta\mathcal{O}$ is given in terms of variance and integrated autocorrelation time as:

$$\delta\mathcal{O} = \frac{\sigma}{\sqrt{N}} \sqrt{2\tau_{\text{auto}}^{\text{int.}}} \quad (4.14)$$

where we have usual definitions i.e., $\sigma = \sqrt{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2}$ and N is the number of measurements.

4.3 One-matrix model: Confirming exact results

4.4 Matrix model closed chain solution

4.5 Three-matrix Yang Mills matrix model

Read Badis review Add Chern-Simon term

4.6 Hoppe's matrix model

This model was first introduced by Hoppe in 1982 and solved later by Kazakov-Kostov-Nekrasov (KKN) in [21]. The partition function is given by:

$$Z = \int \mathcal{D}X \mathcal{D}Y \exp \left[-N \operatorname{Tr}(X^2 + Y^2 - g^2[X, Y]^2) \right] \quad (4.15)$$

At large values of coupling i.e., $g \rightarrow \infty$, this model becomes commuting with $[X, Y] \rightarrow 0$. The presence of commutator term in matrix models is common especially in models which have a dual gravity interpretation. The exact result for average action is:

$$2\langle S_c \rangle + \langle S_q \rangle = N^2 - 1, \quad (4.16)$$

where $S_c = -Ng^2 \operatorname{Tr}[X, Y]^2$ and $S_q = N \operatorname{Tr}(X^2 + Y^2)$

4.7 Generalized Hoppe's model

We can consider a slightly more general two-matrix model which reduces to Hoppe's model mentioned above in a special limit. Such a matrix model generally not solvable. This model was considered in [19] and solved using bootstrap methods.

$$Z = \int \mathcal{D}X \mathcal{D}Y \exp \left[-N \operatorname{Tr}(X^2 + Y^2 - g^2[X, Y]^2 + hX^4 + hY^4) \right] \quad (4.17)$$

For $h = 0$ it can be reduced to Hoppe's matrix model which can be solved via saddle-point analysis or through the reduction to a Kadomtsev-Petviashvili (KP) type equation. For $g = 0$ it reduces to two decoupled one-matrix models and for $g = \infty$, we have $[X, Y] = 0$ and it becomes an eigenvalue problem.

We show in Fig. 7 obtained using Monte Carlo (MC) are consistent with those obtained in [19]. The shown MC results were obtained for $N = 200$ in 445 seconds on a 2019 Macbook.

4.8 Extension of Hoppe's model to D matrices

Matrix model with $SO(D)$ invariance:

$$Z = \int \mathcal{D}X_1 \cdots \mathcal{D}X_d \exp \left[-\sum_i \operatorname{Tr} X_i^2 + \lambda^2 \sum_{i,j} \operatorname{Tr}[X_i, X_j]^2 \right] \quad (4.18)$$

If we consider $X_i \mapsto (1 + \epsilon)X_i$, it must leave Z invariant, one arrives at:

$$d(N^2 - 1) = 2d\langle \operatorname{Tr} X_i^2 \rangle - 4\lambda^2 d(d-1) \langle \operatorname{Tr}[X_i, X_j]^2 \rangle \quad (4.19)$$

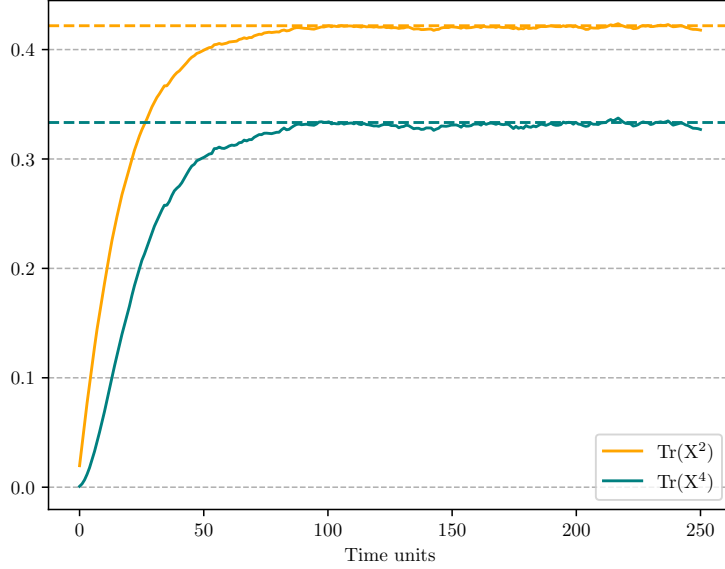


Figure 7. We show the bootstrap results by dashed lines and MC by solid lines

4.9 Dimer Model: Yang-Lee singularity

We start from Equation (4.7) in [22]

$$Z = \int \mathcal{D}X \mathcal{D}M \exp N \text{Tr} \left(-\frac{X^2}{2} + \frac{gX^4}{4} - \frac{M^2}{2} + g\sqrt{\zeta} M X^3 \right) \quad (4.20)$$

and then integrate out the matrix M by completing the squares method to get:

$$Z = \int \mathcal{D}X \exp N \text{Tr} \left(-\frac{X^2}{2} + \frac{gX^4}{4} + g^2 \zeta \frac{X^6}{2} \right) \quad (4.21)$$

Note that if we set $\zeta = 0$, this reduces to the familiar 1MM problem we solved in previous section.

- Exercise: Check that 4.21 follows from 4.20.

4.10 Matrix chain: Ends closed

The matrix chain is a complicated p matrix model which was first considered in [15]. When the chain is connected, the model is not solvable. We use Monte Carlo methods to study this with $N = 100$ which is close to the desired planar limit.

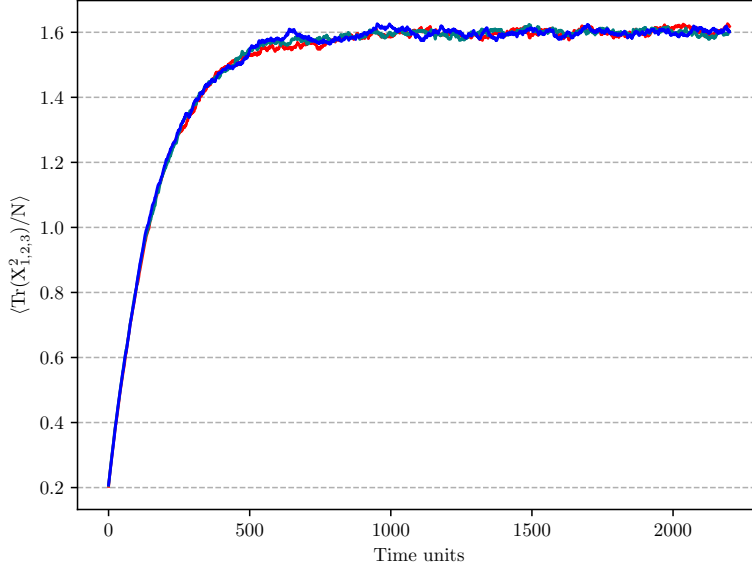


Figure 8. The computed value of normalized $\text{Tr}(X_{1,2,3}^2)$ with MC methods when $g = 1$ and $\kappa = c = 1.35$. It is clear that there is a $SO(3)$ symmetry between scalars and they converge to the same value ~ 1.61 for $N = 100$ we have considered.

$$Z_p(g, c, \kappa) = \int \mathcal{D}M_1 \cdots \mathcal{D}M_p \exp \text{Tr} \left(\sum_{i=1}^p -M_i^2 - gM_i^4 + c \sum_{i=1}^{p-1} M_i M_{i+1} + \kappa M_p M_1 \right) \quad (4.22)$$

$$S(g, c, \kappa) = N \text{Tr} \left(\sum_{i=1}^3 X_i^2 + \sum_{i=1}^3 g X_i^4 - c \sum_{i=1}^2 X_i X_{i+1} - \kappa X_3 X_1 \right) \quad (4.23)$$

SECTION 5

Matrix models for string theory and holography

5.1 $c = 1$ MQM

One of the significant developments in the 1990s was the realization that there is a duality between a special $c = 1$ matrix quantum mechanics (MQM) and two-dimensional bosonic string theory. The action for the matrix model is given by:

$$Z = \int \mathcal{D}X \exp \left[-\beta \int dt \text{Tr}(\dot{X}^2) + V(X) \right] \quad (5.1)$$

