The University of Melbourne,
Department of Mathematics and Statistics

Relating Random Matrix Theory to Queueing Theory

Lauren Trumble

SUPERVISOR: PETER FORRESTER
SECOND READER: NICHOLAS WITTE

Honours Thesis, November 2006.

Contents

Acknowledgements

Firstly I would like to sincerely thank Peter Forrester for being a fantastic supervisor. Peter has always been incredibly patient with me and has always comforted me when I was stressed. Despite Peter being extremely busy this year, he always made time for me and has been inspiring. Thank you Peter I could not have asked for a better supervisor.

To my fellow honours classmates and friends, you have all been wonderful. Thank you for always taking the time out to help me and answer my questions, special thanks to Dave Glove and Hannah.

Thank you also to my parents who have been wonderfully patient and for deep down having faith in me despite me giving them many reasons for doubting that I would ever finish university. Thank you also to Anna and Hayden for their endless disruptions but also for their endless support.

Chapter 1

Introduction

The main theme of this thesis is to relate probability distributions of two seemingly different topics: random matrix theory and queueing processes.

Random matrix theory is used widely in theoretical physics for its accurate prediction of a number of observable quantities. Some examples include the statistical properties of highly excited energy levels of heavy nuclei and the conductance fluctuations in mesoscopic wires, [**?**] and [**?**]. In the application to heavy nuclei, the Hamiltonian is approximated by an $N \times N$ matrix. The Hamiltonian must have real eigenvalues and as a result of an axiom of quantum mechanics, this $N \times N$ matrix must be Hermitian. The entries of the Hermitian matrix are not known and to compensate for this, the entries of the matrix are taken to be random variables from a probability distribution (there may be different distributions for the diagonal elements and the off diagonal elements). However there is one other feature of the physical problem which can be accounted for in the random matrix model. This is the feature that for physical systems with a time reversal symmetry, all the elements of the matrix are taken to be real. The reason for this is that in quantum mechanics, the time reversal operator corresponds to complex conjugation, so for a Hermitian matrix to be invariant under complex conjugation it must be real symmetric. In the theoretical description of the conductance of mesoscopic wires, the relevant quantity is $M^T M$ where M is the transfer matrix describing the passage of the plane wave states through the system. Again a time reversal symmetry would restrict the entries of M to be real, but in this problem there is no other restriction (note that in general, matrices of the form $M^T M$ have all eigenvalues non-negative).

In the application of random matrix theory to queueing processes, both random Hermitian matrices and random positive definite matrices are of relevance. However rather then having real elements as in the applications to physical problems with time reversal symmetry, the elements are complex. In a physical problem, time reversal symmetry can be broken by the application of a magnetic field. Another peculiar feature particular to the queueing application is that it is the distribution of the largest eigenvalue that is the quantity of primary interest, where previously it was the bulk or middle of the range eigenvalues for the nuclear spectrum problem and the smallest eigenvalues for the quantum conductance problem. In chapter **??** the eigenvalue probability density function for complex Hermitian matrices with Gaussian entries is calculated, as is the eigenvalue probability density function for random positive definite matrices $M^T M$ with the elements of M complex Gaussians. The characteristic polynomial for both these classes of random matrices satisfies a random three term recurrence. Computing the largest zero of these polynomials allows the empirical distribution of respective largest eigenvalues to be computed. These are presented in a histogram, after appropriate centering and scaling.

In chapter **??**, the task of computing the exact form of the scaled distribution of the largest eigenvalue is considered, in the $N \to \infty$ limit this distribution is the same for both complex Hermitian matrices and positive definite matrices. The exact form is given in terms of a solution of a special non-linear equation called the Painlevé II equation. To obtain high precision statistical characterisation of the distribution, a power series solution of the differential equation is obtained (using computer algebra) at unit intervals for a large range of the independent variable. From this, the corresponding power series for the distribution itself are obtained at the same unit intervals.

Queueing theory can be applied to many different queueing processes, from people waiting in a bank queue, to products passing through various stages of production. This thesis considers queues in series. The specific model considered here is described by a series of n single servers with unlimited waiting space and a first-in first-out service. Chapter **??** describes this specific queueing model and the different ways it can be represented. One of the basic representations is a matrix of positive entries corresponding to queueing times for each job in each queue. This data can also be represented as a set of growth models which are described by a series of non-intersecting lattice paths. The Robinson-Schensted-Knuth correspondence [**?**] shows a bijection between non-negative matrices and pairs of semi-standard tableaux called Young tableaux. A fundamental observable quantity relating to the queueing process is the exit time of the final job from the final queue. When the service times are random variables chosen from the exponential distribution, an analytic formula can be obtained for the probability distribution of this exit time. This formula is precisely that for the distribution of the largest eigenvalue in the Laguerre Unitary Ensemble of random complex positive definite matrices. Its limiting form is the distribution described in chapter **??**, giving a limit formula for exit times in the queueing process.

Chapter 2

Empirical Distribution of an Eigenvalue Probability Density Function

The basic idea of random matrix theory is to select $N \times N$ matrices according to a given probability measure, then to diagonalise these matrices and record the statistical properties of the eigenvalues. To record accurate statistical properties of the eigenvalues we must repeat the process a large number of times and we present the frequencies in a histogram. This idea will be applied to Hermitian matrices and positive definite matrices with complex Gaussian entries and the chosen statistical property will be the scaled distribution of the largest eigenvalues.

2.1 Complex Hermitian Matrices

Definition 1. *An Hermitian matrix is a square matrix with real and complex entries such that it is equal to its own conjugate transpose* ($X = \overline{X}^T$)*, which is also called the Hermitian adjoint. Equivalently:*

for each entry $x_{jk}(j \leq k)$ *it is required that* $x_{jk} = \overline{x}_{kj}$

For example the matrix

$$\begin{bmatrix} 1 & 3-i & 0 \\ 3+i & 3 & -2i \\ 0 & 2i & 0 \end{bmatrix}$$

is Hermitian. As a result of the definition of a Hermitian matrix, only real entries can exist on the main diagonal. In general matrix diagonalisation involves transforming a square matrix X by a similarity transformation $X \mapsto B^{-1}XB$ to obtain a diagonal matrix L. Let $L = diag(\lambda_1, \ldots \lambda_N)$ and $B = [\vec{b_k}]_{k=1,\ldots N}$ where $\vec{b_k}$ thus denotes the kth column of B. Then the equation $B^{-1}XB = L$

is seen to be equivalent to the eigenvalue equations

$$X\vec{b_k} = \lambda_k \vec{b_k} \qquad (k = 1, \ldots, n)$$

It is known, [**?**] that for an Hermitian matrix X, $B = U$ where U is a unitary matrix such that $\overline{U}^T U = I$, where I is the identity matrix. Thus the eigenvectors are mutually orthogonal and they have their length normalised to unity. In the case that the eigenvalues are distinct, the former property is straightforward to establish.

Proposition 1. *For an Hermitian matrix,*

a) the eigenvalues λ are real and

b) for distinct eigenvalues, λ_1 and λ_2 and corresponding eigenvectors $\mathbf{u_1}$ and $\mathbf{u_2}$ the orthogonality $\mathbf{u_1} \cdot \overline{\mathbf{u_2}} = 0$ holds.

Proof a)
For an eigenvector $\mathbf{u}$ we have the eigenvalue equation

$$X\mathbf{u} = \lambda\mathbf{u} \tag{2.1.1}$$

Taking the conjugate transposes of both sides we get

$$\overline{X}^T \overline{u}^T = \overline{\lambda}\overline{\mathbf{u}}^T$$

As X is an $N \times N$ Hermitian matrix, and thus $X = \overline{X}^T$ it follows that the right hand side can be rewritten

$$\overline{\mathbf{u}}^T X = \overline{\lambda}\overline{\mathbf{u}}^T$$

Acting on both sides with $\mathbf{u}$ gives

$$\mathbf{u}\overline{\mathbf{u}}^T X = \mathbf{u}\overline{\lambda}\overline{\mathbf{u}}^T$$

Multiplying equation **??** through by $\overline{\mathbf{u}}^T$ gives

$$\overline{\mathbf{u}}^T X\mathbf{u} = \overline{\mathbf{u}}^T \lambda\mathbf{u}$$

Hence

$$\mathbf{u}\overline{\lambda}\overline{\mathbf{u}}^T = \overline{\mathbf{u}}^T \lambda\mathbf{u}$$

As $\mathbf{u} \neq 0$, so that the dot product $\overline{\mathbf{u}}^T \cdot \mathbf{u} > 0$ we must have $\overline{\lambda} = \lambda$, so the eigenvalues are real as required.

Proof b)
We begin with the eigenvalue equation

$$X\mathbf{u_1} = \lambda_1\mathbf{u_1}$$

Acting on both sides with $\overline{\mathbf{u}}_2^T$ gives

$$\overline{\mathbf{u}}_2^T X\mathbf{u_1} = \lambda_1\overline{\mathbf{u}}_2^T\mathbf{u_1}$$

As X is Hermitian it follows that the right hand side can be rewritten

$$\begin{aligned}\overline{\mathbf{u}}_2^T X\mathbf{u_1} &= (\overline{X}\overline{\mathbf{u}}_2)^T\mathbf{u_1}\\ &= \lambda_2\overline{\mathbf{u}}_2^T\mathbf{u_1}\end{aligned}$$

Hence

$$\lambda_1\overline{\mathbf{u}}_2^T\mathbf{u_1} = \lambda_2\overline{\mathbf{u}}_2^T\mathbf{u_1}$$

As λ_1 and λ_2 are distinct this requires $\overline{\mathbf{u}}_2^T\mathbf{u_1} = \mathbf{u_1}\cdot\overline{\mathbf{u}}_2 = 0$, which gives us the desired result. □

2.2 The Gaussian Unitary Ensemble

Definition 2. *A random Hermitian $N \times N$ matrix X is said to belong to the Gaussian Unitary Ensemble or GUE if the diagonal entries, x_{jj} and the upper triangular entries, $x_{jk} = u_{jk} + iv_{jk}$ are independently chosen with the following probability density functions*
The diagonal entries

$$\frac{1}{\sqrt{\pi}}e^{-x_{jj}^2}$$

The upper triangle entries

$$\frac{2}{\pi}e^{-2(u_{jk}^2+v_{jk}^2)} = \frac{2}{\pi}e^{-2|x_{jk}|^2}$$

Equivalently, the diagonal entries have Gaussian distribution $N[0, \frac{1}{\sqrt{2}}]$ and the upper triangular entries have Gaussian distribution $N[0, \frac{1}{2}] + iN[0, \frac{1}{2}]$.

The probability distribution of the matrix, $P(X)$, is defined as the product of the probability density functions of all the independent entries

$$P(X) = \prod_{j=1}^{N}\frac{1}{\sqrt{\pi}}e^{-x_{jj}^2}\prod_{1\leq j<k\leq n}\frac{2}{\pi}e^{-2|x_{jk}|^2}$$

Proposition 2.

$$P(X) \propto e^{-Tr(X^2)}$$

where $Tr(X^2)$ is the trace of the matrix X^2.

The trace of a matrix is defined as the sum of all of the diagonal entries.
Proof
For a general $N \times N$ matrix $X = [x_{jk}]_{j,k=1,\ldots,N}$;

$$X^2 = \left[\sum_{p=1}^{N} x_{jp}x_{pk}\right]_{j,k=1,\ldots,N}$$

which comes from the rule for matrix multiplication. For an Hermitian matrix $x_{pk} = \overline{x}_{kp}$, so in this case

$$X^2 = \left[\sum_{p=1}^{N} x_{jp} \overline{x}_{pk} \right]_{j,k=1,\dots,N}$$

Setting $j = k$ gives the diagonal entries. Summing over j the shows

$$\begin{aligned} Tr(X^2) &= \sum_{j=1}^{N} \sum_{p=1}^{N} |x_{jp}|^2 \\ &= \sum_{j=1}^{N} x_{jj}^2 + 2 \sum_{1 \le j < p \le N} |x_{jp}|^2 \end{aligned}$$

the second line was obtained using the fact that x_{jj} is real as well as the fact that $|x_{jp}|^2 = |x_{pj}|^2$. Consequently

$$\begin{aligned} e^{-Tr(X^2)} &= e^{-\sum_{j=1}^{N} x_{jj}^2 - 2\sum_{1 \le j < p \le N} |x_{jp}|^2} \\ &= \prod_{j=1}^{N} e^{-x_{jj}^2} \prod_{1 \le j < p \le N} e^{-2|x_{jp}|^2} \end{aligned}$$

which implies the stated result. □

Proposition 3. *Similarity transformation by a unitary matrix maps Hermitian matrices to Hermitian matrices.*

Proof For X Hermitian and U unitary we want to show that $U^{-1}XU$ is Hermitian. Now:

$$\begin{aligned} (\overline{U^{-1}XU})^T &= (\overline{U}^{-1}\overline{XU})^T \\ &= \overline{U}^T \overline{X}^T (\overline{U}^T)^{-1} \\ &= U^{-1}XU \end{aligned}$$

as required. □

Proposition 4. *$Tr(X^2)$ is unchanged if X undergoes a similarity transformation by a unitary matrix.*

Proof

$$Tr((U^{-1}XU)^2) = Tr(U^{-1}X^2U) = Tr(X^2)$$

where to obtain the final equality the cyclic property of the trace,

$$Tr(AB) = Tr(BA)$$

has been used. □

The result of propositions **??** and **??** show us that:

$$P(U^{-1}XU) = P(X)$$

Because of this unitary invariance, random matrices specified as in definition **??** are given the adjective ”Unitary” in the name Gaussian Unitary Ensemble.

