



Bachelorarbeit

Die Universalitätsklassen AI^\dagger und AII^\dagger in nicht-Hermitescher Zufallsmatrixtheorie

Autor:	Lippl, Jonas
Matrikelnummer:	2101758
geboren am:	02.01.1999
Studiengang:	Physik B.Sc.
Betreuer:	Prof. Dr. Tilo Wettig
Abgabedatum:	01.09.2021

Zusammenfassung

Nach einer kurzen allgemeinen Einführung in die Zufallsmatrixtheorie werden zwei Methoden vorgestellt, durch die sich nicht-Hermiteschen Zufallsmatrizen in drei Universalitätsklassen einteilen lassen. Diese sind die Untersuchung der Abstandsverhältnisse von nächstem Nachbarn zu übernächstem Nachbarn und die Abstandsverteilung nächster Nachbarn. Durch Berechnungen in Python kann man erkennen, dass sich das Ginibre-, das AI^\dagger - und das AII^\dagger -Ensemble in diesen Statistiken unterscheiden. Eine wichtige Kenngröße zur Beschreibung dieser Universalitätsklassen ist die Zufallsverteilung der Eigenwerte. Diese ist allerdings nur für das Ginibre-Ensemble bekannt. Daher wurde versucht, die Zufallsverteilung der Eigenwerte für das AI^\dagger - und das AII^\dagger -Ensemble herzuleiten. Dies gelang leider nicht, jedoch wurden alle untersuchten Ansätze dargelegt und insbesondere die dabei auftretenden Probleme herausgearbeitet. Somit kann diese Arbeit als Grundlage dienen, um in Zukunft neue Versuche zu unternehmen, diese Zufallsverteilung herzuleiten.

Inhaltsverzeichnis

1	Einleitung	1
2	Einführung in die Zufallsmatrixtheorie	3
3	Nicht-Hermitesche Zufallsmatrixtheorie	7
3.1	Abstandsverhältnisse	7
3.2	Abstandsverteilung nächster Nachbarn	12
3.3	Universalitätsklassen	13
3.4	Zufallsverteilung der Eigenwerte des Ginibre-Ensembles	15
3.5	AI^\dagger -Ensemble	20
3.5.1	Schur-Zerlegung	20
3.5.2	Diagonalisieren durch orthogonale Matrizen	21
3.5.3	Takagi-Faktorisierung	23
3.6	AII^\dagger -Ensemble	24
3.6.1	Schur-Zerlegung	26
3.6.2	Eigenwertzerlegung	27
4	Fazit	29
	Literatur	32
	Anhang	33
A	Abstandsverteilung nächster Nachbarn	33
B	Komplexe Abstandsverhältnisse des Ginibre-Ensembles	35
C	Komplexe Abstandsverhältnisse des AI^\dagger-Ensembles	37
D	Komplexe Abstandsverhältnisse des AII^\dagger-Ensembles	38
E	Vergleich der komplexen Abstandsverhältnisse der Universalitätsklassen	40
F	Schur-Zerlegung des AI^\dagger-Ensembles	41
G	Schur-Zerlegung des AII^\dagger-Ensembles	42

1 Einleitung

Eine Zufallsmatrix ist, wie der Name schon sagt, eine Matrix, deren Einträge zufällig sind. Diese Einträge können zum Beispiel aus einer Gauß-Verteilung stammen. Anschließend untersucht man die Wahrscheinlichkeitsverteilung der Eigenwerte oder Eigenvektoren, um Rückschlüsse auf das System zu ziehen. Anwendung findet diese Technik in vielen Bereichen der Physik, aber auch in anderen Branchen wie der Finanzwirtschaft oder der Entwicklung neuronaler Netze. In der Kernphysik wurde damit zunächst von Eugene P. Wigner das Spektrum schwerer Atome untersucht. Wie sich herausstellte, kann man die Energieniveaus durch Zufallsmatrixtheorie modellieren [18]. Im Laufe der Jahre wurde die Zufallsmatrixtheorie auch in anderen Bereichen der Mathematik und Physik verwendet, wie etwa der String Theorie, der Untersuchung chaotischer Systeme [16] oder wie hier an der Universität Regensburg zur Untersuchung nicht-Hermitescher Dirac-Operatoren in QCD-artigen Theorien [11].

Die Theorie hermitescher Zufallsmatrizen ist bereits ausführlich durch Dyson, Wigner und viele weitere untersucht. Im Bereich nicht-Hermitescher Zufallsmatrizen gibt es allerdings noch einige offene Fragen. So wurde von Hamazaki et. al. [8] im Jahr 2019 herausgefunden, dass es neben dem Ginibre-Ensemble genau zwei weitere Universalitätsklassen gibt, die sich bezüglich ihrer Abstandsverteilung nächster Nachbarn und ihrer Abstandsverhältnisse von der des Ginibre-Ensembles unterscheiden. Dies sind die Universalitätsklassen AI^\dagger und AII^\dagger . Wie bereits erwähnt, ist die Zufallsverteilung der Eigenwerte von großer Bedeutung bei der Untersuchung von Zufallsmatrizen. Für das Ginibre-Ensemble wurde diese von Ginibre [16] (15.1.10) und Dyson [16] (A.33) hergeleitet. Die Herleitung nach Dyson wird in dieser Arbeit mit vielen ausführlichen Nebenrechnungen nachvollzogen. Für die beiden anderen Universalitätsklassen AI^\dagger und AII^\dagger ist die Zufallsverteilung der Eigenwerte hingegen noch unbekannt. Deshalb wird versucht, die beiden Verteilungen herzuleiten. Dazu werden alle durchgeführten Ansätze und die dabei auftretenden Probleme ausführlich dargelegt, um für zukünftige Herleitungsversuche die Probleme vorab klarzustellen.

Zur Notation sei angemerkt, dass diese in der Literatur oft unterschiedlich verwendet wird. Ist H eine Matrix, dann bezeichnet H^* die komplex konjugierte, H^T die transponierte und H^\dagger die adjungierte Matrix.

Da der Großteil der Fachliteratur englischsprachig ist, werden von vielen Fachbegriffen die englischen Entsprechungen in Klammern angegeben.

Für viele kleinere Berechnungen befinden sich im Anhang A-G Python-Codes, die auch auf <https://github.com/JonasLippl/Bachelorarbeit> abgerufen werden können.

2 Einführung in die Zufallsmatrixtheorie

Zum besseren Verständnis des Themas soll zu Beginn dieser Arbeit eine kurze Einführung in die Zufallsmatrixtheorie erfolgen. Man kann zunächst eine beliebige $N \times N$ -Matrix mit reellen Zufallswerten, beispielsweise aus einer Gaußschen Wahrscheinlichkeitsverteilung füllen. Berechnet man nun die Eigenwerte, erhält man im Allgemeinen komplexe Zahlen. Um reelle Eigenwerte zu erhalten, muss die Matrix symmetrisiert werden. Dazu wird wie folgt vorgegangen [15].

$$H_s = \frac{(H + H^T)}{2} \quad (1)$$

Matrizen mit Einträgen aus einer bestimmten Zufallsverteilung werden Beispiele eines Ensembles genannt. Die oben beschriebene Matrix mit symmetrischen Einträgen aus einer Gaußschen Wahrscheinlichkeitsverteilung gehört zum Gaußschen Orthogonalen Ensemble (GOE). Zwei Matrizen, die zueinander ähnlich sind, also $H' = U H U^{-1}$, treten in einem Ensemble mit der gleichen Wahrscheinlichkeit

$$\rho(H') dH'_{11}, \dots, dH'_{NN} = \rho(H) dH_{11}, \dots, dH_{NN} \quad (2)$$

auf. Der Name Gaußsches Orthogonales Ensemble ist so gewählt, weil für diese Matrizen die Transformationsmatrix U orthogonal ist.

Das Gaußsche Unitäre Ensemble (GUE) erhält man, indem man die Matrix mit komplexen Zahlen aus einer Gauß-Verteilung füllt und fordert, dass die Matrizen hermitesch sind. Somit gilt $H_u = H_u^\dagger$. Für diese Matrizen ist, wie der Name schon vermuten lässt, die Transformationsmatrix U unitär.

Zufallsmatrizen mit quaternionischen Einträgen müssen selbst-dual sein, damit die Eigenwerte reell sind. Das bedeutet, dass diese $2N \times 2N$ Matrizen die Form

$$H_{symp} = \frac{1}{2} \cdot \left[\begin{pmatrix} X & Y \\ -Y^* & X^* \end{pmatrix} + \begin{pmatrix} X & -Y^* \\ Y & X^* \end{pmatrix} \right] \quad (3)$$

haben müssen, wobei X und Y Matrizen mit komplexen Einträgen sind. Dieses Ensemble nennt man Gaußsches Symplektisches Ensemble (GSE), da U symplektisch ist [15].

Die Einträge der Zufallsmatrizen sind im Falle der Gaußschen Ensembles reelle, komplexe oder quaternionische Zufallszahlen aus einer Gauß-Verteilung. Diese lautet für einen Erwartungswert μ und eine Varianz σ

$$f(H_{ij}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(H_{ij} - \mu)^2}{2\sigma^2}\right). \quad (4)$$

Da die einzelnen Einträge unabhängig voneinander sind, ist die gemeinsame Wahrscheinlichkeitsdichtefunktion (jpdf = joint probability density function) ein Produkt der einzelnen Wahrscheinlichkeitsdichtefunktionen (pdf = probability density

function). Somit ist die jpdf für eine reelle Matrix aus dem Gaußschen Ensemble mit $\mu = 0$ und $\sigma = 1$ ohne spezielle Symmetrie

$$\rho(H_{11}, \dots, H_{NN}) = \prod_{i,j=1}^N \left[\exp(-H_{ij}^2/2) / \sqrt{2\pi} \right] \quad (5)$$

und für das GOE Ensemble

$$\rho((H_s)_{11}, \dots, (H_s)_{NN}) = \prod_{i=1}^N \left[\exp(-(H_s)_{ii}^2/2) / \sqrt{2\pi} \right] \prod_{i < j}^N \left[\exp(-(H_s)_{ij}^2/2) / \sqrt{\pi} \right], \quad (6)$$

da hier die Elemente außerhalb der Hauptdiagonale durch die Symmetrie zweifach auftreten und die Varianz dieser Einträge damit $1/2$ beträgt. Alternativ kann die Zufallsverteilung umgeschrieben werden, indem man gewisse Eigenschaften der Spur verwendet. Die Spur einer Matrix ist die Summe ihrer Diagonalelemente. Für die Spur des Quadrats einer $N \times N$ -Matrix gilt mit der Definition der Matrizenmultiplikation

$$Sp(H^2) = \sum_i^N (H^2)_{ii} = \sum_{1 \leq i, j \leq N} H_{ij} H_{ji}. \quad (7)$$

Durch die Symmetrie $H_{ij} = H_{ji}$ beim GOE vereinfacht sich dieser Ausdruck zu

$$Sp(H_s^2) = \sum_{1 \leq i, j \leq N} (H_{ij})^2 \quad (8)$$

$$= \sum_i^N (H_{ii})^2 + \sum_{1 \leq i < j \leq N} (H_{ij})^2 + \sum_{1 \leq j < i \leq N} (H_{ij})^2 \quad (9)$$

$$= \sum_i^N (H_{ii})^2 + 2 \cdot \sum_{1 \leq i < j \leq N} (H_{ij})^2. \quad (10)$$

Beim Vergleich mit (6), sieht man, dass sich die Zufallsverteilung für das GOE und auch für GUE und GSE schreiben lässt als

$$\rho(H_{11}, \dots, H_{NN}) = \frac{1}{Z_{N,\beta}} e^{-\frac{\beta}{2} Sp(H^2)}. \quad (11)$$

β ist der sogenannte Dyson-Index. Für das GOE ist $\beta = 1$, für das GUE ist $\beta = 2$ und für das GSE ist $\beta = 4$. $Z_{N,\beta}$ ist eine Normalisierungskonstante, die dafür sorgt, dass

$$\int_{H \in G\beta E(N)} \frac{1}{Z_{N,\beta}} e^{-\frac{\beta}{2} Sp(H^2)} = 1. \quad (12)$$

Diese ist

$$Z_{N,\beta} = (2\pi)^{\frac{N}{2}} \prod_{j=1}^N \frac{\Gamma(1 + j\beta/2)}{\Gamma(1 + \beta/2)}. \quad (13)$$

Ähnlich wie hier für die Gaußschen Ensembles kann später auch die jpdf für das AI^\dagger -Ensemble und das AII^\dagger -Ensemble aufgestellt werden [15].