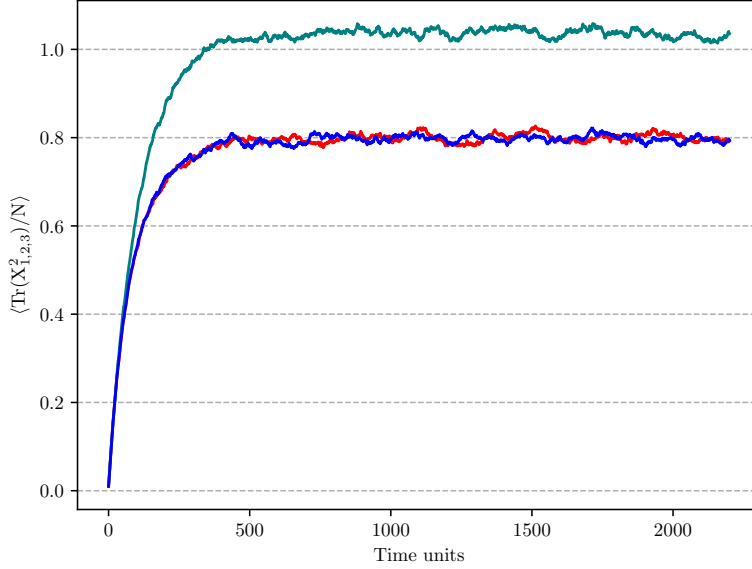


Figure 9. The computed value of normalized $\text{Tr}(X_{1,2,3}^2)$ with MC methods when $g = 1, c = 1.35, \kappa = 0$. The absence of term connecting the ends results in $SO(3)$ symmetry being broken. They converge to two sets of value ~ 0.80 and ~ 1.03 for $N = 100$.

where t runs either from $-\infty$ to ∞ or from 0 to 2π and X is a $N \times N$ Hermitian matrix. The planar graph expansion of Z discretizes string worldsheet in the target space time direction t while an additional spatial direction grows from the eigenvalue coordinate of X . By changing to the eigenvalue basis and introducing an antisymmetric function, it can be show that this model reduces to a fermionic problem with N degrees of freedom in external potential with each ‘particle’ behaving independently. This is in fact very surprising since we initially started with Bose particles and now ran into Fermi statistics [13, 23]. This relation to N free fermions is the foundation of the exact solubility of string theory in two dimensions [24].

- Exercise: Show that if $\Delta(\lambda) = \prod_{i < j} (\lambda_i - \lambda_j)$, then following is true:

$$\sum_i (\lambda_i)^k \frac{\partial^k}{\partial \lambda_i^k} \Delta = \sum_{r=k}^{n-1} \frac{r!}{(r-k)!} \Delta = \frac{n(n-1) \cdots (n-k)}{k+1} \Delta = k! \binom{n}{k+1} \Delta$$

5.2 BFSS matrix model

The non-perturbative features of supersymmetric Yang-Mills (SYM) is thought to play a crucial role in the beyond Standard model (BSM) Physics and also in M/String theory. The non-perturbative study of string theory is based on various matrix models. Up to present, two kinds of non-perturbative formulation of M/string theory are developed. The one is Matrix theory. The typical examples are BFSS Matrix model and the plane wave Matrix model (PWMM) which describes a sector of M theory with the definite light-cone momentum on flat background and pp-wave background, respectively. M theory on a generic weakly curved background is described by BFSS Matrix model with the corresponding vertex operator perturbations added. The trick was to work in an infinite momentum frame (now known as “light-cone frame”) instead of a rest frame. However, even that one dimensional quantum theory is not well defined in the canonical ensemble (fixed temperature) (BFSS) because it has a moduli space. The moduli space in this theory is given by the eigenvalues of nine commuting matrices i.e $[X_i, X_j] = 0$ that transform under the $SO(9)$ R-symmetry. The existence of these flat directions means that, at finite temperature, the partition function is formally divergent, and it was shown that when Monte Carlo simulations of this theory are performed, this divergence eventually causes the simulation to break down. However, even that one dimensional quantum theory is still not well defined in the canonical ensemble because it has a moduli space. The moduli space in this theory is given by the eigenvalues of nine commuting matrices that transform under the $SO(9)$ R-symmetry. The existence of these flat directions means that, at finite temperature, the partition function is formally divergent, and it was shown in that when Monte Carlo simulations of this theory are performed, this divergence eventually causes the simulation to break down. The second kind is the AdS/CFT correspondence, for which the $AdS^5/\mathcal{N} = 4$ SYM correspondence is famously studied following Maldacena.

In a seminal paper [25], Banks-Fischler-Shenker-Susskind proposed a matrix model emphasizing the relation between uncompactified eleven dimensional M-theory in the light cone frame and the planar limit of the supersymmetric matrix quantum mechanics describing $D0$ -branes. Since, in recent years, it has been emphasized how the gauge symmetry can be elevated to being a global symmetry in this case, we will discuss the model in two subsections.

5.2.1 Gauge singlet sector

$$S_{\text{BFSS}} = \frac{N}{4\lambda} \int dt \text{Tr} \left[-(D_t X^i)^2 - \frac{1}{2} [X^i, X^j]^2 + \Psi^\alpha D_t \Psi^\alpha + i \Psi^\alpha \gamma_{\alpha\beta}^j [\Psi^\beta, X^j] \right], \quad (5.2)$$

where D_t is the covariant derivative and summation over spatial indices $i, j = 1, \dots, 9$ and spinor indices $\alpha, \beta = 1, \dots, 16$ is implicit. By dimensional analysis, one concludes that the 't Hooft coupling λ has dimension $[E]^3$. This system is controlled by two dimensionless

parameters: N and $t = T/\lambda^{\frac{1}{3}}$, where T will be referred to as temperature. BFSS model has been well-studied over the decades using Gaussian methods, numerical methods, etc. The supersymmetric mode has a single deconfined phase while if we ignore the fermions, we have a deconfinement transition between confined and deconfined phases. The model possesses moduli space consisting of commuting matrices. This is a problem in general since the thermal partition function is not well-defined and an IR cutoff needs to be introduced in most cases.

5.2.2 Including non-singlet sector

The gauge symmetry constraint implies that the states in the Hilbert space have to satisfy the condition that they should be annihilated by gauge symmetry i.e., they are gauge singlets. It was shown in that as far as the low-temperature region is concerned, the contribution of gauge non-singlet states are exponentially small and hence there is no difference if we elevate the gauge symmetry to be a global symmetry. It was shown that non-singlet states correspond to long strings which are localized around the boundary and are important for understanding the bulk gravity. This proposal was also confirmed numerically by Monte Carlo results.

5.3 IKKT matrix model

Just shortly after the BFSS matrix model was proposed as a description of M-theory, the authors of [26] proposed a reduction of quantum mechanical model down to zero dimensions which is now named after them. This model is related to D-instantons in the sense that it has the same action as that of low-energy effective action of D-instantons. This model was proposed with hope to play a key role in description of Type IIB string theory in DLCQ. The action is written schematically as:

$$S = \frac{N}{4\lambda} \int \text{Tr} \left(\frac{1}{4} [X_\mu, X_\nu]^2 + \bar{\psi} \Gamma^\mu [A_\mu, \psi] \right) \quad (5.3)$$

where X_μ and ψ are $N \times N$ Hermitian matrices. The gauge field is a ten-dimensional vector and the ψ are ten-dimensional Majorana-Weyl spinor field respectively. This model in zero dimensions possess no usual space-time supersymmetry but is dimensional reduction of $\mathcal{N} = 1$ SYM theory in ten dimensions. It is expected that in this model both space and time should be dynamically generated as a result of the dynamics of the large matrices. The space-time should be constructed only from the matrices. IKKT model has no free parameters since λ can be absorbed in the field redefinitions.