2.3 The Eigenvalue Probability Density Function

Associated with the probability density function $P(X)$ is the probability that an $N \times N$ Hermitian matrix lies within a small interval dX of the matrix X. Thus [**?**]:

$$\begin{aligned} P(X \in [X, X+dX]) &= P(X)(dX) \\ &= A_N e^{-Tr(X^2)}(dX) \end{aligned}$$

where A_N is the normalisation which is chosen so that

$$\int P(X)(dX) = 1$$

and

$$(dX) = \prod_{j=1}^{N} dx_{jj} \prod_{j<k} dx_{jk}^R dx_{jk}^I$$

with $x_{jk} = x_{jk}^R + ix_{jk}^I$.

We want to deduce from this the probability density function for the eigenvalues. To do this we require a change of variables from the elements of the matrix to variables relating to the eigenvalues and eigenvectors. For an Hermitian matrix X the diagonalisation formula can be written $X = UL\overline{U}^T$. On the left hand side there are $N + 2(N(N-1)/2) = N^2$ independent variables, where the term N is the number of diagonal variables (which are all real), while $N(N-1)/2$ is the number of upper triangular elements, which must be multiplied by 2 since they are complex. The lower triangular elements are not independent variables, since they are equal to the complex conjugate of the upper triangular elements.

Consider now the right hand side. The matrix L has N independent variables, which are the eigenvalues of X (recall these are all real). How many independent variables are there in an $N \times N$ unitary matrix? A general $N \times N$ complex matrix has $2N^2$ variables. For a unitary matrix the columns must be normalised to unitary and thus

$$\overline{\mathbf{u}}_j.\mathbf{u}_j = 1 \qquad (j = 1, \ldots, N)$$

which gives us a total of $2N$ real equations and thus constraints. Further the different columns of U must be mutually orthogonal

$$\overline{\mathbf{u}}_j.\mathbf{u}_k = 0 \qquad (1 \leq j < k \leq N)$$

which gives us a total of $2(N(N-1)/2)$ real equations and thus extra constraints. There then are a total of N^2 constraints, leaving N^2 independent variables out of the $2N^2$ variables. So both sides of the equation have the same number of independent variables.

To implement the desired change of variables we need the Jacobian of the transformation.
Given a set of equations, written as

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_N(\mathbf{x}) \end{bmatrix}$$

The Jacobian is defined by

$$\begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_N(\mathbf{x})}{\partial x_N} \end{bmatrix}$$

For the change of variables from the elements of X to its eigenvalues and corresponding eigenvectors, the Jocabian [**?**] is such that

$$(dX) = \prod_{1\le j<k\le N} (\lambda_k - \lambda_j)^2 d\lambda_1 \dots d\lambda_N (\overline{U}^T dU)$$

The notation $(U^T dU)$ means the product of all the independent differentials in $U^T dU$ of which there are $N(N-1)$. It remains to change variables in the probability density. Now

$$\begin{aligned} e^{-Tr(X^2)} &= e^{-Tr(UL^2\overline{U}^T)} = e^{-Tr(L^2)} \\ &= e^{-\sum_{j=1}^{N} \lambda_j^2} \end{aligned}$$

where the second equality uses the cyclic property of the trace, while the third equality uses that fact that the square of a diagonal matrix is the square of the diagonal elements. Thus we have the change of variables formula

$$A_N P(X)(dX) = A_N e^{-\sum_{j=1}^{N} \lambda_j^2} \prod_{1\le j<k\le N} (\lambda_k - \lambda_j)^2 d\lambda_1 \dots d\lambda_N \times (\overline{U}^T dU)$$

It is important to note that the eigenvalue dependent portion; $\prod_{1\le j<k\le N}(\lambda_k - \lambda_j)^2 d\lambda_1 \dots d\lambda_N$ factorises from the eigenvector dependent portion; $(\overline{U}^T dU)$. Hence integrating over $(\overline{U}^T dU)$ only contributes a constant, and we read off that the probability density function for the eigenvalues is given by

$$\frac{1}{C} \prod_{l=1}^{N} e^{-\lambda_l^2} \prod_{1\le j<k\le N} (\lambda_k - \lambda_j)^2 \tag{2.3.1}$$

where C is the normalisation.

2.4 The Characteristic Polynomial and Empirical Calculation

For a general $N \times N$ matrix A the characteristic polynomial $p_N(\lambda)$ is defined by

$$p_N(\lambda) = \det(\lambda I_N - A) \tag{2.4.1}$$

This is a polynomial of degree N which will vanish at each of the eigenvalues λ of A. It is known, [**?**] and [**?**], that matrices belonging to the Gaussian Unitary Ensemble have characteristic polynomials that satisfy the random three term recurrence

$$\begin{aligned} &p_k(\lambda) = (\lambda - a_k)p_{k-1}(\lambda) - b_{k-1}^2 p_{k-2}(\lambda) \qquad (k = 1, 2, ...) \\ &\text{where } a_k \in N[0,1], b_k^2 \in \Gamma[k,1] \end{aligned} \tag{2.4.2}$$

Here the notation $\Gamma[s, \sigma]$ refers to the Gamma distribution.
The recurrence is subject to the initial conditions

$$p_{-1}(\lambda) = 0 \qquad\qquad p_0(\lambda) = 1$$

To study the largest eigenvalue we are interested in the statistical properties of the largest zero of $p_N(\lambda)$. Further scaling behaviour for large N is sought.
For matrices that belong to the Gaussian Unitary Ensemble, it is known that the mean of the largest eigenvalue is to leading order equal to $\sqrt{2N}$. Furthermore, the spacing or the difference between the largest eigenvalue and the second largest eigenvalue is of order $\frac{1}{(\sqrt{2}N^{\frac{1}{6}})}$ [**?**]. This suggests studying the corresponding distribution of the largest eigenvalue by changing variables in the following way

$$\lambda \mapsto \sqrt{2N} + \frac{X}{\sqrt{2}N^{\frac{1}{6}}} \tag{2.4.3}$$

where X is the scaled variable. This distribution can be calculated empirically using the following procedure:
For a particular N the characteristic polynomial $p_N(x)$ is computed for the recurrence **??**. The roots are computed from $p_N(x) = 0$, then we read off the largest value of λ. This procedure is repeated M times, for large M resulting in an array of the largest eigenvalues $\{x_1, x_2, \ldots, x_M\}$. This array is then scaled using (**??**) where we introduce the new scaled variable

$$y_j = \sqrt{2}N^{\frac{1}{6}}\left(x_j - \sqrt{2N}\right)$$

for each $j = 1, 2, \ldots, M$.
For a particular positive integer L, consider the intervals $\left(\frac{j}{L}, \frac{j+1}{L}\right)$ with $j = -4 \times L, -4 \times L + 1, \ldots, L$ (most y_j lie in these intervals) and then calculate the number of y_j that are in the interval $\left(\frac{j}{L}, \frac{j+1}{L}\right)$. These numbers must be normalised so that

$$\sum_{\text{subintervals}} (\text{number of } y_j)\,(\text{length of subinterval}) = 1$$

The length of each interval is $\frac{1}{L}$ and the total number of y_j summed over all subintervals is M

$$\sum_{\text{subintervals}} (\text{number of } y_j) = M$$

It is therefore necessary to divide the number of y_j in the interval $\left(\frac{j}{L}, \frac{j+1}{L}\right)$ by $\frac{M}{L}$. This then gives the final height of the interval $\left(\frac{j}{L}, \frac{j+1}{L}\right)$ in the corresponding histogram. The histogram or bar-plot can then be generated. The histogram that was obtained using the described procedure with $N = 50$ and $M = 5000$ is shown in Figure **??**. The code to calculate the largest eigenvalues and the corresponding histogram is given in Appendix **??**.

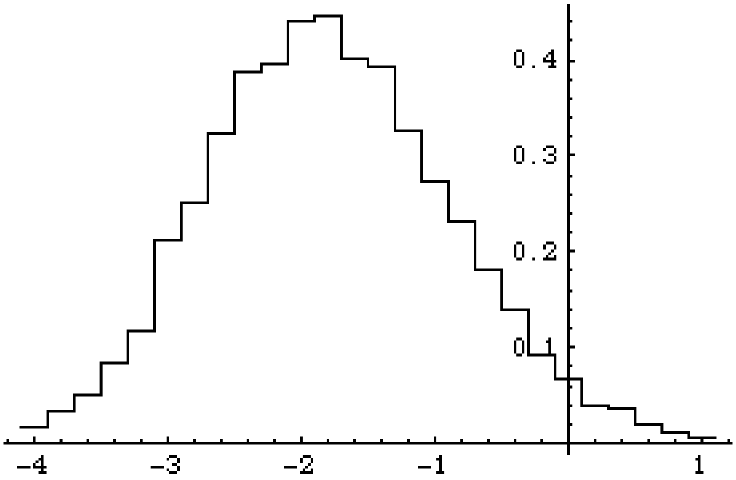

Figure 2.1: Eigenvalue P.d.f. for Matrices from the GUE.

We can see that this distribution looks vaguely normal with a mean at around -2 and we will later show that this is approximately equal to a distribution called the p_2^{soft} distribution.

2.5 The Laguerre Unitary Ensemble

In addition to the Gaussian Unitary Ensemble, the Laguerre Unitary Ensemble, LUE is also of interest for its application to queueing processes.

Definition 3. *Let X be an $N \times N$ matrix where the entries are complex Gaussians specified by $N[0, \frac{1}{\sqrt{2}}]+iN[0, \frac{1}{\sqrt{2}}]$. The Laguerre Unitary Ensemble is formed out of the positive definite matrices $X^T X$.*

Making use of the workings which led to the calculation of the eigenvalue probability density function for matrices from the GUE, the eigenvalue probability density function for the LUE can be shown to be given by

$$\frac{1}{C}\prod_{l=1}^{N} e^{-\lambda_l} \prod_{1\le j<k\le N} (\lambda_k-\lambda_j)^2 \qquad \lambda_l \ge 0 \tag{2.5.1}$$

Further, it is known that in terms of the scaled variable X specified by

$$\lambda = 4N + 2(2N)^{\frac{1}{3}} X \tag{2.5.2}$$

According to a result from [**?**], the characteristic polynomial for the LUE satisfies the following random three term recurrence

$$B_j(x) = (x - v_2^{(j)})B_{j-1}(x) - xv_1^{(j)}B_{j-2}(x) \tag{2.5.3}$$

where $B_{-1}(x) = 0$, $B_0(x) = 1$ and $v_1^{(j)}, v_2^{(j)}$ have distributions $\Gamma[j-1, 1]$ and $\Gamma[N-j+1, 1]$ respectively. For a given value of N, this allows the largest eigenvalue to be computed as the largest root of the characteristic polynomial. From this a histogram giving the distribution of the corresponding scaled variable X, as specified by **??**, can be obtained. This histogram in shown in figure **??** and the corresponding code is shown in Appendix **??**.

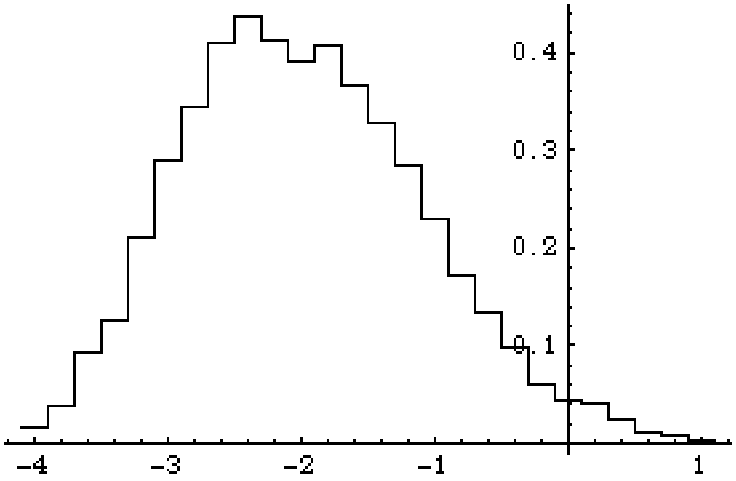

Figure 2.2: Eigenvalue P.d.f. for Matrices from the LUE

This appears to be the same distribution as that in figure **??** for the empirical distribution of the scaled largest eigenvalue in the GUE. In fact it is a known theorem [**?**], that in the limit $N \to \infty$ the distribution of the largest eigenvalue for the LUE agrees with that for the GUE.