3 Nicht-Hermitesche Zufallsmatrixtheorie

Bisher wurden nur hermitesche Matrizen betrachtet, welche ausschließlich reelle Eigenwerte haben. Für viele Probleme reichen diese Matrizen jedoch nicht aus. Zur Untersuchung der spektralen Eigenschaften des Fokker-Planck Operators [2], zur Klassifikation nicht-Hermitescher Dirac-Operatoren in QCD-artigen Theorien [11] oder zur Beurteilung dissipativer Quantensysteme [6] sind nicht-Hermitesche Zufallsmatrizen notwendig. Für nicht-Hermitesche Matrizen muss

$$H^* \neq H^T \quad (14)$$

gelten. Damit sind die Eigenwerte im Allgemeinen komplexe Zahlen.

3.1 Abstandsverhältnisse

Eine wichtige Kenngröße zur Klassifikation von Zufallsmatrix Ensembles sind die Abstandsverhältnisse der Eigenwerte einer Matrix. Damit sind die Verhältnisse aus Abstand des nächsten Nachbarn zum Abstand des übernächsten Nachbarn gemeint. In der Physik wird dies zum Beispiel genutzt, um chaotische von regulären Quantenspektren zu unterscheiden. Für komplexe Eigenwerte $\{\lambda_k\}_{k=1}^N$ berechnet man die Abstandsverhältnisse wie folgt [17].

$$z_k = \frac{\lambda_k^{NN} - \lambda_k}{\lambda_k^{NNN} - \lambda_k} \quad (15)$$

λ_k^{NN} : nächster Nachbar von λ_k

λ_k^{NNN} : übernächster Nachbar von λ_k

Wie sich zeigen wird, unterscheiden sich die Zufallsverteilungen der Abstandsverhältnisse von nicht-Hermiteschen Zufallsmatrizen nur in drei Fällen. Deshalb können diese nicht-Hermiteschen Ensembles in drei Universalitätsklassen unterteilt werden. Zur Berechnung der Abstandsverhältnisse wird der Python-Code in B, C und D verwendet. Für komplexe Eigenwerte sind die Abstandsverhältnisse per Definition komplexe Zahlen $z = re^{i\theta}$ innerhalb des Einheitskreises. Der Vorteil dieser Abstandsverhältnisse ist, dass sie unabhängig von der Größe der Eigenwerte sind. Sie können gut in einer sogenannten Heatmap dargestellt werden. Wie man in den Heatmaps in Abb. 1 erkennen kann, ist die Wahrscheinlichkeitsverteilung $\varrho^{(N)}(z)$ stark anisotrop. Um den Ursprung herum geht $\varrho^{(N)}(z)$ gegen Null, was an der sogenannten Niveauabstoßung (Englisch: level repulsion) liegt. Das bedeutet, dass die Wahrscheinlichkeit dafür, dass zwei Eigenwerte sehr nahe beisammen liegen, gegen Null geht. An (15) erkennt man auch gut, dass $z_k \rightarrow 0$ für $\lambda_k^{NN} \rightarrow \lambda_k$.

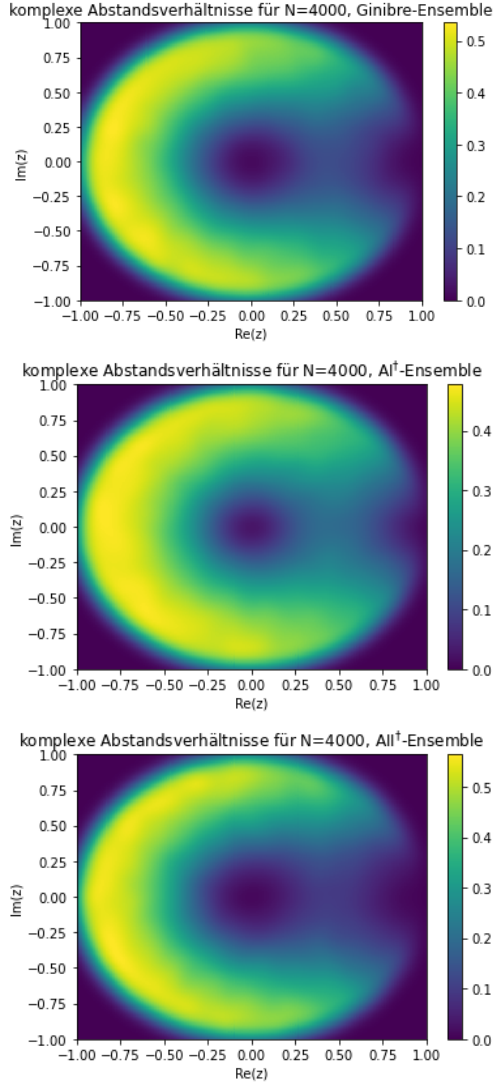


Abb. 1: Komplexe Abstandsverhältnisse des Ginibre-, AI^\dagger - und AII^\dagger -Ensembles berechnet durch 100 4000×4000 -Matrizen.

deutlich von der Matrixgröße ab. Für kleine N dominieren die Eigenwerte am Rande des Spektrums, für große N überwiegen jedoch die Eigenwerte im Inneren des Spektrums [17]. Dies sieht man deutlich, wenn man wieder die Winkel- (Abb. 5) und Radialanteile (Abb. 4) der Abstandsverhältnisse von kleinen 10×10 -Matrizen betrachtet.

In [17] (APPENDIX C) wird gezeigt, wie die Zufallsverteilung der Abstandsver-

Wir werden später bei der Herleitung der Zufallsverteilung der Eigenwerte erkennen, dass die Niveauabstoßung aus der Vandermonde-Determinante resultiert. Eine weitere Konsequenz der Niveauabstoßung ist, dass die nächsten Nachbarn eines Eigenwerts gleichmäßig um diesen herum verteilt sind. Dadurch werden kleine Winkel Θ eher unwahrscheinlich, da sie nur vorkommen würden, wenn λ_k^{NN} und λ_k^{NNN} in derselben Richtung von λ_k aus liegen.

Anhand der Heatmaps, die mit dem Code E generiert werden, kann der Unterschied zwischen den drei Ensembles nur schwer festgestellt werden. Um dies zu gewährleisten, werden die Verteilungen in ihre Radialanteile (16) (Abb. 2) und ihre Winkelanteile (17) (Abb. 3) aufgeteilt.

$$\varrho(r) = \int d\Theta \, r \, \varrho(r, \Theta) \quad (16)$$

$$\varrho(\Theta) = \int dr \, r \, \varrho(r, \Theta) \quad (17)$$

Abschließend gilt es noch Randeffekte zu berücksichtigen. Für Eigenwerte, die am Rand des Spektrums liegen, ist die Wahrscheinlichkeit für kleine Θ wesentlich höher als für Eigenwerte, die sich zentral im Spektrum befinden. Dies liegt daran, dass der nächste und übernächste Nachbar von Eigenwerten am Rand nur in Richtung des Inneren des Spektrums liegen können. Somit hängt besonders die Heatmap und $\varrho(\Theta)$

hältnisse $\varrho^{(N)}(z)$ aus der Zufallsverteilung der Eigenwerte $P^{(N)}(\lambda_1, \dots, \lambda_N)$ berechnet werden kann. Zunächst wird ein Eigenwert $\lambda_1 = x_1 + iy_1$ gewählt, sein nächster Nachbar als $\lambda_2 = x_2 + iy_2$ und sein übernächster Nachbar als $\lambda_3 = x_3 + iy_3$ bezeichnet. Damit lautet das Abstandsverhältnis

$$z = x + iy = \frac{(x_2 - x_1)(x_3 - x_1) + (y_2 - y_1)(y_3 - y_1)}{(x_3 - x_1)^2 + (y_3 - y_1)^2} + i \frac{(x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1)}{(x_3 - x_1)^2 + (y_3 - y_1)^2}. \quad (18)$$

Nun werden die neuen Variablen $u = x_1$, $v = y_1$, $p = x_2 - x_1$, $q = y_2 - y_1$, $s = x_3 - x_1$, $t = y_3 - y_1$, $a_n = x_{n+3} - x_1$, und $b_n = y_{n+3} - y_1$ mit $n = 1, \dots, N-3$ eingeführt. Die Delta-Distribution sorgt dafür, dass der Real- und Imaginärteil in (18) dem Real- und Imaginärteil von z entspricht.

$$\delta\left(x - \frac{ps + qt}{s^2 + t^2}\right) \delta\left(y - \frac{sq - pt}{s^2 + t^2}\right) = (s^2 + t^2) \delta(p - (sx - ty)) \delta(q - (tx + sy)) \quad (19)$$

Die Heaviside-Funktion

$$\Theta((s^2 + t^2) - (p^2 + q^2)) \prod_{j=4}^N \Theta((a_j^2 + b_j^2) - (s^2 + t^2)) \quad (20)$$

stellt sicher, dass alle andern Eigenwerte λ_n weiter von λ_1 entfernt sind als λ_3 . Damit kann nun die Zufallsverteilung $\varrho^{(N)}(z)$ berechnet werden, indem man die Zufallsverteilung der Eigenwerte mit (19) und (20) multipliziert und über die neu eingeführten Variablen integriert. Nach Integration über p und q erhält man

$$\begin{aligned} \varrho^{(N)}(x, y) = & \Theta(1 - (x^2 + y^2)) \int du dv ds dt \prod_{j=1}^{N-3} da_j db_j \Theta((a_j^2 + b_j^2) - (s^2 + t^2)) \cdot \\ & \cdot (s^2 + t^2) \times P^{(N)}(u, u + sx - ty, u + s, u + a_1, \dots, u + a_{n-3}, v, \\ & v + tx + sy, v + t, v + b_1, \dots, v + b_{N-3}). \end{aligned} \quad (21)$$

Dieses Integral für größere Matrizen zu lösen, ist jedoch äußerst kompliziert.

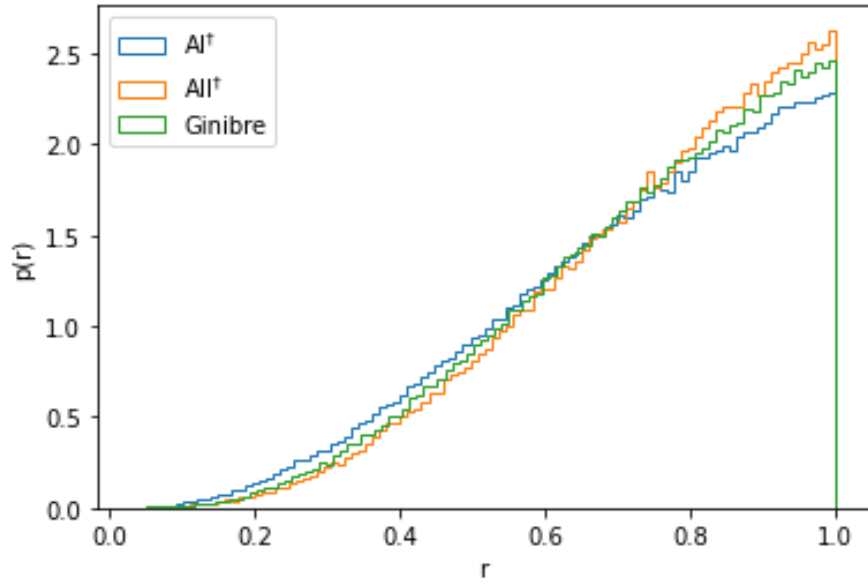


Abb. 2: Radialanteile der Abstandsverhältnisse berechnet durch 100 4000×4000 -Matrizen

