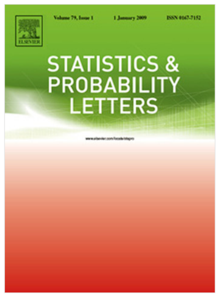
Journal Pre-proof



A note on the exact simulation of a random eigenvalue of a GUE matrix

Luc Devroye, Jad Hamdan

PII: S0167-7152(26)00100-8
DOI: https://doi.org/10.1016/j.spl.2026.110736
Reference: STAPRO 110736

To appear in: *Statistics and Probability Letters*

Received date : 6 July 2025
Revised date : 10 March 2026
Accepted date : 11 March 2026

A note on the exact simulation of a random eigenvalue of a GUE matrix

Luc Devroye* and Jad Hamdan†

*School of Computer Science, McGill University,
†Mathematical Institute, University of Oxford

Abstract. We develop a simple, *exact* algorithm to generate random variables whose density is proportional to squared Hermite functions. As an application, we show how to generate a randomly chosen eigenvalue of a matrix from the Gaussian Unitary Ensemble (GUE) in $O(n^{2/3})$ expected time.

Keywords. Random variate generation, orthogonal polynomials, Hermite functions, rejection method, random matrices, Gaussian unitary ensemble, eigenvalues.

1. Introduction

In this note, we concern ourselves with the exact generation of a random eigenvalue of a matrix in the Gaussian Unitary Ensemble GUE(n), which is given by the Gaussian measure with density

$$(2^{n/2}\pi^{n^2/2})^{-1}e^{-(n/2)\,\mathrm{tr}H^2}$$

on the space of $n \times n$ Hermitian matrices $H = (H_{ij})_{i,j=1}^n$ [3, 11]. That is, the diagonal elements are normal with zero mean and unit variance, and the off-diagonal elements are conjugate complex with independent real and imaginary parts that are both zero-mean normals with variance 1/2. By *exact* generation, we mean that the output distribution matches the target distribution perfectly, with no approximation error. This is sometimes referred to as *perfect simulation* in the literature [22].

The problem of computing eigenvalues of a (deterministic) matrix is classical. This yields a natural first group of methods for the random case: one starts from the matrix H and uses iterative numerical methods to compute the set of eigenvalues, noting that the Abel-Ruffini theorem (see [1], [36]) implies that there is no exact finite-time algorithm that finds roots of polynomials of degree greater than 5. These methods must take time $\Omega(n^2)$. If one is satisfied with a given accuracy level, then some of these approximations can be computed in time $O(n^2)$. Prominent among these numerical methods are the QR algorithm [17, 18, 26, 42] and its variations, Lanczos' algorithm [29], the Rayleigh quotient iteration [24] for Hermitian matrices, or, more generally, power iteration [40].

The second group of methods uses numerical approximation but starts from a different random matrix H' with the property that the vector of eigenvalues has the same distribution as that of H. Following Dumitriu and Edelman [13], we note that one can take H' tridiagonal, with i.i.d. normal $(0,1)$ random variables N_i on the diagonal, and with two identical adjacent diagonals filled with independent random variables $\sqrt{G_1}, \ldots, \sqrt{G_{n-1}}$, where G_i is gamma (i). In other words, $H'_{i,i} = N_i$, and $H'_{i,i+1} = H'_{i+1,i} = \sqrt{G_i}$, $1 \leq i \leq n-1$, and so H' is a random Jacobi matrix. Generating H' takes linear time. In addition, a divide-and-conquer algorithm [9] can compute the spectrum of H' in time $O(n \log n)$ if one only requires the answer up to a fixed precision.

The third group of methods uses stochastic tools to approximate the vector of eigenvalues, based on Markov chain convergence, with rates of convergence dependent upon the mixing times of these Markov chains or Gibbs samplers [25, 19].

The fourth gaggle of methods relies on the observation that for some integrable random matrix ensembles (of which the GUE is an example), the eigenvalues form a so-called determinantal point process (DPP) [33]. This is useful, as the literature on sampling DPPs is vast and includes both approximate [27, 2, 32] and *exact* algorithms [21, 30]. The main exact algorithm, sometimes referred to as HKPV [21], draws the entire vector of eigenvalues of H by sampling from a sequence $\{p_i(x)\}_{i=1}^n$ of appropriately constructed marginals. This procedure has complexity $O(n^3)$ assuming one can sample from each p_i in constant time, but recent surveys [31, 19, 20] report that these sampling subroutines have a poorly understood complexity. Noting that p_1 is precisely the density of a randomly selected eigenvalue of H, we still have to find an efficient and exact sampler for this quantity, which we recall is our primary objective.