In the original model, since it is a ten-dimensional reduction we have, $\mu, \nu = 1 \cdots 10$. It is possible to consider other models where $\mu, \nu = 1 \cdots D$ where $D = 4, 6$. One might worry whether because of the integral measure being over non-compact X , it is divergent. The convergence issues of these models for $D = 4, 6, 10$ was studied by —. In what follows

in these notes, we will ignore the fermionic term and only focus on the commutator/bosonic term. One of the observables (also known as ‘size’ or i.e., extent of scalars) which we compute in these models is:

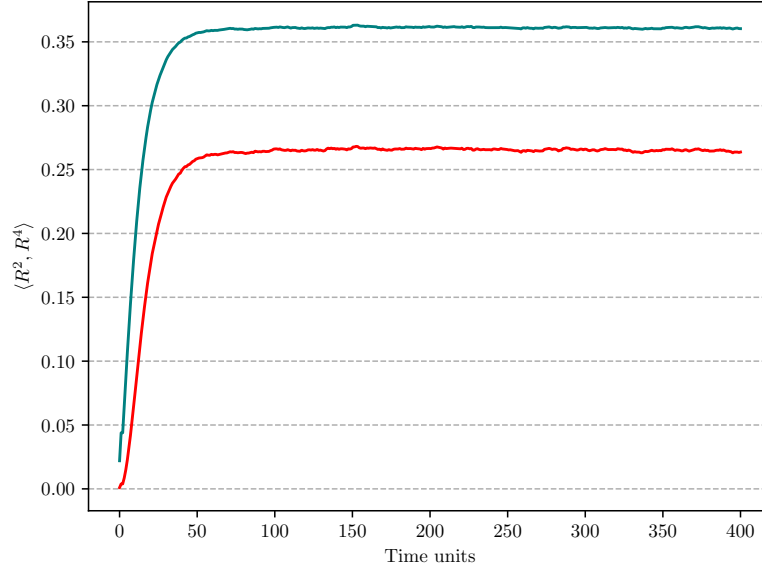


Figure 10. The scalar extents for the bosonic IKKT model. After about 200 trajectories, it seems to converge to the answer. The plot is for $N = 200$ and $\lambda = 1$.

$$\langle R^2 \rangle = \frac{1}{DN} \left\langle \text{Tr} \sum_{i=1}^D X_i^2 \right\rangle, \quad \langle R^4 \rangle = \frac{1}{DN} \left\langle \text{Tr} \sum_{i=1}^D X_i^4 \right\rangle, \quad (5.4)$$

It is known that R^2 should behave as $\sqrt{\lambda}$ in the large N limit from [Nishimura '98](#) but we have not found any previous study which fixes the prefactor. Our numerical results suggest that $\langle R^2 \rangle = 0.361(2)\sqrt{\lambda}$ and $\langle R^4 \rangle = 0.265(3)\lambda$ with $N = 200$ for a wide range of couplings i.e., $\lambda \in [1, 100]$. One of the standard tests we do for the reliability of the numerical results is computing the average action. It can be shown that under a change: $X \rightarrow e^\epsilon X$ if we demand that Z is invariant, then we find:

$$\frac{\langle S \rangle}{N^2 - 1} = \frac{D}{4} \quad (5.5)$$

We show in [Fig—](#) that for $D = 10$ we get the correct answer and hence the reported results above can be fully trusted.

- Exercise (★): Derive (5.5) by doing the change: $X_\mu \mapsto e^\epsilon X_\mu$ and ignoring terms in $\mathcal{O}(\epsilon^2)$

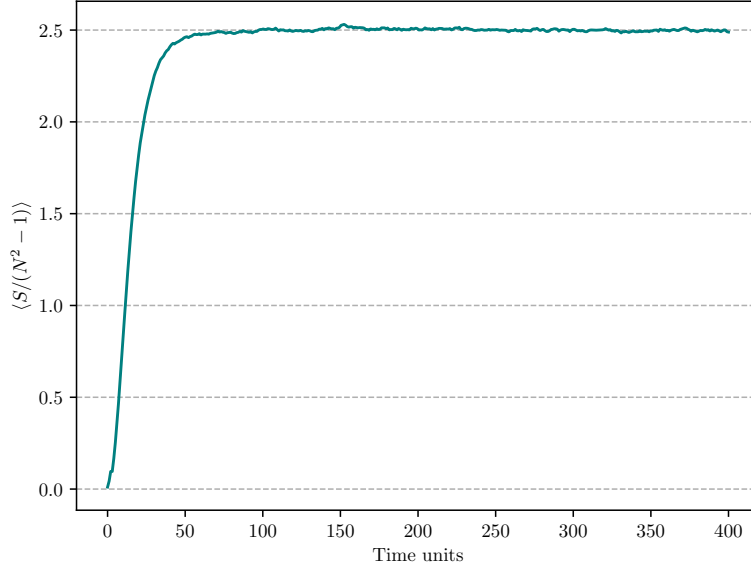


Figure 11. The average action (normalized) for the $D = 10$ bosonic sector of IKKT model.

We also note that if we consider $D = 3$, then as argued in [27] it becomes little tricky and **our results supports this?**.

There has been some recent work which tries to take the master-field approach to the IKKT model [28]. This is a promising direction but it is not yet clear how effective it will be.

- Exercise: Carry out the MC computation for the bosonic IKKT model (10 MM) for the action given in (5.3). After sufficient thermalization cut, check that the results are consistent with exact result obtained using Schwinger-Dyson equations within errors. Refer to appendix for the PYTHON code and to perform Jackknife error analysis with sufficiently large block size.

5.4 BMN matrix model

We now consider another useful matrix model which is an extension of the BFSS model. This model is a massive deformation of BFSS (5.2) and is known as PWMM (Plane-Wave Matrix

Model) or BMN after the authors of [29]. The dual description is on a plane wave background which retains the supersymmetry unlike BFSS which is dual to M-theory on a flat geometry.

5.4.1 Full sector

The action reads

$$S = S_{\text{BFSS}} + \frac{N}{4\lambda} \int dt \text{Tr} \left[-\frac{\mu^2}{3^2} (X^i)^2 - \frac{\mu^2}{6^2} (X^a)^2 + \frac{\mu}{4} \Psi^\alpha (\gamma^{123})_{\alpha\beta} \Psi^\beta - \frac{\sqrt{2}\mu}{3} \epsilon_{ijk} X^i X^j X^k \right], \quad (5.6)$$

where the indices i, j, k run over 1, 2, 3 and the index a runs over 4, ..., 9. The mass parameter μ breaks the $SO(9)$ global symmetry of (5.2) down to $SO(6) \times SO(3)$. This mass deformation of BFSS retains maximal supersymmetry [29]. These extra terms in the BMN model give this model two advantages over the BFSS.

- In the limit of $\mu \rightarrow \infty$ the model becomes weakly coupled and can be studied perturbatively using Gaussian gauged matrix model.
- The mass terms lift the moduli space of the BFSS model and give a discrete set of vacua. This is useful for numerical simulations because it means the theory has well-defined thermal canonical ensemble.

However, for a dual gravity description, the mass term which depends on μ should be small. The lattice coupling will eventually depend on $N, N_t, \lambda, \beta, \mu$. The first four should be familiar with BFSS 1d sixteen supercharge lattice coupling, the addition of μ is what makes the phase structure exciting to study.

$$S_B = \frac{N}{4\lambda} \sum_{n=0}^{N_\tau-1} \text{Tr} \left[(D_\tau X_n^i)^2 + \sum_{i<j} [X_n^i, X_n^j]^2 + \left(\frac{\mu}{3} X_n^I\right)^2 + \left(\frac{\mu}{6} X_n^M\right)^2 + \frac{\sqrt{2}\mu}{3} \epsilon_{IJK} X_n^I X_n^J X_n^K \right], \quad (5.7)$$

where N_τ is the number of lattice sites. The term which is cubic in scalars is often referred to as ‘Myers term’. In fact, there exist some studies of models where this term is added to the usual commutator term.

In those cases, the action becomes:

$$S = N \text{Tr} \left(\sum_{\mu, \nu=1}^{d>3} [X_\mu, X_\nu]^2 + \frac{i\alpha}{3} \epsilon^{abc} X_a X_b X_c \right) \quad (5.8)$$

The bosonic model has a first-order phase transition in the large N limit. For small μ while for large μ there is an additional phase transition between two confined phases.

In [30], it was claimed that for a model with $d > 1$ matrices (bosonic or fermionic), of masses $\omega_j > 0$ with $j = 1 \dots d$, the inverse Hagedorn temperature β_H is given by the solution of:

$$\sum_{j=1}^d e^{-\beta_H \omega_j} = 1. \quad (5.9)$$

This can be used to accurately determine the critical temperature for the full BMN model at weak couplings. If we define, $\mu/T = x$, the critical temperature is the solution of the following equation:

$$3e^{-x/3} + 6e^{-x/6} + \underbrace{8e^{-x/4}}_{\text{fermions}} = 1 \implies \frac{T}{\mu} \approx 0.076 \quad (5.10)$$

5.4.2 Bosonic sector

BMM model has a rich phase structure even with fermions left out. It again has a deconfinement transition which can be calculated exactly using

This estimate works well also for $g \rightarrow 0$ with no fermions by just removing the term corresponding to fermions above. We compare this exact result to our numerical result and find excellent agreement.