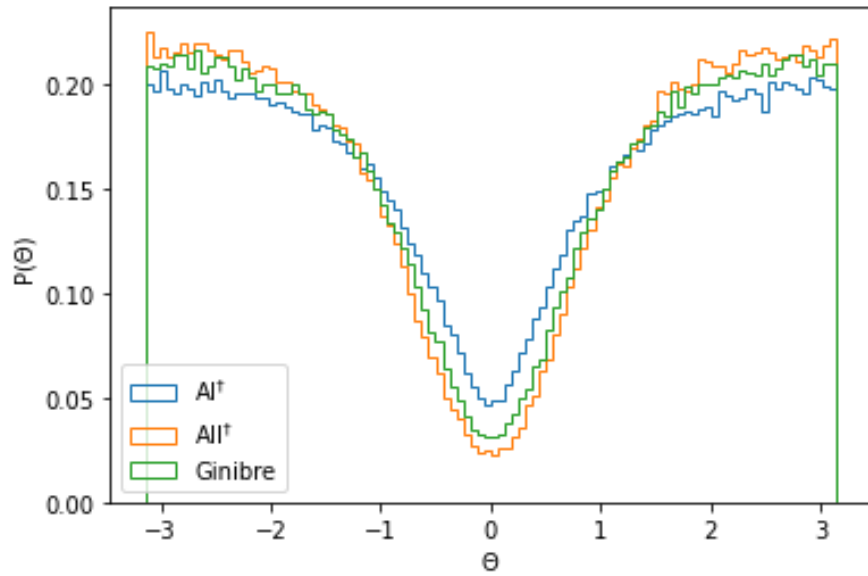


Abb. 3: Winkelanteile der Abstandsverhältnisse berechnet durch 100 4000×4000 -Matrizen

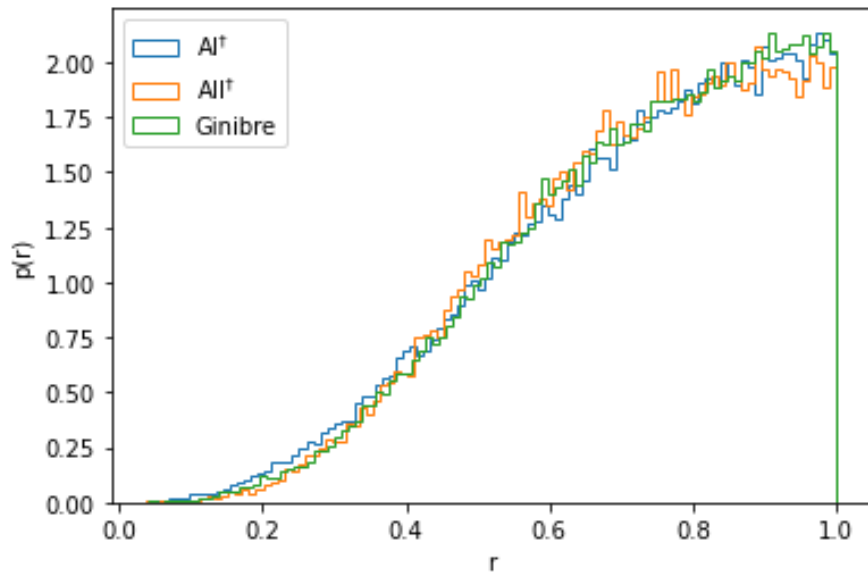


Abb. 4: Radialanteile der Abstandsverhältnisse berechnet durch 10000 10×10 -Matrizen

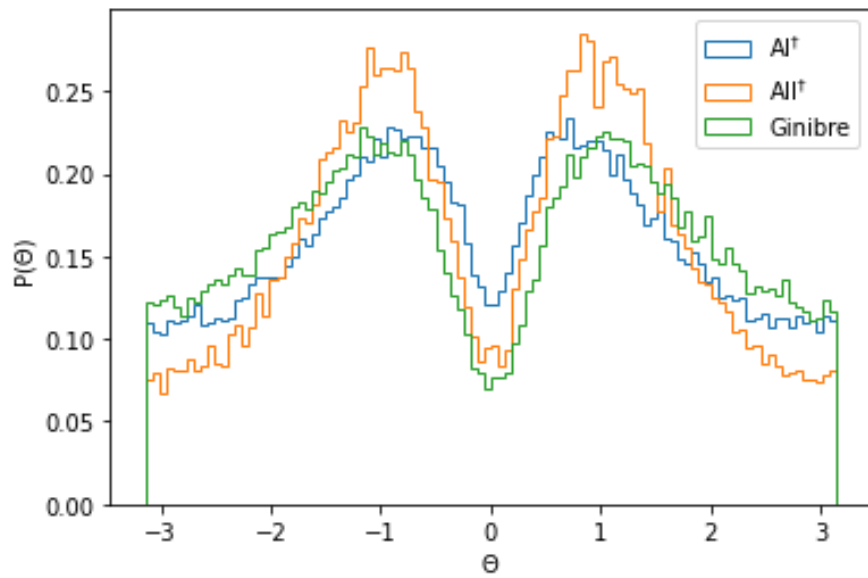


Abb. 5: Winkelanteile der Abstandsverhältnisse berechnet durch 10000 10×10 -Matrizen

3.2 Abstandsverteilung nächster Nachbarn

Eine weitere Möglichkeit, die Zufallsmatrizen in verschiedene Universalitätsklassen zu unterteilen, ist nach [8] die Abstandsverteilung der nächsten Nachbarn zu untersuchen. Dazu berechnet man zunächst den minimalen Abstand eines jeden Eigenwertes λ_α zu den anderen Eigenwerten λ_β .

$$d_{1,\alpha} = \min_\beta |\lambda_\alpha - \lambda_\beta| \quad (22)$$

Damit die Abstandsverteilung normiert ist, muss gelten

$$\int_0^\infty p(s) ds = \int_0^\infty s p(s) ds = 1. \quad (23)$$

Um dies zu gewährleisten, müssen die Abstände entfaltet werden. Dazu verwendet man

$$s_\alpha = d_{1,\alpha} \sqrt{\bar{\rho}} \quad (24)$$

mit

$$\bar{\rho} = \frac{n}{\pi d_{n,\alpha}^2}. \quad (25)$$

$\bar{\rho}$ ist die lokale mittlere Dichte der Eigenwerte. $d_{n,\alpha}$ ist der Abstand zum n -ten Nachbarn vom Eigenwert λ_α . Um genaue Ergebnisse zu erhalten sollte für eine $N \times N$ -Matrix $1 \ll n \ll M$ gelten. Bei der in Abb. 6 durchgeführten numerischen Berechnung wurde $n = 15$ gewählt. Hamazaki et. al. [8] empfehlen einen Wert von $n = 10 - 30$.

Alternativ haben Akemann et al. [1] eine weitere Möglichkeit gefunden, die Entfaltung durchzuführen. Um die mittlere Dichte der Eigenwerte zu nähern kann diese als Summe von Gauß-Verteilungen um jeden Eigenwert z_j geschrieben werden.

$$\bar{\rho}(x, y) = \frac{1}{2\pi\sigma^2 N} \sum_{j=1}^N \exp \left[\frac{-1}{2\sigma^2} |z - z_j|^2 \right] \quad (26)$$

Jeder zum Eigenwert z_i gehörige nächste Nachbarabstand wird nun mit $\sqrt{\bar{\rho}(x_i, y_i)}$ multipliziert. σ muss jetzt noch geschickt gewählt werden, um gute Ergebnisse zu erhalten. In [1] wird hierfür $\sigma = 4.5\bar{s}$ verwendet, wobei \bar{s} der Mittelwert der Nachbarabstände ist.

Wie man in Abb. 6 deutlich erkennt, ist das Maximum der Verteilung für das AI^\dagger -Ensemble niedriger und für das AII^\dagger -Ensemble höher als das Maximum des

Ginibre-Ensembles. Der analytische Ausdruck für die Abstandsverteilung nächster Nachbarn ist für das Ginibre-Ensemble durch [7] bekannt und lautet nach [1] (Gleichung 3)

$$p_{GinUE}(s) = \left(\prod_{k=1}^{\infty} \frac{\Gamma(1+k, s^2)}{k!} \right) \sum_{j=1}^{\infty} \frac{2s^{2j+1}e^{-s^2}}{\Gamma(1+j, s^2)}. \quad (27)$$

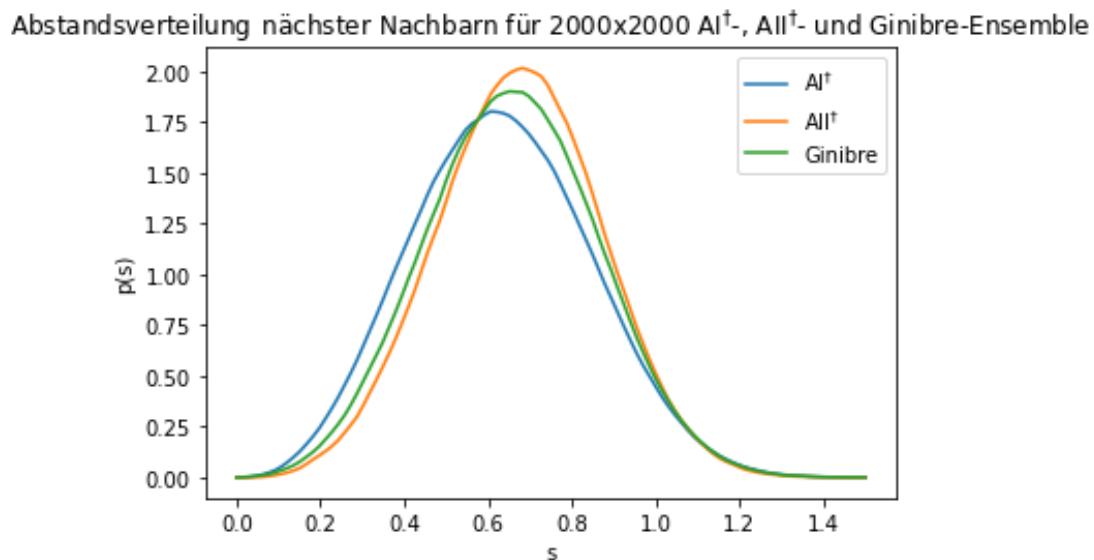


Abb. 6: Abstandverteilung nächster Nachbarn berechnet für 1000 2000×2000 -Matrizen

3.3 Universalitätsklassen

Matrizen lassen sich bezüglich ihrer Abstandsverteilung nächster Nachbarn $p(s)$ oder ihrer Abstandsverhältnisse z_k in Universalitätsklassen unterteilen [8]. In den vorherigen Abschnitten wurden diese beiden Methoden zur Unterteilung ausführlich vorgestellt. Nun soll detaillierter auf die Universalitätsklassen in hermitescher und nicht-Hermitescher Zufallsmatrixtheorie eingegangen werden.

Hermitesche Zufallsmatrizen unterteilt man bezüglich $p(s)$ in drei universelle Symmetrieklassen nach Dyson [4]. Diese nennt man A, AI und AII. In den folgenden Kapiteln wurde die Nomenklatur von [8] übernommen. Matrizen in Klasse A sind hermitesch, ohne weitere Einschränkungen, das heißt

$$H^* = H^T. \quad (28)$$

Blickt man zurück zu Kapitel 2 erkennt man, dass Matrizen aus dem GUE diese Symmetrieklasse bilden. Diese Matrizen erfüllen keine Zeitumkehrsymmetrie. Für die Klasse AI gilt zusätzlich

$$H = H^* = H^T. \quad (29)$$

Matrizen aus Klasse AI sind somit reell und symmetrisch und gehören daher zum GOE (1). Diese Matrizen erfüllen Zeitumkehrsymmetrie (time-reversal symmetry = TRS), welche definiert ist als $T H^* T^{-1} = H$. T ist hier eine unitäre Matrix deren Quadrat $T T^* = +1$ beträgt. Matrizen in Klasse AII erfüllen

$$H = \sum^y H^* \sum^y = \sum^y H^T \sum^y, \quad (30)$$

wobei

$$\sum^y = \begin{pmatrix} 0 & -i\mathbf{1}_{\frac{N}{2} \times \frac{N}{2}} \\ i\mathbf{1}_{\frac{N}{2} \times \frac{N}{2}} & 0 \end{pmatrix}. \quad (31)$$

Somit gehören die Matrizen des GSE zu dieser Symmetrieklasse. Auch hier ist die Zeitumkehrsymmetrie erfüllt, jedoch beträgt das Quadrat der unitären Matrix $T T^* = -1$.