For $n \leq 4$, one can generate H and compute the roots of its characteristic polynomial. Thus, we will describe a method for $n \geq 5$. In Section 2, we develop an algorithm to sample a random uniformly selected GUE(n) eigenvalue and prove that its expected runtime is sublinear in n. We rely on the fact that our problem can be reduced to generating random variables from densities which can be expressed in orthogonal polynomials. We then sample these using the rejection method. Earlier investigations in this direction [5, 6] used a different proposal density and were strictly empirical, leaving the question of explicit rejection bounds open before this work. To our knowledge, no other exact algorithm with sublinear expected time is available today.

1.1. Motivation and applied context

Random matrices have become increasingly central across a remarkable range of disciplines, from the physics of complex quantum systems [43] to modern statistics [4], wireless communications [10], machine learning [35], and data science [28]. The GUE is among the most studied of these models, occupying a central role in mathematical physics [34, 11] and serving as a canonical example of the broad universality class of random Hermitian matrices whose empirical spectral measure converges to the semicircle law [43, 15].

This convergence is typically defined and studied through *linear spectral statistics*, meaning observables of the form

$$\mathcal{L}_n(f) := \frac{1}{n}\sum_{i=1}^{n} f(\lambda_i),$$

where $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of a GUE(n) matrix and f is a test function. Monte Carlo estimation of $\mathbf{E}\mathcal{L}_n(f)$ requires repeated independent draws from the eigenvalue distribution, and it is here that exact simulation becomes essential. Beyond spectral statistics, exact GUE eigenvalue simulation has recently found a concrete application in the debiasing of active polynomial regression algorithms [8]. Their procedure requires sampling from the leverage score distribution, which in their setting coincides exactly with that of a uniformly chosen GUE eigenvalue.

In all such settings, the full matrix is often incidental: the object of interest is a single eigenvalue or a function thereof, which motivates sampling it directly without ever forming the matrix or computing the full spectrum. The present algorithm does precisely this: it returns a single exact eigenvalue in $O(n^{2/3})$ expected time, far below the $O(n^2)$ cost of forming the matrix and the $O(n^3)$ cost of diagonalization, and with no asymptotic approximation involved. For large n, this reduction can be transformative, by lowering hardware requirements and dramatically increasing the sample counts achievable in Monte Carlo studies.

1.2. The RAM model

We operate under the Random Access Machine (RAM) model: real numbers can be stored and operated upon in constant time. We also assume that all the standard operations, as well as the exponential, logarithmic, trigonometric, and gamma functions, can be computed in constant time. Finally, a source capable of producing an i.i.d. sequence of uniform $[0,1]$ random variables is available. The RAM model provides a natural and widely accepted framework for analyzing the complexity of numerical algorithms such as ours. It assumes constant-time access to memory and constant-time arithmetic on fixed-size words, closely matching the abstraction level at which modern processors operate. For floating-point computations and array-based data structures, this model gives reliable asymptotic scaling laws and is the standard in both theoretical computer science and numerical analysis. In our setting, where the principal comparison is between an $O(n^2)$ matrix-generation procedure (followed by $O(n^3)$ diagonalization) and an $O(n^{2/3})$ eigenvalue-sampling method, the RAM model accurately captures the dominant computational trade-offs. Indeed, the large polynomial separation in complexity ensures that the asymptotic advantage is meaningful in practice, not merely theoretical.

At the same time, the RAM model abstracts away features of real hardware that can influence performance. It ignores memory hierarchy effects (cache, bandwidth, latency), assumes uniform cost of memory access, and treats arithmetic operations as constant-time regardless of context. In practice, memory traffic is often a bottleneck, and algorithms with better locality can significantly outperform others even at equal asymptotic complexity. However, in our context, these deviations do not weaken the theoretical conclusion: if anything, they strengthen it. Because matrix-based methods incur substantial memory allocation and data movement costs, the RAM model likely underestimates the practical advantage of an algorithm that avoids matrix construction altogether. Thus, while simplified, the RAM assumption is both appropriate and conservative for evaluating the computational benefit of our approach.