In an old work [31], the critical temperature was computed in the large N limit in $1/D$ expansion for the $\mu = 0$ counterpart. The result is:

$$T(d) = \frac{\sqrt[3]{d}}{\left(\frac{203}{160} \frac{\sqrt{5}}{d} + 1\right) \log(d)}, \quad (5.11)$$

It was shown numerically in [32], that for $d = 9$, the obtained $T_c = 0.895$ compared well to the analytical result obtained i.e., $T_c = 0.894658$. The transition is first-order.

In the bosonic BMN case we consider with $d = 9$, we must retrieve the above result in the limit $\mu \rightarrow 0$. Therefore, it is expected that the critical temperature will behave as:

$$T_c = 0.895(1 + f(\mu)) \quad (5.12)$$

We recently performed numerical computations and found the form with $N = 32, 48$. A more ambitious goal would be to compute the $1/D$ expansion with μ . This might be guessed by varying the number of scalar fields i.e., $d \neq 9$. Other thing we can compute is the distribution ρ of eigenvalues of scalars X . A similar computation can be seen in [33].

It is also well-known for BFSS model at high temperatures where the fermions decouple that $E/N^2 = 6T$ in the notation where $\lambda = 1$ and large N limit has been taken. This is a

standard result, see [34] for additional details.

Now, this calculations is not done for mass deformed matrix model such as bosonic BMN. It would be nice to find an expression like one below using our lattice data.

$$\frac{E}{N^2} = 6T \left(1 + f(\mu) \right) \quad (5.13)$$

SECTION 6

Summary

We have discussed a wide range of matrix models some of which Are analytically solvable I the plaar limit. The purpose is to introduce the reader to Monte Carlo methods and Show the applications by providing codes and clear explanation.

Acknowledgements

The author is indebted to Pedro Vieira for numerous discussions and for encouragement. The author was supported by a postdoctoral fellowship at the Perimeter Institute for Theoretical Physics during 2019-2022. Research at Perimeter Institute is supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Economic Development, Job Creation and Trade. The author would like to thank Department of Science & Technology, Govern- ment of India for KVPY (Kishore Vaigyanik Protsahan Yojana) Scholarship during 2008-2010, European Union for Erasmus Mundus scholarship during 2010-2011, Department of Physics at Syracuse University for support during 2013-2019.

Last edited by RGJ: 2021-09-28 at 16:00:03

SECTION A

Character Expansion

The method of character expansion is well-known to lattice gauge theorists and was first initiated in a seminal work by Migdal [cite](#). The basic idea of character expansion is to utilize a specific transform on group manifold. The method of character expansion was introduced for graph enumeration and solving matrix models in [\[35, 36\]](#). This was done since there were many interesting cases where the HCIZ formula was not immediately applicable. We will consider this method and show how it can be used to solve a two Hermitian matrix model with ABAB interaction. This method has also been applied to dually weighted graphs (DWG). **Check!** These models have deep connections to the 8-vertex models in Statistical physics on dynamical planar graphs and was also shown to be connected to the non-perturbative formulation of three-dimensional Lorentzian quantum gravity [\[37\]](#). See [\[38\]](#) for a review of this method.

Let us consider:

$$Z = \int \mathcal{D}X \exp \text{Tr} \left[-X^2 + V(AX) \right] \quad (\text{A.1})$$

Where A is an external matrix and V is the potential. Because of the general form of the potential HCIZ formula is not applicable. We can do the character expansion as follows:

$$\exp \text{Tr} V(AX) = \sum_R f_R \chi_R(AX) \quad (\text{A.2})$$

Where the coefficients f_R are fiction of the irreducible representation Of the Gaige group.

$$f_R = \int d\Omega \exp \left[\text{tTr} V(\Omega) \right] \chi_R(\Omega^\dagger) \quad (\text{A.3})$$

SECTION B

Orthogonal polynomials

One of the methods we discussed for the solution of large N limit of matrix models was the saddle point approximation. However, this method is not useful to understand the terms subleading in $1/N$. The method of orthogonal polynomials introduced by Bessis [\[? \]](#) is usually used for such computations. In fact, it is also very useful in solving multi-matrix models. These polynomials are defined as:

$$\int d\lambda e^{-V(\lambda)} P_n(\lambda) P_m(\lambda) = \int d\mu(\lambda) P_n(\lambda) P_m(\lambda) = a_n \delta_{mn} \quad (\text{B.1})$$

where $d\mu(\lambda) = d\lambda e^{-V(\lambda)}$ is the measure

The basic idea is to rewrite the Vandermonde determinant appearing after we change from matrix basis to the basis of eigenvalues.

$$\Delta(\lambda) = \det(\lambda_i^{j-1})_{1 \leq i, j \leq N} = \det(P_{j-1}(\lambda_i))_{1 \leq i, j \leq N} \quad (\text{B.2})$$

These polynomials can solve:

$$\exp(Z) = \int dM \exp[-\text{Tr} V(M)] \quad (\text{B.3})$$

In fact, it can be shown that B.3 is equivalent to:

$$\exp(Z) = N! a_0^N \prod_{k=1}^N f_k^{N-k} \quad (\text{B.4})$$

where $f_k := a_k/a_{k-1}$. Hence, solving the matrix model is now equivalent to solving for the normalization appearing in B.1.

As we have shown in ..., we can use this method to solve the unitary matrix model.

SECTION C

Schwinger-Dyson/loop equations

The basic idea of the loop equations of matrix models is to capture the invariance of the model under field redefinitions. This is also sometimes known as ‘Schwinger-Dyson (SD) equations’. One of the exercises in the main text was to derive these equations. Here, we will sketch a proof for the interested reader. We start by noting that the integral of the total derivative vanish and hence:

$$\sum_{i,j} \int dM \frac{\partial}{\partial M_{ij}} \left((M^k)_{ij} e^{-N \text{Tr} V(M)} \right) = 0, \quad (\text{C.1})$$

By computing the derivatives and using large N factorization, we obtain:

$$\left\langle \text{Tr} M^k V'(M) \right\rangle = \sum_{l=0}^{k-1} \langle \text{Tr} M^l \rangle \langle \text{Tr} M^{k-l-1} \rangle \quad (\text{C.2})$$

In the steps above, we have used two identities:

$$\frac{\partial}{\partial M_{ij}} (M^k)_{ij} = \sum_{l=0}^{k-1} (M^l)_{ii} (M^{k-l-1})_{jj} \quad (\text{C.3})$$

and,

$$\frac{\partial}{\partial M_{ij}} e^{-N \text{Tr} V(M)} = -N V'(M)_{ji} e^{-N \text{Tr} V(M)} \quad (\text{C.4})$$

where V' denotes the derivative w.r.t to the matrix. However, these loop equations are not valid when the integration is over some other matrix ensembles (such as orthogonal/symplectic). It is better to start from the eigenvalue integral representation to consider general $\beta \in \mathbb{C}$. We can write moments as:

$$\langle \text{Tr} M^k \rangle = \frac{1}{Z} \int \Delta(\lambda)^\beta d\lambda_1 \cdots d\lambda_N \exp \left(-\frac{N\beta}{2} \sum_i V(\lambda_i) \right) \sum_{i=1}^N \lambda_i^k \quad (\text{C.5})$$

It is easy to derive ‘generalized loop equations’ from here using the fact that inntegral of total derivatives vanishes. We obtain:

$$\underbrace{\left\langle \text{Tr} M^k V'(M) \right\rangle + k \left(\frac{2}{\beta} - 1 \right) \text{Tr} M^{k-1}}_{\text{zero for } \beta = 2} = \sum_{l=0}^{k-1} \langle \text{Tr} M^l \rangle \langle \text{Tr} M^{k-l-1} \rangle \quad (\text{C.6})$$

SECTION D

Mathematica code for solution of one-matrix model

We now give the details to solve the one matrix model in MATHEMATICA. For this we consider the potential:

$$V(Y) = \frac{Y^2}{2} + \frac{gY^4}{4}$$

As we have shown in the text, for this case, the higher moments of trace of Y are related and hence we will just calculate $\text{Tr} Y^2$ in the planar limit (normalized by N). We give the code below for $g = 1$. One is encouraged to try and change g and see how the results change.

```
V[y_]=y^2/2+(g y^4)/4;
G[x_]=Integrate[-1/(2\[Pi]I)Sqrt[x^2-a^2]/Sqrt[y^2-a^2] (N V'[y])/(x-y),{y,-a,a},
Assumptions->{x>a,a>0}];
sol=Series[G[x],{x,\[Infinity], 1}]-N/x//Simplify//Solve[# == 0,a]&//Simplify;
Series[G[x],{x,\[Infinity], 5}]/Normal;
{Coefficient[%, x, -3]}/N;
% /. sol;
%/.{g -> 1}//Chop//N// Grid
```

Python code for solution of one-matrix model.