Nicht-Hermitesche Matrizen unterteilt man analog in Symmetrieklassen nach Ginibre [5]. Hier ersetzen das GinUE, das GinOE und das GinSE die Matrixensembles GUE, GOE und GSE. Auch diese Symmetrieklassen werden mit A, AI und AII bezeichnet. Matrizen aus Klasse A (GinUE) sind nicht-Hermitesch ohne weitere Einschränkungen der Symmetrie, also

$$H^* \neq H^T. \quad (32)$$

Für die Symmetrieklasse AI (GinOE) gilt zusätzlich

$$H = H^* (\neq H^T). \quad (33)$$

Für die Klasse AII (GinSE) gilt neben der nicht-Hermitezität

$$H = \sum^y H^* \sum^y \left(\neq \sum^y H^T \sum^y \right). \quad (34)$$

Beim Vergleich der Abstandsverteilungen nächster Nachbarn fällt jedoch auf, dass diese drei Symmetrieklassen von Ginibre alle dieselbe universelle Statistik $p_{GinUE}(s)$ aufweisen.

Insgesamt gibt es laut [12] 38 Symmetrieklassen von nicht-Hermiteschen Zufallsmatrizen. Jedoch nur zwei davon weichen von der Statistik $p_{GinUE}(s)$ ab. Dies sind die Symmetrieklassen AI^\dagger und AII^\dagger . Matrizen aus AI^\dagger erfüllen

$$H = H^T (\neq H^*). \quad (35)$$

Diese Klasse enthält somit komplexe, symmetrische Matrizen. Statt der Zeitumkehrsymmetrie kann hier eine neue Symmetrie definiert werden. Für eine unitäre Matrix C gilt $C H^T C^{-1} = H$ und $C C^* = +1$. Diese Symmetrie wird mit TRS^\dagger bezeichnet.

Matrizen aus der Symmetrieklasse AII^\dagger erfüllen zusätzlich zur Bedingung der nicht-Hermitezität

$$H = \sum^y H^T \sum^y \left(\neq \sum^y H^* \sum^y \right). \quad (36)$$

Auch diese Matrizen erfüllen TRS^\dagger , jedoch ist hier $C C^* = -1$.

Die drei universellen Statistiken von A, AI^\dagger und AII^\dagger wurden in Abb. 6 berechnet. Die Ergebnisse stimmen mit den Ergebnissen aus [8] (Abb. 7) und aus [10] (FIG. 1) bis auf die Normierung überein. Abschließend sei noch angemerkt, dass auch

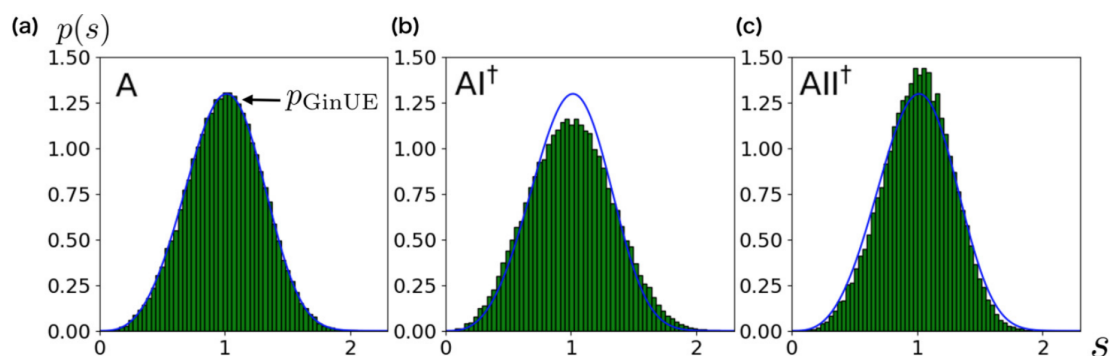


Abb. 7: Abstandsverteilung nächster Nachbarn für die drei Universalitätsklassen A, AI^\dagger und AII^\dagger aus [8].

andere Bezeichnungen für die Universalitätsklassen geläufig sind. Jaiswal et. al. [10] kennzeichnen die drei Universalitätsklassen mit dem Parameter β . Der Universalitätsklasse A entspricht $\beta = 2$, AI^\dagger entspricht $\beta = 1$ und AII^\dagger entspricht $\beta = 4$.

3.4 Zufallsverteilung der Eigenwerte des Ginibre-Ensembles

Die Universalitätsklasse A wird auch Ginibre-Ensemble genannt. Zunächst soll die bereits aus [16] (A.33) bekannte Herleitung der Zufallsverteilung der Eigenwerte aus der Zufallsverteilung der Einträge nachvollzogen werden. Diese Herleitung wurde erstmals von Ginibre durchgeführt. Die hier dargestellte, etwas leichtere Variante, geht auf Freeman John Dyson zurück. Die Zufallsverteilung der Einträge

der Matrix lautet

$$P(H) dH \propto \exp \left(-\beta \sum_{i,j} |H_{ij}|^2 \right) \prod_{i,j} dH_{ij} dH_{ij}^* \quad (37)$$

$$\propto \exp(-\beta \operatorname{Sp}(H^\dagger H)) dH, \quad (38)$$

da jeder der Werte aus einer Gauß-Verteilung stammt und alle Matrixeinträge unabhängig voneinander sind. In der unteren Zeile wurde wieder die Eigenschaft (7) genutzt:

$$\operatorname{Sp}(H^\dagger H) = \sum_i (H^\dagger H)_{ii} = \sum_{i,j} H_{ij}^* H_{ij} = \sum_{i,j} |H_{ij}|^2$$

Zunächst kann die Matrix H mit Hilfe der Schur-Zerlegung geschrieben werden als

$$H = U T U^\dagger. \quad (39)$$

Dabei ist U eine unitäre Matrix, das heißt $U U^\dagger = \mathbb{1}$ und T ist eine obere Dreiecksmatrix, also $T_{jk} = 0$ für $j > k$. Die Einträge $T_{jj} = z_j$ auf der Diagonale entsprechen den komplexen Eigenwerten der Matrix H . Zuerst soll hier eine kurze Betrachtung der Freiheitsgrade erfolgen.

Matrix	Freiheitsgrade
H	Die Matrix H hat N^2 komplexe Einträge und somit $2N^2$ reelle Parameter.
U	Zunächst hat U $2N^2$ reelle Parameter. Durch die Forderung nach Unitarität ergeben sich N^2 Bedingungen. Seien u_i und u_j Spaltenvektoren von U , dann muss gelten: $u_i \cdot u_j = \delta_{ij}$ mit $i, j = 1, \dots, N$. Daher bleiben nur noch N^2 reelle Parameter übrig.
T	Für die Anzahl der reellen Parameter in T ergibt sich mit der Gaußschen Summenformel $2 \cdot \frac{N(N+1)}{2} = N^2 + N$.

Nach der Matrixtransformation beträgt die Anzahl der reellen Parameter also insgesamt $2N^2 + N$. Da die Freiheitsgrade auf der linken Seite die auf der rechten Seite um N übersteigen, besteht die Möglichkeit, N Einschränkungen für U vorzunehmen. Dies ist zulässig, weil die Matrix U nicht eindeutig durch die Schur-Zerlegung festgelegt wird. U kann mit einer beliebigen unitären Diagonalmatrix V multipliziert werden und die resultierende Matrix bleibt unitär. Die neue Transformation

$$(U V)^\dagger H (U V) = V^\dagger T V \quad (40)$$

liefert damit wieder eine Dreiecksmatrix mit den Eigenwerten auf der Hauptdiagonale. Damit eine Diagonalmatrix unitär ist, müssen die Einträge auf der Diagonale komplexe Zahlen auf dem Einheitskreis sein. Nach dem Differenzieren von

$H = U T U^\dagger$, erhält man

$$\begin{aligned} dH &= dU T U^\dagger + U dT U^\dagger + U T dU^\dagger = \\ &= U U^\dagger dU T U^\dagger + U dT U^\dagger + U T dU^\dagger U U^\dagger. \end{aligned} \quad (41)$$

Wird jetzt noch

$$\begin{aligned} U^\dagger U &= \mathbb{1} \\ \rightarrow 0 &= d(\mathbb{1}) = d(U^\dagger U) = dU^\dagger U + U^\dagger dU \\ \rightarrow dU^\dagger U &= -U^\dagger dU \end{aligned} \quad (42)$$

genutzt, kann (41) geschrieben werden als

$$dH = U (dT + U^\dagger dU T - T U^\dagger dU) U^\dagger = U dA U^\dagger. \quad (43)$$

Das hier neu eingeführte dA ist somit definiert als

$$\begin{aligned} dA &= dT + U^\dagger dU T - T U^\dagger dU = \\ &= dT + i(dS T - T dS). \end{aligned} \quad (44)$$

dS ist hier eine hermitesche Matrix, d.h. $dS^T = dS^*$. Wie oben erwähnt, können N Freiheitsgrade in U eingeschränkt werden. Da dS ein Produkt aus U^\dagger und dU ist, können stattdessen auch N Bedingungen für dS festgelegt werden. Daher wird gefordert, dass

$$dS_{jj} = -i(U^\dagger dU)_{jj} = 0, \quad j = 1, 2, \dots, N \quad (45)$$

gilt. Liest man Gleichung (44) nun komponentenweise, erhält man

$$dA_{jk} = dT_{jk} + i(T_{kk} - T_{jj})dS_{jk} + i \sum_{l < k} dS_{jl} T_{lk} - i \sum_{j < l} T_{jl} dS_{lk}. \quad (46)$$

Jetzt folgt ein sehr geschicktes Anordnen der Indizes (j,k), wodurch die Jakobi-Matrix $\left(\frac{\partial A_{jk}}{\partial (S_{jk}, T_{jk})} \right)_{jk}$ möglichst einfach wird. Dazu werden die Matrixelemente bezüglich des Index j in absteigender Reihenfolge notiert und alle Matrixelemente mit gleichem j sollen bezüglich k in aufsteigender Reihenfolge arrangiert werden. Falls $j \leq k$ ist wird $T_{j,k}$ der Variable $A_{j,k}$ zugeordnet und falls $j > k$ ist wird $S_{i,j}$ verwendet. Anschaulich kann man diese Zuordnung folgendermaßen darstellen: Zur Erinnerung, die Jakobi-Matrix für eine Variablentransformation $(x_1, \dots, x_m \rightarrow y_1, \dots, y_m)$ ist definiert als

$$J := \left(\frac{\partial x_i}{\partial y_j} \right)_{i=1 \dots m, j=1 \dots m} = \begin{pmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \dots & \frac{\partial x_1}{\partial y_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial y_1} & \frac{\partial x_m}{\partial y_2} & \dots & \frac{\partial x_m}{\partial y_m} \end{pmatrix}.$$

$$\begin{array}{cccccccc}
A_{N,1} & A_{N,2} & \cdots & A_{N,N-1} & A_{N,N} & A_{N-1,1} & \cdots & A_{N-1,N-2} & A_{N-1,N-1} \\
\downarrow & \downarrow & \cdots & \downarrow & \downarrow & \downarrow & \cdots & \downarrow & \downarrow \\
S_{N,1} & S_{N,2} & \cdots & S_{N,N-1} & T_{N,N} & S_{N-1,1} & \cdots & S_{N-1,N-2} & T_{N-1,N-1}
\end{array}$$

$$\begin{array}{ccccccc}
A_{N-1,N} & A_{N-2,1} & \cdots & A_{1,1} & A_{1,2} & \cdots & A_{1,N} \\
\downarrow & \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \\
T_{N-1,N} & S_{N-2,1} & \cdots & T_{1,1} & T_{1,2} & \cdots & T_{1,N}
\end{array}$$

Durch unsere Anordnung der Variablen entspricht x_1 hier $A_{N,1}$, x_2 entspricht $A_{N,2}$ usw. Bei den neuen Variablen entspricht y_1 $S_{N,1}$, y_2 $S_{N,2}$ usw. Damit sind die oben durch Pfeile zugeordneten Variablentransformationen auf der Diagonale der Jakobimatrix.

$$J = \begin{pmatrix} \frac{\partial A_{N,1}}{\partial S_{N,1}} & \frac{\partial A_{N,1}}{\partial S_{N,2}} & \cdots & \frac{\partial A_{N,1}}{\partial T_{1,N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial A_{1,N}}{\partial S_{N,1}} & \frac{\partial A_{1,N}}{\partial S_{N,2}} & \cdots & \frac{\partial A_{1,N}}{\partial T_{1,N}} \end{pmatrix} \quad (47)$$