Chapter 3

Calculating the p_2^{soft} Distribution

In the previous chapter an empirical determination of the scaled distribution of the largest eigenvalue of large GUE matrices and large LUE matrices were given. In fact in the $N \to \infty$ limit an exact expression for this distribution, to be denoted $p_2^{soft}(s)$ is known [**?**]. It is given by

$$p_2^{soft}(s) = \frac{d}{ds}\left(exp\left(-\int_s^\infty (t-s)u^2(t)dt\right)\right) \tag{3.0.1}$$

where $u(t)$ is specified in terms of the solution of a nonlinear equation. The nonlinear equation in question is

$$\frac{d^2q}{ds^2} = sq + 2q^3 \tag{3.0.2}$$

subject to the boundary condition

$$q(s;\xi) \sim \xi Ai(s) \tag{3.0.3}$$

where $Ai(s)$ denotes the Airy function [**?**]. The equation **??** can be recognised as the $\alpha = 0$ case of the Painlevé equation [**?**]. The function $u(t)$ in **??** is related to $q(s;\xi)$ in equation **??** by

$$u(t) = q(t;1) \tag{3.0.4}$$

3.1 Large t expansion of $u(t)$

According to equations **??** and **??**, the leading order large t behaviour of $u(t)$ is given by the Airy function. The Airy function $Ai(s)$ is defined as the solution of the Airy equation

$$\frac{d^2y}{ds^2} - sy = 0 \tag{3.1.1}$$

which decays as $s \to \infty$. The precise form of $Ai(s)$ as $s \to \infty$ is given by the following asymptotic formula.

Proposition 5. *As $s \to \infty$ [**?**]*

$$Ai(s) \sim \frac{e^{-\frac{2}{3}s^{\frac{3}{2}}}}{2\sqrt{\pi}s^{\frac{1}{4}}} \sum_{n=0}^{\infty} \frac{(-1)^n}{(\frac{2}{3}s^{\frac{3}{2}})^n}\alpha_n \tag{3.1.2}$$

where α_n is determined by the recurrence

$$\alpha_n = \frac{1}{72n}(6n-1)(6n-5)\alpha_{n-1} \qquad\qquad (n = 1, 2....)$$

subject to the initial condition: $\alpha_0 = 1$.

Proof

By simply substituting the expression **??** into **??**, we get

$$\begin{aligned}
& \frac{e^{-\frac{2}{3}s^{\frac{3}{2}}}}{2\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n\alpha_n}{(\frac{2}{3})^n}[s^{-\frac{3n}{2}+\frac{3}{4}} - \left(-\frac{3n}{2}+\frac{1}{4}\right)s^{-\frac{3n}{2}+\frac{3}{4}} \\
- & \left(-\frac{3n}{2}-\frac{1}{4}\right)s^{-\frac{3n}{2}+\frac{3}{4}} + \left(-\frac{3n}{2}-\frac{1}{4}\right)\left(-\frac{3n}{2}-\frac{5}{4}\right)s^{-\frac{3n}{2}-\frac{9}{4}}] \\
= & \frac{e^{-\frac{2}{3}s^{\frac{3}{2}}}}{2\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n\alpha_n}{(\frac{2}{3})^n}s^{-\frac{3n}{2}+\frac{3}{4}}
\end{aligned}$$

This simplifies to:

$$\sum_{n=0}^{\infty}(-1)^n\alpha_n\left(\frac{2}{3}\right)^{-n}s^{-\frac{3n}{2}-\frac{9}{4}}\left(-\frac{3n}{2}-\frac{1}{4}\right)\left(-\frac{3n}{2}-\frac{5}{4}\right) = 3\sum_{n=0}^{\infty}(-1)^n\alpha_{n+1}\left(\frac{2}{3}\right)^{-(n+1)}s^{-\frac{3n}{2}-\frac{9}{4}}(n+1)$$

Equating coefficients gives the recurrence and so verifies the result □

As a correction to the leading $s \to \infty$ behaviour, we write

$$q(s) = \xi Ai(s) + \xi^3 Q(s) \tag{3.1.3}$$

where $|Q(s)| \ll Ai(s)$ for $s \to \infty$, and here ξ^3 has been put in front of $Q(s)$ for convenience which will become obvious later.

Substituting this into **??**, we get

$$\xi Ai''(s) + \xi^3 Q''(s) = s\xi Ai(s) + s\xi^3 Q(s) + 2\left(\xi Ai(s) + \xi^3 Q(s)\right)^3$$

The left hand side of this equation reduces to

$$s\xi Ai(s) + \xi^3 Q''(s)$$

upon using the Airy equation **??**. Equating and dividing through by ξ^3, gives

$$Q''(s) = sQ(s) + 2\left(Ai(s) + \xi^2 Q(s)\right)^3$$

By assumption $|Q(s)| \ll Ai(s)$ for s large so for $s \to \infty$ we can simplify this to read

$$Q''(s) \sim sQ(s) + 2\left(Ai(s)\right)^3 \tag{3.1.4}$$

Proposition 6. *Replace the $\sim$ for an $=$ in equation **??** to obtain the differential equation*

$$Q''(s) = sQ(s) + 2\left(Ai(s)\right)^3 \tag{3.1.5}$$

For $s \to \infty$ this admits the solution

$$Q(s) = \frac{e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}s^{\frac{7}{4}}} \sum_{n=0}^{\infty} \frac{(-1)^n}{(\frac{2}{3}s^{\frac{3}{2}})^n} a_n \tag{3.1.6}$$

where $\{a_n\}$ satisfy the recurrence

$$a_n = \alpha_n^{(3)} + \frac{3}{4}na_{n-1} - \frac{1}{8}(n - \frac{1}{6})(n - \frac{5}{6})a_{n-2} \tag{3.1.7}$$

subject to the initial conditions: $a_{-2} = a_{-1} = 0$.

Proof
We can use equation **??** to calculate an approximation for $(Ai(s))^3$. To do this we will need to introduce some theory for multiplying two power series together. Consider the following two power series $A(x)$ and $B(x)$,

$$A(x) = \sum_{n=0}^{\infty} a_n x^n \qquad\qquad B(x) = \sum_{n=0}^{\infty} b_n x^n$$

Then

$$A(x)B(x) = \sum_{n=0}^{\infty} c_n x^n$$

with

$$c_n = \sum_{p=0}^{n} a_p b_{n-p}$$

For example

$$(A(x))^2 = \sum_{n=0}^{\infty} a_n^{(2)} x^n$$

with

$$a_n^{(2)} = \sum_{p=0}^{\infty} a_p a_{n-p}$$

and

$$(A(x))^3 = A(x)(A(x))^2$$

$$= \left(\sum_{n=0}^{\infty} a_n x^n\right)\left(\sum_{n=0}^{\infty} a_n^{(2)} x^n\right)$$

with

$$a_n^{(3)} = \sum_{p=0}^{n} a_p a_{n-p}^{(2)} = \sum_{p=0}^{n} a_{n-p} a_p^{(2)}$$

Noting from equation **??** that

$$Ai(s) \sim \frac{e^{-\frac{2}{3}s^{\frac{3}{2}}}}{2\sqrt{\pi}s^{\frac{1}{4}}} \sum_{n=0}^{\infty} (-1)^n \alpha_n \left(\frac{3}{2s^{\frac{3}{2}}}\right)^n$$

We can use the previous theory to show that

$$(Ai(s))^2 \sim \frac{e^{-\frac{4}{3}s^{\frac{3}{2}}}}{4\sqrt{\pi}s^{\frac{1}{2}}} \sum_{n=0}^{\infty} (-1)^n \alpha_n^{(2)} \left(\frac{3}{2s^{\frac{3}{2}}}\right)^n$$

with

$$\alpha_n^{(2)} = \sum_{p=0}^{n} \alpha_p \alpha_{n-p}$$

Then

$$(Ai(s))^3 \sim \frac{e^{-2s^{\frac{3}{2}}}}{8\pi^{\frac{1}{2}}s^{\frac{3}{4}}} \sum_{n=0}^{\infty} \frac{(-1)^n}{(\frac{2}{3}s^{\frac{3}{2}})^n} \alpha_n^{(3)}$$

with

$$\alpha_n^{(3)} = \sum_{p=0}^{n} \alpha_p \alpha_{n-p}^{(2)}$$

With these formulas established, we can prove that **??** admits the solution **??** by direct substitution. This shows

$$\begin{aligned}
& \frac{9e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}} \sum_{k=0}^{\infty} (-1)^k a_k \left(\frac{2}{3}\right)^{-k} s^{-\frac{3k}{2}-\frac{3}{4}} + \frac{9e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}} \sum_{k=0}^{\infty} (-1)^k a_k \left(\frac{2}{3}\right)^{-k} (k+1)\, s^{-\frac{3k}{2}-\frac{9}{4}} \\
+ \quad & \frac{e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}} \sum_{k=0}^{\infty} (-1)^k a_k \left(\frac{2}{3}\right)^{-k} \left(-\frac{3k}{2}-\frac{7}{4}\right)\left(-\frac{3k}{2}-\frac{11}{4}\right) s^{-\frac{3k}{2}-\frac{15}{4}} - \frac{e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}} \sum_{k=0}^{\infty} (-1)^k a_k \left(\frac{2}{3}\right)^{-k} s^{-\frac{3k}{2}-\frac{3}{4}} \\
= \quad & \frac{2e^{-2s^{\frac{3}{2}}}}{8\pi^{\frac{1}{2}}} \sum_{n=0}^{\infty} (-1)^n \alpha_n^{(3)} \left(\frac{2}{3}\right) s^{-\frac{3n}{2}-\frac{3}{4}}
\end{aligned}$$

With a few index shifts and some working, this simplifies to

$$\begin{aligned}
& \frac{e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}} \sum_{n=0}^{\infty} (-1)^n a_n \left(\frac{2}{3}\right)^{-n} s^{-\frac{3n}{2}-\frac{3}{4}} \left[8a_n - 6na_{n-1} + \frac{4}{9}a_{n-2}\left(-\frac{3n}{2}+\frac{5}{4}\right)\left(-\frac{3n}{2}+\frac{1}{4}\right)\right] \\
= \quad & \frac{e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}} \sum_{n=0}^{\infty} (-1)^n a_n \left(\frac{2}{3}\right)^{-n} s^{-\frac{3n}{2}-\frac{3}{4}} \left[8\alpha_n^{(3)}\right]
\end{aligned}$$

Equating coefficients gives the recurrence shown in **??** and hence we have proved Proposition **??**.
□

According to Proposition **??**

$$q(s,\xi) \sim \xi Ai(s) + \xi^3 \frac{e^{-2s^{\frac{3}{2}}}}{32\pi^{\frac{3}{2}}s^{\frac{7}{4}}} \sum_{k=0}^{\infty} \frac{(-1)^k}{(\frac{2}{3}s^{\frac{3}{2}})^k} a_k$$

As this is an asymptotic series, there is an optimal value in the k summation to truncate the series to obtain the most accurate approximation to $q(s;\xi)$. Working empirically, this has been calculated in [**?**] to be $k \approx \frac{4}{3}s_0^{\frac{3}{2}}$ for $s_0 \gg 0$.

3.2 Power Series Expansion of $u(t)$

For large s_0, the results of the previous section allow $q(s_0,\xi)$ and $q'(s_0,\xi)$ to be accurately computed. These values can be used to compute the coefficients in the power series expansion

$$q(s;\xi) = \sum_{l=0}^{\infty} c_l (s-s_0)^l \tag{3.2.1}$$

by recurrence.

Proposition 7. *The coefficients c_n in the power series expansion* **??**, *are specified by the recurrence*

$$c_{n+2} = \frac{s_0 c_n + c_{n-1} + 2c_n^{(3)}}{(n+2)(n+1)}$$

where $c_n^{(k)} = \sum_{j=0}^{n} c_{n-j} c_j^{(k-1)}$ are the coefficients of $(q(s,\xi))^k$ about s_0.