We now solve the 1MM using the Monte Carlo method which is the standard in lattice field theory literature. By executing the code given below And waiting about 55-60 seconds on a modern laptop, we get the result shown in Fig. 12 below. We can readily extend this code (by changing NMAT) to study matrix models where the integration is over several different matrices. As we will discuss later, for the case of IKKT model, there are ‘ten’ matrices involved and hence this code will be adjusted accordingly to study that model for given λ and N .

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import random
import math
from matplotlib.pyplot import *
from matplotlib import pyplot as plt
import numpy as np
import random
from scipy.linalg import expm
from numpy import linalg as LA
from numpy.linalg import matrix_power
import time
import datetime
import sys

startTime = time.time()
print ("STARTED:" , datetime.datetime.now().strftime("%d %B %Y, %H:%M:%S"))

if len(sys.argv) < 4:
    print("Usage:", str(sys.argv[0]), "READIN " " SAVE " " NCOL")
    sys.exit(1)

READIN = int(sys.argv[1])
SAVE = int(sys.argv[2])
NCOL = int(sys.argv[3])

NMAT = 1
Nitters_sim = 10 # Should be even
g = 1.
dt = 1e-3
nsteps = int(0.5/dt)
skip = 2.
cut=int(0.25*Nitters_sim)
X = np.zeros((NMAT, NCOL, NCOL), dtype=complex)
mom_X = np.zeros((NMAT, NCOL, NCOL), dtype=complex)
f_X = np.zeros((NMAT, NCOL, NCOL), dtype=complex)
```

```

X_bak = np.zeros((NMAT, NCOL, NCOL), dtype=complex)
HAM, expDS, XSQ, X4SQ, X6SQ, MOM = [], [], [], [], [], []

print ("Matrix integral simulation with number of matrices = %4.0f" %(NMAT))
print ("NCOL=" "%3.0f " ", " " and 'g' = " " "%4.2f" % (NCOL, g))
print ("-----")

def dagger(a):
    return np.transpose(a).conj()

def box_muller():
    PI = 2.0*math.asin(1.0);
    r = random.uniform(0,1)
    s = random.uniform(0,1)
    p = np.sqrt(-2.0*np.log(r)) * math.sin(2.0*PI*s)
    q = np.sqrt(-2.0*np.log(r)) * math.cos(2.0*PI*s)
    return p,q

def copy_fields(b):
    for j in range(NMAT):
        X_bak[j] = b[j]
    return X_bak

def rejected_go_back_old_fields(a):
    for j in range(NMAT):
        X[j] = a[j]
    return X

def refresh_mom():
    for j in range (NMAT):
        mom_X[j] = random_hermitian()
    return mom_X

def random_hermitian():
    tmp = np.zeros((NCOL, NCOL), dtype=complex)

    for i in range (NCOL):
        for j in range (i+1, NCOL):
            r1, r2 = box_muller()
            tmp[i][i] = complex(r1, 0.0)
            tmp[i][j] = complex(r1, r2)/math.sqrt(2)
            tmp[j][i] = complex(r1, -r2)/math.sqrt(2)

    if np.allclose(tmp, dagger(tmp)):
        return tmp

def hamil(X,mom_X):
    ham = action(X)
    for j in range (NMAT):
        ham += 0.50 * np.trace(np.dot(mom_X[j],mom_X[j])).real

```

```

    return ham

def action(X):
    b_action = 0.0
    for i in range (NMAT):
        b_action += 0.50 * np.trace(np.dot(X[i],X[i])).real
        b_action += (g/4.0)* np.trace((matrix_power(X[i], 4))).real
    return b_action*NCOL

def force(X):
    f_X = np.zeros((NMAT, NCOL, NCOL), dtype=complex)
    for i in range (NMAT):
        f_X[i] = (X[i] + (g*(matrix_power(X[i], 3))))*NCOL
    return f_X

def leapfrog(X,dt):
    mom_X = refresh_mom()
    ham_init = hamil(X,mom_X)

    for j in range(NMAT):
        X[j] += mom_X[j] * dt * 0.5 # Half step

    for i in range(1, nsteps+1):
        f_X = force(X)
        for j in range(NMAT):
            mom_X[j] -= f_X[j] * dt # Full step
            X[j] += mom_X[j] * dt

        f_X = force(X)
        for j in range(NMAT):

            mom_X[j] -= f_X[j] * dt
            X[j] += mom_X[j] * dt * 0.5 # Half step

    ham_final = hamil(X,mom_X)
    return X, ham_init, ham_final

if __name__ == '__main__':

    if READIN ==0:
        for i in range (NMAT):
            for j in range (NCOL):
                for k in range (NCOL):
                    X[i][j][k] = complex(0.0,0.0)

    if READIN ==1:
        print ("Reading old config.")
        with open("config_mm.txt") as f2:
            A = np.loadtxt(f2).view(complex)
            f2.close()

```



```

        for i in range (NMAT):
            for j in range (NCOL):
                for k in range (NCOL):
                    X[i][j][k] = A[(NCOL*i)+j][k]

acc_count = 0
for MDTU in range (Niters_sim):

    X_bak = copy_fields(X)
    X, start, end = leapfrog(X, dt)
    change = end - start
    expDS.append(np.exp(-1.0*change))
    if np.exp(-change) < random.uniform(0,1):
        X = rejected_go_back_old_fields(X_bak)
        print(("REJECT: deltaH = " "%8.7f " " startH = " "%8.7f" " endH = " "%8.7f" % (change,
            start, end)))
    else:
        print(("ACCEPT: deltaH = " "%8.7f " "startH = " "%8.7f" " endH = " "%8.7f" % (change,
            start, end)))
        acc_count += 1

    if MDTU%skip == 0:
        tmp0 = np.trace(np.dot(X[0],X[0])).real
        XSQ.append(tmp0/NCOL)
        tmp1 = np.trace(X[0] @ X[0] @ X[0] @ X[0]).real
        X4SQ.append(tmp1/NCOL)

if SAVE ==1:

    print ("Saving config.")
    f1 = open("config_mm.txt", "w")
    for i in range (NMAT):
        np.savetxt(f1, X[i].view(float), delimiter= " ")
    f1.close()

plt.rc('text', usetex=True)
plt.rc('font', family='serif')
MDTU = np.linspace(0, int(Niters_sim/skip), int(Niters_sim/skip), endpoint=True)
plt.ylabel(r'Tr(X$^2$)', fontsize=16)
plt.xlabel('Time units', fontsize=16)
plt.grid(which='major', axis='y', linestyle='--')
plt.figure(1)
plot (MDTU, XSQ, 'teal')
print ("Fraction of MDTU accepted", (acc_count/Niters_sim)*100)

if READIN == 0:
    XSQ = XSQ[cut:]
    X4SQ = X4SQ[cut:]
    X6SQ = X6SQ[cut:]
    expDS = expDS[cut:]

```

```

print("<Tr X^2 / NCOL>", np.mean(XSQ), "+/-", (np.std(XSQ)/np.sqrt(np.size(XSQ) - 1.0)))
print("<Tr X^4 / NCOL>", np.mean(X4SQ), "+/-", (np.std(X4SQ)/np.sqrt(np.size(X4SQ) - 1.0)))
print("exp(-deltaH)", np.mean(expDS), "+/-", np.std(expDS)/np.sqrt(np.size(expDS) - 1.0))
plt.savefig('mm_plot.pdf')
plt.show()
print ("COMPLETED:" , datetime.datetime.now().strftime("%d %B %Y, %H:%M:%S"))
endTime = time.time()
print ("Running time:", round(endTime - startTime, 2), "seconds")
    