Die partiellen Ableitungen kann man mit der Definition des totalen Differentials

$$df = \sum_{i=1}^N \frac{\partial f}{\partial x_i} dx_i \quad (48)$$

leicht an (46) ablesen. Die Diagonalelemente der Matrix sind damit entweder 1 oder $i(T_{kk} - T_{jj})$ und die Elemente über der Diagonale sind alle 0. Deshalb ist die Jakobi-Determinante einfach das Produkt der Diagonalelemente.

$$J = \begin{pmatrix} i(T_{11} - T_{NN}) & 0 & \cdots & 0 \\ iT_{1,2} & i(T_{22} - T_{NN}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \quad (49)$$

Nach [16] (15.1.8) gilt für das lineare Maß $\mu(dH)$, dass das Verhältnis der Volumenelemente der Jakobi-Determinante entspricht, wenn ein Differential linear durch andere Differentiale ausgedrückt wird. Das bedeutet für $\mu(dH)$:

$$\begin{aligned}
\mu(dH) &= \prod_{j,k} dH_{jk} dH_{jk}^* = \prod_{j,k} dA_{jk} dA_{jk}^* = \\
&= \prod_{j < k} |T_{jj} - T_{kk}|^2 dS_{jk} dS_{jk}^* \prod_{j \leq k} dT_{jk} dT_{jk}^* = \\
&= \prod_{j < k} |z_j - z_k|^2 \prod_j dz_j dz_j^* \prod_{j < k} dT_{jk} dT_{jk}^* dS_{jk} dS_{jk}^*
\end{aligned} \quad (50)$$

Hier wurden die N^2 Differentiale $dH_{jk} dH_{jk}^*$ durch die $\frac{N(N+1)}{2}$ Differentiale $dT_{jk} dT_{jk}^*$, die $\frac{N(N-1)}{2}$ Differentiale $dS_{jk} dS_{jk}^*$ und die Jakobi-Determinante $\prod_{j < k} |T_{jj} - T_{kk}|^2$ ersetzt. Die Zufallsverteilung der Einträge (38) enthält den Term $\exp(-\beta \text{Sp}(H^\dagger H))$. Nach der Transformation lautet dieser Term dann $\exp(-\beta \text{Sp}((U T U^\dagger)^\dagger (U T U^\dagger)))$. Zur Vereinfachung des Ausdrucks kann die Eigenschaften der Spur ausgenutzt werden, dass sie invariant unter zyklischer Vertauschung ist ($\text{Sp}(ABC) = \text{Sp}(CAB) = \text{Sp}(BCA)$). Außerdem gilt für die Adjungierte eines Produkts mehrerer Matrizen $(ABC)^\dagger = C^\dagger B^\dagger A^\dagger$. Somit kann $\text{Sp}((U T U^\dagger)^\dagger (U T U^\dagger))$ geschrieben werden als

$$\begin{aligned} \text{Sp}((U T U^\dagger)^\dagger (U T U^\dagger)) &= \text{Sp}(U T U^\dagger U T U^\dagger) = \\ &= \text{Sp}(T^\dagger T U^\dagger U) = \text{Sp}(T^\dagger T) = \sum_j |z_j|^2 + \sum_{j < k} |T_{jk}|^2. \end{aligned} \quad (51)$$

Mit diesen Ergebnissen erhält man

$$\begin{aligned} &\int \exp(-\text{Sp}(H^\dagger H)) \mu(dH) = \\ &= \int \exp\left(-\sum_j |z_j|^2\right) \cdot \exp\left(-\sum_{j < k} |T_{jk}|^2\right) \prod_{j < k} |z_j - z_k|^2 \\ &\quad \times \prod_j dz_j dz_j^* \prod_{j < k} dT_{jk} dT_{jk}^* dS_{jk} dS_{jk}^*. \end{aligned} \quad (52)$$

Nun haben wir schon fast das gewünschte Ergebnis. Jedoch wollen wir die Zufallsverteilung erhalten, in der nur noch die Eigenwerte vorhanden sind. Deshalb integriert man über die störenden Variablen T_{jk} und S_{jk} . Das Integral über T ist ein Gaußintegral. Die Integration über den Realteil eines bestimmten Tupels (j,k) ergibt $\sqrt{\pi}$ und die Integration über den entsprechenden Imaginärteil ergibt ebenfalls $\sqrt{\pi}$. Somit erhalten wir für jeden der $\frac{N(N+1)}{2}$ Matrixeinträge einen Faktor π . Insgesamt also

$$\int \exp\left(-\sum_{j < k} |T_{jk}|^2\right) \prod_{j < k} dT_{jk} dT_{jk}^* = \pi^{\frac{N(N+1)}{2}}. \quad (53)$$

Abschließend gilt es noch das Integral

$$\int \sum_{j < k} dS_{jk} dS_{jk}^* = \text{vol}(U(N)) / \int \prod dS_{jj} = \frac{\text{vol}(U(N))}{(2\pi)^N} \quad (54)$$

zu lösen. Hier ist $\text{vol}(U(N))$ das Volumen der unitären Gruppe, welches laut [14] (Corollary 3.5.2) gegeben ist durch

$$\text{vol}(U(N)) = \frac{(2\pi)^{\frac{N^2+N}{2}}}{\prod_{k=1}^{N-1} k!}. \quad (55)$$

Diese beiden letzten Integrale ergeben somit einen konstanten Vorfaktor vor der Zufallsverteilung der Eigenwerte. Das Integral (52) lässt sich dann mit (53) und (54) schreiben als

$$\int \exp(-Sp(H^\dagger H)) \mu(dH) = \int P_C(z_1, \dots, z_N) \prod_j dz_j dz_j^*, \quad (56)$$

wobei

$$P_C(z_1, \dots, z_N) = \text{Konst.} \cdot \prod_{j < k} |z_j - z_k|^2 \exp\left(-\sum_j |z_j|^2\right) \quad (57)$$

die gewünschte Zufallsverteilung der Eigenwerte ist. Die Faktoren $\prod_{j < k} |z_j - z_k|^2$ werden auch Vandermonde-Determinante genannt. Dieser lässt sich entnehmen, dass $P_C(z_1, \dots, z_N) \rightarrow 0$ geht, wenn $z_j \rightarrow z_k$ geht. Dieses Verhalten wird Niveauabstoßung genannt.

3.5 AI[†]-Ensemble

Das Ziel dieser Arbeit war es, für das AI[†] und das AII[†] Ensemble, aus der Zufallsverteilung der Einträge die Zufallsverteilung der Eigenwerte herzuleiten. Dazu geht man zunächst für das AI[†]-Ensemble von folgender Zufallsverteilung [8] (A26) aus

$$P(H)dH \propto \exp\left[-\beta\left(\sum_i |H_{ii}|^2 + \sum_{i>j} 2|H_{ij}|^2\right)\right] \times \prod_{i \geq j} dH_{ij} dH_{ij}^*. \quad (58)$$

Die Klasse AI[†] wird repräsentiert durch nicht-Hermitesche Matrizen, welche die Eigenschaft

$$H_{ij} = H_{ji} \quad (59)$$

erfüllen. Daraus resultiert auch der Faktor 2 in der Zufallsverteilung, da die Matrixelemente außerhalb der Hauptdiagonale immer zweifach auftreten.

3.5.1 Schur-Zerlegung

Der intuitiv logisch erscheinende Ansatz für das AI[†]-Ensemble ist, analog zum Ginibre-Ensemble, mit einer Schur-Zerlegung zu beginnen. Die Matrix H kann dann geschrieben werden als $H = UTU^\dagger$. Eine solche Zerlegung kann in Python leicht mit F berechnet werden. Beim Ginibre Ensemble wurden N Bedingungen für die Matrix $dS = -i(U^\dagger dU)$ festgelegt, nämlich $dS_{jj} = 0$. Damit stimmten

die Freiheitsgrade vor und nach der Transformation überein. Nun soll für das AI^\dagger -Ensemble dieselbe Überlegung angestellt werden. Die $N \times N$ -Matrix H hat $1 + 2 + \dots + N = \frac{N \cdot (N+1)}{2}$ unabhängige Einträge. Da es sich um komplexe Zahlen handelt, erhält man doppelt so viele, also $N^2 + N$, reelle Freiheitsgrade. Die unitäre Matrix hat wie bei der Herleitung nach Dyson N^2 und die Dreiecksmatrix $N \cdot (N + 1)$ Freiheitsgrade. Somit sind auf der linken Seite der Gleichung $H = U T U^\dagger$, $(2N^2 + N) - (N^2 + N) = N^2$ Freiheitsgrade zu viel vorhanden. Eine Einschränkung von dS gestaltet sich daher als schwierig.

Legt man N^2 Bedingungen für U fest, so wäre U vollständig festgelegt und sämtliche Freiheitsgrade müssen in T enthalten sein. Somit wäre U durch H und T vollständig bestimmt. Eine analoge Vorgehensweise zum Ginibre Ensemble, liefert statt (46) nur

$$dA_{jk} = dT_{jk}, \quad (60)$$

da $dS = -i(U^\dagger dU) = 0$, weil $dU = 0$. Dadurch würde die Vandermonde-Determinante verschwinden. Dies darf natürlich nicht passieren, da sie für die Niveauabstoßung verantwortlich ist, welche auch in den numerischen Berechnungen der Abstandsverteilung nächster Nachbarn beobachtet werden kann. Somit kann dieser Ansatz nicht korrekt sein.

Eine weitere Überlegung wäre, die Matrix unitär zu diagonalisieren, da dann die Freiheitsgrade übereinstimmen würden. Leider ist dies für das AI^\dagger -Ensemble nicht möglich, da eine Matrix nur unitär diagonalisiert werden kann, wenn sie normal ist. Eine Matrix ist genau dann normal, wenn sie mit ihrer Adjungierten kommutiert [9] (Definition 2.5.1).

$$HH^\dagger = H^\dagger H \quad (61)$$

Dies ist für das AI^\dagger -Ensemble im Allgemeinen nicht der Fall, da zum Beispiel für die 2×2 -Matrizen

$$\begin{aligned} \begin{pmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{pmatrix} \cdot \begin{pmatrix} H_{11}^* & H_{12}^* \\ H_{12}^* & H_{22}^* \end{pmatrix} &= \begin{pmatrix} |H_{11}|^2 + |H_{12}|^2 & H_{11}H_{12}^* + H_{12}H_{22}^* \\ H_{12}H_{11}^* + H_{22}H_{12}^* & |H_{12}|^2 + |H_{22}|^2 \end{pmatrix} \neq \\ &\neq \begin{pmatrix} |H_{11}|^2 + |H_{12}|^2 & H_{11}^*H_{12} + H_{12}^*H_{22} \\ H_{12}^*H_{11} + H_{22}^*H_{12} & |H_{12}|^2 + |H_{22}|^2 \end{pmatrix} = \begin{pmatrix} H_{11}^* & H_{12}^* \\ H_{12}^* & H_{22}^* \end{pmatrix} \cdot \begin{pmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{pmatrix} \end{aligned}$$

gilt. Analog kann auch für höhere Dimensionen nachgerechnet werden, dass die Normalität nicht erfüllt ist.

3.5.2 Diagonalisieren durch orthogonale Matrizen

Um das Problem der überschüssigen Freiheitsgrade zu umgehen, soll nun eine andere Transformation genutzt werden. Laut [9] (Theorem 4.4.13) kann man eine komplexe, symmetrische Matrix H falls diese diagonalisierbar ist (d.h. $H = A\Lambda A^{-1}$,

wobei Λ diagonal und A nichtsingulär ist) schreiben als $H = O\Lambda O^T$, wobei O orthogonal ist. Damit gilt $O^T O = \mathbb{1}$. Λ enthält auf der Diagonale die Eigenwerte λ_i der Matrix H .