2. The density of a random GUE eigenvalue

For a distribution on sets of n points, the $k-$point correlation function $\sigma_k(x_1, ..., x_k)$ describes the induced probability distribution on uniformly selected subsets of size $k \leq n$. It is typically normalized so that $\frac{1}{n}\sigma_n(x)$ is the probability density function of a uniformly selected (set of 1) point. It is well-known (see [14]) that for GUE(n) eigenvalues, the $k-$point correlation function is given by a $k \times k$ determinant $\det(K(x_1, x_2))_{1 \leq i,j \leq k}$, where

$$K(x, y) = \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{n-1} \frac{H_k(x) H_k(y) e^{-(x+y)^2/4}}{k!} = \sum_{k=0}^{n-1} \phi_k(x) \phi_k(y).$$

In the expression above, H_k is the k-th *Hermite polynomial*

$$H_k(x) = (-1)^k e^{x^2/2} \left(\frac{\mathrm{d}}{\mathrm{d}x} \right)^k e^{-x^2/2},$$

and ϕ_k is the so-called *Hermite function* $H_k(x)e^{-x^2/4}/\sqrt{k!\sqrt{2\pi}}$ (in the theory of orthogonal functions, $K(x, y)$ is known as the *Christoffel-Darboux* kernel [38]). The probability density for the distribution of a single, uniformly selected eigenvalue from a GUE(n) matrix is thus equal to

$$\frac{1}{n} K(x, x) = \frac{1}{n} \sum_{k=0}^{n-1} \phi_k(x)^2. \tag{1}$$

We remark in passing that, using asymptotic properties of Hermite polynomials (see [34]), one recovers Wigner's famous semicircle law [43] from this expression in the large–n limit.

Next, we note that the Hermite polynomials are orthogonal. In particular, for any $k \in \mathbf{N}$, we have

$$\int_{-\infty}^{\infty} H_k(x)^2 e^{-x^2/2} \mathrm{d}x = \sqrt{2\pi}\, k!$$

(see [37] for a proof), which shows that ϕ_k^2 is a density for every k. It then follows from (1) that if one picks an index $k \in \{0, \ldots, n-1\}$ uniformly at random and generates a random variate described by the density ϕ_k^2, the result would be distributed as a uniformly selected GUE(n) eigenvalue. The remainder of this paper focuses on generating random variables described by the densities ϕ_k^2.

3. Generating random variates with squared Hermite densities

3.1. Notation and preliminaries on Hermite polynomials

The aforementioned Hermite polynomials can alternatively be defined using the recurrence $H_0 = 1$, $H_1(x) = x$, and

$$H_{k+1}(x) = xH_k(x) - kH_{k-1}(x) \tag{2}$$

for any $k \geq 2$, giving us a simple $O(k)$ algorithm to compute H_k for fixed x ([38]). We also have the following theorem, due to Bonan & Clarke [7], which gives an upper bound for ϕ_n^2 on $\mathbf{R}$. Note that we computed the explicit constants in that paper. Other inequalities could also have been used, such as those developed by Foster and Krasikov [16] and Krasikov [23].

Theorem 1. *For any $n \in \mathbf{N}$, let ϕ_n^2 be the square of the n-th Hermite function (defined in the previous section). Then*

$$\sup_{x\in\mathbf{R}} \phi_n^2(x) \leq \frac{8(\pi+1)}{3n^{1/6}}.$$

Furthermore,

$$\phi_n^2(x) \leq \begin{cases} \frac{8\pi}{3\sqrt{4n+2-x^2}}, & \text{if } |x| \leq \sqrt{4n+2}, \\ \frac{2\sqrt{2}B^2}{n^{5/6}\,(\sqrt{4n+2}-x)^4}, & \text{if } |x| > \sqrt{4n+2}, \end{cases}$$

where $B = (\pi+1)^2\sqrt{8(\pi+1)/3}$.

If we let

$$x_1 = x_1(n) = \sqrt{4n+2-\frac{\pi^2}{(\pi+1)^2}\,n^{1/3}}$$

and

$$x_2 = x_2(n) = \sqrt{4n+2} + \sqrt{B}\left(\frac{3}{2\sqrt{2}(\pi+1)}\right)^{1/4} n^{-1/6},$$

it follows directly from the theorem above that the function

$$h_n(x) \stackrel{\text{def}}{=} \begin{cases} \frac{8\pi}{3\sqrt{4n+2-x^2}}, & \text{if } |x| \leq x_1, \\ \frac{8(\pi+1)}{3n^{1/6}}, & \text{if } |x| \in (x_1, x_2], \\ \frac{2\sqrt{2}B^2}{n^{5/6}\,(\sqrt{4n+2}-x)^4}, & \text{if } |x| > x_2, \end{cases}$$

dominates ϕ_n^2 on all of $\mathbf{R}$.