```

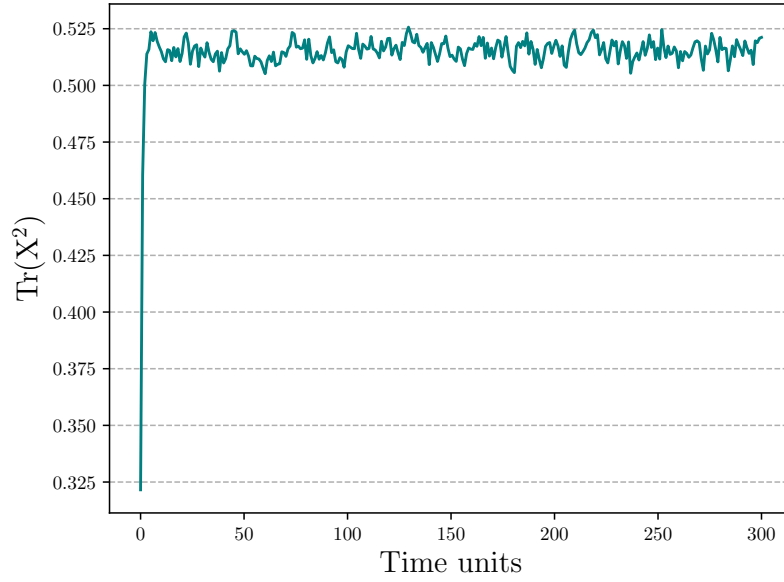


Figure 12. The computed value of $\text{Tr}(X^2)$ with MC methods is consistent with that obtained using analytical saddle-point methods. These results are with $N = 150$ and $g = 1$.

SECTION F

Bosonic sector of IKKT model: Python code

We now provide a sketch of the Monte Carlo code with which we can study the bosonic part of the IKKT model. The details about the model and its relation to the non-perturbative formulations of string theory is discussed in **SECTION XXX**. We have intentionally left out some parts of the code for the reader to fill out from previous study of single matrix model case.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import random
import os
import math
from matplotlib.pyplot import *
from matplotlib import pyplot as plt
import numpy as np
import random
from scipy.linalg import expm
from numpy import linalg as LA
import time
import datetime
import sys
startTime = time.time()
print ("STARTED:" , datetime.datetime.now().strftime("%d %B %Y %H:%M:%S"))

if len(sys.argv) < 5:
    print("Usage:", str(sys.argv[0]), "READIN " " SAVE_or_NOT " " NCOL " " LAM")
    sys.exit(1)

READIN = int(sys.argv[1])
SAVE = int(sys.argv[2])
NCOL = int(sys.argv[3])
LAMBDA = float(sys.argv[4])
COUPLING = float(NCOL/(4.0*LAMBDA))
GENS = NCOL**2 - 1
NSCALAR = 10
dt=5e-4
nsteps = int(1e-2/dt)
Nriters_sim=6
GAP=1
xsq = np.zeros((NSCALAR),dtype=float)
X = np.zeros((NSCALAR, NCOL, NCOL), dtype=complex)
mom_X = np.zeros((NSCALAR, NCOL, NCOL), dtype=complex)
f_X = np.zeros((NSCALAR, NCOL, NCOL), dtype=complex)
X_bak = np.zeros((NSCALAR, NCOL, NCOL), dtype=complex)
HAM, expDS, ACT, scalar = [],[],[],[]

print ("IKKT matrix model simulation with only bosonic term")
print ("NCOL=" "%3.0f " " " and dimensionless coupling = " " %4.2f" % (NCOL, COUPLING))
print ("-----")

def dagger(a):
    return np.transpose(a).conj()

def box_muller():
    PI = 2.0*math.asin(1.0);
    r = random.uniform(0,1)
    s = random.uniform(0,1)
```

```

p = np.sqrt(-2.0*np.log(r)) * math.sin(2.0*PI*s)
q = np.sqrt(-2.0*np.log(r)) * math.cos(2.0*PI*s)
return p,q

def comm(A,B):
    return np.dot(A,B) - np.dot(B,A)

def unit_matrix():
    matrix = np.zeros((NCOL, NCOL), dtype=complex)
    for i in range (NCOL):
        matrix[i][i] = complex(1.0,0.0)
    return matrix

def copy_fields(b):
    for j in range(NSCALAR):
        X_bak[j] = b[j]
    return X_bak

def rejected_go_back_old_fields(a):
    for j in range(NSCALAR):
        X[j] = a[j]
    return X

def refresh_mom():
    for j in range (NSCALAR):
        mom_X[j] = random_hermitian()
    return mom_X

def random_hermitian():
    tmp = np.zeros((NCOL, NCOL), dtype=complex)
    for i in range (NCOL):

        for j in range (i+1, NCOL):
            r1, r2 = box_muller()
            tmp[i][j] = complex(r1, r2)/math.sqrt(2)
            tmp[j][i] = complex(r1, -r2)/math.sqrt(2)

    for i in range (NCOL):
        r1, r2 = box_muller()
        tmp[i][i] = complex(r1, 0.0)
    return tmp

def kinetic_energy(mom_X):
    s = 0.0
    for j in range (NSCALAR):
        s += 0.50 * np.trace(np.dot(dagger(mom_X[j]),mom_X[j]))
    return s.real

def action(X):
    b_action = 0.0
    for i in range (NSCALAR):

```

```

        for j in range (i+1, NSCALAR):
            co = np.dot(X[i],X[j]) - np.dot(X[j],X[i])
            tr = np.trace(np.dot(co,co))
            b_action -= COUPLING*(tr.real)
    return b_action

def force(X):
    tmp_X = np.zeros((NSCALAR, NCOL, NCOL), dtype=complex)
    for i in range (NSCALAR):
        for j in range (NSCALAR):
            if i == j:
                continue
            else:
                temp = comm(X[i], X[j])
                tmp_X[i] -= comm(X[j], temp)
            f_X[i] = 2.0*COUPLING*dagger(tmp_X[i])
    return f_X

def leapfrog(X,mom_X, dt):
    for j in range(NSCALAR):
        X[j] += mom_X[j] * dt/2.0
    f_X = force(X)

    for step in range(nsteps):
        for j in range(NSCALAR):
            mom_X[j] -= f_X[j] * dt
            X[j] += mom_X[j] * dt
            f_X = force(X)

    for j in range(NSCALAR):
        mom_X[j] -= f_X[j] * dt
        X[j] += mom_X[j] * dt/2.0

    return X, mom_X, f_X

def update(X):
    mom_X = refresh_mom()
    KE = kinetic_energy(mom_X)
    ba = action(X)
    start_act = ba + KE
    X_bak = copy_fields(X)
    X, mom_X, f_X = leapfrog(X,mom_X,dt)
    KE = kinetic_energy(mom_X)
    ba = action(X)
    end_act = ba + KE
    change = end_act - start_act
    HAM.append(abs(change))
    expDS.append(np.exp(-1.0*change))

    if np.exp(-change) < random.uniform(0,1):
        X = rejected_go_back_old_fields(X_bak)

```

```

        print(("REJECT: deltaS = " "%8.7f " " startS = " "%8.7f" " endS = " "%8.7f" % (change,
            start_act, end_act)))
    else:
        print(("ACCEPT: deltaS = " "%8.7f " "startS = " "%8.7f" " endS = " "%8.7f" % (change,
            start_act, end_act)))

    ACT.append(ba)

    tmp = 0.0
    for i in range (0,10):
        val = np.trace(np.dot(X[i],X[i])).real/NCOL
        xsq[i] = val
        tmp += val

    tmp /= NSCALAR
    scalar.append(tmp)

    if MDTU%GAP == 0:
        f3.write("%4.8f \n" %(ba/GENS))
        f4.write("%4.8f " "%4.8f " "%4.8f " "%4.8f " "%4.8f " "%4.8f " "%4.8f " "%4.8f " "%4.8f "
            "%4.8f \n" %(xsq[0],xsq[1],xsq[2],xsq[3],xsq[4],xsq[5],xsq[6],xsq[7],xsq[8],xsq[9]))
    return X

if __name__ == '__main__':

    if READIN ==0:
        for i in range (NSCALAR):
            X[i] = random_hermitian()/(NCOL**2)

    if READIN ==1:
        name_f = "config_IKKT/config_IKKT_N{}_l{}_D{}.txt".format(NCOL, LAMBDA, NSCALAR)
        if os.path.isfile(name_f) == True:
            print ("Reading old configuration")
            with open(name_f) as f2:
                A = np.loadtxt(f2).view(complex)
            f2.close()
            for i in range (NSCALAR):
                for a in range (NCOL):
                    for b in range (NCOL):
                        X[i][a][b] = A[(NCOL*i)+a][b]
        else:
            print ("Can't find config. file for this NCOL and LAM")
            print ("Starting from fresh")
            for i in range (NSCALAR):
                X[i] = random_hermitian()