Zunächst sollen wieder die Freiheitsgrade der Matrixtransformation untersucht werden. Die symmetrische Matrix H hat wie zuvor $N^2 + N$ Freiheitsgrade. Die komplexe Matrix O hat N^2 Einträge. Durch die Forderung nach Orthogonalität muss

$$q_i^T \cdot q_j = \delta_{ij} \quad (62)$$

für die Spaltenvektoren gelten. Daher erhalten wir $\frac{N(N+1)}{2}$ Gleichungen, die die Einträge der Matrix erfüllen müssen. Somit haben wir $N^2 - \frac{N(N+1)}{2} = \frac{N(N-1)}{2}$ unabhängige Einträge und damit $N^2 - N$ Freiheitsgrade. Die Diagonalmatrix hat N Einträge und deshalb $2 \cdot N$ Freiheitsgrade. Die Freiheitsgrade nach der Transformation betragen dann insgesamt $N^2 + N$ und stimmen damit mit den Freiheitsgraden der Matrix H überein. Die Wahrscheinlichkeitsverteilung nach der Matrixtransformation setzt sich im Allgemeinen aus der Zufallsverteilung der Eigenwerte und der orthogonalen Matrix $P(S(O, \Lambda))$, der Jakobi-Determinante $|J(S \rightarrow O, \Lambda)|$ und den Differentialen dO und $d\lambda_i$ zusammen.

$$P(H)dH = P(H(O, \Lambda)) |J(H \rightarrow O, \Lambda)| dO \prod_{i=1}^N d\lambda_i \quad (63)$$

Die Determinante der Matrixtransformation entspricht der sogenannten Vandermonde-Determinante und beträgt damit

$$\prod_{j>k} (\lambda_j - \lambda_k)^\beta. \quad (64)$$

Der Exponent β ist jedoch noch unbekannt. Um aus der Zufallsverteilung der Eigenwerte und der orthogonalen Matrizen die Zufallsverteilung, in der nur noch die Eigenwerte enthalten sind, zu erhalten, muss man über die Untermannigfaltigkeit \mathbb{V}_N des Raumes \mathbb{C}^{N^2} integrieren. Diese Untermannigfaltigkeit ist die sogenannte Stiefel-Mannigfaltigkeit über den Körper \mathbb{C}^{N^2} . Somit müsste folgendes Integral gelöst werden,

$$P(\lambda_1, \dots, \lambda_N) = \prod_{j>k} |\lambda_j - \lambda_k|^\beta \int_{\mathbb{V}(\mathbb{C})^{N^2}} dO \exp(-\beta \text{Sp}(O^* \Lambda O^\dagger O \Lambda O^T)), \quad (65)$$

was sich als äußerst schwierig erweist.

3.5.3 Takagi-Faktorisierung

Ein weiterer Versuch, eine passende Matrixtransformation zu finden, war die sogenannte Takagi-Faktorisierung [9] (Corollary 4.4.4). Diese besagt, dass es für eine symmetrische Matrix H eine unitäre Matrix U und eine Diagonalmatrix D mit reellen, nichtnegativen Elementen gibt, sodass man H schreiben kann als $H = U D U^T$. Die Spalten der Matrix U sind Eigenvektoren der Matrix HH^* . Die Einträge der Diagonalmatrix D sind die nichtnegativen Wurzeln der Eigenwerte λ_i von HH^* . Man nennt diese Werte auch die Singulärwerte $\sigma_i = \sqrt{\lambda_i}$ der Matrix H .

Zunächst sollen wieder die Freiheitsgrade der Zerlegung betrachtet werden. Die symmetrische $N \times N$ -Matrix H hat, wie bereits erläutert, $N^2 + N$ Freiheitsgrade. Die unitäre Matrix U hat N^2 Freiheitsgrade und die Diagonalmatrix D hat N Freiheitsgrade, da die Diagonalelemente im Gegensatz zur orthogonalen Diagonalisierung reell sind. Somit stimmt die Anzahl der Freiheitsgrade überein und bei der Transformation müssen keine Einschränkungen vorgenommen werden. Es sollen jedoch die Eigenwerte der Matrix H und nicht die der Matrix HH^* untersucht werden. Deshalb muss folgende Umformung vorgenommen werden.

Sei μ_i ein Eigenwert der Matrix H und λ_i ein Eigenwert der Matrix HH^* , dann folgt daraus

$$\begin{aligned} HH^* \vec{x}_i &= H \mu_i^* \vec{x}_i = \mu_i \mu_i^* \vec{x}_i \\ \rightarrow \lambda_i &= \mu_i \mu_i^* = |\mu_i|^2 \\ \rightarrow D &= \text{diag}(|\mu_1|, \dots, |\mu_N|). \end{aligned} \tag{66}$$

Der Vorteil dieser Zerlegung ist, dass man die Umformung der Spur wie für das Ginibre-Ensemble nutzen kann, d.h.

$$\begin{aligned} Sp(HH^\dagger) &= Sp((U D U^T)^\dagger (U D U^T)) = Sp(U^* D^\dagger U^\dagger U D U^T) = \\ &= Sp(U^T U^* D^\dagger D) = Sp(D^\dagger D) = \sum_{i=1}^N |\mu_i|^2. \end{aligned} \tag{67}$$

Das Problem bei der Takagi-Faktorisierung ist, dass man nur die Beträge der Eigenwerte von H in der Diagonalmatrix erhält. Somit geht wichtige Information über die Phase der Eigenwerte verloren. Damit könnte man nur die Zufallsverteilung der Beträge der Eigenwerte erhalten, jedoch nicht die Zufallsverteilung der Eigenwerte selbst. Nichtsdestotrotz soll diese hier kurz aufgeführt werden. In [3] (Theorem A.11) wird die Zufallsverteilung der Singulärwerte für komplexe symmetrische Gauß-Ensembles hergeleitet. Dazu wird unter anderem die oben angesprochene Takagi-Faktorisierung verwendet. Seien σ_i die Singulärwerte der Matrix

H , dann ist

$$P(\sigma_1, \dots, \sigma_N) = c_N |\Delta(\sigma_1^2, \dots, \sigma_N^2)| \prod_{j=1}^N |\sigma_j| \prod_{j=1}^N \exp\left(-\frac{|\sigma_j|^2}{2}\right) \quad (68)$$

die Zufallsverteilung der Singulärwerte und

$$P(|\mu_1|, \dots, |\mu_N|) = c_N |\Delta(|\mu_1|^2, \dots, |\mu_N|^2)| \prod_{j=1}^N |\mu_j| \prod_{j=1}^N \exp\left(-\frac{|\mu_j|^2}{2}\right) \quad (69)$$

die Zufallsverteilung der Beträge der Eigenwerte. $\Delta(\dots)$ steht hier jeweils für die Vandermonde-Determinante.

3.6 AII[†]-Ensemble

Anschließend soll Selbiges beim AII[†]-Ensemble versucht werden. Dieses besteht aus $N \times N$ -Matrizen, die vier Untermatrizen enthalten. Deshalb muss N geradzahlig sein.

$$H = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (70)$$

Diese weisen folgende Eigenschaften auf:

$$\begin{aligned} a &= d^T \\ b &= -b^T \\ c &= -c^T \end{aligned}$$

Die Einträge der Matrizen a , b , c und d sind komplexe Zahlen. Für den 4×4 -Fall sieht die Matrix H wie folgt aus:

$$H = \begin{pmatrix} a_{11} & a_{12} & 0 & b_{12} \\ a_{21} & a_{22} & -b_{12} & 0 \\ 0 & c_{12} & a_{11} & a_{21} \\ -c_{12} & 0 & a_{12} & a_{22} \end{pmatrix} \quad (71)$$

Eine weitere Möglichkeit nach [10] die Matrizen zu definieren, ist sie mit komplexen Quaternionen zu füllen und zu fordern, dass $H_{ij}^D = H_{ji}$ gilt. Die Quaternionen [13] sind ein Schiefkörper mit den imaginären Einheiten i , j und k . Man kann sie schreiben als

$$q = q_0 + q_1 i + q_2 j + q_3 k. \quad (72)$$

Das H_{ij}^D steht hier für das duale der Quaternionen, welches definiert ist als

$$q^D = q_0 - q_1 i - q_2 j - q_3 k. \quad (73)$$

Von komplexen Quaternionen sind die Vorfaktoren q_0, q_1, q_2 und q_3 jeweils komplexe Zahlen. Zwischen den Quaternionen und 2×2 -Matrizen existiert ein injektiver Ringhomomorphismus, weshalb jede Quaternion geschrieben werden kann als

$$q = q_0 + q_1 i + q_2 j + q_3 k \mapsto \begin{pmatrix} q_0 + i q_1 & q_2 + i q_3 \\ -q_2 + i q_3 & q_0 - i q_1 \end{pmatrix}. \quad (74)$$

Da $H_{ij}^D = H_{ji}$ gelten muss sieht die Matrix für den 2×2 -Fall wie folgt aus:

$$H = \begin{pmatrix} a & c + di + ej + fk \\ c - di - ej - fk & b \end{pmatrix} \quad (75)$$

Nach dem Ersetzen der Quaternionen durch die entsprechenden 2×2 -Matrizen und dem Vertauschen der 2. und 3. Zeilen und Spalten, schreibt sich (75) als

$$\begin{aligned} H &= \begin{pmatrix} a & 0 & c + id & e + if \\ 0 & a & -e + if & c - id \\ c - id & -e - if & b & 0 \\ e - if & c + id & 0 & b \end{pmatrix} = \\ &= \begin{pmatrix} a & c + id & 0 & e + if \\ c - id & b & -e - if & 0 \\ 0 & -e + if & a & c - id \\ e - if & 0 & c + id & b \end{pmatrix}. \end{aligned} \quad (76)$$

Da $a, b, c, d, e, f \in \mathbb{C}$ können die Matrixeinträge in ihre Real- (c_R) und Imaginärteile (c_I) aufgeteilt werden. Dann ergibt beispielsweise $c + id = (c_R - d_I) + i(c_I + d_R)$ und beim Vergleich mit (71) wird sofort klar, dass dies dem komplexen Eintrag a_{12} entspricht. Analog kann mit den anderen Matrixeinträgen verfahren werden. Folglich ist gezeigt, dass die beiden Formulierungen (70) aus [8] und (75) aus [10] identisch zueinander sind.

Für den allgemeinen $N \times N$ -Fall kann die Wahrscheinlichkeitsverteilung der Einträge geschrieben werden als [8] (A32)

$$\begin{aligned} P(H) dH &\propto \exp \left\{ -2\beta \left[\sum_{i \geq j} |a_{ij}|^2 + \sum_{i > j} (|b_{ij}|^2 + |c_{ij}|^2 + |d_{ij}|^2) \right] \right\} \\ &\quad \times \prod_{i \geq j} da_{ij} da_{ij}^* \prod_{i > j} db_{ij} db_{ij}^* dc_{ij} dc_{ij}^* dd_{ij} dd_{ij}^*. \end{aligned} \quad (77)$$

Der Faktor 2 resultiert wieder aus der Symmetrie, da alle Einträge abgesehen vom Vorzeichen zweifach auftreten. Die Matrixelemente auf der Diagonale von a und darunter werden durch die Summanden $\sum_{i \geq j} |a_{ij}|^2$ abgedeckt. Die Elemente über der Diagonale werden in $\sum_{i > j} |d_{ij}|^2$ berücksichtigt. b und c sind schiefsymmetrisch, weshalb man durch den Faktor 2 nur noch die Elemente unter der Diagonale berücksichtigen muss. Alternativ kann die Zufallsverteilung durch die Spur der Matrix ausgedrückt werden als

$$P(H)dH \propto \exp\{-2\beta \operatorname{Sp}(HH^\dagger)\}. \quad (78)$$

Damit bietet es sich wieder an, eine Schur-Zerlegung durchzuführen, da dann nach der Transformation wieder die unitären Matrizen aus der Spur verschwinden würden.

3.6.1 Schur-Zerlegung

Zunächst sollen erneut die Freiheitsgrade untersucht werden. Die Freiheitsgrade der Matrix H berechnet man, indem man die Freiheitsgrade der vier Untermatrizen untersucht. Somit ergeben sich insgesamt $N^2 - N$ Freiheitsgrade für die Matrix H .

Untermatrix	Freiheitsgrade
a:	$2 \cdot \frac{N}{2} \cdot \frac{N}{2} = \frac{N^2}{2}$
b:	$2 \cdot \frac{\frac{N}{2}(\frac{N}{2}-1)}{2} = \frac{N^2}{4} - \frac{N}{2}$
c:	$2 \cdot \frac{\frac{N}{2}(\frac{N}{2}-1)}{2} = \frac{N^2}{4} - \frac{N}{2}$
d:	Wird vollständig durch a festgelegt.