Proof
Let the power series expansion of $(q(s;\xi))^3$ about $s = s_0$ be

$$(q(s;\xi))^3 = \sum_{n=0}^{\infty} c_n^{(3)} (s-s_0)^n \tag{3.2.2}$$

Now we can substitute equation **??** and equation **??** into equation **??** and then equate like powers of $(s-s_0)$ to obtain the recurrence for c_n.
Firstly to calculate sq we need to write it as

$$\begin{aligned} sq &= (s-s_0)q + s_0 q \\ &= s_0 \sum_{l=0}^{\infty} c_l (s-s_0)^l + (s-s_0) \sum_{l=0}^{\infty} c_l (s-s_0)^l \end{aligned}$$

Substituting into **??** we get

$$\sum_{n=0}^{\infty} n(n-1)c_n(s-s_0)^{n-2} = s_0 \sum_{l=0}^{\infty} c_l(s-s_0)^l + (s-s_0)\sum_{l=0}^{\infty} c_l(s-s_0)^l + 2\sum_{l=0}^{\infty} c_l^{(3)}(s-s_0)^l$$

After some index shifts we obtain

$$\sum_{n=0}^{\infty} c_{n+2}(n+2)(n+1)(s-s_0)^n = s_0 \sum_{n=0}^{\infty} c_n(s-s_0)^n + \sum_{n=0}^{\infty} c_{n-1}(s-s_0)^n + 2\sum_{n=0}^{\infty} c_n^{(3)}(s-s_0)^n$$

Equating coefficients of $(s-s_0)^n$ gives the sought recurrence. □

In a numerical computation the series in **??** must be truncated. Because of this only values of s sufficiently close to s_0 will accurately reproduce $q(s;\xi)$. In addition, with $\xi = 1$ the series **??** only has a finite radius convergence, and s_0 cannot be used outside this range, independent of the number of terms summed over. In relation to this later point, the radius of convergence R can be calculated from the nth root formula

$$\frac{1}{R} = \lim_{n\to\infty} |c_n|^{(\frac{1}{n})}$$

With $s_0 = 50$, after calculating $\{c_n\}_{n=0,1,\ldots,n_0}$ with $n_0 = 600$, it was estimated that R is between 44 and 46.

To obtain an accurate evaluation, a sequence of power series based on the points $s_0 - is[n]$, $is[n] = n$ ($n = 0, 1, \ldots, nn$) with $nn = 90$) were successively computed, starting from the power series about $s_0 = 50$. Here each successive power series $q[s,j]$ is deduced from Proposition **??** with initial condition obtained from the previous power series,

$$c_0 = q[s_0 - is[j], j-1], \qquad c_1 = q'[s_0 - is[j], j-1]$$

for $j = 1, \ldots, nn$. We regard $u(s)$ as being specified by $q[s,j]$ for $s_0 - is[j+1] < s \leq s_0 - is[j]$.

3.3 Power Series expansion of p_2^{soft}

Since $u(t) \to 0$ rapidly as $t \to \infty$, it follows from **??** that as $s \to \infty$

$$p_2^{soft}(s) \sim \int_s^{\infty} u^2(t)dt$$

Making use of equations **??** and **??** we then have

$$\begin{aligned} p_2^{soft}(s) &\sim \frac{1}{4\pi}\int_s^{\infty} t^{-\frac{1}{2}} e^{-\frac{4}{3}t^{\frac{3}{2}}} dt \\ &< \frac{1}{4\pi}\int_s^{\infty} t^{\frac{1}{2}} e^{-\frac{4}{3}t^{\frac{3}{2}}} dt \\ &= -\frac{1}{8\pi}\int_s^{\infty} \left(\frac{d}{dt} e^{-\frac{4}{3}t^{\frac{3}{2}}}\right) dt \\ &= \frac{1}{8\pi} e^{-\frac{4}{3}s^{\frac{3}{2}}} \end{aligned}$$

This bound allows us to estimate the error in replacing the infinite terminal in **??** by a finite value s_0, thus obtaining

$$p_2^{soft}(s) \approx \frac{d}{ds}\left(exp\left(-\int_s^{s_0}(t-s)u^2(t)dt\right)\right) \tag{3.3.1}$$

Let the power series about $s_0 - is[j]$ of $u^2(t)$ be given by

$$u^2(t) = \sum_{n=0}^{\infty} c[n,2,j](t-(s_0-is[j]))^n$$

We use this in the range

$$s_0 - is[j+1] < t \leq s_0 - is[j]$$

Next we define

$$A(s,j) = \int_s^{s_0} tu^2(t)dt = \sum_{k=0}^{j-1}\int_{s_0-is[k+1]}^{s_0-is[k]} tu^2(t)dt + \int_s^{s_0-is[j]} tu^2(t)dt$$

where $s_0 - is[j+1] < s \leq s_0 - is[j]$. Here the key quality is the last integral, in the sense that knowledge of it allows the computation of the integrals in the summation. This last integral can be computed in terms of the power series of $u^2(t)$ about $s_0 - is[j]$.

Proposition 8. *The quantity*

$$a(s,j) = \int_s^{s_0-is[j]} tu^2(t)dt$$

can be written as

$$\sum_{n=1}^{\infty} d[n,j]\,(s-(s_0-is[j]))^n$$

where

$$d[n,j] = -\left(\frac{c[n-2,2,j]}{n} + (s_0-is[j])\frac{c[n-1,2,j]}{n}\right)$$

with $c[-1,2,j] = 0$

Proof

We begin by noting

$$\int_s^{s_0-is[j]} tu^2(t)dt = \int_s^{s_0-is[j]}(t-(s_0-is[j]))u^2(t)dt + (s_0-is[j])\int_s^{s_0-is[j]} u^2(t)dt$$

and substitute the power series to obtain

$$\begin{aligned} & \int_s^{s_0-is[j]} tu^2(t)dt \\ = \quad & \int_s^{s_0-is[j]}\sum_{n=0}^{\infty}(t-(s_0-is[j]))^{n+1}c[n,2,j]dt + (s_0-is[j])\int_s^{s_0-is[j]}\sum_{n=0}^{\infty}(t-(s_0-is[j]))^n c[n,2,j]dt \\ = \quad & -\sum_{n=0}^{\infty}\frac{c[n,2,j]}{n+2}(s-(s_0-is[j]))^{n+2} - (s_0-is[j])\sum_{n=0}^{\infty}\frac{c[n,2,j]}{n+1}(s-(s_0-is[j]))^{n+1} \end{aligned}$$

After some index shifts and simplification this last line reads

$$-\sum_{n=1}^{\infty}\left(\frac{c[n-2,2,j]}{n}+(s_0-is[j])\frac{c[n-1,2,j]}{n}\right)(s-(s_0-is[j]))^n$$

and the result follows. □

Using this result we can compute $A(s,j)$ according to

$$A(s,j)=\sum_{k=0}^{j-1}(a(s_0-is[k+1],k))+a(s,j)$$

We can now do a similar analysis on

$$B(s,j)=\int_s^{s_0}u^2(t)dt=\sum_{k=0}^{j-1}\int_{s_0-is[k+1]}^{s_0-is[k]}u^2(t)dt+\int_s^{s_0-is[j]}u^2(t)dt$$

Proposition 9. *The quantity*

$$b(s,j)=\int_s^{s_0-is[j]}u^2(t)dt$$

can be written in the form

$$\sum_{n=1}^{\infty}b[n,j]\,(s-(s_0-is[j]))^n$$

where

$$b[n,j]=-\left(\frac{c[n-1,2,j]}{n}\right)$$

Proof

Use the power series expansion of $u^2(t)$ about $s_0-is[j]$ to write

$$\int_s^{s_0}u^2(t)dt=\int_s^{s_0}\sum_{n=0}^{\infty}c[n,2,j](t-(s_0-is[j]))^n dt$$

After computing the integral and some index shifts and simplifying we obtain

$$\int_s^{s_0}u^2(t)dt=-\sum_{n=0}^{\infty}\frac{c[n-1,2,j]}{n}(s-(s_0-is[j]))^n$$

which is the sought result. □

Using this result we can compute $B(s,j)$ according to

$$B(s,j)=\sum_{k=0}^{j-1}(b(s_0-is[k+1],k))+b(s,j)$$

The quantity appearing in **??** is

$$-\int_s^{s_0}(t-s)u^2(t)dt$$

The above results can be used to compute the power series of this quantity about $s_0 - is[j]$. Firstly we need to write s as

$$s = (s - (s_0 - is[j])) + (s_0 - is[j])$$

Then we have

$$-\int_s^{s_0} (t-s)u^2(t)dt = -A(s,j) + ((s - (s_0 - is[j])) + (s_0 - is[j]))\, B(s,j)$$

At this stage in the calculation the power series expansion about $s_0 - is[j]$ of the exponent in **??** is known. The next step is to deduce from that the power series about $s_0 - is[j]$ of the exponential. This is done automatically in the Mathematica code. The final step, as required by **??**, is to differentiate the power series. This we do automatically, after first storing the coefficients in the power series exponential.

The Mathematica code which implements this computation is given in Appendix **??**. A plot of of the distribution p_2^{soft} which is produced by the code is given in Figure **??**.

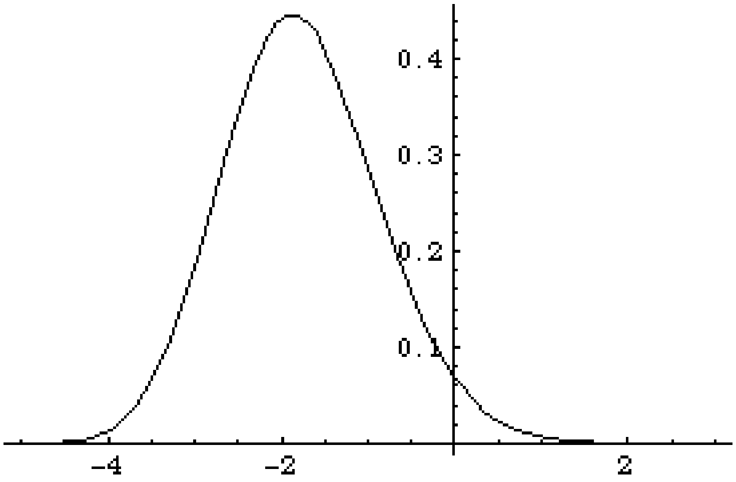

Figure 3.1: The p_2^{soft} distribution

The code can be used to compute statistical properties of the distribution. For this define the moments

$$\mu(n) = \int_{-\infty}^{\infty} t^n p^{soft}(t)dt$$

Statistical quantities of interest are the mean μ and the variance σ^2 defined in terms of the moments by

$$\mu = \mu(1), \qquad \sigma^2 = \mu(2) - (\mu(1))^2 .$$

Also of interest are the higher order statistical quantities

$$\begin{aligned} \gamma_1 &= \frac{\mu(3) - 3\mu(2)\mu(1) + 2(\mu(1))^3}{\sigma^3} \\ \gamma_2 &= \frac{\mu(4) - 4\mu(3)\mu(1) + 6\mu(2)(\mu(1))^2 - 3(\mu(1))^4}{\sigma^4} - 3 \end{aligned}$$

referred to as the skewness and (excess) kurtosis respectively. These quantities have been calculated using Mathematica (the code is shown in Appendix **??**) and are shown in table **??**.

	mean	variance	skewness	kurtosis
p_2^{soft}	-1.77108680	0.81319479	0.22408420	0.09344809

Table 3.1: Statistical properties of p_2^{soft}

Looking back at the p.d.f. for the eigenvalues for GUE matrices in Figure **??** and comparing this to the distribution p_2^{soft} in Figure **??**, we can see a closeness. Plotting the two on the same axes demonstrates that the two are indeed asymptotically equal, see Figure **??**.

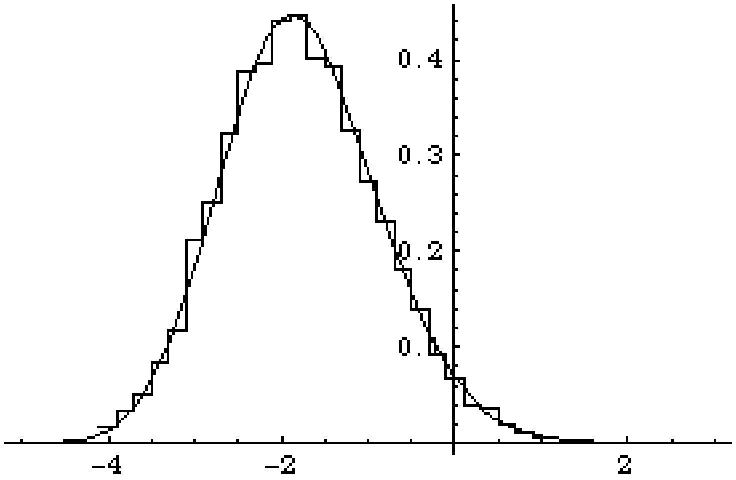

Figure 3.2: The p_2^{soft} distribution and the eigenvalue p.d.f. or GUE matrcies

Chapter 4

A Model for Queueing Processes

A queueing process is defined as a time evolution of jobs through a number of queues. A random matrix whose entries are all non-negative integers can be used to specify a queueing process. Label the rows of the matrix from the bottom up and label the columns of the matrix from left to right.

$$\begin{array}{cc}
\begin{array}{c}3\\2\\1\end{array} & \begin{bmatrix}0&0&1\\3&2&1\\2&1&0\end{bmatrix}\\
 & \begin{array}{ccc}1&2&3\end{array}
\end{array}$$

Table 4.1: Matrix A

Each entry $x_{i,j}$ of the matrix represents the amount of time it takes $queue\ i$ to process $job\ j$, once it reaches the server at the front of the queue. Observing the queueing process defined by matrix A, it takes $queue$ 2, 3 units of time to process job 1. We can observe the evolution of a queueing process using a step by step diagram showing each job ($J1$, $J2$, $J3$) moving through each queue ($Q1$, $Q2$, $Q3$). The evolution of the queueing process defined by matrix A is shown in table **??**.