The following lemma will be useful for analyzing our algorithm's expected runtime.

Lemma 2. *Let h_n be defined as above. Then*

$$\int_{\mathbf{R}} h_n(x)\,\mathrm{d}x = O(1), \qquad \int_{x_1}^{\infty} h_n(x)\,\mathrm{d}x = O(n^{-1/3}).$$

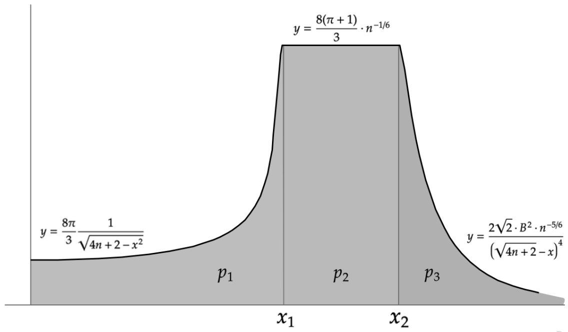


Figure 1. The function h_n, defined piecewise. As in Algorithm 1 (section 3.2), p_1, p_2 and p_3 denote the area under the curve in $[0, x_1)$, $[x_1, x_2)$ and $[x_2, \infty)$ respectively.

Proof. Let $n \in \mathbf{N}$ be arbitrary, then

$$\begin{aligned}
&\int_{\mathbf{R}} h_n \\
&= \int_0^{x_1} \frac{8\pi}{3} \frac{1}{\sqrt{4n+2-x^2}} \mathrm{d}x + \int_{x_1}^{x_2} \frac{8(\pi+1)}{3n^{1/6}} \mathrm{d}x + \int_{x_2}^{\infty} \frac{2\sqrt{2}B^2 n^{-5/6}}{(\sqrt{4n+2}-x)^4} \mathrm{d}x \\
&= \frac{8\pi}{3} \arcsin \frac{x_1}{\sqrt{4n+2}} + \frac{8(\pi+1)}{3n^{1/6}} |x_2 - x_1| + \frac{\sqrt{B}\,2\sqrt{2}}{3n^{1/3}} \left(\frac{2\sqrt{2}(\pi+1)}{3} \right)^{3/4}
\end{aligned}$$

and the lemma follows from the fact that $|x_2 - x_1| = O(n^{-1/6})$. ■

We will also need the following representation of Hermite polynomials due to van Veen [39].

Theorem 3. *For any* $n, k \in \mathbf{N}$, $x \in [0, 2\sqrt{n+1}]$,

$$H_n(x) = A_n(x)\big(B_n(x) + \mu R_n(x)\big)$$

where

$$\begin{aligned}
A_n(x) &\overset{\text{def}}{=} \frac{\Gamma(n+1)}{\pi} \frac{e^{(n+1)/2 + x^2/4}}{(n+1)^{n/2}}, \\
B_n(x) &\overset{\text{def}}{=} \frac{\sqrt{\pi}}{\{(n+1)\sin(\alpha)\}^{1/2}} * \sin\left\{ \frac{(n+1)}{2}\big(\sin(2\alpha) - 2\alpha\big) + \frac{\alpha}{2} + \frac{3\pi}{4} \right\}, \\
R_n(x) &\overset{\text{def}}{=} \frac{1}{3(n+1)\sin(\alpha)^2}, \\
\mu &\overset{\text{def}}{=} 3.1 + 1.1 * (\sin(\alpha/2 + 3 * \pi/4)/\sin(\alpha))^2,
\end{aligned}$$

and

$$\alpha = \alpha(x) = \arccos \frac{x}{2\sqrt{n+1}}.$$

This representation is then extended to $|x| \leq 2\sqrt{n+1}$ using the fact that H_n is even. On said domain, for any $n, k \in \mathbf{N}$, we use $V_n \overset{\text{def}}{=} A_n * B_n$ (as defined in the theorem above) to

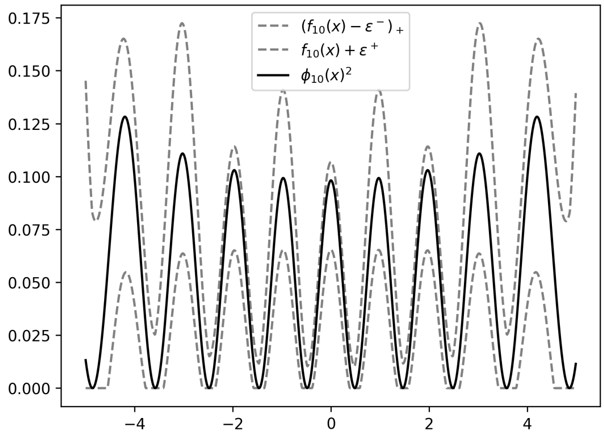


Figure 2. The function ϕ_{10}^2 (bold) bounded from below and above by $(f_{10} - \epsilon^-)_+$ and $(f_{10} + \epsilon^+)$ (dashed).