Berechnet man numerisch mit G die Schur-Zerlegung, so fällt auf, dass die unitäre Matrix N Nullen enthält. Sie ist von der Form

$$U = \begin{pmatrix} U_{1,1} & \dots & U_{1,\frac{N}{2}} & 0 & \dots & 0 \\ U_{2,1} & \dots & U_{2,\frac{N}{2}} & U_{2,\frac{N}{2}+1} & \dots & U_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & U_{\frac{N}{2}+1,\frac{N}{2}+1} & \dots & U_{\frac{N}{2}+1,N} \\ U_{\frac{N}{2}+2,1} & \dots & \dots & \dots & \dots & U_{\frac{N}{2}+2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ U_{N,N} & \dots & \dots & \dots & \dots & U_{N,N} \end{pmatrix}.$$

Somit hat sie $N^2 - N$ komplexe Einträge. Kommen noch die N^2 Bedingungen hinzu, damit die Matrix unitär ist, bleiben am Ende $N^2 - 2N$ Freiheitsgrade für U übrig. Auch die Matrix T muss genauer betrachtet werden, als es zuvor bei Ginibre oder

AI^\dagger der Fall war. Da hier alle Eigenwerte doppelt auftreten, hat man $\frac{N(N-1)}{2} + \frac{N}{2}$ unabhängige Einträge und somit N^2 Freiheitsgrade. Insgesamt ergibt sich für die Differenz vor und nach der Transformation $N^2 - N$. Eine Einschränkung, sodass die Anzahl der Freiheitsgrade übereinstimmt, konnte nicht gefunden werden.

3.6.2 Eigenwertzerlegung

Da die Schur-Zerlegung wegen dem Überschuss am Freiheitsgraden nicht funktioniert, kann alternativ die Eigenwertzerlegung angesetzt werden [9] (Theorem 1.3.7). Dazu verwendet man

$$H = Q\Lambda Q^{-1}. \quad (79)$$

Λ ist hier eine Diagonalmatrix mit den Eigenwerten von H auf der Hauptdiagonale. Da alle Eigenwerte doppelt auftreten, ist $\Lambda = \text{diag}(\lambda_1, \lambda_1, \dots, \lambda_{N/2}, \lambda_{N/2})$. Die Matrix Q hat als Spaltenvektoren die Eigenvektoren von H . Somit haben wir in Λ $N/2$ komplexe Einträge und damit N Freiheitsgrade und in Q $N/2$ linear unabhängige Eigenvektoren mit je N komplexen Einträgen und somit N^2 Freiheitsgrade. Insgesamt haben wir also vor und nach der Transformation $N^2 + N$ Freiheitsgrade. Nun kann ein ähnliches Vorgehen wie bei Kapitel 3.5.2 gewählt werden. Eine Variablentransformation von $H \rightarrow (Q, \Lambda)$ wird durchgeführt. Damit wird aus der Zufallsverteilung der Einträge

$$P(H)dH = P(H(Q, \Lambda)) |J(H \rightarrow Q, \Lambda)| dQ \prod_{i=1}^N \lambda_i. \quad (80)$$

Um die Matrix Q zu eliminieren, muss darüber integriert werden.

$$P(\lambda_1, \dots, \lambda_N) = |J(H \rightarrow Q, \Lambda)| \int dQ \exp(-\beta \text{Sp}(Q^{-1\dagger} \Lambda^\dagger Q^\dagger Q \Lambda Q^{-1})) \quad (81)$$

Hier ist $|J(H \rightarrow \{Q, \Lambda\})|$ wieder die Vandermonde-Determinante. Eine Integral über dieses Produkt aus sechs Matrizen scheint kaum lösbar zu sein.

4 Fazit

Auch wenn es in dieser Arbeit letztlich nicht gelungen ist, die gewünschten Zufallsverteilungen der Eigenwerte für das AI^\dagger - und das AII^\dagger -Ensemble herzuleiten, konnten zumindest verschiedene Ansätze und die damit einhergehenden Probleme aufgezeigt werden. Zusammen mit den einführenden Kapiteln zur nicht-Hermiteschen Zufallsmatrixtheorie und den Python-Codes, bildet diese Arbeit eine gute Grundlage für zukünftige Versuche, diese Herleitungen durchzuführen. So kann der- oder diejenige vorab die dabei auftretenden Probleme erkennen und neue Ansätze versuchen. Nach diesen Ausführungen scheint es am besten zu sein, die Schur-Zerlegung zu nutzen, da sich dann die Spur stark vereinfacht. Ohne diese Vereinfachung steht man vor kaum lösbaren Integralen. Das Hauptproblem, welches bei der Schur-Zerlegung gelöst werden muss, ist die Anzahl an überschüssigen Freiheitsgraden. Gelingt dies, könnte eine ähnliche Rechnung wie bei der Herleitung der Zufallsverteilung der Eigenwerte des Ginibre-Ensembles möglich sein.

Literatur

- [1] Gernot Akemann u. a. “Universal Signature from Integrability to Chaos in Dissipative Open Quantum Systems”. In: *Phys. Rev. Lett.* 123.25 (Dez. 2019). DOI: 10.1103/physrevlett.123.254101. arXiv: 1910.03520v4 (siehe S. 12, 13).
- [2] J. T. Chalker und Z. Jane Wang. “Diffusion in a Random Velocity Field: Spectral Properties of a Non-Hermitian Fokker-Planck Operator”. In: *Phys. Rev. Lett.* 79 (10 Sep. 1997), S. 1797–1800. DOI: 10.1103/PhysRevLett.79.1797. arXiv: cond-mat/9704198v1 (siehe S. 7).
- [3] Ryan C Chen u. a. “Spectral statistics of non-Hermitian random matrix ensembles”. In: *Random Matrices: Theory and Applications* 8.02 (2019), S. 1950005. arXiv: 1803.08127v2 (siehe S. 23).
- [4] Freeman J Dyson. “The threefold way. Algebraic structure of symmetry groups and ensembles in quantum mechanics”. In: *Journal of Mathematical Physics* 3.6 (1962), S. 1199–1215 (siehe S. 13).
- [5] Jean Ginibre. “Statistical ensembles of complex, quaternion, and real matrices”. In: *Journal of Mathematical Physics* 6.3 (1965), S. 440–449 (siehe S. 14).
- [6] Rainer Grobe und Fritz Haake. “Universality of cubic-level repulsion for dissipative quantum chaos”. In: *Phys. Rev. Lett.* 62 (25 Juni 1989), S. 2893–2896. DOI: 10.1103/PhysRevLett.62.2893 (siehe S. 7).
- [7] Rainer Grobe, Fritz Haake und Hans-Jürgen Sommers. “Quantum Distinction of Regular and Chaotic Dissipative Motion”. In: *Phys. Rev. Lett.* 61 (17 Okt. 1988), S. 1899–1902. DOI: 10.1103/PhysRevLett.61.1899 (siehe S. 13).
- [8] Ryusuke Hamazaki u. a. “Universality classes of non-Hermitian random matrices”. In: *Physical Review Research* 2.2 (Juni 2020). DOI: 10.1103/physrevresearch.2.023286. arXiv: 1904.13082v7 (siehe S. 1, 12, 13, 15, 20, 25).
- [9] Roger A Horn und Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012 (siehe S. 21, 23, 27).
- [10] Ambuja Bhushan Jaiswal, Ravi Prakash und Akhilesh Pandey. *Universality classes of quantum chaotic dissipative systems*. 2019. arXiv: 1904.12484 (siehe S. 15, 24, 25).
- [11] Takuya Kanazawa und Tilo Wettig. “New universality classes of the non-Hermitian Dirac operator in QCD-like theories”. In: *Physical Review D* 104.1 (Juli 2021). DOI: 10.1103/physrevd.104.014509. arXiv: 2104.05846v1 (siehe S. 1, 7).

- [12] Kohei Kawabata u. a. “Symmetry and Topology in Non-Hermitian Physics”. In: *Physical Review X* 9.4 (Okt. 2019). DOI: 10.1103/physrevx.9.041015. arXiv: 1812.09133v4 (siehe S. 14).
- [13] Jürg Kramer und Anna-Maria Von Pippich. *Von den natürlichen Zahlen zu den Quaternionen*. Springer, 2013 (siehe S. 24).
- [14] Sergei K Lando, Alexander K Zvonkin und Don Bernard Zagier. *Graphs on surfaces and their applications*. Bd. 75. Springer, 2004 (siehe S. 19).
- [15] Giacomo Livan, Marcel Novaes und Pierpaolo Vivo. *Introduction to random matrices: theory and practice*. Bd. 26. Springer, 2018 (siehe S. 3, 5).
- [16] Madan Lal Mehta. *Random matrices*. Elsevier, 2004 (siehe S. 1, 15, 18).
- [17] Lucas Sá, Pedro Ribeiro und Tomaž Prosen. “Complex Spacing Ratios: A Signature of Dissipative Quantum Chaos”. In: *Physical Review X* 10.2 (Apr. 2020). DOI: 10.1103/physrevx.10.021019. arXiv: 1910.12784v2 (siehe S. 7, 8).
- [18] Eugene P Wigner. “Random matrices in physics”. In: *SIAM review* 9.1 (1967), S. 1–23 (siehe S. 1).

A Abstandsverteilung nächster Nachbarn

```
import math
import numpy as np
import numpy.linalg as linalg
from scipy.spatial import KDTree
from matplotlib import pyplot as plt
from scipy.stats import kde

N=2000      #Dimension der Matrix: NxN
samplesize=1000    #Anzahl der Beispiele die berechnet werden
M=int(N/2)
rea_u=np.array([0.0]*N)
ima_u=np.array([0.0]*N)
rea_h=np.array([0.0]*N)
ima_h=np.array([0.0]*N)
rea_g=np.array([0.0]*N)
ima_g=np.array([0.0]*N)
s_AId=[]    #speicher für die spacing-ratios des AId-Ensembles
s_AIIId=[]  #speicher für die spacing-ratios des AIId-Ensembles
s_g=[]      #speicher für die spacing-ratios Ginibre-Ensembles

for k in range(0, samplesize):
    print(k)
    g=np.random.randn(N,N)/np.sqrt(N)    #Ginibre-Matrix

    u=np.random.randn(N,N)+complex(0,1)*np.random.randn(N,N)
    u=(u*u.transpose())/(2*np.sqrt(N))    #AId-Matrix

    a=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
    d=a.transpose()
    v=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
    v=(v*v.transpose())/2
    b=np.triu(v, 1) - np.tril(v, -1)
    w=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
    w=(w*w.transpose())/2
    c=np.triu(w, 1) - np.tril(w, -1)
    h = complex(0,1)*np.ones((N,N))
    h[0:M,0:M]=a
    h[0:M,M:N]=b
    h[M:N,0:M]=c
```

```

h[M:N,M:N]=d
h=h*1/np.sqrt(N)      #AIId-Matrix

ew_g=linalg.eigvals(g)
ew_u=linalg.eigvals(u)
ew_h=linalg.eigvals(h)

for i in range(0, N):
    rea_g[i]=ew_g[i].real      #Ginibre
    ima_g[i]=ew_g[i].imag
    rea_u[i]=ew_u[i].real      #AId
    ima_u[i]=ew_u[i].imag
    rea_h[i]=ew_h[i].real      #AIId
    ima_h[i]=ew_h[i].imag

tree_g=KDTree(np.c_[rea_g, ima_g])      #Ginibre
tree_u=KDTree(np.c_[rea_u, ima_u])      #AId
tree_h=KDTree(np.c_[rea_h, ima_h])      #AIId

for i in range(0, N):      #Ginibre
    nn,index=tree_g.query([rea_g[i],ima_g[i]], k=15)
    s_g.append(nn[1]*math.sqrt(10/(nn[9]*nn[9]*math.pi)))
for i in range(0, N):      #AId
    nn,index=tree_u.query([rea_u[i],ima_u[i]], k=15)
    s_AId.append(nn[1]*math.sqrt(10/(nn[9]*nn[9]*math.pi)))
"""
Für das AIId-Ensemble muss man den 20. nächsten Nachbarn (
    k=30)
zum Entfalten verwenden, da hier alle Eigenwerte doppelt
auftreten im Gegensatz zum AId- und Ginibre Ensemble (k
    =15).
"""
for i in range(0, N):      #AIId
    nn,index=tree_h.query([rea_h[i],ima_h[i]], k=30)
    s_AIIId.append(nn[2]*math.sqrt(10/(nn[19]*nn[19]*math.pi
        )))

prob_density_g = kde.gaussian_kde(s_g)      #Ginibre
prob_density_AId = kde.gaussian_kde(s_AId)      #AId
prob_density_AIIId = kde.gaussian_kde(s_AIIId)      #AIId

x = np.linspace(0,1.5,100)
y_g = prob_density_g(x)