Observe that all jobs start in $queue$ 1 and each job cannot be processed until it reaches the front of the queue when all jobs in front of it have been processed and have moved onto the next queue. Note also that job 1 disappears from the diagram at time $t = 5$ because job 1 takes time 0 to be processed by $queue$ 3.
We can see that it takes 9 units of time for all jobs to be processed by the queues.

	Q1	**Q2**	**Q3**
$t = 0$	$J1$		
	$J2$		
	$J3$		

	Q1	**Q2**	**Q3**
$t = 1$	$J1$		
	$J2$		
	$J3$		

	Q1	**Q2**	**Q3**
$t = 2$	$J2$	$J1$	
	$J3$		

	Q1	**Q2**	**Q3**
$t = 3$		$J1$	
		$J2$	
		$J3$	

	Q1	**Q2**	**Q3**
$t = 4$		$J1$	
		$J2$	
		$J3$	

	Q1	**Q2**	**Q3**
$t = 5$		$J2$	
		$J3$	

	Q1	**Q2**	**Q3**
$t = 6$		$J2$	
		$J3$	

	Q1	**Q2**	**Q3**
$t = 7$		$J3$	

	Q1	**Q2**	**Q3**
$t = 8$			$J3$

	Q1	**Q2**	**Q3**
$t = 9$			

Figure 4.1: The Queueing Process defined by matrix A

4.1 Computing Queueing Times

Let $T_{i,j}$ denote the time it takes $job\ i$ to leave $queue\ j$. In most cases our objective is to determine $T_{N,N}$ which is the total time for all jobs to be completed. For matrix A $T_{3,3} = 9$. We would like to be able to generate formulas for calculating this and general values of $T_{i,j}$. If we consider again our diagram for the queueing process defined by matrix A, we can observe that to calculate how long it takes for job 3 to leave $queue$ 3 we must consider two scenarios:
Either $queue$ 3 is empty when job 3 arrives at $queue$ 3 and so can be processed straight away or job 2 is in $queue$ 3 and so job 3 must wait until job 2 is processed before it can be processed itself. If we consider the first scenario we need to determine the time it takes for job 3 to leave $queue$ 2 this time is defined by $T_{3,2}$. If we now consider the second scenario we need to determine how long it takes for job 2 to leave $queue$ 3, which is defined by $T_{2,3}$. If the time it takes for job 3 to leave $queue$ 2 is greater than the time it takes for job 2 to leave $queue$ 3 then scenario one occurs, otherwise scenario two occurs. Therefore the time it takes for job 3 to leave $queue$ 3 will be the maximum of $T_{2,3}$ and $T_{3,2}$ plus the time it takes for $queue$ 3 to process job 3, which is denoted by $x_{3,3}$. If we consider this idea more generally we can determine our first equation for computing $T_{i,j}$.

Theorem 4.1.1.

$$T_{i,j} = max(T_{i,j-1}, T_{i-1,j}) + x_{i,j} \tag{4.1.1}$$

where $T_{i,j-1}$ is the time it takes for job i to leave queue j-1 and $T_{i-1,j}$ is the time it takes for job i-1 to leave queue j.

This recurrence relation uniquely determines $T_{i,j}$ given the boundary conditions: $T_{i,0} = T_{0,j} = 0$. This recurrence allows the us to compute all values of $T_{i,j}$ which will form a matrix $[T_{i,j}]$. This can be done by computing $T_{i,j}$ row by row. We can also determine values for $T_{i,j}$ by studying the matrix that defines the queuing process. If we consider matrix A and continue along the same ideas as the previous theorem. To obtain a value for $T_{3,3}$, we know we must add the entry $x_{3,3}$ to the maximum of $T_{3,2}$ and $T_{2,3}$. In terms of matrix A, to get to the entry $x_{3,3}$ we must pass through one of the entries $x_{3,2}$ or $x_{2,3}$. Similarly to calculate a value for $T_{2,2}$ we must calculate the values $T_{1,2}$ and $T_{2,1}$. If we continue this idea more generally we will obtain another formula for $T_{i,j}$.

Theorem 4.1.2. *Let $(1,1)u/rh(i,j)$ refer to a path through the integer entries of the matrix. The path must start at entry $x_{1,1}$ and finish at entry $x_{i,j}$. The path must consist of steps of either one step up or one step right horizontal.*
Then

$$T_{i,j} = max \sum_{p \in P} x_{ij} \tag{4.1.2}$$

where P is the set of all one step up and one step right horizontal paths that begin at the entry $x_{1,1}$ and end at the entry $x_{i,j}$.

An example of $(1,1)u/rh(3,3)$ path for matrix A is

$$\begin{bmatrix} 0 & & 0 & \rightarrow & 1 \\ & & \uparrow & & \\ 3 & & 2 & & 1 \\ & & \uparrow & & \\ 2 & \rightarrow & 1 & & 0 \end{bmatrix}$$

The $maximum$ $(1,1)u/rh(3,3)$ path for matrix A is

$$\begin{bmatrix} 0 & & 0 & & 1 \\ & & & & \uparrow \\ 3 & \rightarrow & 2 & \rightarrow & 1 \\ \uparrow & & & & \\ 2 & & 1 & & 0 \end{bmatrix}$$

For the queueing process defined by matrix A, the time it takes for all jobs to be processed by the three queues is calculated by adding the entries x_{ij} in the maximum $(1,1)u/rh(i,j)$ path for matrix A

$$\begin{aligned} T_{3,3} &= 2+3+2+1+1 \\ &= 9 \end{aligned}$$

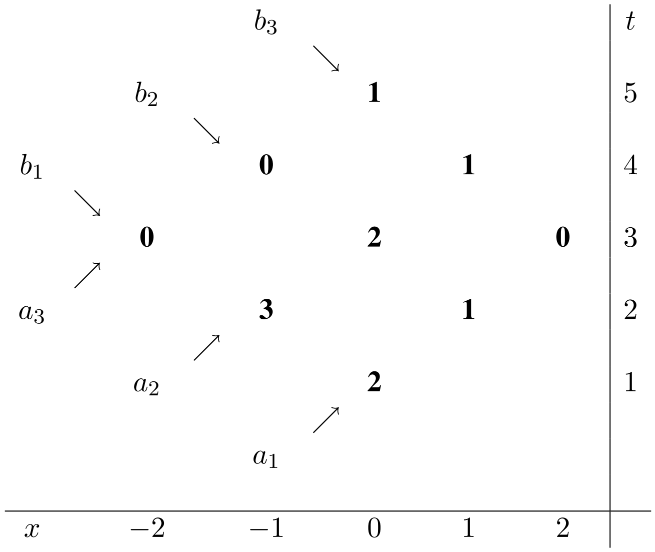

Figure 4.2:

This is of course the same as the value we determined previously using theorem **??**.

4.2 Robinson-Schensted-Knuth Correspondence

There is a correspondence between integer matrices and pairs of Young tableaux called the Robinson-Schensted-Knuth Correspondence. A Young tableau is a semi-standard tableau which is strictly increasing along a row and weakly increasing down a column. Given an $N \times N$ matrix of non-negative integers corresponding to a queueing process we can obtain a sequence of growth models, represented by weighted lattice paths, from which we can obtain a Young tableau. A Young tableau can be used to obtain a value for $T_{N,N}$.

Firstly we rotate Matrix A 45° anti-clockwise. Label the new rows of the rotated matrix by $t = 1, ..., 2N - 1$ and label the new columns by $x = -(N-1)..., 0, ..., (N-1)$, this is shown in Figure **??**.

The entries x_{ij} represent the heights of the weighted nucleation event at corresponding times. A nucleation event is of width one, height x_{ij} and weight $(a_i b_j)^{x_{ij}}$ and it is centered above its corresponding x value. As $t \mapsto t+1$, each nucleation event grows one unit in both horizontal directions and the new nucleation events for $t+1$ are placed above all previous nucleation events. If there is overlap between nucleation events the overlap is moved down to the line below and then continues to grow with time. So at time one there is a nucleation event of width one, height two and centered over $x = 0$. At time two the previous nucleation event from time one is now three units wide, it is still two units high but there is now a new nucleation event of height one unit centered above $x = 1$ and another nucleation event of height three centered above $x = -1$. This growth process continues until $t = 2N - 1$ and the event the occurs at $t = 2N - 1$ is called the final nucleation event. For matrix A the sequence of growth models is shown in Figure **??**.

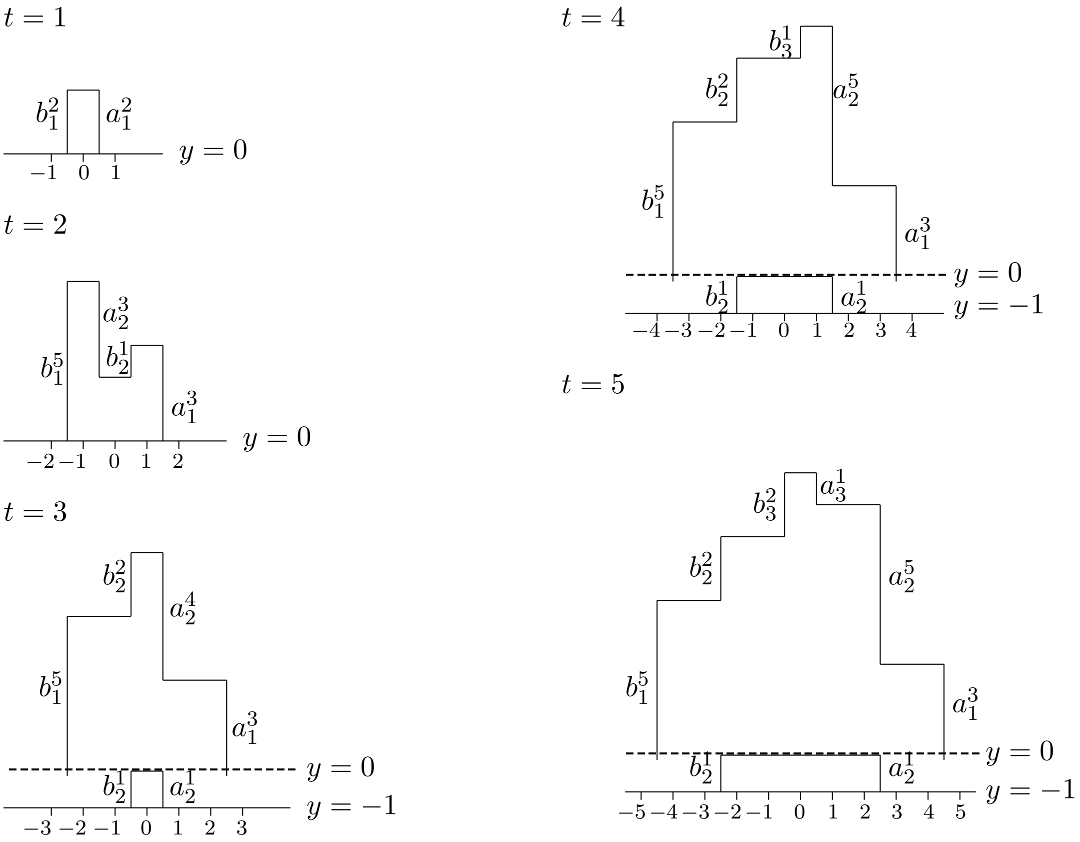

Figure 4.3: Sequence of growth models for matrix A

Using the growth models in Figure **??** we can determine recurrences for the heights of the paths and so relate the height of the first profile to the final queuing time $T_{N,N}$. The path at $y = 0$ is called the level-1 path and more generally the path at $y = -(N - 1)$ is called the nth path.

From the sequence of growth models, represented by a series of non-intersecting lattice paths, we can obtain a pair of Young tableaux. Each row of the tableau corresponds to a different non-intersecting lattice path that appears in the final growth model at $t = 2N - 1$. Row 1 of the tableau corresponds to the lattice path at $y = 0$ and generally the Nth row of the tableau is represented by the path at $y = 1 - N$. So in this case there are two rows of each Young tableaux which corresponds to the two lattice paths in the growth model one at $y = 0$ and one at $y = -1$.

Tableau L

1	1	1	1	1	2	2	3	3
2								

Tableau R

1	1	1	2	2	2	2	2	3
2								

Figure 4.4: The corresponding Young tableaux for matrix A

Two semi-standard tableaux are needed to completely describe a sequence of growth models, one describes all weighted lattice paths left of $x = 0$ and the other describes all lattice paths right of $x = 0$. For matrix A the pair of Young tableaux are shown in Figure **??**.