approximate H_n and define the following approximation to ϕ_n^2:

$$f_n(x) \stackrel{\text{def}}{=} \frac{V_n(x)^2 e^{-x^2/2}}{\sqrt{2\pi}\, n!}$$

(we define $f_n = 0$ if $|x| > 2\sqrt{n+1}$). Note that for any $|x| < 2\sqrt{n+1}$,

$$|\phi_n^2(x) - f_n(x)| = \left(\left(\mu R_n(x)\right)^2 + 2\mu R_n(x) B_n(x)\right) \frac{A_n^2(x) e^{-x^2/2}}{\sqrt{2\pi}\, n!}.$$

Using the fact that $|\mu| < 4.2$, it follows that

$$(\phi_n^2(x) - f_n(x))_+ \le \epsilon^+(x) \stackrel{\text{def}}{=} \frac{A_n^2(x) e^{-x^2/2}}{\sqrt{2\pi}\, n!}\left((8.4)(B_n(x))_+ R_n(x) + (4.2)^2 R_n(x)^2\right)$$
$$(\phi_n^2(x) - f_n(x))_- \le \epsilon^-(x) \stackrel{\text{def}}{=} \frac{A_n^2(x) e^{-x^2/2}}{\sqrt{2\pi}\, n!}(8.4)\left(\left(B_n(x) R_n(x)\right)_+ \vee \left(B_n(x) R_n(x)\right)_-\right)$$

(we used $a \wedge b$ (resp. $a \vee b$) to denote $\min(a, b)$ (resp. $\max(a, b)$) and $(x)_+ = (x \vee 0)$, $(x)_- = (-x \vee 0)$). Note that $\epsilon^{+/-}$ are positive by definition and

$$\left(f_n - \epsilon^-\right)_+ \le \phi_n^2 \le \left(f_n + \epsilon^+\right) \wedge h_n \tag{3}$$

for any $x \in \mathbf{R}$. Lastly, we define $\Delta_\epsilon(x)$ to be the gap between the upper and lower bound for $\phi_n^2(x)$ and note that there exists a constant C for which

$$\Delta_\epsilon(x) \le C\left(|R_n(x)|^2 + |R_n(x) B_n(x)|\right) \frac{A_n^2(x) e^{-x^2/2}}{\sqrt{2\pi}\, n!}$$

and the following proposition holds.

Proposition 4. *For large enough n,*

$$\int_0^{x_1} \Delta_\epsilon(x)\, \mathrm{d}x = O\left(n^{-1/3}\right).$$

Proof. See the appendix. ■

3.2. Generating random variates with density $h_n / \int h_n$

Our main algorithm in the following section requires the generation of random variates with density $h_n / \int h_n$. This can be done in constant expected time using the following algorithm, which is a straightforward application of the inversion method (see [12] for an exposition of this method).

Algorithm 1:

Set $B \leftarrow (\pi+1)^2 \sqrt{8(\pi+1)/3}$.

Set $p_1 \leftarrow \int_0^{x_1} h_n = \frac{8\pi}{3} \arcsin \frac{x_1}{\sqrt{4n+2}}$.

Set $p_2 \leftarrow \int_{x_1}^{x_2} h_n = \frac{8(\pi+1)}{3} n^{-1/6} \left| x_2 - x_1 \right|$.

Set $p_3 \leftarrow \int_{x_2}^{\infty} h_n = \sqrt{B} \cdot \left(\frac{2\sqrt{2}}{3} \right)^{7/4} (\pi+1)^{3/4} n^{-1/3}$.

Generate S, where $\mathbf{P}\{S=1\} = \mathbf{P}\{S=-1\} = 1/2$.

Generate U uniformly on $[0,1]$.

If $U < p_1/(p_1+p_2+p_3)$ then

Generate V uniformly on $[0,1]$.

$X \leftarrow \sqrt{4n+2} \left| \sin(V * \arcsin(x_1/\sqrt{4n+2})) \right|$

else if $p_1/(p_1+p_2+p_3) \leq U < (p_1+p_2)/(p_1+p_2+p_3)$ then

Generate V uniformly on $[0,1]$.