```

```

X = X/(NCOL**2)

f3 = open("action_SD.txt", "w")
f4 = open("extent_of_space_R2.txt", "w")
for MDTU in range (Niters_sim):
    X = update(X)
f3.close()
f4.close()

if SAVE ==1:
    print ("Saving config.")
    name_f = "config_IKKT/config_IKKT_N{}_l{}_D{}.txt".format(NCOL, LAMBDA, NSCALAR)
    f1 = open(name_f, "w")
    for i in range (NSCALAR):
        np.savetxt(f1, X[i].view(float), delimiter= " ")
    f1.close()

ACT = [x/GENS for x in ACT]

print("<S> = ", np.mean(ACT), "+/-", (np.std(ACT)/np.sqrt(np.size(ACT) - 1.0)))
print("<exp(-deltaH)> = ", np.mean(expDS), "+/-", np.std(expDS)/np.sqrt(np.size(expDS) - 1.0))
print ("COMPLETED:" , datetime.datetime.now().strftime("%d %B %Y %H:%M:%S"))
endTime = time.time()

plt.rc('text', usetex=True)
plt.rc('font', family='serif')
MDTU = np.linspace(0, Niters_sim, Niters_sim, endpoint=True)
plt.ylabel(r'$\langle \text{angle } R^2 \rangle$')
plt.xlabel('Time units')
plot(MDTU, scalar, 'teal')
plt.grid(which='major', axis='y', linestyle='--')
plt.show()
plt.ylabel(r'$\langle S/(N^2-1) \rangle$')
plt.xlabel('Time units')
plot(MDTU, ACT, 'teal')
plt.grid(which='major', axis='y', linestyle='--')
plt.savefig('IKKT_plot_S.pdf')
plt.show()
print ("Running time:", round(endTime - startTime, 2), "seconds")

```

SECTION G

Solutions to selected Exercises

★ Solution to Exercise 5:

We want to evaluate:

$$I(\alpha) = \lim_{\alpha \rightarrow 0} \int_{-\infty}^{\infty} e^{-\frac{1}{\alpha} f(x)} dx, \quad (\text{G.1})$$

We use the Taylor expansion about the saddle point x_0 and throw away higher-order terms to get:

$$f(x) = f(x_0) + f''(x_0)(x - x_0)^2 + \dots \quad (\text{G.2})$$

Using G.2 in G.3 along with the famous Gaussian integral result i.e., $\int_{-\infty}^{\infty} e^{-\alpha x^2} dx = \sqrt{\frac{\pi}{\alpha}}$ we get the desired result:

$$I(\alpha) = \sqrt{\frac{2\pi\alpha}{f''(x_0)}} e^{-f(x_0)/\alpha}. \quad (\text{G.3})$$

To compute the higher order corrections, we have to use the (\dots) in (G.2). With some algebra, it is easy to evaluate this.

★ Solution to Exercise :

We now show that $\det(V) = \prod_{i < j} (\lambda_i - \lambda_j)$ where V is:

$$V = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{N-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & \dots & \lambda_N^{N-1} \end{pmatrix} = \lambda_i^{j-1}$$

We first note that the determinant is unchanged if we make the change to all columns except the first given by:

$$\lambda_i^{j-1} \rightarrow \lambda_i^{j-1} - \lambda_1 \lambda_i^{j-2} \quad (\text{G.4})$$

then we have,

$$\det(V') = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \lambda_2 - \lambda_1 & \lambda_2(\lambda_2 - \lambda_1) & \dots & \lambda_2^{N-2}(\lambda_2 - \lambda_1) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \lambda_N - \lambda_1 & \lambda_N(\lambda_N - \lambda_1) & \dots & \lambda_N^{N-2}(\lambda_N - \lambda_1) \end{vmatrix} \quad (\text{G.5})$$

By using Laplace Expansion formula for determinants, along the first row we find that $\det(V') = \det(V'')$ where V'' is:

$$\det(V'') = \begin{vmatrix} \lambda_2 - \lambda_1 & \dots & \dots & \lambda_2^{N-2}(\lambda_2 - \lambda_1) \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_N - \lambda_1 & \dots & \dots & \lambda_N^{N-2}(\lambda_N - \lambda_1) \end{vmatrix} \quad (\text{G.6})$$

Taking the factors common in each row, we get:

$$\det(V) = \det(V'') = (\lambda_2 - \lambda_1) \cdots (\lambda_N - \lambda_1) \begin{vmatrix} 1 & \cdots & \cdots & \lambda_2^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & \cdots & \lambda_N^{N-2} \end{vmatrix} \quad (\text{G.7})$$

If we iterate this with this smaller matrix and carry on, we will see that we will get:

$$\det(V) = \prod_{i < j} (\lambda_j - \lambda_i). \quad (\text{G.8})$$

To define a Vandermonde matrix and compute determinant, we can execute following command in MATHEMATICA :

```
V = Table[Subscript[\[Alpha], i]^j, {i, 1, 5}, {j, 0, 4}];
Det@V // Simplify
```

★ Solution to Exercise ...:

We consider $X_\mu \rightarrow (1 + \epsilon)X_\mu + \mathcal{O}(\epsilon^2)$ and $\mathcal{D}X \rightarrow (1 + \epsilon D(N^2 - 1))\mathcal{D}X$ with the path integral:

$$Z = \int \mathcal{D}X e^{-S} = \int \mathcal{D}X \exp \left[-\frac{1}{4g^2} \text{Tr}[X_\mu, X_\nu]^2 \right] \quad (\text{G.9})$$

The transformation changes Z by:

$$Z = Z + \epsilon \{ D(N^2 - 1)Z - 4\langle S \rangle Z \} = Z(1 + \epsilon \{ D(N^2 - 1) - 4\langle S \rangle \}) \quad (\text{G.10})$$

If we demand that Z remains invariant, the term in the paranthesis should vanish and we get the desired result:

$$\frac{D}{4} = \frac{\langle S \rangle}{N^2 - 1} \quad (\text{G.11})$$

★ Executing following command in MATHEMATICA will check that Wigner distribution is observed. The deviation from the semi-circle distribution can be seen for small n .⁹

```
n = 1000;
scaledSpectrum=Flatten[RandomVariate[scaledSpectrum\[ScriptCapitalD][n], 100]];
Show[Histogram[scaledSpectrum, {0.05}, PDF, ChartStyle -> LightOrange],
Plot[PDF[WignerSemicircleDistribution[1], x], {x, -1.5, 1.5}, PlotLegends -> None,
```

⁹Please see <https://www.wolfram.com/language/11/random-matrices> for more

`PlotStyle -> ColorData[27, 1]], ImageSize -> Medium]`

★ Solution to Exercise **XXX**:

Consider (3.29) with $k = 1$,

$$\left\langle \text{Tr} \left(M^2 + g M^4 \right) \right\rangle = 1 \quad (\text{G.12})$$

This then implies,

$$\text{Tr} M^4 \equiv t_4 = \frac{1 - \text{Tr} M^2}{g} \equiv \frac{1 - t_2}{g} \quad (\text{G.13})$$

We can extend this to $\text{Tr} M^6 \equiv t_6$ which can be obtained in terms of t_2 as:

$$t_6 = \frac{2t_2 - \frac{(1-t_2)}{g}}{g}. \quad (\text{G.14})$$

★ Solution to Exercise 5:

We now show that $\langle e^{-\Delta H} \rangle = 1$ The Hamiltonian of the system is defined as:

$$H(P, X) = \frac{1}{2} P^2 + S(X) \quad (\text{G.15})$$

Areas of phase space is preserved i.e.,

$$dP dX = dP' dX' \quad (\text{G.16})$$

Now consider the partition function:

$$\begin{aligned} Z &= \int dP' dX' e^{-H'} \\ &= \int dP dX e^{-H} e^{H-H'} \end{aligned} \quad (\text{G.17})$$

Dividing (G.17) by Z we get,

$$\langle e^{H-H'} \rangle = \langle e^{-\Delta H} \rangle = 1 \quad (\text{G.18})$$

★ Solution to Exercise **xxx**:

Here we give the code to compute errors using Jackknife method.