The length of each row corresponds to the maximum displacement of the corresponding profile. In the filling of the tableaux, the number of times each number is repeated gives the heights of the various vertical segments in the corresponding profiles, reading from the left for tableau L, and from the right for tableau R.

4.3 A Recurrence for h_1

It has already been noted that $T_{N,N}$ satisfies the recurrence **??**. Here it will be established that

$$h_1(N, N) = T_{N,N} \tag{4.3.1}$$

by showing that $h_1(n_1, n_2)$ satisfies the same recurrence.

Proposition 10. $h_1(n_1, n_2)$ *satisfies the following recurrence*

$$h_1(n_1, n_2) = max(h_1(n_1, n_2 - 1), h_1(n_1 - 1, n_2)) + x_{n_1 n_2} \tag{4.3.2}$$

Proof
Denote by $h_l(n_1, n_2)$, the maximum displacement of the level-1 path in the growth model corresponding to a matrix $A = [x_{ij}]$, where A has been truncated so that it only includes the first n_1 rows and n_2 columns. Now $h_l(n_1, n_2)$ is formed from the displacement due to the final nucleation event, plus the displacement due to the $x_{n_1 n_2}$ growth profile at $x = 0$ which can be calculated as the maximum of the height at $x = 1$, denoted by $d(-1)$ and the height at $x = -1$, denoted $d(1)$. Hence

$$h_1(n_1, n_2) = max(d(-1), d(1)) + x_{n_1 n_2}$$

Consider the displacement of the level-1 path at $x = -1$. If we deleted the $(n - 1)th$ row of the matrix A, we would obtain a growth model that would finish at $t = 2n - 2$ and the maximum displacement of the level-1 path at this time would be the displacement $d(-1)$ because no nucleation event occurs above $x = 0$, or $x = 1$ due to the deletion of the column. Therefore $d(-1)$ equals $h_1(n_1, n_2 - 1)$. The same argument can be used to show that $d(1) = h_1(n_1 - 1, n_2)$ We therefore obtain the desired recurrence. □

Using the growth models in figure **??** namely the growth model for $t = 4$ and Matrix A

$$\begin{aligned} h_1(N,N) &= max(h_1(N-1,N), h_1(N-1,N)) + x_{NN} \\ &= max(7+8) + 1 \\ &= 9 \end{aligned}$$

This is the same as the value we determined previously using theorem **??**.

4.4 Schur Polynomials

In a Young tableaux, T let the number of times k appears in the filling be denoted m_k. If the weights of a single vertical segment in the corresponding lattice path segments are $x_1, \ldots, x_N$ for $k = 1, 2, \ldots, N$ respectively, then the weight $m(T)$ of this particular filling is

$$m(T) = x_1^{m_1}, x_2^{m_2}, ..., x_n^{m_N} = \prod_{i=1}^{N} (x_i)^{m_i}$$

For example, the Young tableau

1	1	1	1	1	2	2	3	3
2	3							

has weight

$$m(T) = x_1^5 x_2^3 x_3^3$$

The Schur polynomial $s_\lambda(x_1, \ldots, x_N)$ is defined as the sum over all weights associated with a Young tableau of shape λ and fillings from 1 to N

$$s_\lambda(x_1, \ldots, x_N) = \sum_{fillings} x^T = \sum_{fillings} x_1^{m_1} \ldots x_N^{m_N} \tag{4.4.1}$$

This definition is of direct relevance to specifying the total weight of lattice paths in the growth model. Further progress also relies on an alternative expression for the Schur polynomial.

Proposition 11. *Schur polynomials can be expressed as a ratio of determinants*

$$s_\lambda(x_1,\ldots,x_N)=\frac{\det[x_j^{N-k+\lambda_k}]_{j,k=1\ldots,N}}{\det[x_j^{N-k}]_{j,k=1,\ldots,N}} \tag{4.4.2}$$

Proof

For notational convenience, the case $N=3$ will be considered. Equation **??** then reads

$$s_\lambda(x_1,x_2,x_3)=\frac{\begin{vmatrix} x_1^{2+\lambda_1} & x_1^{1+\lambda_2} & x_1^{\lambda_3} \\ x_2^{2+\lambda_1} & x_2^{1+\lambda_2} & x_2^{\lambda_3} \\ x_3^{2+\lambda_1} & x_3^{1+\lambda_2} & x_3^{\lambda_3} \end{vmatrix}}{\begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix}} \tag{4.4.3}$$

We can show this by introducing a recurrence for the Schur polynomials. If we consider a Young tableau with shape $\lambda=(\lambda_1,\lambda_2,\lambda_3)$, then we can also consider smaller Young tableaux which can be thought of as contained inside the larger Young tableau. The smaller Young tableaux have shape $\mu=\mu_1,\mu_2$ and must have $\lambda_1\geq\mu_1\geq\lambda_2\geq\mu_2\geq\lambda_3$. The smaller Young tableaux are obtained by deleting all the $3's$ in the larger Young tableau. This gives that

$$s_\lambda(x_1,x_2,x_3)=\sum_{\substack{\mu_1,\mu_2:\\ \lambda_1\geq\mu_1\geq\lambda_2\geq\mu_2\geq\lambda_3}} s_\mu(x_1,x_2)x_3^{|\lambda|-|\mu|} \tag{4.4.4}$$

and this together with the initial condition $s_\lambda(x_1,x_2,x_3)=1$ for $\lambda=\emptyset$ uniquely specifies s_λ. Therefore the strategy we will use to show propostion **??**, is to show that both sides of **??** satisfy the recurrence **??**.

To show this, we start by setting $x_3=1$ and subtracting the third row in each determinant from the other two rows in the same determinant to obtain

$$s_\lambda(x_1,x_2,1)=\frac{\begin{vmatrix} x_1^{2+\lambda_1}-1 & x_1^{1+\lambda_2}-1 & x_1^{\lambda_3}-1 \\ x_2^{2+\lambda_1}-1 & x_2^{1+\lambda_2}-1 & x_2^{\lambda_3-1} \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^2-1 & x_1-1 & 0 \\ x_2^2-1 & x_2-1 & 0 \\ 1 & 1 & 1 \end{vmatrix}}$$

Now dividing the first row through by $x_1 - 1$ and the second by $x_2 - 1$ gives

$$s_\lambda(x_1, x_2, 1) = \frac{\begin{vmatrix} \sum_{n=0}^{1+\lambda_1} x_1^n & \sum_{n=0}^{\lambda_2} x_1^n & \sum_{n=0}^{\lambda_3 - 1} x_1^n \\ \sum_{n=0}^{1+\lambda_1} x_2^n & \sum_{n=0}^{\lambda_2} x_2^n & \sum_{n=0}^{\lambda_3 - 1} x_2^n \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 + 1 & 1 & 0 \\ x_2 + 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix}}$$

Subtracting the second column of each determinant from the first column and subtracting the third column of each determinant from the second we obtain

$$s_\lambda(x_1, x_2, 1) = \frac{\begin{vmatrix} \sum_{n=1+\lambda_2}^{1+\lambda_1} x_1^n & \sum_{n=\lambda_3}^{\lambda_2} x_1^n & \sum_{n=0}^{\lambda_3 - 1} x_1^n \\ \sum_{n=1+\lambda_2}^{1+\lambda_1} x_2^n & \sum_{n=\lambda_3}^{\lambda_2} x_2^n & \sum_{n=0}^{\lambda_3 - 1} x_2^n \\ 0 & 0 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & 1 & 0 \\ x_2 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}}$$

Expanding by the final row and simplifying shows

$$s_\lambda(x_1, x_2, 1) = \frac{\begin{vmatrix} \sum_{\mu_1=\lambda_2}^{\lambda_1} x_1^{1+\mu_1} & \sum_{\mu_2=\lambda_3}^{\lambda_2} x_1^{\mu_2} \\ \sum_{\mu_1=\lambda_2}^{\lambda_1} x_2^{1+\mu_1} & \sum_{\mu_2=\lambda_3}^{\lambda_2} x_2^{\mu_2} \end{vmatrix}}{\begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix}}$$

or equivalently

$$s_\lambda(x_1, x_2, 1) = \sum_{\mu_1=\lambda_2}^{\lambda_1} \sum_{\mu_2=\lambda_3}^{\lambda_2} \frac{\begin{vmatrix} x_1^{1+\mu_1} & x_1^{\mu_2} \\ x_2^{1+\mu_1} & x_2^{\mu_2} \end{vmatrix}}{\begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix}}$$

Hence we have shown that

$$s_\lambda(x_1, x_2, 1) = \sum_{\mu_1=\lambda_2}^{\lambda_1} \sum_{\mu_2=\lambda_3}^{\lambda_2} s_\mu(x_1, x_2) \tag{4.4.5}$$

We note from equation **??** that

$$s_\lambda(cx_1, cx_2, cx_n) = c^{|\lambda|} s_\lambda(x_1, x_2, x_3) \tag{4.4.6}$$

Hence we can replace x_1 by $\frac{x_1}{x_3}$ and x_2 by $\frac{x_2}{x_3}$ in equation **??** to deduce

$$s_\lambda(x_1,x_2,x_3) = \sum_{\mu_1=\lambda_2}^{\lambda_1} \sum_{\mu_2=\lambda_3}^{\lambda_2} s_\mu(x_1,x_2)x_3^{|\lambda|-|\mu|}$$

which is the sought recurrence. The formula **??** also gives us the initial condition $s_\lambda(x_1,x_2,x_3) = 1$ for $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$. □

A property which can be seen from equation **??** is the symmetry of Schur polynomials. Schur polynomials satisfy the following

$$s_\lambda(x_1,\ldots,x_n) = s_\lambda(x_{\sigma(1)},\ldots,x_{\sigma(n)})$$

for any permutation σ. This is saying that the weights in equation **??** can be permuted without changing the corresponding Schur polynomial s_λ.

Suppose the variables x_j in equation **??** are specialized to $x_j = q^{j-1}$. This gives

$$s_\lambda(1,q,\ldots,q^{N-1}) = \frac{\det[q^{(j-1)(N-k+\lambda_k)}]_{j,k=1,\ldots,N}}{\det[q^{(j-1)(N-k)}]_{j,k=1,\ldots,N}} \tag{4.4.7}$$

The determinants in this expression can be evaluated by using the following identity, known as the Vandermonde determinant formula.

Proposition 12.

$$\det[x_k^{j-1}]_{j,k=1,\ldots,N} = \prod_{1\leq<k\leq N}(x_k - x_j) \tag{4.4.8}$$

Proof
Again for notational convenience, consider the case $N = 3$. Then equation **??** reads

$$\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ x_1^2 & x_2^2 & x_3^2 \end{vmatrix} = (x_3 - x_2)(x_3 - x_1)(x_2 - x_1)$$

Now we observe that on the left hand side, if $x_k = x_{k'}$ for $k \neq k'$, then two columns are equal and so the determinant vanishes. It follows that the right hand side is a factor. Further, replacing x_k by αx_k $(k = 1,2,3)$ shows both sides are homogeneous of degree 3 and so are equal up to a constant. That the constant is unity follows by comparing coefficients of $x_3^2 x_2 x_1^0$ on both sides. □
Applying equation **??** to equation **??** with $x_k = q^{(N-k+\lambda_k)}$ in the numerator and $x_k = q^{(N-k)}$ in

the denominator shows

$$\begin{aligned} s_\lambda(1, q, \ldots, q^{N-1}) &= \prod_{1\le j<k\le N} \frac{q^{N-k+\lambda_j} - q^{N-j+\lambda_j}}{q^{N-k} - q^{N-j}} \\ &= q^{\sum_{j=1}^{N}(j-1)\lambda_j} \prod_{1\le j<k\le N} \frac{1 - q^{\lambda_j-\lambda_k+k-j}}{1 - q^{k-j}} \end{aligned}$$

Taking the limit $q \to 1$ it follows that

$$s_\lambda(x_1, \ldots, x_N)|_{x_1=\ldots=x_N=1} = \prod_{1\le j<k\le N} \frac{\lambda_j - \lambda_k + k - j}{k - j} \tag{4.4.9}$$

4.5 A formula for $Pr(T_{N,N} \le l)$

In formulating the correspondence between weighted integer matrices and weighted lattice paths, each weight x_{ij} was taken to be proportional to $(a_i b_j)^{x_{ij}}$. A probabilistic model giving rise to this weighting is obtained by choosing each entry x_{ij} of the matrix (or equivalently each service time in the queueing model) with the geometrical distribution

$$Pr(x_{i,j} = k) = (a_i b_j)^k (1 - aib_j), \qquad |a_i b_j| < 1 \tag{4.5.1}$$

With this probability distribution on the entries of the matrix, we want to know what is the probability that the maximum displacements of the corresponding lattice paths are given by the partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N)$? This is equal to

$$\prod_{i,j=1}^{N} (1 - a_i b_j) \times (\text{weight of the left lattice paths}) \times (\text{weight of the right lattice paths})$$