$X \leftarrow x_1 + (x_2 - x_1) * V$

else

Generate V uniformly on $[0,1]$.

$X \leftarrow \sqrt{4n+2} + \sqrt{B} \left(\frac{3}{2\sqrt{2}(\pi+1)} \right)^{1/4} n^{-1/6} * V^{-1/3}$

return $X * S$

3.3. A first eigenvalue algorithm that is linear in n

For any n, we now know how to generate from $h_n / \int h_n$ and that h_n dominates ϕ_n^2. The following rejection algorithm (see [12] and [41]) will be used as a stepping stone towards our main algorithm, and will be shown to run in linear time in the following section.

Algorithm 2:

Repeat until Accept

Generate X from $h_n / \int h_n$

Generate U uniformly on $[0,1]$

Accept $\leftarrow \left[U * h_n(X) \leq \phi_n^2(X) \right]$

return X

Note that the comparison requires us to compute $\phi_n^2(X)$, which we do using the recurrence relation for Hermite polynomials (2) given earlier.

3.4. Main algorithm and runtime analysis

We can refine this second algorithm using van Veen's estimate for H_n stated in Theorem 3. Let ϕ_n^2, f_n, h_n, and $\epsilon^{+/-}$ be defined as above.

```
Algorithm 3:
    Repeat forever
        Generate X from h_n/∫h_n
        Generate U uniformly on [0,1]
        If        U*h_n(X) ≤ (f_n(X) − ε^−(X))_+ then return X
        else if U*h_n(X) ≤ (f_n(X) + ε^+(X)) then
                if U*h_n(X) ≤ φ_n^2(X) then return X
```

We emphasize that f_n, h_n and $\epsilon^{+/-}$ can be computed in $O(1)$ time, and note that in the last line, ϕ_n^2 is computed using the recurrence formula given above, as in Algorithm 2.

Fix $n \in \mathbf{N}$, and let N be the number of iterations of Algorithm 3 when generating from ϕ_n^2. Let T_i be the time taken by the i-th iteration, where $1 \le i \le N$, then the total runtime of the algorithm is $\sum_{i=1}^{N} T_i$. Since N is a stopping time, applying Wald's identity (see [44]) yields

$$\mathbf{E}\left\{\sum_{i=1}^{N} T_i\right\} = \mathbf{E}\{N\}\mathbf{E}\{T_1\}.$$

We have

$$\mathbf{E}\{N\} = \int h_n = O(1),$$

and using the fact that ϕ_n^2 can be computed in $O(n)$ time with the recurrence for H_n explained above, and that one can sample from $h_n/\int h_n$ in constant time using Algorithm 2 (cf. 3.2), we conclude that

$$\mathbf{E}\{T_1\} = O\left(n \cdot \left(\int_0^{x_1} \Delta_\epsilon(x)\,\mathrm{d}x + \int_{x_1}^{\infty} h_n(x)\,\mathrm{d}x\right)\right).$$

It then follows from Proposition 4 and Lemma 2 that $\mathbf{E}\{T_1\} = O(n^{2/3})$, and, in turn, that this refined algorithm is sublinear with an expected runtime of the same order. On the other hand, replacing f_n with 0 and ϵ^+ with ϕ_n^2 reduces the above algorithm to Algorithm 2 and shows that the latter runs in linear expected time.

4. Appendix: Proof of Proposition 4.

We denote $f(x) = O(g(x))$ by $f \ll g$. It suffices to show that

$$I(0, x_1) = \int_0^{x_1} \left(|R_n(x)B_n(x)| + R_n(x)^2\right) \frac{A_n^2(x)e^{-x^2/2}}{\sqrt{2\pi}\, n!}\, dx$$

is of the desired order, since $\Delta_\epsilon \ll I(0, x_1)$. We begin with an application of Stirling's approximation, which yields

$$\frac{A_n(x)^2 e^{-x^2/2}}{\sqrt{2\pi} n!} \ll \sqrt{n}.$$

If $|x| \leq \sqrt{3(n+1)}$, we have $\sin\alpha > 1/2$, $B_n(x) = O(n^{-1/2})$ and $R_n(x) = O(n^{-1})$, so that

$$I\left(0, \sqrt{3(n+1)}\right) \ll \sqrt{n}\int_0^{\sqrt{3(n+1)}} \left(|B_n(x)R_n(x)| + R_n(x)^2\right) \mathrm{d}x = O\left(\frac{1}{\sqrt{n}}\right).$$

To bound $I(\sqrt{3(n+1)}, x_1)$, use Stirling's approximation as above,

$$I(\sqrt{3(n+1)}, x_1) \ll \int_{\sqrt{3(n+1)}}^{x_1} \frac{1}{n\sin(\alpha(x))^{5/2}} + \frac{1}{n^{3/2}\sin(\alpha(x))^4}\,\mathrm{d}x,$$

apply the following technical lemma (taking $\xi = 1/3$) and conclude that the integral is $O(n^{-1/3})$.