Last edited by RGJ: 2021-09-28 at 16:00:03

```

#!/usr/bin/python
# -*- coding: utf-8 -*-
#!/usr/bin/python3

import sys
import itertools
from math import *
data = []; data_tot = 0. ; Data = [] ; data_jack = []

if len( sys.argv ) > 2:
    filename = sys.argv[1]
    therm_cut = int(sys.argv[2])
    blocksize = int(sys.argv[3])
if len( sys.argv ) <= 2:
    print("NEED 3 ARGUMENTS : filename therm-cut blocksize ")
    sys.exit()

file = open(filename, "r")
for line in itertools.islice(file, therm_cut, None):

    line = line.split()
    data_i = float(line[0])
    data.append(data_i)
    data_tot += data_i
    n = len(data)

n_b = int(n/blocksize)
B = 0.

for k in range(n_b):
    for w in range((k*blocksize)+1,(k*blocksize)+blocksize+1):
        B += data[w-1]
    Data.insert(k,B)
    B = 0

''' Do the jackknife error estimates '''

for i in range(n_b-1):
    data_jack.append((data_tot - Data[i]) / (n - blocksize))
    data_av = data_tot / n # Do the overall averages
    data_av = data_av
    data_jack_av = 0.; data_jack_err = 0.
for i in range(n_b-1):
    dR = data_jack[i]
    data_jack_av += dR
    data_jack_err += dR**2

data_jack_av /= n_b-1
data_jack_err /= n_b-1

```

```
data_jack_err = sqrt((n_b - 2) * abs(data_jack_err - data_jack_av**2))
print(" %8.7f " " %6.7f" " %6.2f" % (data_jack_av, data_jack_err, n_b))
```

References

- [1] E. Wigner, “On the statistical distribution of the widths and spacings of nuclear resonance levels,” 1951.
- [2] M. Mehta, “On the statistical properties of the level-spacings in nuclear spectra,” *Nuclear Physics* **18** (1960) 395–419.
<https://www.sciencedirect.com/science/article/pii/0029558260904132>.
- [3] M. Gaudin, “Sur la loi limite de l’espacement des valeurs propres d’une matrice aléatoire,” *Nuclear Physics* **25** (1961) 447–458.
<https://www.sciencedirect.com/science/article/pii/0029558261901766>.
- [4] M. L. Mehta, *Random Matrices*. 3rd ed., 2004.
- [5] G. Akemann, J. Baik, and P. Di Francesco, *The Oxford Handbook of Random Matrix Theory*. Oxford Handbooks in Mathematics. Oxford University Press, 9, 2011.
- [6] R. U. Haq, A. Pandey, and O. Bohigas, “Fluctuation properties of nuclear energy levels: Do theory and experiment agree?,” *Phys. Rev. Lett.* **48** (Apr, 1982) 1086–1089.
<https://link.aps.org/doi/10.1103/PhysRevLett.48.1086>.
- [7] P. Di Francesco, P. H. Ginsparg, and J. Zinn-Justin, “2-D Gravity and random matrices,” *Phys. Rept.* **254** (1995) 1–133, [arXiv:hep-th/9306153](https://arxiv.org/abs/hep-th/9306153).
- [8] B. Eynard, T. Kimura, and S. Ribault, “Random matrices,” [arXiv:1510.04430](https://arxiv.org/abs/1510.04430) [math-ph].
- [9] G. ’t Hooft, “A Planar Diagram Theory for Strong Interactions,” *Nucl. Phys. B* **72** (1974) 461.
- [10] E. Brezin and S. R. Wadia, eds., *The Large N expansion in quantum field theory and statistical physics: From spin systems to two-dimensional gravity*. 1994.
- [11] Y. Y. Goldschmidt, “1/N Expansion in Two-dimensional Lattice Gauge Theory,” *J. Math. Phys.* **21** (1980) 1842.
- [12] M. Marino, “Les Houches lectures on matrix models and topological strings,” 10, 2004.
[arXiv:hep-th/0410165](https://arxiv.org/abs/hep-th/0410165).
- [13] E. Brezin, C. Itzykson, G. Parisi, and J. B. Zuber, “Planar Diagrams,” *Commun. Math. Phys.* **59** (1978) 35.
- [14] A. B. Balantekin, “Character expansions, Itzykson-Zuber integrals, and the QCD partition function,” *Phys. Rev. D* **62** (2000) 085017, [arXiv:hep-th/0007161](https://arxiv.org/abs/hep-th/0007161).
- [15] S. Chadha, G. Mahoux, and M. L. Mehta, “A Method of Integration Over Matrix Variables. 2,” *J. Phys. A* **14** (1981) 579.
- [16] P. D. Anderson and M. Kruczenski, “Loop Equations and bootstrap methods in the lattice,” *Nucl. Phys. B* **921** (2017) 702–726, [arXiv:1612.08140](https://arxiv.org/abs/1612.08140) [hep-th].

- [17] H. W. Lin, “Bootstraps to strings: solving random matrix models with positivite,” *JHEP* **06** (2020) 090, [arXiv:2002.08387 \[hep-th\]](#).
- [18] X. Han, S. A. Hartnoll, and J. Kruthoff, “Bootstrapping Matrix Quantum Mechanics,” *Phys. Rev. Lett.* **125** no. 4, (2020) 041601, [arXiv:2004.10212 \[hep-th\]](#).
- [19] V. Kazakov and Z. Zheng, “Analytic and Numerical Bootstrap for One-Matrix Model and ”Unsolvable” Two-Matrix Model,” [arXiv:2108.04830 \[hep-th\]](#).
- [20] M. Hanada, “Markov Chain Monte Carlo for Dummies,” [arXiv:1808.08490 \[hep-th\]](#).
- [21] V. A. Kazakov, I. K. Kostov, and N. A. Nekrasov, “D particles, matrix integrals and KP hierarchy,” *Nucl. Phys. B* **557** (1999) 413–442, [arXiv:hep-th/9810035](#).
- [22] M. Staudacher, “The Yang-lee Edge Singularity on a Dynamical Planar Random Surface,” *Nucl. Phys. B* **336** (1990) 349.
- [23] V. A. Kazakov and A. A. Migdal, “Recent Progress in the Theory of Noncritical Strings,” *Nucl. Phys. B* **311** (1988) 171.
- [24] I. R. Klebanov, “String theory in two-dimensions,” [arXiv:hep-th/9108019](#).
- [25] T. Banks, W. Fischler, S. H. Shenker, and L. Susskind, “M theory as a matrix model: A Conjecture,” *Phys. Rev. D* **55** (1997) 5112–5128, [arXiv:hep-th/9610043](#).
- [26] N. Ishibashi, H. Kawai, Y. Kitazawa, and A. Tsuchiya, “A Large N reduced model as superstring,” *Nucl. Phys. B* **498** (1997) 467–491, [arXiv:hep-th/9612115](#).
- [27] T. Hotta, J. Nishimura, and A. Tsuchiya, “Dynamical aspects of large N reduced models,” *Nucl. Phys. B* **545** (1999) 543–575, [arXiv:hep-th/9811220](#).
- [28] F. R. Klinkhamer, “A first look at the bosonic master-field equation of the IIB matrix model,” [arXiv:2105.05831 \[hep-th\]](#).
- [29] D. E. Berenstein, J. M. Maldacena, and H. S. Nastase, “Strings in flat space and pp waves from $\mathcal{N} = 4$ super-Yang–Mills,” *JHEP* **0204** (2002) 013, [hep-th/0202021](#).
- [30] K. Furuuchi, E. Schreiber, and G. W. Semenoff, “Five-brane thermodynamics from the matrix model,” [arXiv:hep-th/0310286](#).
- [31] G. Mandal, M. Mahato, and T. Morita, “Phases of one dimensional large N gauge theory in a 1/D expansion,” *JHEP* **02** (2010) 034, [arXiv:0910.4526 \[hep-th\]](#).
- [32] G. Bergner, N. Bodendorfer, M. Hanada, E. Rinaldi, A. Schäfer, and P. Vranas, “Thermal phase transition in Yang-Mills matrix model,” *JHEP* **01** (2020) 053, [arXiv:1909.04592 \[hep-th\]](#).
- [33] V. G. Filev and D. O’Connor, “The BFSS model on the lattice,” *JHEP* **05** (2016) 167, [arXiv:1506.01366 \[hep-th\]](#).
- [34] N. Kawahara, J. Nishimura, and S. Takeuchi, “High temperature expansion in supersymmetric matrix quantum mechanics,” *JHEP* **12** (2007) 103, [arXiv:0710.2188 \[hep-th\]](#).

- [35] P. Di Francesco and C. Itzykson, “A Generating function for fatgraphs,” *Ann. Inst. H. Poincaré Phys. Theor.* **59** (1993) 117–140, [arXiv:hep-th/9212108](#).
- [36] V. A. Kazakov, M. Staudacher, and T. Wynter, “Character expansion methods for matrix models of dually weighted graphs,” *Commun. Math. Phys.* **177** (1996) 451–468, [arXiv:hep-th/9502132](#).
- [37] J. Ambjorn, J. Jurkiewicz, R. Loll, and G. Vernizzi, “Lorentzian 3-D gravity with wormholes via matrix models,” *JHEP* **09** (2001) 022, [arXiv:hep-th/0106082](#).
- [38] V. A. Kazakov, “Solvable matrix models,” 2, 2000. [arXiv:hep-th/0003064](#).