```



```

y_AId = prob_density_AId(x)
y_AIId = prob_density_AIId(x)

plt.plot(x,y_AId, label="AI${N}{DAGGER}$")
plt.plot(x,y_AIId, label="AII${N}{DAGGER}$")
plt.plot(x,y_g, label="Ginibre")
plt.title("Abstandsverteilung nächster Nachbarn für %ix%i AI$
    ^{\N{DAGGER}}$-, AII${N}{DAGGER}}$- und Ginibre-Ensemble"%(
    N,N))
plt.xlabel('s')
plt.ylabel('p(s)')
plt.legend()
plt.show()

```

B Komplexe Abstandsverhältnisse des Ginibre-Ensembles

```

import cmath
import numpy as np
import numpy.linalg as linalg
from scipy.spatial import KDTree
from matplotlib import pyplot as plt
from scipy.stats import kde

N = 4000      #Dimension der matrix: NxN
samplesize=100    #Anzahl der Beispiele die berechnet werden
rea=np.array([0.0]*N)    #speicher für die Real- und
    Imaginärteile der Eigenwerte
ima=np.array([0.0]*N)
s=[]    #speicher für die spacing-ratios
betrag=[]
theta=[]

for k in range(0, samplesize):

    u=np.random.randn(N,N)+complex(0,1)*np.random.randn(N,N)
    u=u*np.sqrt(N)

    ew=linalg.eigvals(u)

```

```

for i in range(0, N):
    rea[i]=ew[i].real
    ima[i]=ew[i].imag

tree=KDTree(np.c_[rea, ima])    #Initialisieren des KDTrees

for i in range(0, N):
    nn,index=tree.query([rea[i],ima[i]], k=3)
    s.append(((rea[index[1]]+complex(0,1)*ima[index[1]])-(
        rea[index[0]]+complex(0,1)*ima[index[0]]))/((rea[
        index[2]]+complex(0,1)*ima[index[2]])-(rea[index
        [0]]+complex(0,1)*ima[index[0]])))

re = np.real(s)
im = np.imag(s)

nbins = 400    # 400 x 400 Punkte
fig = plt.figure()
ax = fig.add_subplot(111)
k = kde.gaussian_kde([re, im])    #kernel-density estimate
xi, yi = np.mgrid[-1:1:nbins*1j, -1:1:nbins*1j]
zi = k(np.vstack([xi.flatten(), yi.flatten()]))
B = zi.reshape(xi.shape)
plot = ax.pcolormesh(xi, yi, B, shading='auto')
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])
plt.colorbar(plot)
plt.title("komplexe Abstandsverhältnisse für N=%i, Ginibre-
    Ensemble"%N)
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')
plt.show()

for i in range(0, N*samplesize):
    r,theta_s=cmath.polar(s[i])
    theta.append(theta_s)
    betrag.append(r)

with open('ginibre.npy', 'wb') as f:    #Speichern der Werte zur
    weiteren Auswertung
    np.save(f, betrag)
    np.save(f, theta)

```

C Komplexe Abstandsverhältnisse des AI^\dagger -Ensembles

```
import cmath
import numpy as np
import numpy.linalg as linalg
from scipy.spatial import KDTree
from matplotlib import pyplot as plt
from scipy.stats import kde

N = 4000      #Dimension der matrix: NxN
samplesize=100    #Anzahl der Beispiele die berechnet werden
rea=np.array([0.0]*N)    #Speicher für die Real- und
    Imaginärteile der Eigenwerte
ima=np.array([0.0]*N)
s=[]    #Speicher für die Abstandsverhältnisse
betrag=[]
theta=[]

for k in range(0, samplesize):

    u=np.random.randn(N,N)+complex(0,1)*np.random.randn(N,N)
    u=(u*u.transpose())/(2*np.sqrt(N))

    ew=linalg.eigvals(u)

    for i in range(0, N):
        rea[i]=ew[i].real
        ima[i]=ew[i].imag

    tree=KDTree(np.c_[rea, ima])

    for i in range(0, N):
        nn,index=tree.query([rea[i],ima[i]], k=3)
        s.append(((rea[index[1]]+complex(0,1)*ima[index[1]])-(
            rea[index[0]]+complex(0,1)*ima[index[0]]))/((rea[
            index[2]]+complex(0,1)*ima[index[2]])-(rea[index
            [0]]+complex(0,1)*ima[index[0]])))

re = np.real(s)
im = np.imag(s)

nbins = 400    # 400 x 400 punkte
```

```

fig = plt.figure()
ax = fig.add_subplot(111)
k = kde.gaussian_kde([re, im]) #kernel-density estimate
xi, yi = np.mgrid[-1:1:nbins*1j, -1:1:nbins*1j]
zi = k(np.vstack([xi.flatten(), yi.flatten()]))
B = zi.reshape(xi.shape)
plot = ax.pcolormesh(xi, yi, B, shading='auto')
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])
plt.colorbar(plot)
plt.title("komplexe Abstandsverhältnisse für N=%i, AI $\sim$ {\N{
    DAGGER}}$-Ensemble"%N)
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')
plt.show()

for i in range(0, N*samplesize):
    r, theta_s = cmath.polar(s[i])
    theta.append(theta_s)
    betrag.append(r)

with open('AI $\sim$ d.npy', 'wb') as f:
    np.save(f, betrag)
    np.save(f, theta)

```

D Komplexe Abstandsverhältnisse des AII[†]-Ensembles

```

import cmath
import numpy as np
import numpy.linalg as linalg
from scipy.spatial import KDTree
from matplotlib import pyplot as plt
from scipy.stats import kde

N = 4000 #Dimension der matrix: NxN. Hier sind aufgrund der
        Symmetrie nur gerade Zahlen zulässig!
samplesize=100 #Anzahl der Beispiele die berechnet werden
M=int(N/2)
rea=np.array([0.0]*N) #Speicher für die Real- und Imaginärteile
        der Eigenwerte
ima=np.array([0.0]*N)

```

```

s=[] #speicher für die spacing-ratios
betrag=[] #Speicher für die Beträge der Abstandsverhältnisse
theta=[]

for k in range(0, samplesize):

    a=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
    d=a.transpose()
    v=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
    v=(v*v.transpose())/2
    b=np.triu(v, 1) - np.tril(v, -1)
    w=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
    w=(w*w.transpose())/2
    c=np.triu(w, 1) - np.tril(w, -1)

    u = complex(0,1)*np.ones((N,N))
    u[0:M,0:M]=a
    u[0:M,M:N]=b
    u[M:N,0:M]=c
    u[M:N,M:N]=d
    u=u*1/np.sqrt(N)

    ew=linalg.eigvals(u)

    for i in range(0, N):
        rea[i]=ew[i].real
        ima[i]=ew[i].imag

    tree=KDTree(np.c_[rea, ima])

    for i in range(0, N):
        nn,index=tree.query([rea[i],ima[i]], k=5)
        s.append(((rea[index[2]]+complex(0,1)*ima[index[2]])-(
            rea[index[0]]+complex(0,1)*ima[index[0]]))/((rea[
            index[4]]+complex(0,1)*ima[index[4]])-(rea[index
            [0]]+complex(0,1)*ima[index[0]])))

re = np.real(s)
im = np.imag(s)

nbins = 400 # 400 x 400 punkte
fig = plt.figure()
ax = fig.add_subplot(111)

```

```

k = kde.gaussian_kde([re, im])
xi, yi = np.mgrid[-1:1:nbins*1j, -1:1:nbins*1j]
zi = k(np.vstack([xi.flatten(), yi.flatten()]))
B = zi.reshape(xi.shape)
plot = ax.pcolormesh(xi, yi, B, shading='auto')
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])
plt.colorbar(plot, ax=ax)
plt.title("komplexe Abstandsverhältnisse für N=%i, AIIN{\N{
    DAGGER}}$-Ensemble"%N)
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')
plt.show()

for i in range(0, N*samplesize):
    r, theta_s = cmath.polar(s[i])
    theta.append(theta_s)
    betrag.append(r)

with open('AIId.npy', 'wb') as f:
    np.save(f, betrag)
    np.save(f, theta)

```

E Vergleich der komplexen Abstandsverhältnisse der Universalitätsklassen

```

import numpy as np
from matplotlib import pyplot as plt

with open('AId.npy', 'rb') as f:
    betrag_AIId = np.load(f)
    theta_AIId = np.load(f)

with open('AIId.npy', 'rb') as h:
    betrag_AIIId = np.load(h)
    theta_AIIId = np.load(h)

with open('ginibre.npy', 'rb') as g:
    betrag_gi = np.load(g)
    theta_gi = np.load(g)

```

```

'''
Radialkomponente
'''
plt.hist(betrag_AId, bins=100,density=True, alpha = 1,histtype=
u'step', label='AI$^{\N{DAGGER}}$',stacked=True)
plt.hist(betrag_AIID, bins=100,density=True, alpha = 1,histtype=
u'step', label='AII$^{\N{DAGGER}}$',stacked=True)
plt.hist(betrag_gi, bins=100,density=True, alpha = 1,histtype=u
'step', label='Ginibre',stacked=True)
plt.legend(loc='upper left')
plt.xlabel('r')
plt.ylabel('p(r)')
plt.show()
'''

Winkelkomponente
'''
plt.hist(theta_AId, bins=100,density=True, alpha = 1, histtype=
u'step', label='AI$^{\N{DAGGER}}$',stacked=True)
plt.hist(theta_AIID, bins=100,density=True, alpha = 1,histtype=
u'step', label='AII$^{\N{DAGGER}}$',stacked=True)
plt.hist(theta_gi, bins=100,density=True, alpha = 1, histtype=u
'step',label='Ginibre',stacked=True)
plt.legend(loc='best')
plt.xlabel('\u0398')
plt.ylabel('P(\u0398)')
plt.show()

```

F Schur-Zerlegung des AI^\dagger -Ensembles

```

import numpy as np
from scipy.linalg import schur

N=2 #Dimension der matrix: NxN
S=np.random.randn(N,N)+complex(0,1)*np.random.randn(N,N)
S=(S*S.transpose())/2

T,U=schur(S, output='complex')

print("S=UTU^{\N{DAGGER}}\n")
print("Zufallsmatrix S: \n",S)

```

```
print("\nDreiecksmatrix T: \n",T)
print("\nUnitäre Matrix U: \n",U)
```

G Schur-Zerlegung des AII^\dagger -Ensembles

```
import numpy as np
from scipy.linalg import schur

N =10 #Dimension der matrix: NxN
M=int(N/2)

a=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
d=a.transpose()

v=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
v=(v*v.transpose())/2
b=np.triu(v, 1) - np.tril(v, -1)
w=np.random.randn(M,M)+complex(0,1)*np.random.randn(M,M)
w=(w*w.transpose())/2
c=np.triu(w, 1) - np.tril(w, -1)

S = complex(0,1)*np.ones((N,N))
S[0:M,0:M]=a
S[0:M,M:N]=b
S[M:N,0:M]=c
S[M:N,M:N]=d

T, U=schur(S, output='complex')

print("S=UTU^\N{DAGGER}\n")
print("Zufallsmatrix S: \n",S)
print("\nDreiecksmatrix T: \n",T)
print("\nUnitäre Matrix U: \n",U)
```


Eidesstattliche Erklärung zur Prüfungsleistung

Ich habe die Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und bisher keiner anderen Prüfungsbehörde vorgelegt. Außerdem bestätige ich hiermit, dass die vorgelegten Druckexemplare und die vorgelegte elektronische Version der Arbeit identisch sind, dass ich über wissenschaftlich korrektes Arbeiten und Zitieren aufgeklärt wurde und dass ich von den §24 Abs. 5 vorgesehenen Rechtsfolgen Kenntnis habe.

Regensburg, 01.09.2021

Ort, Datum

Unterschrift