Lemma 5. *Let $x = \sqrt{4(n+1) - \gamma n^\xi}$ for some $\xi \in (0,1)$ and any $\gamma \in \mathbf{R}_{>0}$ for which $\gamma n^\xi < 4(n+1)$. Then for $\beta > 2$,*

$$\int_0^x \sin(\alpha(t))^{-\beta}\,\mathrm{d}t \ll n^{(\xi-1/2)+\beta(1-\xi)/2}.$$

for n large enough.

Proof of Proposition 4. Recall that $\alpha(t) = \arccos\left(t/\sqrt{4(n+1)}\right)$ for $0 \leq t < \sqrt{4(n+1)}$, giving

$$-\sin\alpha\,\mathrm{d}\alpha = \frac{\mathrm{d}t}{\sqrt{4(n+1)}}$$

and in turn, for any $x = \sqrt{4(n+1) - \gamma n^\xi}$ where $\gamma \in \mathbf{R}_{>0}$, $\xi \in (0,1)$,

$$\int_0^x \sin\left(\alpha(t)\right)^{-\beta}\mathrm{d}t = -\int_{\arccos\left(x/\sqrt{4(n+1)}\right)}^{\pi/2} \sqrt{4(n+1)}(\sin\alpha)^{1-\beta}\mathrm{d}\alpha.$$

Since $\alpha \in (0, \pi/2]$, we have

$$\frac{2}{\pi} \leq \frac{\sin(\alpha)}{\alpha} \leq 1,$$

from which it follows that the integral above is smaller than

$$C\sqrt{4(n+1)} * \left(\frac{1}{\arccos\left(x/\sqrt{4(n+1)}\right)}\right)^{\beta-2} \tag{4}$$

for a constant C that can depend on β, assuming $\beta > 2$. As $\arccos(x) \geq \sqrt{1-x^2}$ for any $|x| \leq 1$, if $x = \sqrt{4(n+1)(1-\delta)}$ for $\delta < 1$, (4) is bounded from above by

$$C\sqrt{4(n+1)} * \left(\frac{1}{\sqrt{1 - x^2/4(n+1)}}\right)^{\beta-2} \leq C\sqrt{4(n+1)}\,\delta^{-(\beta-2)/2} \ll \sqrt{n\delta^{-(\beta-2)}}.$$

The lemma then follows after setting $\delta = n^{\xi-1}$ for any $\xi \in (0,1)$.

5. Acknowledgments

L. D. is supported by NSERC Grant RGPIN-2024-04164 and J. H. is supported by the EP-SRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation (EP/S023925/1).

References

[1] N. H. Abel. Mémoire sur les équations algébriques, ou l'on démontre l'impossibilité de la résolution de l'équation générale du cinquième degré. In *Œuvres Complètes de Niels Henrik Abel*, pages 28–33. Grøndahl & Søn, 1824.

[2] R. H. Affandi, A. Kulesza, E. Fox, and B. Taskar. Nyström approximation for large-scale determinantal processes. In C. M. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 85–98, Scottsdale, Arizona, USA, 2013. PMLR.

[3] G. W. Anderson, A. Guionnet, and O. Zeitouni. *An Introduction to Random Matrices*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2010.

[4] J. Baik, G. Ben Arous, and S. Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Annals of Probability*, 33(5):1643–1697, 2005.

[5] N. Baskerville. A novel sampler for Gauss-Hermite determinantal points processes. *Available at SSRN 4330239*, 2022.

[6] N. P. Baskerville. A novel sampler for Gauss-Hermite determinantal point processes with application to monte carlo integration. *arXiv:2203.08061*, 2022.

[7] S. S. Bonan and D. S. Clarke. Estimates of the Hermite and the Freud polynomials. *Journal of Approximation Theory*, 63:210–224, 1990.

[8] C. Camaño, R. A. Meyer, and K. Shu. Debiasing polynomial and Fourier regression. In *Proceedings of the 2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 340–351. Society for Industrial and Applied Mathematics, 2026.