But

$$\text{weight of the left lattice paths} = s_\lambda(b_1, b_2, \ldots, b_N)$$

and

$$\text{weight of the right lattice paths} = s_\lambda(a_1, a_2, \ldots, a_N)$$

We know $T_{N,N}$ corresponds to λ_1, so summing over all λ with $\lambda_1 \le l$ gives $Pr(T_{N,N} \le l)$. Explicitly

$$Pr(T_{N,N} \le l) = \prod_{i,j=1}^{N} (1 - a_i b_j) \sum_{\lambda:\lambda_1 \le l} s_\lambda(a_1, \ldots, a_N) s_\lambda(b_1, \ldots b_N) \tag{4.5.2}$$

Suppose now that $a_i = b_j = q^{\frac{1}{2}}$ for $(i, j = 1, \ldots, N)$ so equation **??** reads

$$Pr(x_{ij} = k) = (1 - q)q^k \tag{4.5.3}$$

and **??** now reads

$$Pr(T_{N,N} \leq l) = (1-q)^{N^2} \sum_{\lambda:\lambda_1 \leq l} q^{|\lambda|} \left(s_\lambda(x_1,\ldots,x_N)|_{x_1=\ldots=x_N=1}\right)^2 \tag{4.5.4}$$

where use has been made of **??**. Making use of equation **??** this reads

$$Pr(T_{N,N} \leq l) = \frac{(1-q)^{N^2}}{\prod_{1\leq j<k\leq N}(k-j)} \sum_{\lambda:\lambda_1 \leq l} q^{|\lambda|} \left(\prod_{1\leq j<k\leq N}(\lambda_j - \lambda_k + k - j)\right)^2 \tag{4.5.5}$$

To make further progress, we consider the limit $q \to 1$ in equation **??**. This can be done by setting $q = e^{-\frac{1}{L}}$ and taking the limit $L \to \infty$. Because the probability varies slowly on a length scale L, it suggests introducing the scaled variables

$$y = \frac{k}{L} \qquad\qquad X_{ij} = \frac{x_{ij}}{L}$$

into equation **??**. Notice that as k tends to $k+1$, we have $y \to y + \frac{1}{L}$, which we write as $y \to y + dy$. We can then regard X_{ij} as a continuous variable and we interpret $X_{ij} = y$ to mean $X_{ij} \in [y, y+dy]$ since we want to account for all values $X_{ij} \in [\frac{k}{L}, \frac{(k+1)}{L}]$. Consequently the limit $L \to \infty$ of **??** is

$$Pr(X_{ij} \in [y, y+dy]) = e^{-y} dy \tag{4.5.6}$$

The entries of the matrix are now postive real numbers chosen from this distribution, which is an example of the gamma distribution $\Gamma[1,1]$.

The corresponding limit of equation **??** is obtained by replacing l by sL, $T_{N,N}$ by $T_{N,N}L$ and setting $x_j = \frac{\lambda_j}{L}$. Taking $L \to \infty$ then gives

$$Pr(T_{N,N} \leq s) = \frac{1}{\prod_{j=1}^{N} \Gamma[j]} \int_0^s dx_1 \ldots \int_0^s dx_N \prod_{j=1}^{N} e^{-x_j} \prod_{1\leq j<k\leq N} (x_k - x_j)^2 \tag{4.5.7}$$

where use of has been made of the Rieman integral formula for a definite integral [**?**]

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} f\left(\frac{j}{N}\right) = \int_0^1 f(t) dt \tag{4.5.8}$$

Comparing the expression in **??** with the equation **??** shows that the integrand is precisely the eigenvalue probability density function for the LUE. The integral over $[0,y]^N$ of this gives the probability that all the eigenvalues are in the interval $[0,y]$, or equivalently that none of the eigenvalues are in the interval (y,∞). Hence

$$Pr(T_{N,N} \leq y) = Pr\left(\text{the interval } (y,\infty) \text{ contains no eigenvalues in the LUE}\right) \tag{4.5.9}$$

Differentiating with respect to y, the right hand side then gives us the distribution of the largest eigenvalue in the LUE. But we know that in terms of the scaled variable specified by **??**, for $N \to \infty$, this distribution is equal to the distribution, $p_2^{soft}(s)$. Thus we have the limit theorem, [**?**]

$$\lim_{N\to\infty} Pr\left(\frac{T_{N,N} - 4N}{2(2N)^{\frac{1}{3}}} \leq X\right) = \int_{-\infty}^{X} p_2^{soft}(s) ds \tag{4.5.10}$$

This result can be illustrated by empirically computing the distribution of $\frac{(T_{N,N}-4N)}{2(2N)^{\frac{1}{3}}}$ with $T_{N,N}$ being computed from the recurrence in theorem **??**. This result was also shown by constructing a histogram for $T_{N,N}$ and comparing this with the curve of the p_2^{soft} distribution. The recurrence in theorem **??** was used to compute a large number (we used M=5000) of values for $T_{N,N}$ (we used N = 50) which were presented in a histogram for $T_{N,N}$. The histogram is shown in Figure **??** and the corresponding code is shown in Appendix **??**.

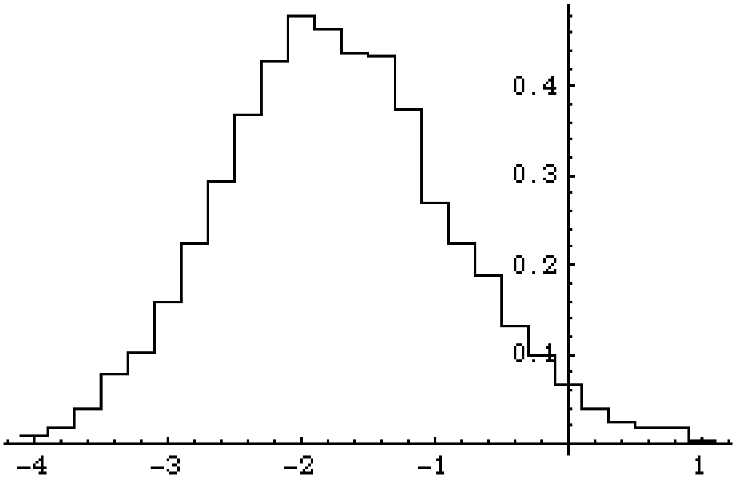

Figure 4.5: The distribution of $T_{N,N}$

Comparison of the histogram for $T_{N,N}$ with the curve for the p_2^{soft} distribution displays strong evidence for the result in equation **??**.

Chapter 5

Concluding Remarks and Ideas for Further Research

This thesis has investigated the inter-relationship between the probability density function for the distribution of the largest eigenvalue in certain random matrix ensembles, and the distribution of the exit time for a certain queueing process.

The random matrix ensembles considered were the Gaussian unitary ensemble of random Hermitian matrices and the Laguerre unitary ensemble of random complex positive definite matrices. These matrices are specified in terms of the probability density function for the elements, which to obtain a tractable model (ie. one that agrees with analytic computations) are taken to be Gaussians. Changing variables from the elements to the eigenvalues and eigenvectors, gives the eigenvalue probability density function. In this study the interest has been in the largest eigenvalue. To compute this from the eigenvalue probability density function requires integrating over each of the eigenvalues up to a definite value, say s, then differentiating with respect to s. If the distributions are appropriately centred and scaled as demonstrated in chapter **??**, the large N limiting forms are both equal to the p_2^{soft} distribution discussed in chapter **??**. Thus two different matrix ensembles have the same limiting behaviour for the distribution of the largest eigenvalue. This is an example of the phenomenon of universality in random matrix theory, whereby limiting statistical quantities depend on the symmetry of the matrix, or whether the elements are real or complex, but not on the details of the distribution of these features. For example, it is expected that p_2^{soft} will again be the limiting distribution of the scaled largest eigenvalue in a complex Hermitian matrix where the off diagonal elements are chosen with equal probability from the discrete set of four elements $\pm 1 \pm i$ [**?**].

In the Queueing process a key quantity is the matrix of the service times. The Robinson-Schensted-Knuth correspondence discussed in chapter **??**, gives a bijection between any matrix with non-negative integers and a pair of semi-standard tableau. When the entries of the matrix of service times are taken to be independent random variables chosen from the exponential distribution, an analytic formula can be obtained for the probability distribution of the exit time of all N jobs from all N queues. This formula is precisely that obtained in chapter **??** for the probability distribution of the largest eigenvalue in the Laguerre unitary ensemble. As the limiting form of this distribution is given by p_2^{soft}, the limiting form of the probability density function for the exit time from the queueing system is also given by p_2^{soft}. Here the tractability of the analysis relied on the entries of the matrix of service times begin chosen from the exponential distribution. Just as the scaled eigenvalue distribution of a random matrix is expected to be insensitive to the details of the distribution of the elements, it should be the case that the limiting service time distribution is always p_2^{soft}, independent of the particular distribution of the service times. To make this a precise mathematical theorem is an outstanding question in the field.

Bibliography

[1] J. Baik, P. Deift and K. Johansson, *On the distribution of the length of the longest increasing subsequence of random permutations*, J. Am. Math. Soc. vol 12, pg. 1119-1178, 1999.

[2] Y. Baryshnikov, *GUEs and queues*, Prob The Rel Fields, vol 119, issue 2, pg 256, 2001.

[3] C.W.J. Beenakker, *Random-matrix theory of mesoscopic fluctuations in conductors and superconductors* Phys. Rev. B. vol 47. pg 15763-15775, 1993.

[4] R. Conte, *The Painlevé property one century later*, Springer-Verlag New York, 1999.

[5] P. Deift, *Orthogonal polynomials and random matrices:a Rieman-Hilbert approach*, Courant Institute of Mathematical Sciences, New York 1999.

[6] I. Dumitriu and A. Edelman *Matrix models for beta ensembles* J. Math. Phys., vol 43, pg 5830-5847, 2002.

[7] M.V. Fedoryuk, *Asymptotic analysis of linear ordinary differential equations*, Springer-Verlag Berlin Heidelberg, Printed in the United States of America, 1993.

[8] P. J. Forrester, *Log-gases and random matrices*, www.ms.unimelb.edu.au/ matjpf/matjpf.html.

[9] P. J. Forrester and E. M. Rains, *Interpretations of some parameter dependent generalizations of classical matrix ensembles*, Probab. Theory Relat. Fields, vol 131, 2005.

[10] J.B. French, P.A. Mello and A.Pandey, *Statistical properties of many-particle spectra. II. Two-point correlations and fluctuations*, Annals. of Phys., vol 113, pg 277-292, 1978.

[11] D. E. Knuth, *Permutations, matrices and generalized Young Tableaux*, Pacific J. Math, vol 34, pg 709-727, 1970.

[12] P. Lancaster, *Theory of matrices*, Academic Press, New York, 1969.

[13] M. L. Mehta, *Random matrices*, Elsevier, Amsterdam, the Netherlands, 2004.

[14] M. L. Mehta, *Random matrices and the statistical theory of energy levels*, Academic Press, 1967.

[15] J. M. Ortega, *Matrix theory a second course*, Plenum Press, New York, 1987.

[16] D. Petz and F. Hiai, *The semicircle law, free random variables and entropy*, The American Mathematical Society, 2000.

[17] M. Prähofer and H, Sphon, *Exact scaling functions for one-dimensional stationary KPZ growth*, asXiv:cond-mat/021519 vol 1, 2002.

[18] Stewart, J, *Calculus concepts and contexts*, Wadsworth Group. Brooks/Cole, U.S.A., 2001.

[19] C.A. Tracy and H. Widom, Introduction to random matrices, In G.F. Helmick, *Geometric and quantum aspects of integrable systems*, vol 424 of Lecture notes in physics:pages 407-424. Springer New York 1993.

[20] E. Wigner, *On the distribution of roots of certain symmetric matrices* Annals of Mathematics, vol.67, pg. 325-327, 1958.

[21] E. Wigner *Characteristic vectors of bordered matrices with infinite dimensions*, Annals of Mathematics, vol 62, pg. 548-564, 1955.

[22] J. Wishart, *The generalized product moment distribution in samples from a normal multivariate population*, Biomatrika, vol. 20 A, pg 32-52, 1928.