[9] E. Coakley and V. Rokhlin. A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices. *Applied and Computational Harmonic Analysis*, 34:379–414, 2013.

[10] R. Couillet and M. Debbah. *Random Matrix Methods for Wireless Communications*. Cambridge University Press, 2011.

[11] P. Deift. *Orthogonal Polynomials and Random Matrices: a Riemann-Hilbert Approach*. American Mathematical Society, 2000.

[12] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, New York, 1986.

[13] I. Dumitriu and A. Edelman. Matrix models for beta ensembles. *Journal of Mathematical Physics*, 43(11):5830–5847, 2002.

[14] A. Edelman and M. L. Croix. The singular values of the GUE (less is more). *Random*

Matrices: Theory and Applications, 04, No. 04, 2014.

[15] L. Erdős, H.-T. Yau, and J. Yin. Bulk universality for generalized Wigner matrices. *Probability Theory and Related Fields*, 154(1–2):341–407, 2012.

[16] W. Foster and I. Krasikov. Explicit bounds for hermite polynomials in the oscillatory region. *London Mathematical Society Journal of Computational Mathematics*, 3:307–314, 2000.

[17] J. Francis. The QR transformation. *The Computer Journal*, 4(3):265–271, 1961.

[18] J. Francis. The QR transformation II. *The Computer Journal*, 4(4):332–345, 1961.

[19] G. Gautier, R. Bardenet, and M. Valko. Fast sampling from β-ensembles. *Statistics and Computing*, 31(1):1–20, 2021.

[20] M. Guillaume and J. Gautier. *On sampling determinantal point processes.* PhD thesis, Centrale Lille Institut, 2020.

[21] J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal processes and independence. *Probability Surveys*, 3:206–229, 2006.

[22] M. L. Huber. *Perfect Simulation.* Number 148 in Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, 2015.

[23] I. Krasikov. New bounds on the Hermite polynomials. *arXiv:math/04011310v1*, 2004.

[24] R. Kress. *Numerical Analysis.* Springer, 1991.

[25] M. Krishnapur, B. Rider, and B. Virág. Universality of the stochastic Airy operator. *Communications on Pure and Applied Mathematics*, 69(1):145–199, 2016.

[26] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics*, 1(3):637–657, 1963.

[27] A. Kulesza and B. Taskar. *Determinantal Point Processes for Machine Learning.* Now Publishers Inc., Hanover, MA, USA, 2012.

[28] D. Kunisky. Lecture notes on random matrix theory in data science and statistics, 2024. Yale University lecture notes, last updated December 2024. Available at `http://www.kunisky.com/static/teaching/2024fall-rmt/rmt-notes-2024.pdf`.

[29] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

[30] C. Launay, B. Galerne, and A. Desolneux. Exact sampling of determinantal point

processes without eigendecomposition. *Journal of Applied Probability*, 57(4):1198–1221, 2020.

[31] F. Lavancier and E. Rubak. On simulation of continuous determinantal point processes. *Statistics and Computing*, 33:1–19, 2023.

[32] C. Li, S. Jegelka, and S. Sra. Fast mixing markov chains for strongly rayleigh measures, dpps, and constrained sampling. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4195–4203, Red Hook, NY, USA, 2016. Curran Associates Inc.

[33] O. Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7:83–122, 1975.

[34] M. L. Mehta. *Random Matrices*. Elsevier, 3rd edition, 2004.

[35] J. Pennington and P. Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[36] P. Ruffini. *Riflessioni Intorno Alla Soluzione Delle Equazioni Algebraiche*. Presso la Società Tipografica, 1813.

[37] G. Sansone. *Orthogonal Functions*. Robert E. Krieger Publishing Company, Inc., 1977.

[38] G. Szegö. *Orthogonal Polynomials*, volume 23 of *Colloquium Publications*. American Mathematical Society, 1931.

[39] S. C. van Veen. Asymptotische Entwicklung und Nullstellenabschätzung der Hermiteschen Funktionen. *Mathematische Annalen*, 105:408—-436, 1931.

[40] R. von Mises and H. Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *Zeitschrift für Angewandte Mathematik und Mechanik*, 9:152–164, 1929.

[41] J. von Neumann. Various techniques used in connection with random digits. *Journal of Research of the National Bureau of Standards, Applied Mathematics Series*, 3:36–38, 1951.

[42] D. S. Watkins. Understanding the QR algorithm. *SIAM Review*, 24(4):427–440, 1982.

[43] E. P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *Annals of Mathematics*, 62(3):548–564, 1955.

[44] D. Williams. *Probability with Martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, 1991.