Appendix A

Code for Calculating the Eigenvalue P.d.f. for Matrices from the GUE

Some notations used:
T=N=size of the matrix
$p[k,x]$ = $p_k(\lambda)$ where $\lambda = x$

```
<< Statistics‘ContinuousDistributions‘
Initial Conditions
M = 5000;
T = 50;
ev = {};
Recurrence
For [m = 1, m < M + 1, m++,
  Clear[p];
  p[0, x_] = 1;
  p[1, x_] = x - Random[ NormalDistribution[0, 1]];
  For[k = 2, k < T + 1, k++,
    p[k_, x_] := p[k, x] = Expand[(x - Random[NormalDistribution[0,
          1]])*p[k - 1,
   x] - Random[GammaDistribution[k - 1, 1]]*p[k - 2, x]]];
  tt = NSolve[p[T, x] == 0, x];
  cc = (x /. tt)/N[Sqrt[2]];
  cc = (Abs[cc] - Sqrt[2*T])*(Sqrt[2]*T^(1/6));
  dd = (Abs[-cc] - Sqrt[2*T])*(Sqrt[2]*T^(1/6));
  AppendTo[ev, {cc[[T]], dd[[1]]}];
  lis = Flatten[ev, 1];
  ]
Drawing the Histogram
nev[int_] := Round[N[int*lis]]/int
frlis2[int_] := (qq = nev[int];
  Table[{{N[(i - 1/2)/int], N[(
    int*Count[qq,
      i/int])/(2*M)]}, {N[(i + 1/2)/int], N[(
          int*Count[qq, i/int])/(2*M)]}}, {i, -4*int, int}])
barplot[int_] := (lis2 = Flatten[frlis2[int], 1]; Graphics[{{Line[lis2]}}])
Show[barplot[5], Axes -> True]
```

Appendix B

Code for Calculating the Eigenvalue P.d.f. for Matrices from the LUE

```
<< Statistics`ContinuousDistributions`
M = 5000;
T = 50;
ev = {};
For[m = 1, m < M + 1, m++,
  Clear[p];
  p[0, x_] = 1;
  p[1, x_] = x - Random[GammaDistribution[T, 1]];
  For[k = 2, k < T + 1, k++,
    p[k_, x_] := p[k, x] = Expand[(x -
                  Random[GammaDistribution[T - k + 1, 1]])*p[k - 1, x] -
            x*Random[GammaDistribution[k - 1, 1]]*p[k - 2, x]]];
  tt = NSolve[p[T, x] == 0, x];
  cc = (x /. tt);
  ec = (cc - 4*T)/(2*(2*T)^(1/3));
  AppendTo[ev, ec[[T]]];
  lis = Flatten[ev, 1];
  ]
Drawing the Histogram
 nev[int_] := Round[N[int*lis]]/int
frlis2[int_] := (qq = nev[int];
  Table[{{N[(i - 1/2)/int], N[(
    int*Count[qq,
      i/int])/(M)]}, {N[(i + 1/2)/int], N[(int*
          Count[qq, i/int])/(M)]}}, {i, -4*int, int}])
barplot[int_] := (lis2 = Flatten[frlis2[int], 1]; Graphics[{{Line[lis2]}}])
Show[barplot[5], Axes -> True]
```

Appendix C

Code for Calculating the p_2^{soft} Distribution

Some notations used:
$al[n] = \alpha_n$
$al3[n] = \alpha_n^{(3)}$
$bigQ[s] = Q(s)$
$imu[s, j] = \int_s^{s_0 - is[j]} tu^2(t)dt$
$smu[s, j] = \int_s^{s_0} tu^2(t)dt$
$bmu[s, j] = \int_s^{s_0 - is[j]} u^2(t)dt$
$sub[s, j] = \int_s^{s_0} u^2(t)dt$

```
Initial conditions
sa = 300;
al[0] = N[1, 300];
a[-2] = 0;
a[-1] = 0;
Recursive definitions
al[n_] := al[n] = (6*n - 1)*(6*n - 5)*al[n - 1]/(72*n);
al3[n_] := al3[n] = Sum[al[n - l]*
          Sum[al[l - k]*al[k], {k, 0, l}],
        {l, 0, n}];
a[n_] := a[n] = al3[n] + (3/4)*n*a[n - 1] -
        (1/8)*(n - 1/6)*(n - 5/6)*a[n - 2];
Tabulation
sa = 300;
s0 = 50;
um = 600;
n0 = Min[um, Floor[(4/3)*s0^(3/2)]];
cc = Table[a[n], {n, 0, n0}];
cg = Get["lau1.50.dat"];
```

```
For[i = 0, i < n0 + 1, i++, a[i] = cg[[i + 1]]];
Evaluation of Q(s0), Q’(s0)
bigQ[t0_] = (Exp[-2*t0^(3/2)]/(32*N[Pi, sa]^(3/2)*
              t0^(7/4)))*
      Sum[(-1)^k*a[k]/(2*t0^(3/2)/3)^k, {k, 0, n0}];
q[s_] := AiryAi[s] + bigQ[s];
c0 = q[s0];
c1 = q’[s0];
Recursive definitions
n1 = 200;
c[-1, 1, 0] = 0;
c[0, 1, 0] = c0;
c[1, 1, 0] = c1;
c[n_, 2, 0] := c[n, 2, 0] =
      Sum[c[n - j, 1, 0]*c[j, 1, 0], {j, 0, n}];
c[n_, 3, 0] := c[n, 3, 0] =
      Sum[c[n - j, 1, 0]*c[j, 2, 0], {j, 0, n}];
c[n_, 1, 0] := c[n, 1, 0] = (2*c[n - 2, 3, 0] + s0*c[n - 2, 1, 0] +
          c[n - 3, 1, 0])/(n*(n - 1))
q[s_, 0] := Sum[c[n, 1, 0]*(s - s0)^n, {n, 0, n1}];
qd[s_, 0] := Sum[n*c[n, 1, 0]*(s - s0)^(n - 1), {n, 1, n1}];
Table[c[n, 1, 0], {n, 0, n1}];
Table[c[n, 2, 0], {n, 0, n1}];
Estimate the radius of convergence
Table[N[1/Abs[c[n, 1, 0]]^(1/n), 10], {n, 180, 200}]
Recursive generation of power series
is[n_] = n;
nn = 90;
k0 = q[s0 - is[1], 0];
k1 = qd[s0 - is[1], 0];
q[s_, j_] := Sum[c[l, 1, j]*(s - s0 + is[j])^l,
      {l, 0, n1}];
qd[s_, j_] := Sum[l*c[l, 1, j]*(s - s0 + is[j])^(l - 1),
      {l, 1, n1}];
For[j = 1, j < nn + 1, j++,
  c[-1, 1, j] = 0;
  c[0, 1, j] = k0;
  c[1, 1, j] = k1;
  c[n_, 2, j] := c[n, 2, j] =
      Sum[c[n - l, 1, j]*c[l, 1, j], {l, 0, n}];
  c[n_, 3, j] := c[n, 3, j] =
      Sum[c[n - l, 1, j]*c[l, 2, j], {l, 0, n}];
  c[n_, 1, j] := c[n, 1, j] = (2*c[n - 2, 3, j] + (s0 - is[
        j])*c[n - 2, 1, j] +
            c[n - 3, 1, j])/(n*(n - 1));
  Table[c[n, 1, j], {n, 0, n1}];
  k0 = q[s0 - is[j + 1], j];
  k1 = qd[s0 - is[j + 1], j];
  ]
 partq[x_] = If[x  s0, q[x], 0];
fiq[x_] = If[x  s0 - is[nn], q[x, nn], 0];
```

```
For[m = 0, m < nn, m++,
  partq[x_, m] = If[s0 - is[m + 1]  x  s0 - is[m],
      q[x, m], 0]]
fq[x_] := partq[x] + fiq[x] + Sum[partq[x, m], {m, 0, nn - 1}]
Plot[fq[x], {x, -30, 5}]
Clear[d]
d[1, p_] = -(s0 - is[p])*c[0, 2, p];
d[n_, p_] := d[n, p] = -c[n - 2, 2, p]/n -
         (s0 - is[p])*c[n - 1, 2, p]/n;
For[j = 1, j < nn + 1, j++,
  Table[d[n, j], {n, 1, n1}]
  ]
imu[s_, p_] := Sum[d[n, p]*(s - (s0 - is[p]))^n, {n, 1, n1}];
smu[s_, p_] := Sum[imu[s0 - is[k + 1], k], {k, 0, p - 1}] + imu[s, p];
For[j = 1, j < nn + 1, j++,
  Table[c[n, 2, j], {n, 1, n1}]
  ]
bmu[s_, p_] := -Sum[c[n - 1, 2, p]/n*(s - (s0 - is[p]))^n, {n, 1, n1}];
sub[s_, p_] := (s - (s0 - is[p]))*(Sum[bmu[s0 - is[k + 1],
          k], {k, 0, p - 1}] + bmu[s, p]) +
      (s0 - is[p])*
        (Sum[bmu[s0 - is[k + 1], k], {k, 0, p - 1}] + bmu[s, p]);
Clear[p]
p[n_, m_] := p[n, m] = -
    Coefficient[x*smu[x +
      s0 - is[m], m], x, n + 1] + Coefficient[x*sub[x + s0 - is[m], m], x,
                n + 1];
For[nt = 0, nt < nn + 1, nt++, Table[p[n, nt], {n, 0, n1}]];
sup[s_, v_] := Sum[p[n, v]*(s - (s0 - is[v]))^n, {n, 0, n1}];
sj[t_, m_] := (g[x] = Series[
        Exp[sup[x, m]], {x, s0 - is[m], 199}]; g[x] = Normal[g[x]] /. x -> t)
eh[n_, m_] := eh[n, m] = Coefficient[x*sj[x + s0 - is[m], m], x, n + 1]
For[nt = 0, nt < nn + 1, nt++, Table[eh[n, nt], {n, 0, n1}]];
dsu[s_, v_] := Sum[n*eh[n, v]*(s - (s0 - is[v]))^(n - 1), {n, 1, n1}];
pr[x_] = If[x  s0, dsu[x - s0, 0], 0];
fr[x_] = If[x  s0 - is[nn], dsu[x - s0 + is[nn], nn], 0];
For[m = 0, m < nn, m++,
  pux[x_, m] = If[s0 - is[m + 1]  x  s0 - is[m],
      dsu[x, m], 0]]
psoft[x_] := pr[x] + fr[x] + Sum[pux[x, m], {m, 0, nn - 1}]
Plot[psoft[x], {x, -5, 3}]
Calculating Moments
tr[jp_, alpha_] :=
  Sum[(-1)^(n - 1)*n*eh[n, jp]*(s0 - is[jp])^(alpha + n)*
      Beta[1 - (s0 - is[jp + 1])/(s0 - is[jp]), n, alpha + 1],
    {n, 1, n1}]
tz[alpha_] := -
    Sum[(n/(
  alpha + n))*eh[n, s0](is[s0] - is[s0 + 1])^(alpha + n), {n, 1, n1}]
ts[jp_, alpha_] := -
    Sum[n*eh[n, jp]*(s0 - is[jp])^(alpha + n)*
```

```
        Beta[1 - (s0 - is[jp])/(s0 - is[jp + 1]),
          n, -alpha - n], {n, 1, n1}]
mu[alpha_] := Sum[tr[jp,
    alpha], {jp, 0, 49}] + tz[alpha] + Sum[ts[jp, alpha], {jp, 51, 90}];
```

Mean

```
mu[1]
```

Variance

```
var = mu[2] - (mu[1])^2
```

Standard Deviation

```
sd=(var)^(1/2)
```

Skewness and Kurtosis

```
skew = (mu[3] - (3*(mu[2])*(mu[1])) + (2*(mu[1])^3))/((sd)^3)
kur = (mu[4] - (4*(mu[3])*(mu[
          1])) + (6*(mu[2])*(mu[1])^2) - 3*(mu[1]^4))/((sd)^4) - 3
```

The psoft distribution and the GUE eigenvalue histogram.

```
Show[Out[83], Out[65]]
```

Appendix D

Code for Calculating the Empirical Distribution for $T_{N,N}$

```
<< Statistics`ContinuousDistributions`
n = 50;
m = 5000;
ev = {};
For[jj = 1, jj < n + 1, jj++,
    T[0, jj] = 0; T[jj, 0] = 0];
 For[p = 1, p < m + 1, p++,
  For[j = 1, j < n + 1, j++,
    For[k = 1, k < n + 1, k++,
      x[j, k] = Random[GammaDistribution[1, 1]];
      ]];
  For[i = 1, i < n + 1, i++,
    For[ij = 1, ij < n + 1, ij++,
      T[i, ij] = Max[T[i - 1, ij], T[i, ij - 1]] + x[i, ij]]];
  qt[p] = T[n, n]; AppendTo[ev, qt[p]];
  lis = Flatten[ev, 1]
  ]
 lis1 = (ev - 4*n)/(2*(2*n)^(1/3));
 nev[int_] := Round[N[int*lis1]]/int
 frlis2[int_] := (qq = nev[int];
  Table[{{N[(i - 1/2)/int], N[(
    int*Count[qq,
      i/int])/(m)]}, {N[(i + 1/2)/int], N[(int*
          Count[qq, i/int])/(m)]}}, {i, -4*int, int}])
barplot[int_] := (lis2 = Flatten[frlis2[int], 1]; Graphics[{{Line[lis2]}}])
Show[barplot[5], Axes -> True]
```